Exploring Input-Output Analysis with APL

Abstract

This volume is a computer software-supplemented version of the *Input-Output Analysis Computational Workbook*, which is a collection of annotated exercises to accompany the Third Edition of *Input-Output Analysis: Foundations and Extensions* —a textbook and desk reference for students and scholars in the input–output research and applications community. Basic concepts of the computer language APL are introduced and used to illustrate solutions to the computational exercise problems aligned with the chapters of the textbook.

Table of Contents

Introduction

This volume is a computer software-supplemented version of the *Input-Output Analysis Computational Workbook*, which is a collection of annotated exercises to accompany the Third Edition of *Input-Output Analysis: Foundations and Extensions*[1]––a textbook and desk reference for students and scholars in the input–output research and applications community.

The textbook, upon which both the basic and computer software-supplemented versions of the workbook are based, includes extensively referenced and indexed coverage of most subtopics in the field. It is an ideal introduction to the subject for advanced undergraduate and graduate students in a wide variety of fields, including economics, regional science, regional economics, city, regional and urban planning, environmental planning, public policy analysis, and public management.

The workbook is an expanded discussion of exercise problems aligned with the book chapters, illustrating major concepts and key analytical approaches as well as exploring applications using examples and selected real-world data. This software-supplemented version of the workbook provides one source of computer software for experimenting with the workbook exercises, as well as an introduction to software for the workbook written in the computer language APL, which is a powerful, array-oriented, interactive, scientific computing language especially well-suited to input-output analysis (IOA) calculations.

Many scientific computing software packages available today are suitable to work through the exercises in the workbook and, for those familiar with such packages, the workbook exercises can be easily formulated using them. The software introduced and used in this volume (written in APL) is tuned very specifically only to the needs of the workbook, which includes many calculations beyond the most basic input-output calculations so it is an efficient path for exploring many of the complex mathematical formulations developed in the text. However, in this volume, only the features necessary to analyze the IOA exercises considered in the workbook are developed which will barely scratch the surface of the language's capabilities. For more extensive coverage of APL's features and modern implementations, see the references and the appendix to this volume.

The third edition of the textbook, which has been fully revised and updated to reflect important developments in the field since earlier editions, is also supported by an accompanying website with supplemental appendices including further information for more advanced readers, the exercise problems and solutions explored in more detail in this volume, and a sampling of real-world data sets for experimenting with IOA (http://cambridge.com/millerandblair), including some larger data sets necessary for some exercise problems. We begin with an overview of the field of input-output analysis.

[1] Ronald E. Miller and Peter D. Blair. 2021. *Input-Output Analysis: Foundations and Extensions*, (Third Edition). Cambridge: Cambridge University Press; and the accompanying volume, *The Input-Output Analysis Computational Workbook: Annotated Exercises*.

Overview of Input-Output Analysis

Professor Wassily Leontief's 1971 presidential address to the American Economic Association was entitled "Theoretical Assumptions and Non-observed Facts." The address took many in the economics profession to task for failing to underscore the necessity of empirically verifying economic theory. This was a longstanding concern of Leontief's about how much of the economics profession had evolved in the post-World War II period and one that he was particularly focused on in developing his own research on systematically analyzing the interdependence of industries in an economy.

Leontief characterized his work as expressing mathematically the efforts of 18th century French economist, Francois Quesnay, to produce a diagrammatic representation of how expenditures can be traced through an economy in a systematic way, known as the *Tableau Économique.* Leontief referred to the analytical framework he had been devising since the 1930s as *input-output analysis* (IOA), referring to the essence of his approach of capturing from observed economic data for a specific geographic region (e.g., a nation, state, or county) the activity of a group of industries that both produce goods and services (outputs) and consume goods and services from other industries (inputs) in the process of producing each industry's own output. In recognition of this work, Leontief received the 1973 Nobel Memorial Prize in Economic Sciences. Today, the basic concepts of IOA set forth by Leontief are key if not central components of many types of economic analysis and, indeed, IOA and its extensions over the last three-quarters of a century remain one of the most widely applied methods in economics.

The number of industries considered in an IOA model may vary from only a few, to hundreds or thousands. The observed data are the flows among or transactions of products between each of an economy's industries (as a producer/seller) and each of the industries (as a purchaser/buyer) over a standard time-period, usually a year. In more contemporary terms, depending upon the level of industry and geographic aggregation and accounting for the role of imports, IOA equations quantify essentially the complete and detailed supply chains for all products and services in the economy.

As noted at the beginning of this overview, one of Leontief's central concerns was the degree to which the transactions table presented an empirically accurate and stable picture of economic activity and what time-period was suitable for sufficiently and faithfully capturing the production characteristics of the economy. Leontief often referred to production functions of industries in his model as production *recipes*, found by normalizing each column of the transactions table by the value of total output of the corresponding industry in the economy to produce a matrix of technical coefficients.

Mathematically, in its simplest form, IOA is based on a matrix of interindustry transactions, $\mathbf{Z}$, the rows of which correspond with producing industry sectors in the economy and the columns to those same industries as consumers of industrial products from across the economy, usually measured value terms such as dollars. The most common form of IOA is called an open model in which a schedule of *final* consumption is specified of industrial products in the economy, i.e., consumption outside the network of interindustry production, such as the total of personal consumption, government expenditures, capital expenditures, and exports. For this vector of total final demand, $\mathbf{f}$, the total industrial production for all sectors in the economy, including both deliveries to interindustry and final consumers, is specified as $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$ where $\mathbf{x}$ is the vector of total industrial outputs for all sectors in the economy. The production recipes or technical coefficients are defined by normalizing each column of $\mathbf{Z}$ by the value of total production for the industry designated by the column, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$, i.e., elements of this matrix of technical coefficients or direct requirements designate the dollars' worth of input from each industry in the economy consumed directly to produce one dollar's worth of the output for the industry designated by the column. The total production accounting can then also be written as $\mathbf{x} = \mathbf{Ax} + \mathbf{f}$ or, rearranging terms, as $\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{f}$ or $\mathbf{x} = \mathbf{Lf}$ where $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$. The matrix, $\mathbf{L}$, is known as the Leontief inverse or matrix of total requirements.

Since, mathematically, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A}^2 + \mathbf{A}^3 + ... + \mathbf{A}^n$, we can interpret the terms of this power series expression as the "rounds" of industrial production necessary to ultimately supply the final consumption. That is, the production necessary to directly supply final consumption is $\mathbf{Af}$. The production necessary to supply the inputs to that direct production (i.e., induced by the direct production) is $\mathbf{A}(\mathbf{Af})$ or $\mathbf{A}^2\mathbf{f}$ and subsequent "rounds" of induced production are $\mathbf{A}^3\mathbf{f}$, $\mathbf{A}^4\mathbf{f}$,..., $\mathbf{A}^n\mathbf{f}$ so that the value of total industry production in the economy, including the final consumption itself as well as the direct and all the induced production necessary to supply that final consumption, is $\mathbf{x} = \mathbf{f} + \mathbf{A}^2\mathbf{f} + \mathbf{A}^3\mathbf{f} + ... + \mathbf{A}^n\mathbf{f} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{f}$. In a Leontief economy, for any newly projected increment of final consumption, $\Delta\mathbf{f}$, the additional total industrial production, $\Delta\mathbf{x}$ necessary to satisfy that new increment of final consumption is then found by $\Delta\mathbf{x} = \mathbf{L}\Delta\mathbf{f}$.

The matrix of technical coefficients, $\mathbf{A}$, incorporates the central assumptions of the basic IOA model, i.e., that the interindustry flows from one industry to another for a given time-period depend entirely on the total output the consuming industry for that same time-period, i.e., industries exhibit a linear production function defined by fixed technical coefficients for that time-period. Thus, in a basic Leontief economy, an industry uses inputs in fixed proportions and ignores returns to scale.

Throughout Leontief's research career he spent much of his effort exploring the robustness of these assumptions and devising extensions and enhancements to the basic model to accommodate the situations when such assumptions were less suitable and to identifying characteristics of the approach and application that most influenced error and uncertainty in its use. Extensions to IOA became an important area of research in economics, many efforts of which focused on the practical challenges of implementing IOA, including managing the prodigious data and computational requirements, effects of industrial and geographic aggregation, and devising methods to characterize secondary industrial production, final consumer consumption, and the role of capital investment and use. Other extensions enabled IOA to be focused on analyzing structural change in the economy or on specific sectoral issues such as analyzing energy use, environment impacts, and labor utilization.

With growing confidence in the utility of IOA in many different types of economic analysis, much more attention was paid by governments to assembling local, regional, and national data suitable for IOA. A pivotal development in broad implementation of IOA was a widespread, if not essentially uniform adoption of a standardized System of National Accounts (SNA) for economic activity. Work spearheaded by British economist Richard Stone, for which he received the 1984 Nobel Memorial Prize in Economic Sciences, and subsequently promulgated by the United Nations, the SNA enabled systematic tracking of economic activities on a national and international scale. Since the late 1950s most developed nations and many developing ones routinely construct IOA tables along with governments or related agencies for many regions and localities and, increasingly, IOA efforts capture transactions between regions or nations in multiregional models, some at a global scale.

Perhaps the three most significant early limitations to widespread use of IOA were: (1) the lack of reliable data from which to construct the basic interindustry accounts, (2) the lack of uniform standards in the kinds and scale of data collected for IOA, and (3) the extraordinary computational requirements of IOA relative to computer capacity at the time. In the earliest days of IOA, the computational requirements were dominant constraints, limiting its application, even if the necessary data were available, to scores of industries rather than the hundreds or thousands today. Even the most basic of IOA applications involves a large system of linear equations. While conceptually straightforward, computational solution at the time was challenging for even the most powerful computers of the day.

The constraint on computational capacity at the time put IOA front and center in use of the earliest electronic computers, but with the exponential growth in computing capacity over the last half century, such limitations have all but evaporated today. With standardized and much more readily available data, supplemented with methods for utilizing alternative sources of data, IOA is experiencing a fresh resurgence of interest in its utility for many economic issues, especially for global issues requiring large multiregional models. As a result, long avoided because its data and computational burden was often considered a bridge too far, IOA has re-emerged as a central tool in economics and, increasingly, in other areas such as accounting for pollution emissions and mitigation (and related ecosystem models), social accounting models, and many others.

In the last decade IOA's integration with other modeling frameworks has blossomed as well, including links with econometrics, resource planning, demographic modeling, and many others. Leontief's original framework conceived of industry production functions as measured in physical units, such as specifying the technical coefficients in terms of tons of coal or bushels of wheat, as inputs, required per dollars' worth of an industry's output or per ton of steel output. However, the data collection requirements and other constraints rendered implementation of the framework measured in physical units too unwieldy, certainly at the time and even today to a lesser extent. But, while the basic methodology for IOA evolved, in both theory and application, largely through measuring all quantities in value terms with implicit fixed prices, its use expressed in physical units was always considered desirable, both to moderate the impact of prices in analysis and to allow IOA to relate more easily with other modeling frameworks.

The generalization of IOA techniques to a broader conceptual level, such as accounting for economic activity beyond its primary focus on interindustry production, also originated with simpler attempts to link IOA models and other national income accounting techniques. Such generalizations enabled extension of IOA to explore the roles of labor, households, and the social institutions of the economy. Extensions to IOA, such as social accounting matrices and other related constructs, capture many different socioeconomic characteristics of an economy associated with interindustry activity, and enable analysis, for example, of income from employment and its disposition, labor costs, and the demographics of the work force that comprise the market for the supply and demand of labor.

Even late in his own life, Leontief continued to explore ways in which his framework could be implemented more widely, e.g., using physical units rather than value terms to facilitate wider use. These techniques involved many measurable quantities associated with interindustry activity, such as employment, energy use, and environmental pollution. Integration with ecosystem models, for example, addresses the interface between the economy and ecosystems, enabling systematic analysis of such contemporary issues as consumption accounting of global carbon emissions, measuring the energy and environmental resource "footprint" of nations, or the environmental emissions embodied in international trade.

Today, IOA is a well-established and widely utilized tool for analyzing economic activity at any geographic scale, most recently at a global scale. Enabled by increasingly standardized data characteristics and availability of data as well as the formidable computational capacity available today, IOA will continue to grow in its use and utility for addressing many types of economic policy and planning issues. Our text captures most of the important features and extensions of IOA since its conception and its initial applications nearly a century ago. The computational exercise problems in this workbook illustrate many of these features.

Input-Output's Computational History

IOA is a computationally intensive field today, but in the field's infancy, computational tools were not available to make practical even the most basic of necessary calculations for IOA. Today there are few computational impediments to constructing and using even the largest IOA models. The evolution of IOA can be characterized *computationally* in six development periods that roughly align with the development of IOA itself:

(1) *Birth*. In the 1930s Professor Leontief conceived of the IOA framework and introduced it to the world of economic thought, establishing the very specific needs for the computing capacity needed to implement it. However, computer capacity at the time did not yet exist to carry out even the most basic calculations and only very specialized electro-mechanical calculators were available. Leontief made use of the first large scale mechanical computing machinery in 1935 and later the first commercial electro-mechanical computer, the IBM Automatic Sequence Controlled Calculator (called the Mark I) developed for military applications that preceded electronic computers. Leontief's IOA was employed to help government preparations for the US entry into World War II, as well as during the war for operational planning and planning for postwar recovery in the US and in Europe. These tasks were very simple implementations of IOA but involved massive manual computational effort for the time.

(2) *Infancy*. In the 1940s computer capabilities finally developed that were sufficient to carry out at least basic IO computations. Electronic computers were still in their infancy. The very first programmable, electronic, general-purpose digital computer, Electronic Numerical Integrator and Computer (ENIAC), appeared at the University of Pennsylvania in 1945. Computers of ENIAC's generation, such as the SEAC (Standards Eastern Automatic Computer), developed by the National Bureau of Standards for the Air Force, were applied to Leontief's IOA through a government interagency project funded by the Air Force's Planning Research Division, known as Scientific Computation of Optimum Programs (SCOOP). Project SCOOP commissioned Leontief to update the 1939 US interindustry transactions table to 1947 to help with economic planning following World War II.
(3) *Toddler*. In the 1950s and 1960s, computing capacity appeared gradually for widespread applications. This new capacity was used to develop many national and regional applications of IOA that materialized in the 1950s, including perhaps most notably Project SCOOP's greatly expanded activities as the Korean War erupted in 1950. That effort, focused on possible obstructions to wartime mobilization, led to considerable progress in the ability to work with large-scale input–output tables of more than 500 industrial sectors. And with introduction of Richard Stone's UN-sponsored standardized system of national accounts (SNA) in 1961, implementation of IOA data and analysis began to proliferate world-wide.
(4) *Pre-school*. In the 1970s and 1980s computing capacity became generally available to enable widespread IOA research activities and applications. Use of this capacity and databases which coincided with increasing adoption of SNA for IOA data across the globe, albeit in the 1970s often at substantial expense which initially slowed the pace. In the 1980s, with the introduction of desktop computers and increasing availability of software tuned to IOA needs, computing costs plummeted and cost concerns began to become less dominant in implementing IOA databases and modeling capabilities. Many sources have attempted to chronicle the history of the cost of computing, which is challenging to address in any meaningful way since it is affected by so many factors, but, generally, computing power available per dollar has increased by a factor of ten roughly every four years over the last quarter of a century. Empowered with growing computational capacity, extension of IOA to address more narrow policy concerns, including environmental, resource, and regional issues developed as well at national, regional, and even metropolitan geographic levels.
(5) *Young adult.* In the 1990's and 2000's the cost of computing capacity continued to drop quickly, becoming much more affordable and widely accessible. Software with capabilities well-matched to IOA needs and the rapidly proliferating extensions became widely accessible and increasingly affordable as well.
(6) *Adulthood*. Since the decade following 2010, the cost of computing capacity has all but evaporated as a significant factor in implementing IOA analysis models and databases relative to other costs, such as those associated with compiling suitable data. Powerful desktop computers can often be used for even very large IOA models. Software tuned to most IOA needs continues to develop commercially and many specialized packages implementing at least basic IOA calculations are now widely available.

As reported in the textbook: "Since the early days of input-output analysis, exponential increases in computing capacity and reductions in the cost of computing have removed many of the practical obstacles to manipulating and inverting even very large matrices, so the computing shortcuts just described [in the text], along with many others, are no longer necessary. . . . In 1939 it reportedly took 56 hours to invert a 42-sector table (on Harvard's Mark II computer; see Leontief, 1951a, p. 20). In 1947, 48 hours were needed to invert a 38-sector input–output matrix. However, by 1953 the same operation took only 45 minutes. (Morgenstern, 1954, p. 496; also, see Lahr and Stevens, 2002, p. 478.) By 1969 a 100-sector matrix could be inverted in between 10 and 36 seconds, depending on the computer used. (Polenske, 1980, p. 15.) Today, inversion of matrices of with thousands of sectors takes only seconds on even desktop computers." (Miller and Blair, 2021, p. 32.)

Again, from the text: "There is no simple way to characterize the combination of features contributing to the historical evolution of computing power. But as a simplistic comparison we note that when the first edition of this text appeared in 1985, a typical microprocessor (those used in early personal computers) could execute on the order of 1.25 million instructions per second (MIPS). This was vastly more computing power than was available in the 1950s and 60s when input-output was emerging as a widely applied tool, and it was even more capable than a mainframe computer in the mid-1970s when the commonly used IBM 370 was often referred to as a "1 MIPS machine. When the second edition of this text appeared in 2009, a typical microprocessor could execute 175,000 MIPS and, at the writing of the current edition, a typical microprocessor includes multiple processor cores that enable multiple "threads" of computations to be executed simultaneously, executing over 300,000 MIPS. And this does not include equally exponentially increasing capacity of computer storage and other features of modern computers. As a result, in the life of this textbook (since the first edition), computing capacity has all but vanished as a significant constraint on applying input-output analysis." (Miller and Blair, 2021, p. 32)

Computer Tools for IOA's Computational Needs

From the interwoven history of computing and of IOA, modern computers and software today meet or exceed most computational needs for IOA. For the most part, general modern mathematical software packages, such as MATLAB, R, STRATA, Scilab, or Mathematica are easily adapted to such needs. Spreadsheet interfaces, such as Microsoft Excel, and associated database software allow orderly editing and management of IOA data. Specialized IOA software packages such as ICIO, IO-Snap, IOT, and IRIOS (included in the references) as well as many others continue to make IOA and its successor frameworks, such as global multiregional input-output models or computational general equilibrium models, accessible and easy to implement. However, for methodological development as well as implementation of such formulations beyond the most basic of familiar calculations, a software gap still remains, at least for easy implementation and experimentation of more complex IOA and related formulations, between specialized IOA packages and generalized scientific computing platforms.

The collection of computer software tools developed here is designed primarily to accompany our textbook, but its modular components can serve more generally to make even complex IOA formulations easier to construct and adapt for experimentation. Because computing capacity has advanced so quickly these tools may also be suitable for even larger IOA implementations, as some exercises in this volume (Chapter 15) will illustrate, such as basic implementation and experimentation with the World Input-Output Database for 43 countries defined as regions along with another region characterizing the rest of the world and 56 industry sectors, comprising nearly a 2,500-sector multiregional input-output table, or experimenting with the 405-sector U.S. input-output tables, both of which are easily and interactively exercised with the tools developed here.

An Overview of APL (A Programming Language) as Applied to IOA

In the history of computer software, few computer languages developed historically have been aligned as well with the original needs of IOA as APL, named after the 1962 book *A Programming Language* by Harvard mathematician Kenneth E. Iverson (Iverson, 1962). In the late 1950s, Iverson began developing a notation for manipulating mathematical arrays that IBM, the dominant computer maker at the time, ultimately used to characterize developing computer systems. What became known as "Iverson notation" used special graphic symbols to represent many frequently-used mathematical functions and operators. The Association for Computing Machinery awarded Iverson the Turing Award, often referred to as the "Nobel Prize of Computing," for this work in 1979.

Iverson notation was first introduced as a commercially-available computer language called APL by IBM in 1967. The language design was faithful to Iverson's original conceptual design as an array-oriented, interactive computer language and it became widely used in scientific applications in the 1970 and 80s. Dialects of the language were developed on many hardware platforms of the time. APL became especially attractive for configuring and solving problems rapidly and accurately, which somewhat by accident aligned well with the extensive methodological development of IOA ongoing at that time as well.

In the 1990s, as spreadsheets, object-oriented programming, and a proliferation of short-lived specific use-oriented languages came and went, APL's fortunes faded amid the many offerings being developed for similar purposes, despite APL's special inherent advantages for application areas such as IOA. Many industry observers attribute the general market decline at the time largely to the specialized character set, limiting its implementation to platforms that could accommodate it, as well as a higher initial learning curve relative to other developing languages of the time and the failure of APL developers to keep pace with many user expectations. Today, however, APL is being rediscovered, especially in application spaces tuned to its inherent strengths. This document illustrates the use of modern APL in IOA calculations, even very complex ones, using the computational workbook exercises of Miller and Blair (2021).

Following a very brief introduction to APL itself below, a number of apps, known as *functions* in APL, for basic IOA calculations are developed and illustrated. Then a broad range of APL functions for IOA are developed and illustrated in the course of exploring the exercise problems. As noted earlier, APL is certainly not necessary to navigate through the computational workbook, but with the functions developed in this document that navigation could be more efficient. APL is a very powerful computer language but, in this volume, it will only be necessary to scratch the surface of the range of features inherent in the language—only those features necessary to carry out IOA calculations, as mathematically complex as some of them turn out to be. Interested readers are invited to select from among the many books and manuals available for general introductions to APL, several of which are summarized in the appendix to this volume. Most notably, Legrand (2009) is a comprehensive introduction to the commonly used modern APL platform, Dyalog APL. The APL code developed in this volume, however, will work on essentially any modern dialect of APL.

Brief Introduction to APL for Input-Output Analysis

As noted earlier, APL is an array-oriented, interactive computer language. The principal data objects in APL are arrays of data known as *variables*. Basic operations on variables can be executed interactively as if they were being completed by a calculator with many powerful basic operations on variables built into the language as so-called *primitive functions* and *operators*. Variables, functions, and operators all exist in the basic organizational structure of the language known as a *workspace*.

A distinctive feature of APL is its use of special characters denoting primitive functions and operators built into the language that are used to construct APL code, which is structured as chains of functions, operators, and variables, known as *expressions*. A complete expression that accomplishes a task is known as a *statement*. To begin, the APL operator known as *assignment*, denoted with the character `←`, is used to define a variable, as in the APL *statement* `A←3 4 5 6 7`, which associates the values of a vector of five integers, 3, 4, 5, 6, and 7, with the named variable `A`. Subsequently, simply typing `A`, like a calculator, returns the contents of the variable `A`:

```
      A←3 4 5 6 7
      A
3 4 5 6 7
```

We can subsequently use defined variables in constructing expressions using primitive functions, such as

```
      1+A
4 5 6 7 8
```

This expression, using a primitive function not surprisingly known as *addition,* denoted by `+`, adds the integer 1 to each element of the array (vector) `A` to yield a new five-element vector of integers, 4, 5, 6, 7, and 8, the result of which can be assigned to a new variable name, `B`, as in

```
      B←1+A
      B
4 5 6 7 8
```

If we request the contents of a variable that is not yet defined, the result is not surprisingly an error, as in

```
      C
VALUE ERROR: Undefined name: C
      C
      ^
```

As a matter of notation for what follows, note that in many instances throughout this document matrices and vectors in mathematical equations, which are generally shown in boldface type, have the same names (characters) as the variable names to which they are assigned in APL. Distinguishing between mathematical matrices/vectors and APL variables with the same names can be confusing, so to help we will distinguish them by their font. That is, when APL variables are defined within the text, they will be shown in the APL font (as illustrated in the expressions above), but if a mathematical vector or matrix is assigned a corresponding APL variable name with exactly the same characters, the variable name will appear in boldface type, e.g., as **A** rather than in the APL font as `A`.

The range of operations built into APL as primitive functions is much broader than most other computer languages. Primitive functions and more advanced operators are the building blocks of APL code. Many APL functions can be expressed with one argument known as *monadic* functions or with two arguments known as *dyadic* functions. For example, the primitive function *reciprocal*, denoted by the character `÷` , in its monadic form, yields the reciprocal of the single argument while the dyadic form, *divide*, yields the result of the left argument divided by the right argument, as in

```
      ÷2
0.5
      10÷2
5
```

As an additional example, using the variables already defined, the element-by-element division of the array (vector, in this case) `A` by the array `B` is

```
      A÷B
0.75 0.8 0.8333333333 0.8571428571 0.875
```

APL expressions are evaluated by the APL interpreter from right to left rather than with an implicit order of operations as in most computer languages (e.g., in most languages, multiplication and division operations are often by default carried out before addition and subtraction operations). In APL the right to left order of operations is, instead, the default, but can be altered by enclosing an expression in parentheses. For example, again using the variables defined so far, the following two expressions yield different results:

```
      A÷B+2
0.5 0.5714285714 0.625 0.6666666667 0.7
      (A÷B)+2
2.75 2.8 2.833333333 2.857142857 2.875
```

So far, we have illustrated how a string of data is created and assigned to a variable. We can also create higher order arrays from a string of data with the dyadic function `⍴` known as *reshape* as in

```
      Z←3 3⍴6 3 21 23 24 12 11 5 17
      Z
 6  3 21
23 24 12
11  5 17
```

The is, of course, a 3×3 matrix, and can be used in operations as we did with vectors earlier, such as

```
      10×Z
 60  30 210
230 240 120
110  50 170
```

The monadic form of `⍴` yields the *shape* of the right argument, as in

```
      ⍴Z
3 3
```

Or, using variables already defined,

```
      ⍴A+B
5
```

Note that in APL a distinction is made between *arrays* (one dimensional vectors, two-dimensional matrices, three dimensional cubes, and so on) and *scalars*, which are single values such as a number like 3.14159 (or a single non-numeric character described later). For example, if we define a variable `Pi←3.14159`, APL interprets it as a scalar with "no shape." That is, if you query the shape of `Pi`, an "empty vector" will be returned. As we have already seen, scalars can be used in operations with arrays of any shape, such as

```
      Pi3←Pi×1 2 3
      Pi3
3.14149 6.28298 9.42447
```

but to use two (or more) arrays in an operation they must be of *conformable* shape. For example, for scalar multiplication (element-by-element multiplication) arrays must be the same shape. For example,

```
      Pi3×10 20 30
31.4149 125.6596 282.7341
      Pi3×10 20
LENGTH ERROR: Mismatched left and right argument shapes
      Pi3×10 20
         ^
```

Most primitive functions and operators in APL operate on scalars, vectors, and higher order arrays in a common manner, with some notable exceptions. For example, one notable exception of particularly important use in IOA is the monadic function known as *matrix inverse*, denoted by the character `⌹`, known as *domino*, which produces the inverse of a square matrix presented as the right argument, as in

```
       ⌹Z
¯0.1914191419  ¯0.0297029703    0.2574257426
 0.1424642464   0.07095709571  ¯0.2260726073
 0.08195819582 ¯0.001650165017 ¯0.04125412541
```

This is the inverse of the square matrix Z defined above. Both the monadic and dyadic versions of this function (the dyadic version is known as *matrix divide*) have many additional powerful features. Note another important feature of this matrix that is distinctive to APL. Some numbers in the matrix include an upper-case negative sign, denoting a negative number, which is distinguished from the normal location of a negative sign that is interpreted in APL as a primitive function. The dyadic form of this function is mathematical *subtraction* of the right argument from the left and the monadic form is *negation*, as in (using variables defined earlier)

```
      B-4
0 1 2 3 4
      -B
¯4 ¯5 ¯6 ¯7 ¯8
```

With only what we have constructed so far, we can write the following intriguing APL statement:

```
      ⌹(I←3 3⍴1,3⍴0)-A←(Z←3 3⍴6 3 21 23 24 12 11 5 17)÷3 3⍴x←100 50 200
1.118124179  0.1556520249 0.1385160222
0.5165790827 2.019549323  0.1917090307
0.1908760305 0.2394280383 1.130500181
```

For IOA it is easy to interpret this APL statement in the following way, once again interpreting the computations from right to left. Assign values to a 3-element vector of total outputs, named **x**, reshape it into a 3×3 matrix with the elements of **x** duplicated in each row. Then define another 3×3 matrix of interindustry transactions and assign it to the variable **Z** (in parentheses overriding the order of execution and using the same values of **Z** as specified above) and divide that matrix, element-by-element, by the reshaped matrix of the vector of total outputs defining each row. In IOA terms, this is the definition of the technical coefficients matrix, **A**, so we assign the result to the variable **A**. Next, we create a string of integers that, when reshaped into a 3×3 matrix, yields the identity matrix, **I**, and assign it to the variable of the same name. Finally, we compute the inverse of the square matrix generated by subtracting, element-by-element, **A** from **I**, which yields the Leontief inverse, $(\mathbf{I}-\mathbf{A})^{-1}$—all in one APL statement. While this certainly illustrates the power of APL's primitive operators as they might be used in IOA, it would not be the general approach to fashioning code to solve IOA problems. Rather, in APL, user-defined functions can be constructed in a variety of ways and used in the same manner as primitive functions to accomplish various IOA tasks, such as $\mathbf{A}=\mathbf{Z}\hat{\mathbf{x}}^{-1}$ or $\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}$.

We next illustrate the use of *user-defined* APL functions for carrying out basic input-output calculations. User-defined functions are assigned names, like variables, and, like primitive functions, can be either monadic with a single argument or dyadic with left and right arguments.[2] User-defined functions can produce an explicit result, as in the case of primitive functions generating a result than can be stored in a variable. So, for example, a dyadic function denoted

[2] The process of creation and editing user-defined functions depends on the APL implementation. One example is described later in this volume (and in further detail in the appendix) for the commonly-used APL system, Dyalog APL.

by the name AMAT to take an interindustry transactions matrix as the left argument, `Z`, and a total outputs vector, x, as the right argument to produce the technical coefficients matrix, `A`, delivered as an explicit result of the user-defined function might look like the following:

```
[0] A←Z AMAT x
[1] A←Z÷(2ρn←ρx)ρx
```

The first line (line `[0]`), known as the function *header*, assigns names to the left and right arguments and to an explicit result. The second line contains the APL statement executing the creation of the matrix of technical coefficients for the variables **Z** and **x**, appearing the function's arguments and producing the explicit result, as in

```
      Z←3 3ρ6 3 21 23 24 12 11 5 17
      x←100 50 200
      Z AMAT x
0.06 0.06 0.105
0.23 0.48 0.06
0.11 0.1  0.085
```

For later use, we can save the explicit result in a defined variable as

```
      A←Z AMAT x
      A
0.06 0.06 0.105
0.23 0.48 0.06
0.11 0.1  0.085
```

As variables and user-defined functions accumulate in an APL workspace, a number of so-called *system commands* can be used to keep track and for other workspace management chores. System commands are all distinguished by beginning with a right parenthesis. For example, the system command for listing the user-defined functions in the workspace is

```
       )fns
AMAT
```

Similarly, to list the variables defined in the current workspace, the system command is

```
       )vars
A       B       I       Z       x
```

APL variables defined in the workspace—those recognized by the system command `)vars`— are called *global variables*. A useful feature of user-defined functions is the use of so-called *local variables* which are variables that are used only within the function while it is being executed. For example, the variable n in the function `AMAT` is defined within the function as the shape of **x** (the number of elements in the vector **x**) but, if designated as a local variable (or as an argument), will only be recognized and used during execution of the function `AMAT` as defined in the function itself. In APL, local variables are designated as a list of names separated from the rest of the function header by a semicolon, as in

```
[0] A←Z AMAT x;n
[1] A←Z÷(2ρn←ρx)ρx
```

Note that in a user-defined function if a global variable with the same name as that of a local variable exists, the function will recognize only the local variable during its execution which will not affect the global variable.

We now illustrate several additional APL features that we will find essential in using APL for IOA. First, we can refer to specific elements of an array by specifying its position or so-called *index* within square brackets. For example, to retrieve the first element of the vector `x`, or its second and third elements we can write

```
      x←100 50 20
      x[1]
100
      x[2 3]
50 20
```

For matrices or higher order arrays the index references for each dimension are separated by a semicolon, as in for the A defined above, the upper left element, the second and third elements of the second row, or the third row, respectively, are identified by

```
      A[1;1]
0.06
      A[2; 2 3]
0.48 0.6
      A[3;]
0.11 0.1 0.85
```

To combine two arrays into one we can use the dyadic APL operator known as *catenate*, designated with a comma, which takes the left argument and combines it with the right argument, as in

```
      x←100 50 200
      y←300 400 500
      x,y
100 50 200 300 400 500
```

Hence, the length (shape) of the result is 6 since each of arguments to be combined are of length 3. The monadic form of catenate operator, known as *ravel*, converts an array of any size or shape to string of the elements used to construct the array, specified as a vector. For example,

```
      A
0.06 0.06 0.105
0.23 0.48 0.06
0.11 0.1  0.085

      ,A
0.06 0.06 0.105 0.23 0.48 0.06 0.11 0.1 0.085
```

We can extract an element from an array either by specifying its index, as above, or by specifying its relative position by using a pair of primitive functions, *take* and *drop*, designated by the characters `↑` and `↓`. For example, we can take the first two elements of the vector x by the following expression

```
      2↑x
100 50
```

or retrieve the last two elements by

```
      ¯2↑x
50 200
```

or retrieve the upper left 2×2 partition of the matrix A as

```
       2 2↑A
0.06 0.06
0.23 0.48
```

With these added primitive functions and operators, we can easily assemble a monadic user-defined function to produce the Leontief inverse for an arbitrary matrix of technical coefficients defined as the right argument with the following:

```
[0] L←LINV A;n;I
[1] I←(2ρn)ρ1,(n←1↑ρA)ρ0
[2] L←⌹I-A
```

Hence, the APL expressions for computing the matrices of technical coefficients and of total requirements (Leontief inverse) can be defined in a more orderly fashion with user-defined APL functions as

```
      Z
 6  3 21
23 24 12
11  5 17
      x
100 50 200
      L←LINV A←Z AMAT x
      A
0.06 0.06 0.105
0.23 0.48 0.06
0.11 0.1  0.085
      L
1.1181242  0.15565202 0.13851602
0.51657908 2.0195493  0.19170903
0.19087603 0.23942804 1.1305002
```

Different implementations of APL have somewhat different features and capabilities for transferring data into and out of workspaces, but most of the primitive operators are the same. In this volume we provide user-defined functions and illustrate additional primitive functions as needed for most of the common tasks in IOA successively as we work though the chapter exercises included in the Workbook. In a final chapter of this manual, we illustrate use of the same functions in several larger sized real-world applications.

The collection of functions presented here were originally developed in a variety of APL systems. The versions presented are intended to work in essentially any APL system available today. Modern APL systems have many additional features as well. To experiment with APL, the appendix to this volume includes information on the installation and use of some distinctive features of a powerful and well-supported APL system known as Dyalog APL that is currently available without charge for personal and non-commercial use at http://www.dyalog.com. Other APL systems, however, are just as suitable for the functions developed in this volume.

For the balance of this volume, we will use the computational exercises assembled for the workbook accompanying the text, as a means to explore IOA as well as well as to continue to develop features of APL useful to facilitating that exploration. Modern implementations of APL, such as Dyalog APL above, have many more powerful features that are worth exploring but, in this volume, we develop and use only features to facilitate explore IOA and that are common on essentially all versions of APL available today.

Chapter 2, Foundations of Input–Output Analysis

Chapter 2 introduces Leontief's conceptual input–output framework and explains how to develop the fundamental mathematical relationships from the interindustry transactions table. The key assumptions associated with the basic Leontief model and implications of those assumptions are recounted and the economic interpretation of the basic framework is explored. The basic framework is illustrated with a highly aggregated model of the US economy. In addition, the "price model" formulation of the input–output framework is introduced to explore the role of prices in input–output models. Appendices to this chapter include a fundamental set of mathematical conditions for input–output models, known as the Hawkins–Simon conditions. The exercise problems for this chapter explore applications of the basic mathematical relationships of input-output analysis.

Problem 2.1: Basic Input-Output Relationships

This problem explores the relationships of the fundamental input-output analysis identities developed in chapter 2: $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$ and $\mathbf{x} = \mathbf{Ax} + \mathbf{f}$ where $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$.

Problem 2.1 Overview

Consider a two-sector economy (agriculture and manufacturing), the basic data for which are the matrix of interindustry transactions, **Z**, and vector of total outputs, **x**, expressed in dollar values, specified as:

$$\mathbf{Z} = \begin{bmatrix} 500 & 350 \\ 320 & 360 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 1,000 \\ 800 \end{bmatrix}$$

Rearranging terms in the first input-output identity, $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$, to $\mathbf{f} = \mathbf{x} - \mathbf{Zi}$ makes it easy to calculate the vector of final demands, **f**, for this economy as

$$\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 1,000 \\ 800 \end{bmatrix} - \begin{bmatrix} 500 & 350 \\ 320 & 360 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1,000 \\ 800 \end{bmatrix} - \begin{bmatrix} 850 \\ 680 \end{bmatrix} = \begin{bmatrix} 150 \\ 120 \end{bmatrix}$$

To illustrate the process of impact analysis, i.e., computing the impact on industrial production in the economy resulting from a new final demands presented to the economy, we specify new final demands as f_1 increased by \$50 and f_2 decreased by \$20, so that the vector of new final demands is $\mathbf{f}^{new} = \begin{bmatrix} 200 \\ 100 \end{bmatrix}$. To determine the production of total output for each sector in this economy necessary to support these new levels of final demand, we first invoke the basic Leontief model assumptions defining the matrix of technical coefficients or direct requirements:

$$\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} 500 & 350 \\ 320 & 360 \end{bmatrix} \begin{bmatrix} 1/1000 & 0 \\ 0 & 1/800 \end{bmatrix} = \begin{bmatrix} .5 & .4375 \\ .32 & .45 \end{bmatrix}$$

We can compute a "round-by-round" approximation of the impacts of the new final demands on this economy to intuitively illustrate the effect of these new final demands on total industrial production throughout the economy by computing, first, the direct requirements to satisfy the new final demand vector added to the final demands themselves, $\mathbf{f}^{new} + \mathbf{A}\mathbf{f}^{new}$, then added the production necessary to supply that first "round" of direct requirements, $\mathbf{A}(\mathbf{A}\mathbf{f}^{new})$, and so on. Mathematically, as discussed in chapter 2, this is expressed as the infinite power series $\mathbf{x}^{new} = \mathbf{f}^{new} + \mathbf{A}\mathbf{f}^{new} + \mathbf{A}(\mathbf{A}\mathbf{f}^{new}) + \cdots = (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots)\mathbf{f}^{new}$. This infinite power series ultimately converges to the total amount of production, required directly and indirectly through the successive rounds of intermediate industrial production, to support the new final demands.

Terminating the power series is an approximation of the "exact" values found by rearranging $\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{f}$ as $\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{f}$ or $\mathbf{x} = \mathbf{L}\mathbf{f}$ where $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^n$ with increasing precision of the approximation as n increases. The matrix of total requirements, $\mathbf{L}$, is often referred to as the Leontief inverse.

For this economy, computing the "round by round" requirements for the first five terms yields only a rough approximation of the total outputs in the economy necessary to satisfy the new final demands: $\tilde{\mathbf{x}}^{new} = (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^4)\mathbf{f}^{new} = \begin{bmatrix} 650.81 \\ 453.98 \end{bmatrix}$, compared with the "exact" value, $\mathbf{x}^{new} = \mathbf{L}\mathbf{f}^{new} = \begin{bmatrix} 1{,}138.90 \\ 844.40 \end{bmatrix}$ where $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 4.07 & 3.24 \\ 2.37 & 3.7 \end{bmatrix}$. It is a rough approximation because, in this particular case, the power series converges very slowly, e.g., for $n = 25$, the approximation is $\tilde{\mathbf{x}}^{new} = (\mathbf{I} + \mathbf{A} + \cdots + \mathbf{A}^{25})\mathbf{f}^{new} = \begin{bmatrix} 1{,}122.80 \\ 831.60 \end{bmatrix}$, compared again with the "exact" value, $\mathbf{x}^{new} = \begin{bmatrix} 1{,}138.90 \\ 844.40 \end{bmatrix}$, and it is not until $n = 57$ that $\tilde{\mathbf{x}}^{new} = \mathbf{x}^{new}$ for both elements within 0.1. This feature of slow convergence, however, is not always the case depending upon the characteristics of $\mathbf{A}$. For example, if $\overline{\mathbf{A}} = .01 \times \mathbf{A}$, for the same vector of final demands, convergence, i.e., when $\tilde{\mathbf{x}}^{new} = \mathbf{x}^{new}$ for both elements within 0.1, occurs at $n = 6$. This result is analogous to the result in ordinary algebra, $1/(1-a) = 1 + a + a^2 + a^3 + \cdots + a^n$ for a scalar a where $|a| < 1$. For example, if $a = .427$, this series converges to within 0.001 at $n = 8$ while, for .0427, i.e., $.01a$, the series converges at $n = 2$.

Computational Notes

In order to solve this problem with APL, we need to introduce a number of additional features of APL programming. The first is a new monadic primitive function, designated with the character `/`, known as *reduction*. Reduction specifies an operation (often referred to as the operand) and produces a derived, combined function that is then applied to an array supplied as the argument. For example, the APL expression `+/x` takes the dyadic function addition and applies it repeatedly to the elements of the array (vector in this case) x, presented as the right argument.

So, “plus reduction” of a three-element vector x is simply the sum of all the elements of x, which can be found as `x[1]+x[2]+x[3]`, but much more simply as `+/x`. For example,

```
      x←10 20 30
      x[1]+x[2]+x[3]
60
      +/x
60
```

For higher order arrays, reduction applies to a specific dimension. For example, with matrices, the rows are considered the first dimension and the columns the second. By default, the reduction operator assumes the last dimension, so that for a matrix `Q`, the default operation would result in the rows sums of `Q`, as in

```
      Q
1 2 3
4 5 6
7 8 9
      +/Q
6 15 24
```

To specify the dimension (overriding the default), the index of the dimension follows the operator before specifying the argument, so, for example, to compute the column sums of `Q`, the expression would be

```
    +/[1]Q
12 15 18
```

and, of course, the default expression could be expressed equivalently as

```
    +/[2]Q
6 15 24
```

Just as there is a default notation for reduction along the *last* dimension of an array, invoked for “plus reduction” by `+/`, there is also a default notation for reduction along the *first* dimension, denoted by the character `⌿`, known as *reduce first.* We invoke “plus reduction” by `+⌿`, which for the matrix `Q` is

```
      +⌿Q
12 15 18
```

The reduction operator can be used with many other primitive functions as well. For example, “times reduction,” or the successive multiplicative product of all elements of a vector is found as

```
      x←10 20 30
      x[1]×x[2]×x[3]
6000
      ×/x
6000
```

Also useful is the dyadic form denoted by the reduction function character, known as *compression*, which takes as the left argument a vector of zeroes or ones (referred to as a *logical* array) specifying which elements of a vector specified as the right argument are to be returned as an explicit result, as in

```
      1 0 1 0/1 2 3 4
1 3
```

If compression retrieves none of the elements of the right argument the function returns an empty or null vector as the explicit result, i.e., as noted earlier, a vector with no elements which has shape 0, as in

```
          ⍴1 2 3 4
4
      ⍴1 0 1 0/1 2 3 4
2
      ⍴0 0 0 0/1 2 3 4
0
```

In IOA, among the most common mathematical operations necessary is matrix multiplication. For example, consider the following two arrays:

```
      Q
1 2 3
4 5 6
7 8 9
      R
1 1 1
1 0 1
0 0 1
```

The matrix product, **QR**, could be accomplished by computing the result for each element of the matrix product. Traditionally, and in most computer languages, this would mean specifying each element in **QR**, such as, for the upper left element of **QR**, as the sum of the element-by-element multiplication of the first row of **Q** by the first column of **R**, and so on for every element of **QR** as in

```
      +/Q[1;]×R[;1]
3
```

The operation could be replicated for each the 9 combinations of rows and columns of **Q** and **R** to specify the corresponding elements of **QR**, which would be the process in most computer languages. In APL, this process can be accomplished much more efficiently by the operator *inner product* which combines two primitive functions operating on two arrays specified as left and right arguments. The symbols for the two primitive functions employed are separated by a period. For example, matrix multiplication of **Q** and **R**, using the inner product would be

```
      Q+.×R
 3 1  6
 9 4 15
15 7 24
```

Finally, another APL feature necessary for solving this problem is the mechanism for branching and looping in APL functions. While many if not most tasks can be accomplished in APL by means of operations on arrays, sometimes a sequence of operations in a function must be altered and the order of execution of statements directed by branching and looping, as is common in most computer languages. When necessary, to accomplish this in APL, a unique symbol is the "branch to" character `→`. In many modern versions of APL this character can be replaced with the equivalent clause, `:GoTo.` The argument specified to the right of the "branch to" character is a *label* designating the destination of the next APL statement in a user-defined function to be interpreted. The label is placed at the beginning of the statement and followed by a colon.

The right argument of a "branch to" statement can be and often is another APL expression. If the result of that expression is a null vector (as defined earlier) then control transfers to the line following the "branch to" statement in the function.

The "branch to" statement can be executed conditionally as well, i.e., going to a labeled location (statement) in an APL function when a particular condition is satisfied, which can be accomplished with a family of primitive *logical operators* that compare arrays specified as arguments. For example, using the logical operator *equals*, designated not surprisingly by the character `=`, we can query whether or not the left argument is equal to the right argument, the result of which is a 1 if the condition is true and a 0 if the condition is false, as in

```
      3=3
1
      3=5
0
      3=1 2 3 4 5
0 0 1 0 0
```

We can use a logical expression to specify the condition for conditional branching in a function. For example, consider the following expression:

```
→(i=4)/L1
```

This specifies that if the value of `i` is 4 then control will transfer to the statement located at label `L1`. If the value of `i` is not 4 then the "branch to" statement will transfer control by default to the next line in the function, since the result of the compression operation is an empty vector. As an aside, if the result of an APL expression appears as the operand of "branch to" statement is a vector, only the first element of the vector will be used.

So, for this problem, collecting these new primitive functions and operators described so far, consider the following APL function named `RINV` to calculate the round-by-round approximation of the Leontief inverse as $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^n$:

```
[0] LN←n RINV A;i;AN;m
[1] LN←(AN←A)+(2⍴m)⍴1,(m←1↑⍴A)⍴0
[2] i←2
[3] L1:LN←LN+AN←AN+.×A
[4] →(n≥i←i+1)/L1
```

In this function, the number of terms to be calculated in the power series approximation of the Leontief inverse, n, is specified as the left argument and the matrix of technical coefficients is the right argument **A**. The explicit result returned by the function is the nth order approximation of the Leontief inverse, defined as `LN`. Note how this function operates. First, `LN` is initialized as $\mathbf{I}+\mathbf{A}$ (a first order approximation of **L**) by first determining the number of rows or columns of the **A**, found by retrieving the first element of the shape of **A** and storing it as the local variable `m`. The value of m is used to specify an identity matrix of the appropriate size ($m \times m$) and then adding it to **A**, the result of which is also stored as the "current" value of $\mathbf{I}+\mathbf{A}+\mathbf{A}^2+\cdots+\mathbf{A}^n$, named `AN`.

We then specify that we will be calculating the 2nd order approximation of the power series by defining the local variable i and assigning it the value of 2. Then we calculate the value of $\mathbf{A}^2$ by matrix multiplication of `AN` (the initial value of which was A) and adding the result to first order approximation of the power series $\mathbf{I}+\mathbf{A}$ (saved initially as the variable `LN`) and then saving the result ovewriting the previous value of `LN`, which now becomes the 2nd order approximation of the power series, $\mathbf{I}+\mathbf{A}+\mathbf{A}^2$.

Specifying `LN` in this way allows us to iteratively add successive terms of $\mathbf{A}^n$ to the result until we reach the specified number of terms of the approximation, n. The APL statement to be successively computed is designated with the label `L1` and the "branch to" statement defines the number of times the statement is executed. That is, for each iteration, the value of `i` is incremented by 1 and tested to see if it is equal to or less than the desired number of successive terms n. If so, then the control is transferred again to the operative statement at `L1` and, if not, the control is transfer to the next line of the function which does not exist, concluding execution of the function and returning the current value of `AN` as the explicit result.

To illustrate the use of the function `RINV`, consider the following, by now familiar matrix of technical coefficients

```
      A
0.06 0.06 0.105
0.23 0.48 0.06
0.11 0.1  0.085
```

We compute the exact value of the Leontief inverse with the function, LINV, developed earlier:

```
      LINV A
1.118124179  0.1556520249 0.1385160222
0.5165790827 2.019549323  0.1917090307
0.1908760305 0.2394280383 1.130500181
```

We can compute successive approximations of the Leontief inverse as

```
      2 RINV A
1.08895 0.1029 0.123825
0.3608  1.7302 0.11805
0.14895 0.1631 1.109775
      3 RINV A
```

```
1.10262475 0.1271115 0.131038875
0.4325795  1.863949  0.15173025
0.16852525 0.1982025 1.119756625
      4 RINV A
1.109807406  0.1402748925 0.1345405931
0.4713533675 1.935823315  0.1701548587
0.1788713188 0.2172243775 1.124766614
```

We can compare the successive approximations with the exact value to determine the accuracy of the approximations.

Finally, another pair of primitive functions worth introducing here use to help with solving this problem are designated by the characters *ceiling* (`⌈`) and *floor* (`⌊`). In their dyadic forms these primitive functions compare the left and right arguments and return as the result which is the larger or smaller of the two arguments, respectively. So, for example

```
      10⌊20
10
      10⌈20
20
      10⌈1 5 15 25
10 10 15 25
```

If we use the ceiling function in a reduction operation, the result is the largest element of a vector presented as the argument and the floor reduction operation yields the smallest element of the vector, as in

```
      x←1 5 15 25
      ⌈/x
25
      ⌊/x
1
```

We can use ceiling reduction to compare successive approximations of **L** using `RINV` with the exact value of **L**. To do this we can compute the largest of the element-by-element differences between the successive approximation of **L** and **L** itself. In this case we find that at $n = 12$ all elements of the `RINV` approximation of **L** are within .001 of the exact value of **L**.

```
    .001≥⌈/⌈/(LINV A)-11 RINV A
0
     001≥⌈/⌈/(LINV A)-12 RINV A
1
    12 RINV A
1.118065703  0.155543786  0.1384881424
0.5162608026 2.01896019   0.191557284
0.1907916027 0.2392717631 1.130459928
```

We now have assembled APL tools sufficient to solve Problem 2.1. The first question is to compute the vector of final demands given the specified vector of total outputs and matrix if interindustry transactions:

```
      f←(x←1000 800)-+/Z←2 2⍴500 350 320 360
      f
150 120
```

Next the task is to compute the new production in each sector generated by changes in final demand involving an increase in sector 1 by 50 and a decrease in sector 2 by 20.

```
      Δx←(L←LINV A←Z AMAT x)+.×Δf←f+50 ¯20
      Δx
1138.888889 844.4444444
```

Next the problem is to produce an approximation to this answer by using the first five terms in the power series approximation.

```
      (5 RINV A)+.×f+50 ¯10
741.7648711 537.8563969
```

Finally, we already computed the exact answer using the actual Leontief inverse above.

```
      (LINV A)+.×f+50 ¯20
1138.888889 844.4444444
```

Problem 2.2: Basic Input-Output Relationships Expanded

This problem explores a more extensive example of basic input-output relationships.

Problems 2.2 Overview

We specify interindustry sales and industry total outputs in a three-sector national economy for year *t*, given in the following table, where values are shown in thousands of dollars. (S_1, S_2, and S_3 designate the three industry sectors).

	Interindustry Sales S_1	S_2	S_3	Total Output
S_1	350	0	0	1,000
S_2	50	250	150	500
S_3	200	150	550	1,000

From the table, the matrix of interindustry transactions, $\mathbf{Z}^t$, and the vector of total outputs, $\mathbf{x}^t$, are defined as $\mathbf{Z}^t = \begin{bmatrix} 350 & 0 & 0 \\ 50 & 250 & 150 \\ 200 & 150 & 550 \end{bmatrix}$, and $\mathbf{x}^t = \begin{bmatrix} 1,000 \\ 500 \\ 1,000 \end{bmatrix}$. The matrix of technical coefficients for year *t*, $\mathbf{A}^t$, and the corresponding matrix of total requirements, $\mathbf{L}^t$, are then found as

$$\mathbf{A}^t = \mathbf{Z}^t(\hat{\mathbf{x}}^t)^{-1} = \begin{bmatrix} .35 & 0 & 0 \\ .05 & .5 & .15 \\ .2 & .3 & .55 \end{bmatrix}, \text{ and } \mathbf{L}^t = (\mathbf{I} - \mathbf{A}^t)^{-1} = \begin{bmatrix} 1.538 & 0 & 0 \\ .449 & 2.5 & .833 \\ .983 & 1.667 & 2.778 \end{bmatrix}.$$

Suppose that government tax policy changes generate final demands for the products delivered by sectors 1, 2, and 3 projected for next year (year $t + 1$) to be 1,300, 100, and 200 for the three sectors, respectively (also measured in thousands of dollars). The corresponding total outputs that would be necessary from the three sectors to meet this projected new demand, assuming that there is no change in the technological structure of the economy (that is, assuming that the **A** matrix does not change from year t to year $t+1$), would be $\mathbf{x}^{t+1} = \mathbf{L}^t\mathbf{f}^{t+1} = \begin{bmatrix} 2{,}000 \\ 1{,}000 \\ 2{,}000 \end{bmatrix}$ for

$\mathbf{f}^{t+1} = \begin{bmatrix} 1{,}300 \\ 100 \\ 200 \end{bmatrix}$. The original vector of final demands for year t is computed as

$\mathbf{f}^t = \mathbf{x}^t - \mathbf{Z}^t\mathbf{i} = \begin{bmatrix} 650 \\ 50 \\ 100 \end{bmatrix}$, from which we can observe that $\mathbf{f}^{t+1} = 2\mathbf{f}^t$, so it can be easily verified that

that $\mathbf{x}^{t+1} = 2\mathbf{x}^t$ since $\mathbf{x}^{t+1} = \mathbf{L}^t\mathbf{f}^{t+1} = 2\mathbf{L}^t\mathbf{f}^t = 2\mathbf{x}^t$, illustrating the linearity of the Leontief model.

Computational Notes

We have all the APL tools we need to solve this problem.

```
      Z←3 3⍴350 0 0 50 250 150 200 150 550
      x←1000 500 1000
      L←LINV A←Z AMAT x
      A
0.35 0    0
0.05 0.5 0.15
0.2  0.3 0.55
      L
1.538461538  0           0
0.4487179487 2.5         0.8333333333
0.9829059829 1.666666667 2.777777778
```

For the new vector of final demands, we compute the corresponding vector of total outputs as

```
      L+.×1300 100 200
2000 1000 2000
```

The original vector of final demands is found by

```
      f←x-+/Z
      f
650 50 100
```

Problem 2.3: Open and Closed Leontief Models

This problem illustrates the distinctions between the open and closed Leontief models.

Problem 2.3 Overview

Using the data of problem 2.1, the interindustry transactions matrix and vector of total outputs, respectively, were defined as $\mathbf{Z} = \begin{bmatrix} 500 & 350 \\ 320 & 360 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 1{,}000 \\ 800 \end{bmatrix}$.

Suppose that the part of the original final demands attributable to household (consumption) expenditures for this economy are \$90 from sector 1 and \$50 from sector 2 with the remaining parts of final demand reported as exports of products 1 and 2. Suppose, further, that (1) payments from sectors 1 and 2 for household labor services were \$100 and \$60, respectively; (2) that total household (labor) income in the economy was \$300; (3) that household purchases of labor services was \$40; and (4) that any new final demands presented to the economy are for exports.

This additional information allows us to expand **Z** and **x** of the basic data for two-sector model from Problem 2.1, to be $\mathbf{Z}^c = \begin{bmatrix} 500 & 350 & 90 \\ 320 & 360 & 50 \\ 100 & 60 & 40 \end{bmatrix}$ and $\mathbf{x}^c = \begin{bmatrix} 1{,}000 \\ 800 \\ 300 \end{bmatrix}$. This illustrates the process known as closing the model to households. The result is a three-sector representation of the economy for which the matrices of direct and total requirements, respectively, are

$$\mathbf{A}^c = \mathbf{Z}^c(\hat{\mathbf{x}}^c)^{-1} = \begin{bmatrix} .5 & .438 & .3 \\ .32 & .45 & .167 \\ .1 & .075 & .133 \end{bmatrix} \text{ and } \mathbf{L}^c = (\mathbf{I} - \mathbf{A}^c)^{-1} = \begin{bmatrix} 5.820 & 5.036 & 2.983 \\ 3.686 & 5.057 & 2.248 \\ 0.990 & 1.019 & 1.693 \end{bmatrix}.$$

We can now find the impacts in terms of required new production for sectors 1 and 2 of the new final demands specified in Problem 2.1, but this time using the Leontief inverse for the new, expanded matrix of technical coefficients of dimension 3×3. The vector of new final demands (now attributed solely to exports) is $\tilde{\mathbf{f}}^c = \begin{bmatrix} 200 \\ 100 \\ 0 \end{bmatrix}$, and we compute the resulting new vector of total outputs necessary to support those final demands as $\tilde{\mathbf{x}}^c = \mathbf{L}^c\tilde{\mathbf{f}}^c = \begin{bmatrix} 1{,}667.5 \\ 1{,}242.9 \\ 300 \end{bmatrix}$. Since in Problem 2.1 we found $\mathbf{x}^o = \mathbf{L}^o\mathbf{f}^o = \begin{bmatrix} 1{,}138.9 \\ 844.4 \end{bmatrix}$ for $\mathbf{f}^0 = \begin{bmatrix} 200 \\ 100 \end{bmatrix}$, the increases in outputs for both sectors 1 and 2 using the closed model reflect increased interindustry production resulting from the inclusion of households as an endogenous sector in the 3-sector model.

Computational Notes

We have all the APL tools to solve this problem. First, compute **A** and **L**, which in APL we define as `A2` and `L2`, respectively.

```
      Z2←3 3⍴500 350 90 320 360 50 100 60 40
      x2←1000 800 300
      L2←LINV A2←Z2 AMAT x2
      A2
0.5  0.4375 0.3
0.32 0.45   0.1666666667
```

```
0.1  0.075  0.1333333333
      L2
5.819663567  5.03604639  2.982969387
3.6861352    5.056942848 2.248458886
0.9904921116 1.01870233  1.692613102
```

The vector of total outputs for the new vector of final demands is then

```
      L2+.×200 100 0
1667.537352 1242.921325 299.9686553
```

Problem 2.4: The Hawkins-Simon Conditions

This problem explores the Hawkins-Simon conditions for the Leontief model developed in chapter 2.

Problem 2.4 Overview

Consider an economy organized into three industries: (1) lumber, (2) machinery, and (3) paper characterized by the following:

- A consulting firm estimates that last year the lumber industry had an output valued at \$50 (assume all monetary values are in units of \$100,000), 5 percent of which the industry consumed itself; 70 percent of the lumber industry's output was consumed by final demand; 20 percent by the paper industry; and 5 percent by the machinery industry.
- The machinery industry consumed 15 percent of its own products, out of a total of \$l00; 25 percent went to final demand; 30 percent to the lumber industry; 30 percent to the paper industry.
- Finally, the paper industry produced \$50, of which it consumed 10 percent; 80 percent went to final demand; 5 percent went to the lumber industry; and 5 percent to the machinery industry.

Using this information the matrix of interindustry transactions and the vector of total outputs for this economy are $\mathbf{Z} = \begin{bmatrix} 2.5 & 10 & 2.5 \\ 2.5 & 5 & 2.5 \\ 30 & 30 & 15 \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} 35 \\ 40 \\ 25 \end{bmatrix}$, respectively, so the vector of total outputs, $\mathbf{x}$, and the matrix of technical coefficients, $\mathbf{A}$, are then $\mathbf{f} = \begin{bmatrix} 35 \\ 40 \\ 25 \end{bmatrix}$,

$\mathbf{x} = \mathbf{Zi} + \mathbf{f} = \begin{bmatrix} 50 \\ 50 \\ 100 \end{bmatrix}$ and $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .05 & .2 & .025 \\ .05 & .1 & .025 \\ .6 & .6 & .15 \end{bmatrix}$. The Hawkins-Simon conditions require positivity of all principal minors of $(\mathbf{I} - \mathbf{A}) = \begin{bmatrix} .95 & -.2 & -.025 \\ -.05 & .9 & -.025 \\ -.6 & -.6 & .85 \end{bmatrix}$. Here the three first-order principal minors are the main diagonal elements, 0.95, 0.9 and 0.85; the three second-order principal minors are 0.845, 0.75 and 0.793, and the third-order principal minor is just the determinant $|\mathbf{I}-\mathbf{A}| = 0.687$, so all the principal minors are positive (see Appendix A of the text for discussion of minors in matrix operations).

The Leontief inverse for this economy is $\mathbf{L} = (\mathbf{I}-\mathbf{A})^{-1} = \begin{bmatrix} 1.092 & .269 & .04 \\ .084 & 1.154 & .036 \\ .830 & 1.005 & 1.23 \end{bmatrix}$. If we anticipate an economic recession reflected in decreased final demands for lumber, machinery, and paper of 25, 10, and 5 percent, respectively. The vector of new final demands is then $\mathbf{f}^{new} = \begin{bmatrix} (.75)f_1 \\ (.90)f_2 \\ (.95)f_3 \end{bmatrix} = \begin{bmatrix} 26.25 \\ 36.00 \\ 23.75 \end{bmatrix}$ and the corresponding vector of total outputs supporting this change in final demand is found by $\mathbf{x}^{new} = (\mathbf{I}-\mathbf{A})^{-1}\mathbf{f}^{new} = \begin{bmatrix} 39.317 \\ 44.606 \\ 87.181 \end{bmatrix}$ for $\mathbf{f}^{new} = \begin{bmatrix} (.75)f_1 \\ (.90)f_2 \\ (.95)f_3 \end{bmatrix} = \begin{bmatrix} 26.25 \\ 36.00 \\ 23.75 \end{bmatrix}$. The new matrix of interindustry transactions is $\mathbf{Z}^{new} = \mathbf{A}(\hat{\mathbf{x}}^{new}) = \begin{bmatrix} 1.97 & 8.92 & 2.18 \\ 1.97 & 4.46 & 2.18 \\ 23.59 & 26.76 & 13.08 \end{bmatrix}$, so the vectors of value-added inputs and of intermediate outputs, respectively, are then computed as $\mathbf{v}^{new} = (\mathbf{x}^{new})' - \mathbf{i}'(\mathbf{Z}^{new}) = \begin{bmatrix} 11.795 & 4.461 & 69.744 \end{bmatrix}$ and $\mathbf{u}^{new} = (\mathbf{Z}^{new})\mathbf{i} = \begin{bmatrix} 13.067 \\ 8.606 \\ 63.431 \end{bmatrix}$.

Computational Notes

We can compute the Hawkins Simon conditions using the method of cofactors by first finding the determinant of the matrix $(\mathbf{I}-\mathbf{A})$ but it is a tedious calculation and we won't show all the steps here. Instead, included in the appendix to this volume, is a monadic user-defined function `DETER` which returns the determinant (if one exists) of a square matrix presented as the argument.

Another useful primitive APL function for this problem is *transpose* designated by the character `⍉`. The monadic form of transpose for a matrix simply interchanges the rows and columns as in

```
      Q←2 3⍴1 2 3 4 5 6
      Q
1 2 3
4 5 6
      ⍉Q
1 4
2 5
3 6
```

The dyadic version specifies the order of transposition of the dimension (mostly applicable to arrays of 3 dimensions or more) but as you will see below of other use as well. For example, with a matrix, the expression `2 1⍉Q` is equivalent to the expression `⍉Q` and the expression `1 2⍉Q` is equivalent to `Q`, as if the original matrix was not transposed at all.

```
      2 1⍉Q
1 4
2 5
3 6
      1 2⍉Q
1 2 3
4 5 6
```

For matrices this isn't a very interesting result (it is for higher order arrays) but it does provide a mechanism to retrieve the elements of the principal diagonal of a matrix. For example, consider the matrix **R** defined by

```
      R←3 3⍴ 1 2 3 4 5 6 7 8 9
      R
1 2 3
4 5 6
7 8 9
```

The principal diagonal of **R** can easily be found by the dyadic transpose as

```
      1 1⍉R
1 5 9
```

Another helpful APL feature for this problem is the dyadic operator *compression* as applied to matrices. If the right-hand argument is a matrix, then the left-hand argument specifies the columns to be extracted. For example, if we wish to extract the first two columns from the matrix R, the APL expression would be

```
      1 1 0/R
1 2
4 5
7 8
```

To compress along the rows, the character `/` is replaced by `⌿`, so to extract the first two rows of R the APL expression would be

```
      1 1 0⌿R
1 2 3
4 5 6
```

These operations can be combined as well, so, for example, we can extract the upper left 2×2 submatrix of **R** (equivalent to `2 2↑R`) with the successive matrix compression operations

```
      1 1 0⌿1 1 0/R
1 2
4 5
```

This provides a convenient method for computing cofactor matrices in testing the Hawkins Simon conditions. For this problem to test the Hawkins Simon conditions, we first compute the determinant of $(\mathbf{I}-\mathbf{A})$ by

```
     Z←3 3⍴2.5 10 2.5 2.5 5 2.5 30 30 15
     f←35 40 25
     x←50 50 100
     A←Z AMAT x
     A
0.05 0.2 0.025
0.05 0.1 0.025
0.6  0.6 0.15
     DET←DETER (3 3⍴1,3⍴0)-A
     DET
0.68675
```

Then, the principal minors of $(\mathbf{I}-\mathbf{A})$ are just the diagonal elements, as in

```
      I←3 3⍴1,3⍴0
      IA←I-A
      IA
 0.95 ¯0.2 ¯0.025
¯0.05  0.9 ¯0.025
¯0.6  ¯0.6  0.85
       1 1⍉IA
0.95 0.9 0.85
```

And the second order principal minors are found by using the compression operator as

```
     DETER 1 1 0⌿1 1 0/IA
0.845
     DETER 1 0 1⌿1 0 1/IA
0.7925
     DETER 0 1 1⌿0 1 1/IA
0.75
```

The Leontief inverse for this problem is found by

```
      L←LINV A
      L
1.0921005   0.26938478 0.040043684
0.083727703 1.1539862  0.036403349
0.82999636  1.0047324  1.2304332
```

Hence, the total outputs necessary to support the new vector of final demands is found by

```
      x2←L+.×f2←f-f×.25 .1 .05
      f2
26.25 36 23.75
      x2
39.316527 44.605934 87.180561
```

Finally, it is convenient to use several functions we have developed already to create a function that diagonalizes a vector, i.e., places the elements of a vector along the main diagonal of a matrix with all the of non-diagonal elements set to zero. We can easily do this with an APL expression that element-by-element multiplies an appropriately size identity matrix by a matrix composed of the elements of the vector to be diagonalized repeated in each row. For example,

```
      x←50 50 100
      x
50 50 100
      (3 3ρ1,3ρ0)× 3 3ρx
50  0   0
 0 50   0
 0  0 100
```

We can generalize this expression with a user-defined function `DIAG`, defined by

```
[0] R←DIAG x;n
[1] R←((2ρn)ρ1,nρ0)×(2ρn←ρx)ρx
```

This is a monadic function that receives a vector **x** as the argument and returns as the function's explicit result a square matrix with the values of the vector **x** placed along the principal diagonal of the matrix with zeroes elsewhere in the matrix.

Our new function `DIAG` allows us to generate the transactions matrix corresponding to a matrix of technical coefficients matrix and vector of total output by means of the input-output identity $\mathbf{Z} = \mathbf{A}\hat{\mathbf{x}}$ in APL as

```
      A+.×DIAG x
 2.5 10  2.5
 2.5  5  2.5
30   30 15
```

For the problem at hand, we use the new vector of total outputs, `x2`, to generate the new matrix of interindustry transactions

```
      Z2←A+.×DIAG x2
      Z2
 1.9658264  8.9211867  2.179514
 1.9658264  4.4605934  2.179514
23.589916  26.76356   13.077084
```

And the corresponding vectors of value-added and intermediate outputs are easily obtained, respectively, as

```
      w2←x2-+⌿Z2
      w2
11.794958 4.4605934 69.744448
      u2←+/Z2
      u2
13.066527 8.6059337 63.430561
```

Problem 2.5: Impact Analysis

This problem assembles an input-output transactions table and explores the Hawkins-Simon conditions along with impact analysis for a new vector of final demands for the defined economy.

Problem 2.5 Overview

Consider a simple two-sector economy containing industries *A* and *B*. Industry *A* requires \$2 million worth of its own product and \$6 million worth of Industry *B*'s output in the process of supplying \$20 million worth of its own product to final consumers. Similarly, Industry *B* requires \$4 million worth of its own product and \$8 million worth of Industry *A*'s output in the process of supplying \$20 million worth of its own product to final consumers.

Using these data, we define the matrix of interindustry transactions and vector of final demands as $\mathbf{Z} = \begin{bmatrix} 2 & 8 \\ 6 & 4 \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} 20 \\ 20 \end{bmatrix}$, respectively, so the corresponding vector of total outputs is computed as $\mathbf{x} = \mathbf{f} + \mathbf{Zi} = \begin{bmatrix} 30 \\ 30 \end{bmatrix}$. Hence, the matrix of direct requirements is found by $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .067 & .267 \\ .2 & .133 \end{bmatrix}$ and the Hawkins-Simon conditions are satisfied as positive values for the determinant and the principal minors of the matrix $(\mathbf{I} - \mathbf{A})$, i.e., $|\mathbf{I} - \mathbf{A}| = 0.756$, $(1 - a_{11}) = 0.993$, and $(1 - a_{22}) = 0.867$. The matrix of total requirements is then found as $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.147 & .353 \\ .265 & 1.235 \end{bmatrix}$ and, for new vector of final demands, $\mathbf{f}^{new} = \begin{bmatrix} 15 \\ 18 \end{bmatrix}$, the corresponding vector of total outputs is computed as $\mathbf{x}^{new} = \mathbf{Lf}^{new} = \begin{bmatrix} 23.559 \\ 26.206 \end{bmatrix}$ and related interindustry activity (matrix of interindustry transactions) is $\mathbf{Z}^{new} = \mathbf{A}\hat{\mathbf{x}}^{new} = \begin{bmatrix} 1.571 & 6.988 \\ 4.712 & 3.494 \end{bmatrix}$.

Computational Notes

We have already developed all the APL tools to complete this problem. First, we have

```
      Z←2 2⍴2 8 6 4
      f←20 20
      x←f++/Z
       x
30 30
```

The matrx of technical coefficients and the Leontief inverse along with the determinant of $(\mathbf{I}-\mathbf{A})$ and the principal minors are found by

```
      I←2 2⍴1,2⍴0
      L←LINV A←Z AMAT x
      A
0.066666667 0.26666667
0.2         0.13333333
      L
1.1470588  0.35294118
0.26470588 1.2352941
      DETER I-A
0.75555556
      1 1⍉I-A
0.93333333 0.86666667
```

And, finally, the total outputs and accompanying interindustry transactions for the new vector of final demands is found by

```
      x2←L+.×f2←15 18
      Z2←A+.×DIAG x2
      x2
23.558824 26.205882
      Z2
1.5705882 6.9882353
4.7117647 3.4941176
```

Problem 2.6: Round-by-Round Approximation of L for Impact Analysis

While computer advances have considerably reduced the computational constraints for many applications of input-output analysis, it has also made possible the construction and use of much larger scale input-output models with thousands of sectors specified. This problem illustrates, on a small scale, practical considerations in working with very large input-output models for determining when using round-by-round calculations for impact analysis is a cost-effective substitute for using the direct computation of the Leontief inverse in impact analysis.

Problem 2.6 Overview

Recall that the power series, $\tilde{\mathbf{x}}=\sum_{i=0}^{r}\mathbf{A}^i\mathbf{f}$, is a substitute for using the direct computation of the Leontief inverse in impact analysis, $\mathbf{x}=\mathbf{Lf}$ where $\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}$.

Consider the following transactions table, $\mathbf{Z}$, and total outputs vector, $\mathbf{x}$, for a two-sector economy:

$$\mathbf{Z} = \begin{bmatrix} 6 & 2 \\ 4 & 2 \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} 20 \\ 15 \end{bmatrix}$$

For this economy, the vectors of value-added inputs and final demands are computed as $\mathbf{v}' = \mathbf{x}' - \mathbf{i}'\mathbf{Z} = \begin{bmatrix} 10 & 11 \end{bmatrix}$ and $\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 20 \\ 20 \end{bmatrix}$, respectively. With $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .3 & .133 \\ .2 & .133 \end{bmatrix}$ we show first that the Hawkins-Simon conditions are satisfied by positive values for the determinant and the principal minors of the matrix $(\mathbf{I} - \mathbf{A})$: $|\mathbf{I} - \mathbf{A}| = 0.58$, $(1 - a_{11}) = 0.7$, and $(1 - a_{22}) = 0.867$, so we consider the economy to be "well behaved." The r-order round-by-round approximation of $\mathbf{x} = \mathbf{Lf} = \begin{bmatrix} 20 \\ 15 \end{bmatrix}$ is found as: $\tilde{\mathbf{x}} = \sum_{i=0}^{r} \mathbf{A}^i \mathbf{f}$ (remember that $\mathbf{A}^0 = \mathbf{I}$), shown in the following table.

Round-by-round approximation of total outputs for $r = 1, 2, \cdots, 10$

r	1	2	3	4	5	**6**	7	8	9	10
$\tilde{x}_1$	16.800	18.720	19.488	19.795	19.918	**19.967**	19.987	19.995	19.998	19.999
$\tilde{x}_2$	12.600	14.040	14.616	14.846	14.939	**14.975**	14.990	14.996	14.998	14.999

We see from the table that $x_j - \tilde{x}_j < 0.05$ for both sectors ($j = 1, 2$) at $r = 6$.

The specified cost of performing impact analysis on the computer using the round-by-round method is then computed as $C_r = c_1 r + c_2(r - 1.5)$ where r is the order of the approximation (c_1 is the cost of an addition operation and c_2 is the cost of a multiplication operation). Also, we assume further that $c_1 = 0.5c$, that the cost of computing $(\mathbf{I} - \mathbf{A})^{-1}$ exactly rather than via successive approximation is given by $C_e = 20c_2$, and that the cost of using this inverse in impact analysis (multiplying it by a final-demand vector) is given by $C_f = c_2$.

If we want to determine whether to use the round-by-round method or to compute the exact inverse and then perform impact analysis, i.e., to determine the least-cost method for computing the solution, for one final demand vector, the equation defining the computation cost is $C_r = 0.5c_2 r + c_2 r - 1.5c_2 = (0.5r + r - 1.5)c_2 = (1.5r - 1.5)c_2 = 1.5(r - 1)c_2$ and, from the table, for $x_j - \tilde{x}_j < 0.2$ then $r = 5$ so $C_r = 6c_2$.

Hence, for one final demand vector, the cost of the round-by-round approximation is $C_r = 6c_2$ which is less than the cost of using the exact inverse $C_e + C_f = 21c_2$, it is much more cost effective to use the round-by-round round method. For five final demand vectors, however, $C_r = 5(6c_2) = 30c_2 > C_e + 4C_f = 20c_2 + 4c_2 = 24c_2$, so it is more cost effective to use the exact method. For four final demand vectors, it turns out, the total cost of computation is $C_r = 4(6c_2) = 24c_2 = C_e + 4C_f = 20c_2 + 4c_2 = 24c_2$, i.e., the costs of both methods are identical so at least in terms of cost effectiveness we are indifferent as to which method to employ.

Computational Notes

We have all of the APL tools needed to solve this problem. First, the basic IOA data are

```
      Z←2 2⍴6 2 4 2
      x←20 15
      w←x-+⌿Z
      f←x-+/Z
      w
10 11
      f
12 9
```

The Hawkins-Simon conditions are satisfied

```
      I←2 2⍴1,2⍴0
      IA←I-A←Z AMAT x
      A
0.3 0.13333333
0.2 0.13333333
      IA
 0.7 ¯0.13333333
¯0.2  0.86666667
      DETER IA
0.58
      1 1⍉IA
0.7 0.86666667
```

For convenience we create a table R comparing successive approximations of x necessary to support the specified vector of final demands with the exact value computed with the Leontief inverse. To generate the table, we can use the following dyadic function

```
[0] R←f RINVTEST A;i
[1] R←9 3⍴0
[2] i←1
[3] L1:R[i;]←(i+1),((i+1)RINV A)+.×f
[4] →(9≥i←i+1)/L1
```

For this problem, we use RINVTEST with the defined values for **f** and **A** to generate the table of successive approximations of **x**.

```
      f RINVTEST A
 2 18.72     14.04
 3 19.488    14.616
 4 19.7952   14.8464
 5 19.91808  14.93856
 6 19.967232 14.975424
 7 19.986893 14.99017
 8 19.994757 14.996068
 9 19.997903 14.998427
10 19.999161 14.999371
```

The exact value of **x** is found with

```
      (LINV A)+.×f
20 15
```

From the table we can see that for the successive approximation of **x**, all elements of the approximation come within .05 of the corresponding exact value at $n = 6$.

Problem 2.7: Impact Analysis of an Eight-Sector Economy

This problem explores computation of the Leontief inverse and impact analysis for an eight-sector economy (practical only with computer tools).

Problem 2.7 Overview

Consider the following matrix of interindustry transactions, **Z**, and vector of total outputs, **x**, for an eight-sector economy:

$$\mathbf{Z} = \begin{bmatrix} 8,565 & 8,069 & 8,843 & 3,045 & 1,124 & 276 & 230 & 3,464 \\ 1,505 & 6,996 & 6,895 & 3,530 & 3,383 & 365 & 219 & 2,946 \\ 98 & 39 & 5 & 429 & 5,694 & 7 & 376 & 327 \\ 999 & 1,048 & 120 & 9,143 & 4,460 & 228 & 210 & 2,226 \\ 4,373 & 4,488 & 8,325 & 2,729 & 2,9671 & 1,733 & 5,757 & 14,756 \\ 2,150 & 36 & 640 & 1,234 & 165 & 821 & 90 & 6,717 \\ 506 & 7 & 180 & 0 & 2,352 & 0 & 18,091 & 26,529 \\ 5,315 & 1,895 & 2,993 & 1,071 & 13,941 & 434 & 6,096 & 46,338 \end{bmatrix}$$

$$\mathbf{x}' = \begin{bmatrix} 37,610 & 45,108 & 46,323 & 41,059 & 209,403 & 11,200 & 55,992 & 161,079 \end{bmatrix}$$

We compute the matrices of direct requirements, **A**, and the matrix of total requirements, **L**, as the following:

$$\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .228 & .179 & .191 & .074 & .005 & .025 & .004 & .022 \\ .040 & .155 & .149 & .086 & .016 & .033 & .004 & .018 \\ .003 & .001 & .000 & .010 & .027 & .001 & .007 & .002 \\ .027 & .023 & .003 & .223 & .021 & .020 & .004 & .014 \\ .116 & .099 & .180 & .066 & .142 & .155 & .103 & .092 \\ .057 & .001 & .014 & .030 & .001 & .073 & .002 & .042 \\ .013 & 0 & .004 & 0 & .011 & 0 & .323 & .165 \\ .141 & .042 & .065 & .026 & .067 & .039 & .109 & .288 \end{bmatrix}$$

$$\mathbf{L} - (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.339 & .296 & .312 & .172 & .034 & .058 & .030 & .067 \\ .089 & 1.214 & .209 & .153 & .038 & .057 & .025 & .051 \\ .013 & .009 & 1.011 & .019 & .034 & .008 & .018 & .013 \\ .065 & .056 & .034 & 1.306 & .038 & .041 & .021 & .041 \\ .265 & .215 & .320 & .174 & 1.207 & .230 & .229 & .240 \\ .100 & .029 & .045 & .059 & .011 & 1.089 & .018 & .074 \\ .109 & .049 & .068 & .035 & .054 & .030 & 1.547 & .372 \\ .321 & .162 & .210 & .117 & .135 & .103 & .269 & 1.506 \end{bmatrix}$$

For a case where final demands in sectors 1 and 2 increase by 30 percent while in sector 5 they decrease by 20 percent with all other final demands unchanged, we first compute the base final demands as $\mathbf{f} = \mathbf{x} - \mathbf{Ax}$, then $\mathbf{x}^{new} = \mathbf{Lf}^{new}$, for $\mathbf{f} = \mathbf{x} - \mathbf{Ax} = \begin{bmatrix} 3,994 \\ 19,269 \\ 39,348 \\ 22,625 \\ 137,571 \\ -653 \\ 8,327 \\ 82,996 \end{bmatrix}$, and, applying the changes indicated, $\mathbf{f}^{new} = \begin{bmatrix} 5,192 \\ 25,050 \\ 39,348 \\ 22,625 \\ 110,057 \\ -653 \\ 8,327 \\ 82,996 \end{bmatrix}$ to yield $\mathbf{x}^{new} = \mathbf{Lf}^{new} = \begin{bmatrix} 39,998 \\ 51,181 \\ 45,455 \\ 40,404 \\ 177,756 \\ 11,182 \\ 54,929 \\ 158,687 \end{bmatrix}$.

Computational Notes

We can enter the data for the vector of total outputs and matrix of interindustry transactions as a string of data and reshape the transactions data to form the 8×8 matrix by

```
x←37610 45108 46323 41059 209403 11200 55992 161079
Z←8565 8069 8843 3045 1124 276 230 3464
Z←Z,1505 6996 6895 3530 3383 365 219 2946
Z←Z,98 39 5 429 5694 7 376 327
Z←Z,999 1048 120 9143 4460 228 210 2226
Z←Z,4373 4488 8325 2729 29671 1733 5757 14756
Z←Z,2150 36 640 1234 165 821 90 6717
Z←Z,506 7 180 0 2352 0 18091 26529
```

```
Z←Z,5315 1895 2993 1071 13941 434 6096 46338
Z←8 8⍴Z
```

Now, using the functions we have already developed, it is straightforward to compute

```
L←LINV A←Z AMAT x
f←x-+/Z
```

For larger matrices in APL, it is often convenient to use the *format* operator denoted by the character ⍕ to specify the precision of output displayed. The simplest use of the format operator takes as the left argument the specification of the number of spaces for a column of formatted data and the number of significant digits to the right of the decimal point and the array to be formatted as the right argument.[3] For example, for a vector

```
      R←0.22773199 0.17888179 0.19089869 0.074161572
      R
0.22773199 0.17888179 0.19089869 0.074161572
```

We might express R more succinctly as one of the following

```
      10 4⍕R
    0.2277    0.1789    0.1909    0.0742
      8 5⍕R
 0.22773 0.17888 0.19090 0.07416
```

For this problem we use the format operator to show **A** and **L** in this way.

```
      8 4⍕A
  0.2277  0.1789  0.1909  0.0742  0.0054  0.0246  0.0041  0.0215
  0.0400  0.1551  0.1488  0.0860  0.0162  0.0326  0.0039  0.0183
  0.0026  0.0009  0.0001  0.0104  0.0272  0.0006  0.0067  0.0020
  0.0266  0.0232  0.0026  0.2227  0.0213  0.0204  0.0038  0.0138
  0.1163  0.0995  0.1797  0.0665  0.1417  0.1547  0.1028  0.0916
  0.0572  0.0008  0.0138  0.0301  0.0008  0.0733  0.0016  0.0417
  0.0135  0.0002  0.0039  0.0000  0.0112  0.0000  0.3231  0.1647
  0.1413  0.0420  0.0646  0.0261  0.0666  0.0388  0.1089  0.2877
      8 4⍕L
  1.3394  0.2960  0.3115  0.1721  0.0337  0.0584  0.0299  0.0669
  0.0887  1.2139  0.2091  0.1527  0.0382  0.0571  0.0247  0.0514
  0.0129  0.0089  1.0111  0.0195  0.0340  0.0080  0.0175  0.0128
  0.0646  0.0562  0.0343  1.3055  0.0384  0.0405  0.0208  0.0409
  0.2648  0.2155  0.3196  0.1735  1.2070  0.2302  0.2294  0.2395
  0.0999  0.0288  0.0454  0.0589  0.0111  1.0891  0.0178  0.0743
  0.1093  0.0493  0.0684  0.0350  0.0538  0.0300  1.5472  0.3718
  0.3214  0.1624  0.2099  0.1175  0.1351  0.1025  0.2686  1.5061
```

For the new vector of total outputs, we compute

```
      f2←f+f×0.3 0.3 0 0 ¯0.2 0 0 0
      x2←L+.×f2
      8 0⍕f2
```

[3] The format operator has many other features as well, detailed in Legrand (2009).

```
  5192    25050    39348    22625   110057      ¯653      8327    82996
   8 0⍕x2
 39998    51181    45455    40404   177756    11182    54929   158687
```

Problem 2.8: Changes in Relative Prices Resulting From Value-Added Changes

The problem explores changes in relative prices in an input-output formulation resulting from changes in value-added inputs.

Problem 2.8 Overview

Consider a two-sector input-output table measured in millions of dollars:

	Manuf.	Services	Final Demand	Total Output
Manufacturing	10	40	50	100
Services	30	25	85	140
Value Added	60	75	135	
Total Output	100	140		

Using the table data, we define the matrix of interindustry transactions, $\mathbf{Z}^0 = \begin{bmatrix} 10 & 40 \\ 30 & 25 \end{bmatrix}$, the vector of total outputs, $\mathbf{x}^0 = \begin{bmatrix} 100 \\ 140 \end{bmatrix}$, and the vector of total value-added inputs $(\mathbf{w}^0)' = \begin{bmatrix} 60 & 75 \end{bmatrix}$, and we can then compute

$$\mathbf{A}^0 = \mathbf{Z}^0(\hat{\mathbf{x}}^0)^{-1} = \begin{bmatrix} 10 & 40 \\ 30 & 25 \end{bmatrix} \begin{bmatrix} 1/100 & 0 \\ 0 & 1/140 \end{bmatrix} = \begin{bmatrix} .1 & .286 \\ .3 & .179 \end{bmatrix}$$ and $\mathbf{L}^0 = (\mathbf{I} - \mathbf{A}^0)^{-1} = \begin{bmatrix} 1.257 & .437 \\ .459 & 1.377 \end{bmatrix}$. For this formulation we will need the matrix transposes of $\mathbf{A}^0$ and $\mathbf{L}^0$, i.e., $(\mathbf{A}^0)' = \begin{bmatrix} .1 & .3 \\ .286 & .179 \end{bmatrix}$ and $(\mathbf{L}^0)' = [\mathbf{I} - (\mathbf{A}^0)']^{-1} = \begin{bmatrix} 1.257 & .459 \\ .437 & 1.377 \end{bmatrix}$. The value-added coefficients are computed as $\mathbf{v}_c^0 = (\mathbf{w}^0)'(\hat{\mathbf{x}}^0)^{-1} = \begin{bmatrix} 60/100 & 75/140 \end{bmatrix} = \begin{bmatrix} .6 & .536 \end{bmatrix}$ for which the normalized prices are found, not surprisingly, as $\tilde{\mathbf{p}}^0 = (\mathbf{L}^0)'(\mathbf{v}_c^0)' = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ or, perhaps more intuitively, since the transpose of a product of matrices is the product of the transposes of the individual matrices in the reverse order (see Appendix A) and the row vector, $(\tilde{\mathbf{p}}^0)' = \mathbf{v}_c^0\mathbf{L}^0 = \begin{bmatrix} 1 & 1 \end{bmatrix}$. This is not surprising since, beginning with the basic accounting identity, $\mathbf{w} = \mathbf{x}' - \mathbf{i}'\mathbf{Z}$, which we can express as $\mathbf{w} = \mathbf{x}' - \mathbf{i}'\mathbf{A}\hat{\mathbf{x}}$, we first postmultiply through by $\hat{\mathbf{x}}^{-1}$ to obtain $\mathbf{v}_c = \mathbf{w}\hat{\mathbf{x}}^{-1} = \mathbf{x}'\hat{\mathbf{x}}^{-1} - \mathbf{i}'\mathbf{A}\hat{\mathbf{x}}\hat{\mathbf{x}}^{-1} = \mathbf{i}' - \mathbf{i}\mathbf{A} = \mathbf{i}'(\mathbf{I} - \mathbf{A})$. Then, postmultiplying through by $(\mathbf{I} - \mathbf{A})^{-1}$, the result is $\mathbf{v}_c(\mathbf{I} - \mathbf{A})^{-1} = \mathbf{i}'(\mathbf{I} - \mathbf{A})(\mathbf{I} - \mathbf{A})^{-1}$ which reduces to the general result, $\mathbf{v}_c\mathbf{L} = \mathbf{i}'$.

If labor costs in the services sector increase, causing a 25 percent increase in value added inputs required per unit of services and labor costs in manufacturing decrease by 25 percent, the new value-added coefficients, reflecting the changes are $\mathbf{v}_c^1 = \begin{bmatrix} 1.25 & 0 \\ 0 & .75 \end{bmatrix} \mathbf{v}_c^0 = \begin{bmatrix} .75 & .402 \end{bmatrix}$, so the prices for the new period 1 relative to the current period 0 are $(\tilde{\mathbf{p}}_0^1)' = \mathbf{v}_c^1 \mathbf{L}^0 = \begin{bmatrix} 1.127 & .881 \end{bmatrix}$.

Computational Notes

We have developed all the APL tools we need to solve this problem. First, we can compute all the basic quantities.

```
Z0←2 2⍴10 40 30 25
f0←50 85
x0←f0++/Z0
A0←Z0 AMAT x0
L0←LINV A0
L0T←LINV AT0←⍉A0
p0←L0T+.×vc0←60 75÷100 140
p1←L0T+.×vc1←1.25 0.75×vc0
```

```
         Z              f0       x0
   10         40        50       100
   30         25        85       140

         A0
    0.100      0.286
    0.300      0.179
         AT0
    0.100      0.300
    0.286      0.179
         L0
    1.257      0.437
    0.459      1.377
         LT0
    1.257      0.459
    0.437      1.377
      vc0,     p0
    0.600      1.000
    0.536      1.000
      vc1,     p1
    0.750      1.127
    0.402      0.881

         A1
    0.127      0.284
    0.298      0.139


         AT1
    0.127      0.298
    0.284      0.139
```

```
      L1
 1.291     0.425
 0.446     1.308
      LT1
 1.291     0.446
 0.425     1.308

   Z,                f1,       x1
14         35        56        113
34         17        75        123
  vc11,     p11
 0.575     1.000
 0.578     1.000
```

Problem 2.9: Product Prices Changes Generated by National Corporate Income Tax

This problem explores changes in relative product prices resulting from a change in value-added inputs generated by a national corporate income tax.

Problem 2.9 Overview

We use the 2003 U.S. direct requirements table given in Table 2.6. For the matrix of direct requirements, **A**, given in the table, the transpose of the Leontief inverse is

$$\mathbf{L}' = (\mathbf{I} - \mathbf{A}')^{-1} = \begin{bmatrix} 1.262 & 0.009 & 0.008 & 0.229 & 0.149 & 0.238 & 0.024 \\ 0.006 & 1.075 & 0.003 & 0.119 & 0.085 & 0.293 & 0.024 \\ 0.013 & 0.012 & 1.005 & 0.262 & 0.137 & 0.270 & 0.023 \\ 0.057 & 0.034 & 0.006 & 1.342 & 0.156 & 0.292 & 0.037 \\ 0.004 & 0.019 & 0.007 & 0.069 & 1.089 & 0.271 & 0.028 \\ 0.007 & 0.003 & 0.011 & 0.086 & 0.060 & 1.412 & 0.030 \\ 0.007 & 0.007 & 0.025 & 0.126 & 0.085 & 0.314 & 1.034 \end{bmatrix}.$$

Suppose that the new corporate income tax generates increases in the total value-added inputs of 10 percent for primary industries (agriculture and mining), of 15 percent for construction and manufacturing, and of 20 percent for all other sectors. The vector of value-added coefficients for the original input output economy is found as

$\mathbf{v}_c^0 = \mathbf{i} - \mathbf{i}'\mathbf{A} = \begin{bmatrix} .486 & .633 & .580 & .470 & .699 & .629 & .640 \end{bmatrix}'$, so that $\tilde{\mathbf{p}}^0 = \mathbf{L}'\mathbf{v}_c^0 = \mathbf{i}$. We define the vector of value-added growth factors, reflecting the value-added changes indicated, as

$\mathbf{d} = [1.1 \quad 1.1 \quad 1.15 \quad 1.15 \quad 1.2 \quad 1.2 \quad 1.2]'$ so that we can find the new vector of value-added coefficients by $\mathbf{v}_c^1 = \hat{\mathbf{d}}\mathbf{v}_c^0 = \begin{bmatrix} 1.1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.2 \end{bmatrix} \begin{bmatrix} .486 \\ .633 \\ .58 \\ .47 \\ .699 \\ .629 \\ .64 \end{bmatrix} = \begin{bmatrix} .534 \\ .696 \\ .667 \\ .540 \\ .839 \\ .754 \\ .768 \end{bmatrix}$.

Hence the new prices are found as

$$\tilde{\mathbf{p}}^1 = \mathbf{L}'\mathbf{v}_c^1 = \begin{bmatrix} 1.133 & 1.129 & 1.163 & 1.163 & 1.197 & 1.197 & 1.195 \end{bmatrix}'.$$

Computational Notes

We have developed all the APL tools we need to solve this problem.

```
vc0←1-+⌿A
LT←INV⍉A
p0←LT+.×vc0
vc1←vc0×d←1.1 1.1 1.15 1.15 1.2 1.2 1.2
p1←LT+.×vc1
```

A

0.201	0.000	0.001	0.034	0.000	0.002	0.001
0.001	0.066	0.004	0.022	0.015	0.000	0.003
0.003	0.000	0.001	0.002	0.004	0.007	0.021
0.125	0.068	0.180	0.232	0.034	0.041	0.073
0.086	0.053	0.091	0.095	0.065	0.032	0.053
0.090	0.167	0.133	0.126	0.165	0.271	0.187
0.009	0.013	0.010	0.020	0.019	0.018	0.023

LT

1.262	0.009	0.008	0.229	0.149	0.238	0.024
0.006	1.075	0.003	0.119	0.085	0.293	0.024
0.013	0.012	1.005	0.262	0.137	0.270	0.023
0.057	0.034	0.006	1.342	0.156	0.292	0.037
0.004	0.019	0.007	0.069	1.089	0.271	0.028
0.007	0.003	0.011	0.086	0.060	1.412	0.030
0.007	0.007	0.025	0.126	0.085	0.314	1.034

vc0	p0
0.486	1.000
0.633	1.000
0.580	1.000
0.470	1.000
0.699	1.000
0.629	1.000
0.640	1.000
vc1	p1
0.534	1.133
0.696	1.129

```
0.667     1.163
0.540     1.163
0.839     1.197
0.754     1.197
0.768     1.195
```

Problem 2.10: "Scrubbing" Imports from "U.S.-Style" IO Models

This problem explores the process of opening a "U.S.-style" input-output economy (adopting the accounting conventions of national input-output tables assembled in the United States) to imports by "scrubbing" from the assembled interindustry transactions matrix the portion of interindustry transactions that represent competitive imports from outside the economy and reassigning them as value-added imports (noncompetitive imports are already treated as value added inputs).

Problem 2.10 Overview

Consider an input-output economy with three sectors: agriculture, services, and personal computers. The matrix of interindustry transactions and vector of total outputs are given, respectively, by $\mathbf{Z} = \begin{bmatrix} 2 & 2 & 1 \\ 1 & 0 & 0 \\ 2 & 0 & 1 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 5 \\ 2 \\ 2 \end{bmatrix}$ so that the associated vector of final demands is $\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$. Notice, first, that this is a closed economy where all industry outputs become inputs. That is, with the given vector of total outputs, $\mathbf{x}$, the vector of total value-added inputs is found by $\mathbf{v}' = \mathbf{x}' - \mathbf{i}'\mathbf{Z} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$ and, of course, the gross domestic product is $\mathbf{v}'\mathbf{i} = \mathbf{i}'\mathbf{f} = 0$. For this economy, $\mathbf{A} = \begin{bmatrix} .4 & 1 & .5 \\ .2 & 0 & 0 \\ .4 & 0 & .5 \end{bmatrix}$, so we can compute $|\mathbf{I} - \mathbf{A}| = 0$. This means that $(\mathbf{I} - \mathbf{A})$ is a singular matrix and $\mathbf{L}$ does not exist.

Suppose that we determine *all* the inputs for the personal computers sector are imported. We can create a domestic transactions matrix by "opening" the economy to imports, i.e., transfer the value of all inputs to personal computers to final demand. For a "U.S. style" input-output table, competitive imports are included in the matrix of transactions and a corresponding negative entry for imports is included in final demand.

To "scrub" this transactions table of competitive imports we need to, first, subtract the value of imports from the first and third entries in the column for personal computers, then, add those amounts to the first and third entries of final demand. We define $\mathbf{D}$ as the matrix of domestic transactions where the values of competitive imports are subtracted to remove them from the matrix of interindustry transactions, $\mathbf{Z}$, and $\mathbf{g}$ as the new vector of final demands where the values of competitive imports are added to remove imports from final demand, $\mathbf{f}$, (recall that they were included originally in final demand as negative values).

Thus, $\mathbf{Z}=\begin{bmatrix}2 & 2 & 1\\ 1 & 0 & 0\\ 2 & 0 & 1\end{bmatrix}$ becomes $\mathbf{D}=\begin{bmatrix}2 & 2 & 1-1\\ 1 & 0 & 0\\ 2 & 0 & 1-1\end{bmatrix}=\begin{bmatrix}2 & 2 & 0\\ 1 & 0 & 0\\ 2 & 0 & 0\end{bmatrix}$ and **f** becomes

$\mathbf{g}=\begin{bmatrix}0+1\\ 1\\ -1+1\end{bmatrix}=\begin{bmatrix}1\\ 1\\ 0\end{bmatrix}$. The vector of total outputs, **x**, is unchanged, but the new vector of total value added is $\bar{\mathbf{v}}'=\mathbf{x}'-\mathbf{iD}=\begin{bmatrix}0 & 0 & 2\end{bmatrix}$ and, hence, gross domestic product is $\mathbf{v}'\mathbf{i}=\mathbf{i}'\mathbf{f}=2$. We then can compute the matrix of direct requirements as $\bar{\mathbf{A}}=\begin{bmatrix}.4 & 1 & 0\\ .2 & 0 & 0\\ .4 & 0 & 0\end{bmatrix}$, for which $\left|\mathbf{I}-\bar{\mathbf{A}}\right|=0.4$ so the matrix $(\mathbf{I}-\bar{\mathbf{A}})$ is non-singular and the matrix of total requirements can be computed as

$\bar{\mathbf{L}}=(\mathbf{I}-\bar{\mathbf{A}})^{-1}=\begin{bmatrix}2.5 & 2.5 & 0\\ .5 & 1.5 & 0\\ 1 & 1 & 1\end{bmatrix}$.

Computational Notes

We have developed all the APL tools we need to solve this problem.

```
Z←3 3⍴2 2 1 1 0 0 2 0 1
x←5 2
f←x-+/Z
I←3 3⍴1,3⍴0
A←Z AMAT x
DET←DETER I-A
v←x-+⌿Z
D←3 3⍴2 2 0 1 0 0 2 0 0
g←1 1 0
L2←LINV A2←D AMAT x
vb←x-+⌿D
```

```
        Z,                     f,         x
2         2         1          0          5
1         0         0          1          2
2         0         1         ¯1          2


v
0         0         0

A
     0.400     1.000     0.500
     0.200     0.000     0.000
     0.400     0.000     0.500
```

```
DET
0

          D                              g
          2          2          0          1
          1          0          0          1
          2          0          0          0
vb
          0          0          2

A2
     0.400      1.000      0.000
     0.200      0.000      0.000
     0.400      0.000      0.000
L2
     2.500      2.500      0.000
     0.500      1.500      0.000
     1.000      1.000      1.000
```

Chapter 3, Input–Output Models at the Regional Level

Chapter 3 extends the basic input–output framework to analysis of regions and the relationships between regions. First, "single-region" models are presented, and the various assumptions employed in formulating regional models versus national models are explored. Next, the structure of an interregional input–output (IRIO) model, designed to expand the basic input–output framework to capture transactions between industrial sectors in regions, is presented. An important simplification of the IRIO model designed to deal with the most common of data limitations in constructing such models is known as the multiregional input–output (MRIO) model. This chapter introduces the basic MRIO formulation and explores the implications of its simplifying assumptions along with the features of the balanced regional model which captures the distinction between industrial production for regional versus national markets. Finally, the chapter summarizes the fast-growing range of applications of MRIO analysis to multinational and global economic models and issues. The exercise problems for this chapter explore various characteristics of regional, IRIO, and MRIO model configurations and their applications.

Problem 3.1: Regional Purchase Coefficients

This problem explores the use of regional purchase coefficients to analyze regional interindustry activity. We begin with the data from Problem 2.2, which describes a small national economy that contains firms producing in each of the three industry sectors.

Problem 3.1 Overview

Suppose that for a regional economy within this national economy, the technological structure of production of firms within the region is estimated to be the same as that reflected in the national data, but that there is need to import into the region (from producers elsewhere in the country) some of the inputs used in production in each of the regional sectors. In particular, the percentages of required inputs from sectors 1, 2, and 3 that come from within the region are 60, 90, and 75, respectively, which defines the vector of regional purchase coefficients as

$\mathbf{p} = \begin{bmatrix} 0.60 \\ 0.90 \\ 0.75 \end{bmatrix}$. Using the matrix of technical coefficients, **A**, from Problem 2.2, we compute the

regional direct requirements matrix as $\mathbf{A}^R = \hat{\mathbf{p}}\mathbf{A} = \begin{bmatrix} .210 & 0 & 0 \\ .045 & .450 & .135 \\ .150 & .225 & .413 \end{bmatrix}$ and

$(\mathbf{I} - \mathbf{A}^R)^{-1} = \begin{bmatrix} 1.266 & 0 & 0 \\ .202 & 2.007 & .461 \\ .401 & .759 & 1.879 \end{bmatrix}$ is the regional total requirements matrix. If new final

demands for the outputs of the regional producers are projected to be 1300, 100, and 200, respectively, or $\mathbf{f}^{new} = \begin{bmatrix} 1,300 \\ 100 \\ 200 \end{bmatrix}$, the total regional production necessary to support those final demands is computed as the vector of regional total outputs, $\mathbf{x}^{new} = (\mathbf{I} - \mathbf{A}^R)^{-1}\mathbf{f}^{new} = \begin{bmatrix} 1,645.57 \\ 555.346 \\ 973.257 \end{bmatrix}$.

Computational Notes

We have developed all the APL tools needed to solve this problem.

```
Z←3 3ρ350 0 0 50 250 150 200 150 550
x←1000 500 1000
R←DIAG 0.6 0.9 0.75
x2←(L←INV AR←R+.×A←Z AMAT x)+.×f2←1300 100 200

      Z                          x
      350        0         0     1000
       50      250       150      500
      200      150       550     1000

      R
      0.6      0.0       0.0
      0.0      0.9       0.0
      0.0      0.0       0.8
      A,                   AR,                  L
      0.350  0.000  0.000  0.210  0.000  0.000  1.266  0.000  0.000
      0.050  0.500  0.150  0.045  0.450  0.135  0.202  2.007  0.461
      0.200  0.300  0.550  0.150  0.225  0.413  0.401  0.769  1.879

      f2,      x2
      1300.0   1645.6
       100.0    555.3
       200.0    973.3
```

Problem 3.2: The Interregional Input-Output (IRIO) Model

This problem explores the basic structure of an interregional input-output (IRIO) model.

Problem 3.2 Overview

The following table shows sales (in dollars) between and among two industry sectors in two regions, *r* and *s*.

		Region *r*		Region *s*	
		Industry 1	Industry 2	Industry 1	Industry 2
Region *r*	Industry 1	40	50	30	45
	Industry 2	60	10	70	45
Region *s*	Industry 1	50	60	50	80
	Industry 2	70	70	50	50

In addition, sales to final demand purchasers for each region are designated, respectively for regions r and s, are $\mathbf{f}^r = \begin{bmatrix} 200 \\ 200 \end{bmatrix}$ and $\mathbf{f}^s = \begin{bmatrix} 300 \\ 400 \end{bmatrix}$.

These data are sufficient to create a two-region IRIO model connecting regions r and s. Using the data from the table, $\mathbf{Z}$ is defined as the matrix of IRIO transactions, the corresponding vector of final demands is found as $\mathbf{f} = \begin{bmatrix} \mathbf{f}^r \\ \hdashline \mathbf{f}^s \end{bmatrix}$, and the vector of total outputs is found as

$\mathbf{x} = \mathbf{f} + \mathbf{Zi} = \begin{bmatrix} 365 \\ 385 \\ \hdashline 540 \\ 640 \end{bmatrix}$. Consequently, the matrix of technical coefficients is found as $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$

which can be partitioned into $\mathbf{A} = \left[\begin{array}{c:c} \mathbf{A}^{rr} & \mathbf{A}^{rs} \\ \hdashline \mathbf{A}^{sr} & \mathbf{A}^{ss} \end{array}\right] = \left[\begin{array}{cc:cc} 0.110 & 0.130 & 0.056 & 0.070 \\ 0.164 & 0.026 & 0.130 & 0.070 \\ \hdashline 0.137 & 0.156 & 0.093 & 0.125 \\ 0.192 & 0.182 & 0.093 & 0.078 \end{array}\right]$. If, because of a stimulated region r economy, household demand increased by \$280 for the output of sector 1 in region r and by \$360 for the output of sector 2 in region r, the vector of *changes* in final demand is $\Delta\mathbf{f} = \begin{bmatrix} 280 \\ 360 \\ \hdashline 0 \\ 0 \end{bmatrix}$. Computing, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \left[\begin{array}{cc:cc} 1.205 & 0.202 & 0.115 & 0.123 \\ 0.263 & 1.116 & 0.189 & 0.131 \\ \hdashline 0.273 & 0.262 & 1.177 & 0.200 \\ 0.330 & 0.289 & 0.179 & 1.156 \end{array}\right]$, the matrix of total requirements, then we can compute $\Delta\mathbf{x} = \begin{bmatrix} \Delta\mathbf{x}^r \\ \hdashline \Delta\mathbf{x}^s \end{bmatrix} = \mathbf{L}\Delta\mathbf{f} = \begin{bmatrix} 409.98 \\ 475.67 \\ \hdashline 170.62 \\ 196.24 \end{bmatrix}$, defining the new necessary gross outputs from each of the sectors in each of the two regions to satisfy this new final demand. Note that the increased outputs in region s for sector 1 of 170.62 and 196.24 for sector 2 are attributable solely to the interregional feedback effects associated with the new final demands in region r.

Computational Notes

We have developed all the APL tools needed to solve this problem.

```
Z←4 4⍴40 50 30 45 60 10 70 45 50 60 50 80 70 70 50 50
f←200 200 300 400
x←f++/Z
```

```
L←LINV A←Z AMAT x
x2←L+.×f2←280 360 0 0
```

```
       Z,                                      f,       x
       40       50       30       45       200      365
       60       10       70       45       200      385
       50       60       50       80       300      540
       70       70       50       50       400      640

A,                                              L
0.110    0.130    0.056    0.070    1.205    0.202    0.115    0.123
0.164    0.026    0.130    0.070    0.263    1.116    0.189    0.131
0.137    0.156    0.093    0.125    0.273    0.262    1.177    0.200
0.192    0.182    0.093    0.078    0.330    0.289    0.179    1.156

f2.          x2
   280.000   409.981
   360.000   475.665
     0.000   170.619
     0.000   196.240
```

Problem 3.3: The Multiregional Input-Output (MRIO) Model

This problem explores the basic structure of the multiregional input-output (MRIO) model.

Problem 3.3 Overview

Suppose that you have assembled the following information on (1) the dollar values of purchases of each of two goods in each of two regions and (2) on the shipments of each of the two goods between regions:

Purchases in Region r		Purchases in Region s	
$z_{11}^{r}=40$	$z_{12}^{r}=50$	$z_{11}^{s}=30$	$z_{12}^{s}=45$
$z_{21}^{r}=60$	$z_{22}^{r}=10$	$z_{21}^{s}=70$	$z_{22}^{s}=45$
Shipments of Good 1		Shipments of Good 2	
$z_{1}^{rr}=50$	$z_{1}^{rs}=60$	$z_{2}^{rr}=50$	$z_{2}^{rs}=80$
$z_{1}^{sr}=70$	$z_{1}^{ss}=70$	$z_{2}^{sr}=50$	$z_{2}^{ss}=50$

These data are sufficient to generate the necessary matrices for a two-region MRIO model involving regions r and s. There will be six necessary matrices—$\mathbf{A}^{r}$, $\mathbf{A}^{s}$, $\hat{\mathbf{c}}^{rr}$, $\hat{\mathbf{c}}^{rs}$, $\hat{\mathbf{c}}^{sr}$, and $\hat{\mathbf{c}}^{ss}$. All of these will be 2×2 matrices, configured from the transactions and trade shipments for each region. First, from the table we can construct the matrix of total transactions for each region as $\mathbf{Z}=\begin{bmatrix}\mathbf{Z}^{r} & 0\\ 0 & \mathbf{Z}^{s}\end{bmatrix}=\left[\begin{array}{cc:cc}40 & 50 & 0 & 0\\ 60 & 10 & 0 & 0\\ \hdashline 0 & 0 & 30 & 45\\ 0 & 0 & 70 & 45\end{array}\right]$. These transactions for each region, $\mathbf{Z}^{r}$ or $\mathbf{Z}^{s}$, include the inputs from all regions to support production in that region. We can configure the shipments of goods 1 and 2 in a matrix defined as $\mathbf{Q} = \left[\begin{array}{cc|cc} z_1^{rr} & 0 & z_1^{rs} & 0 \\ 0 & z_2^{rr} & 0 & z_2^{rs} \\ \hline z_1^{sr} & 0 & z_1^{ss} & 0 \\ 0 & z_2^{sr} & 0 & z_2^{ss} \end{array}\right] = \left[\begin{array}{cc|cc} 50 & 0 & 60 & 0 \\ 0 & 50 & 0 & 80 \\ \hline 70 & 0 & 70 & 0 \\ 0 & 50 & 0 & 50 \end{array}\right]$ so the vector of row sums of $\mathbf{Q}$ is $\mathbf{x} = \mathbf{Qi} = \left[\begin{array}{c} 110 \\ 130 \\ \hline 140 \\ 100 \end{array}\right]$, which is the vector of total deliveries of commodities of each type for each region to all regions, and the vector of the column sums of $\mathbf{Q}$ is $\mathbf{q} = \mathbf{i}'\mathbf{Q} = \begin{bmatrix} 120 & 100 & 130 & 130 \end{bmatrix}$, which is the vector of total availability from all regions of each commodity in each region. Hence, we can define matrix of technical coefficients as

$\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \left[\begin{array}{c|c} \mathbf{A}^r & 0 \\ \hline 0 & \mathbf{A}^s \end{array}\right] = \left[\begin{array}{cc|cc} .364 & .385 & 0 & 0 \\ .545 & .077 & 0 & 0 \\ \hline 0 & 0 & .214 & .45 \\ 0 & 0 & .5 & .45 \end{array}\right]$ and the matrix of trade coefficients as

$\mathbf{C} = \mathbf{Q}\hat{\mathbf{q}}^{-1} = \left[\begin{array}{c|c} \hat{\mathbf{c}}^{rr} & \hat{\mathbf{c}}^{rs} \\ \hline \hat{\mathbf{c}}^{sr} & \hat{\mathbf{c}}^{ss} \end{array}\right] = \left[\begin{array}{cc|cc} .417 & 0 & .462 & 0 \\ 0 & .5 & 0 & .615 \\ \hline .583 & 0 & .538 & 0 \\ 0 & .5 & 0 & .385 \end{array}\right]$. Now we compute the matrix of multiregional total requirements as $(\mathbf{I} - \mathbf{CA})^{-1}\mathbf{C} = \left[\begin{array}{cc|cc} 0.971 & 0.556 & 1.024 & 0.524 \\ 0.882 & 1.197 & 0.889 & 1.251 \\ \hline 1.297 & 0.714 & 1.264 & 0.677 \\ 0.663 & 1.010 & 0.673 & 0.854 \end{array}\right]$. If the projected demands for the coming period are $\mathbf{f}^{r(new)} = \begin{bmatrix} 50 \\ 50 \end{bmatrix}$ and $\mathbf{f}^{s(new)} = \begin{bmatrix} 40 \\ 60 \end{bmatrix}$, then $\mathbf{f}^{new} = \left[\begin{array}{c} \mathbf{f}^{r(new)} \\ \hline \mathbf{f}^{s(new)} \end{array}\right] = \left[\begin{array}{c} 50 \\ 50 \\ \hline 40 \\ 60 \end{array}\right]$. The corresponding vector of new total outputs for each sector in each region; $\mathbf{x}^r$ and $\mathbf{x}^s$, necessary to satisfy this new vector of final demands is found as the vector of new total outputs,

$$\mathbf{x}^{new} = \left[\begin{array}{c} \mathbf{x}^{r(new)} \\ \hline \mathbf{x}^{s(new)} \end{array}\right] = (\mathbf{I} - \mathbf{CA})^{-1}\mathbf{C}\mathbf{f}^{new} = \left[\begin{array}{c} 148.778 \\ 214.539 \\ \hline 191.718 \\ 161.772 \end{array}\right].$$

Computational Notes

To solve this problem, it is helpful to develop an APL function to manage MRIO data more efficiently. First, we presume that MRIO data are stored in three dimensional arrays. For example, the array of *interregional* flows, which we will denote with the variable QQ includes a separate "sheet" for each commodity with each sheet including source regions designated in the rows and destination regions designated in the columns. For this problem, since there are two commodities and two regions, the shape of QQ is $2 \times 2 \times 2$, and would appear in APL as

```
QQ
50      60
70      70

50      80
50      50
```

Similarly, we presume that the array of *intraregional* flows, which we will denote with the variable ZZ, includes a separate "sheet" for each region with each sheet including producing industries (of commodities) designated in the rows and consuming industries designated in the columns. For this problem, since there are two regions and two commodities, the shape of ZZ is $2 \times 2 \times 2$, and would appear in APL as

```
ZZ
40      50
60      10

30      45
70      45
```

Note that when working with arrays of dimension larger than two it is important to keep track of how the indices reference locations in the array. For QQ and ZZ, the first dimension comprises the "sheets" of the three-dimensional array, the second the rows of each sheet, and the third the columns of each sheet. For example, you can retrieve the first "sheet" of QQ by

```
QQ[1;;]
50 60
70 70
```

In addition, the behavior of operators is sensitive to the shape of the array. For example, reduction operations presume that the operation will be carried out on the last dimension of the array unless otherwise specified. For example, the expression `+/QQ` would sum along the columns of each sheet of QQ, as in

```
+/QQ
110 140
130 100
```

In our case, this expression is equivalent to `+/[3]QQ` for the three-dimensional array. Similarly, the expression `+/[2]QQ` would sum along the rows of each "sheet" of QQ, and `+/[1]QQ` (equivalently `+⌿QQ`) would sum across the first dimension of QQ, i.e., sum across the "sheets." For this problem the expression `+/QQ` is important since it results in a matrix in which each column is the total output vector of each region, summing the deliveries of each commodity to all regions including itself. We will denote that matrix by XX. Hence, to specify the basic data for the MRIO model we have

```
ZZ←2 2 2ρ40 50 60 10 30 45 70 45
QQ←2 2 2ρ50 60 70 70 50 80 50 50
XX←+/QQ
```

To convert the basic data to the more conventional two-dimensional format, several new APL functions are useful. The first in a monadic function GENQ that receives as its argument a three-dimensional MRIO commodity flows array (such as QQ) and returns as its explicit result the interregional commodity flows matrix in the familiar two-dimensional format.

```
[ 0]  CC←GENQ QQ;i;j;l;m;ns;nr;CQ
[ 1] ⍝Generate big MRIO Q (comm flows)
[ 2] ⍝from 3D interregional comm flows
[ 3]  CQ←((ns←1↑ρQQ),2ρnr←¯1↑ρQQ)ρ0×i←1
[ 4] L1:CQ[i;;]←QQ[i;;]
[ 5]  →(ns≥i←i+1)/L1
[ 6]  CC←((2ρnr)×ns)ρl←m←0×i←j←1
[ 7] L2:CC[m+ιns;l+ιns]←DIAG,CQ[;i;j]
[ 8]  l←l+ns
[ 9]  →(nr≥j←j+1)/L2
[10]  j←1+l←0×m←m+ns
[11]  →(nr≥i←i+1)/L2
```

In our case this is

```
      Q←GENQ QQ
      Q
50  0 60  0
 0 50  0 80
70  0 70  0
 0 50  0 50
```

The second is a similar monadic function GENZ that receives as its argument a three-dimensional intraregional interindustry transactions array (such as ZZ) and returns as its explicit result the intraregional transactions matrix in the familiar two-dimensional format.

```
[0]  ZZ←GENZ ZZZ;i;j;l;M;nr;ns
[1] ⍝Generate MRIO Big ZZZ from
[2] ⍝3 D interregional flows ZZZ
[3]  nr←1↑ρZZZ ⋄ ns←¯1↑ρZZZ
[4]  ZZ←((2ρnr)×ns)ρl←0×j←1
[5] L2:M←ZZZ[j;;]
[6]  ZZ[l+ιns;l+ιns]←M
[7]  l←l+ns
[8]  →(nr≥j←j+1)/L2
```

In our case this is

```
      Z←GENZ ZZ
```

```
Z
40 50  0  0
60 10  0  0
 0  0 30 45
 0  0 70 45
```

We can now generate the vectors of total industry outputs and total commodity outputs as

```
      x←+/Q
      x
110 130 140 100
      q←+⌿Q
      q
120 100 130 130
```

We can now compute C as

```
      C←Q AMAT q
      C
0.41666667 0   0.46153846 0
0          0.5 0          0.61538462
0.58333333 0   0.53846154 0
0          0.5 0          0.38461538
```

and A as

```
      A←Z AMAT x
      A
0.36363636 0.38461538  0          0
0.54545455 0.076923077 0          0
0          0           0.21428571 0.45
0          0           0.5        0.45
```

and, finally, **L** as

```
      L←(LINV C+.×A)+.×C
      L
0.97111491 0.55644409 1.0238482  0.52410525
0.88159175 1.1970088  0.88866106 1.2510417
1.2974875  0.71361698 1.2639161  0.67676551
0.66302868 1.0098041  0.67294214 0.8535466
```

So, for the new vector of final demands, defined as `f2`, wc compute the corresponding vector of total outputs as

```
      f2←50 50 40 60
      x2←L+.×f2

      f2
50 50 40 60
      x2
148.77819 214.53897 191.7178 161.77212
```

Problem 3.4: Distinctive Features of Regional Input-Output Data

This problem illustrates several important features of regional input-output data.

Problem 3.4 Overview

Suppose that a federal government agency for a three-region country has collected the following data on input purchases for two sectors, (1) manufacturing and (2) agriculture, measured in dollars for previous year. These flows are not specific with respect to region of origin; that is, they can be described as of the $z_{ij}^{\bullet s}$ sort rather than of the z_{ij}^{rs} sort. The three regions are denoted by *A*, *B,* and *C*.

	Region *A*		Region *B*		Region *C*	
	1	2	1	2	1	2
1	200	100	700	400	100	0
2	100	100	100	200	50	0

Also, gross (total) outputs for each of the two sectors in each of the three regions are known and specified by the vectors:

$$\mathbf{x}^A = \begin{bmatrix} 600 \\ 300 \end{bmatrix},\ \mathbf{x}^B = \begin{bmatrix} 1{,}200 \\ 700 \end{bmatrix} \text{ and } \mathbf{x}^C = \begin{bmatrix} 200 \\ 0 \end{bmatrix}$$

From the table we can define total regional interindustry transactions for each region as: $\mathbf{Z}^A = \begin{bmatrix} 200 & 100 \\ 100 & 100 \end{bmatrix}$, $\mathbf{Z}^B = \begin{bmatrix} 700 & 400 \\ 100 & 200 \end{bmatrix}$ and $\mathbf{Z}^C = \begin{bmatrix} 100 & 0 \\ 50 & 0 \end{bmatrix}$. We can then construct matrices of regional technical coefficients as $\mathbf{A}^r = \mathbf{Z}^r(\hat{\mathbf{x}}^r)^{-1}$ for regions $r = A$, *B* and *C* as $\mathbf{A}^A = \begin{bmatrix} 0.333 & 0.333 \\ 0.167 & 0.333 \end{bmatrix}$, $\mathbf{A}^B = \begin{bmatrix} 0.583 & 0.571 \\ 0.083 & 0.286 \end{bmatrix}$, and $\mathbf{A}^C = \begin{bmatrix} 0.500 & 0 \\ 0.250 & 0 \end{bmatrix}$. It is also straightforward to assemble the matrix of national transactions as the sum of all the regional transactions matrices, $\mathbf{Z}^N = \mathbf{Z}^A + \mathbf{Z}^B + \mathbf{Z}^C = \begin{bmatrix} 1{,}000 & 500 \\ 250 & 300 \end{bmatrix}$, and the vector of national total outputs as the sum of regional total output vectors, $\mathbf{x}^N = \mathbf{x}^A + \mathbf{x}^B + \mathbf{x}^C = \begin{bmatrix} 2{,}000 \\ 1{,}000 \end{bmatrix}$. Hence, the national technical coefficients matrix is found by $\mathbf{A}^N = \mathbf{Z}^N(\hat{\mathbf{x}}^N)^{-1} = \begin{bmatrix} .500 & .500 \\ .125 & .300 \end{bmatrix}$.

Since origin-destination data on shipments of each good have not been specified it is not yet possible to construct these data as an IRIO or MRIO model, but using the data specified, if the federal government is considering spending \$5,000 on manufactured goods and \$4,500 on agricultural products next year, we can define the vector of changes in final demand as $\mathbf{f}^{new} = \begin{bmatrix} 5{,}000 \\ 4{,}500 \end{bmatrix}$. Using $\mathbf{A}^N$ and $\mathbf{f}^{new}$, we compute $(\mathbf{I} - \mathbf{A}^N)^{-1} = \begin{bmatrix} 2.435 & 1.739 \\ 0.435 & 1.739 \end{bmatrix}$ and we find that

$\mathbf{x}^{new} = (\mathbf{I} - \mathbf{A}^N)^{-1}\mathbf{f}^{new} = \begin{bmatrix} 20,000 \\ 10,000 \end{bmatrix}$. Note that the original national total outputs vector was $\mathbf{x}^N = \begin{bmatrix} 2,000 \\ 1,000 \end{bmatrix}$ and the corresponding national final demand vector is $\mathbf{f}^N = \begin{bmatrix} 500 \\ 450 \end{bmatrix}$, found as $\mathbf{f}^N = \mathbf{x}^N - \mathbf{Z}^N\mathbf{i}$ or as $\mathbf{f}^N = \mathbf{f}^A + \mathbf{f}^B + \mathbf{f}^C$ where $\mathbf{f}^r = \mathbf{x}^r - \mathbf{Z}^r\mathbf{i}$ for regions $r =$ A, B and C. This simply illustrates the linearity of the input-output model, since $\mathbf{x}^{new} = 10\mathbf{x}^N$ follows directly from $\mathbf{f}^{new} = 10\mathbf{f}^N$. (See also the solution to Problem 2.2.)

Computational Notes

Using the new APL tools developed in Problem 3.4, first we formulate the available basic data for the matrix of interindustry transactions `ZZ` and vector of total outputs `XX`.

```
 ZZ←3 2 2ρ200 100 100 100 700 400 100 200 100 0 50 0
 XX←2 3ρ600 1200 200 300 700 0
```

In this case there are three regions with two industries in each region, so `XX` is a matrix of total regional outputs, i.e., with two industries designating the rows and the regions designating the columns.

```
ZZ
     200      100
     100      100

     700      400
     100      200

     100        0
      50        0
XX
     600     1200      200
     300      700        0
```

We can generate the three matrices of regional technical coefficients by

```
 A1←ZZ[1;;] AMAT XX[;1]
 A2←ZZ[2;;] AMAT XX[;2]
 A3←ZZ[3;;] AMAT XX[;3]
      A1
0.33333333 0.33333333
0.16666667 0.33333333
      A2
0.58333333  0.57142857
0.083333333 0.28571429
      A3
0.5  1
0.25 1
```

The national data can be constructed by summing across all the regions

```
 xn←+/XX
 ZN←+/[1]ZZ
```

```
 LN←LINV AN←ZN AMAT xn
 fN←xn-+/ZN

xn
    2000.0    1000.0
fN
     500.0     450.0
ZN
    1000.0     500.0
     250.0     300.0
AN
     0.500     0.500
     0.125     0.300
LN
     2.435     1.739
     0.435     1.739
```

For the new vector of final demands, we have

```
 f2←5000 4500
 x2←LN+.×f2

f2
    5000.0    4500.0
x2
   20000.0   10000.0
```

Problem 3.5: Interregional Linkages in IRIO Models

This problem illustrates the key features of an interregional input-output (IRIO) model configuration, especially the role of interregional linkages.

Problem 3.5 Overview

Consider the following two-region, three-sector interregional input-output transactions table:

		North			South			
		Agric.	Mining	Const.& Manuf.	Agric.	Mining	Const.& Manuf.	Total Output
North	Agriculture	277,757	3,654	1,710,816	8,293	26	179,483	3,633,382
	Mining	319	2,412	598,591	15	112	30,921	743,965
	Construction & Manufacturing	342,956	39,593	6,762,703	45,770	3,499	1,550,298	10,931,024
South	Agriculture	7,085	39	98,386	255,023	3,821	1,669,107	3,697,202
	Mining	177	92	15,966	365	3,766	669,710	766,751
	Construction & Manufacturing	71,798	7,957	2,017,905	316,256	36,789	8,386,751	14,449,941

Using the table's data to define the IRIO transactions matrix, $\mathbf{Z} = \left[\begin{array}{c|c} \mathbf{Z}^{NN} & \mathbf{Z}^{NS} \\ \hline \mathbf{Z}^{SN} & \mathbf{Z}^{SS} \end{array}\right]$, the total regional final demand vector is found as $\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \left[\begin{array}{c} \mathbf{f}^N \\ \hline \mathbf{f}^S \end{array}\right] = \left[\begin{array}{r} 1{,}453{,}353 \\ 111{,}595 \\ 2{,}186{,}205 \\ \hline 1{,}663{,}741 \\ 76{,}675 \\ 3{,}612{,}485 \end{array}\right]$. Hence the matrices of regional technical coefficients for the North and South regions, respectively are

$$\mathbf{A}^{NN} = \mathbf{Z}^{NN}(\hat{\mathbf{x}}^N)^{-1} = \begin{bmatrix} 0.076 & 0.005 & 0.157 \\ 0.000 & 0.003 & 0.055 \\ 0.094 & 0.053 & 0.619 \end{bmatrix} \text{ and } \mathbf{A}^{SS} = \mathbf{Z}^{SS}(\hat{\mathbf{x}}^S)^{-1} = \begin{bmatrix} 0.069 & 0.005 & 0.116 \\ 0.000 & 0.005 & 0.046 \\ 0.086 & 0.048 & 0.580 \end{bmatrix};$$

the matrices of interregional trade coefficients between the two regions are found as

$$\mathbf{A}^{SN} = \mathbf{Z}^{SN}(\hat{\mathbf{x}}^N)^{-1} = \begin{bmatrix} 0.002 & 0.000 & 0.009 \\ 0.000 & 0.000 & 0.001 \\ 0.020 & 0.011 & 0.185 \end{bmatrix} \text{ and } \mathbf{A}^{NS} = \mathbf{Z}^{NS}(\hat{\mathbf{x}}^S)^{-1} = \begin{bmatrix} 0.002 & 0.000 & 0.012 \\ 0.000 & 0.000 & 0.002 \\ 0.012 & 0.005 & 0.107 \end{bmatrix}.$$

If we assume that a constrained availability of imported oil (upon which the economy is totally dependent) has forced the construction and manufacturing industry (sector 3) to reduce total output by 10 percent in the South and 5 percent in the North and, further, that interindustry relationships remain the same, i.e., the technical coefficients matrix remains unchanged, the corresponding amounts of output available for final demand in the economy are found by first assembling the IRIO coefficients matrix as $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}^{NN} & \mathbf{A}^{NS} \\ \hline \mathbf{A}^{SN} & \mathbf{A}^{SS} \end{array}\right]$. The new constrained total outputs vector can be computed as $\mathbf{x}^{new} = \hat{\mathbf{r}}\mathbf{x} = \left[\begin{array}{r} 3{,}633{,}382 \\ 743{,}965 \\ 10{,}384{,}473 \\ \hline 3{,}697{,}202 \\ 766{,}751 \\ 13{,}004{,}947 \end{array}\right]$ where the vector $\mathbf{r}$ is defined as $\mathbf{r} = \left[\begin{array}{ccc|ccc} 1 & 1 & .95 & 1 & 1 & .9 \end{array}\right]$, reflecting the specified reduced total outputs for construction and manufacturing in the two regions. The corresponding new vector of final demands is found by

$$\mathbf{f}^{new} = \mathbf{x}^{new} - \mathbf{A}\mathbf{x}^{new} = \begin{bmatrix} 1,556,842 \\ 144,617 \\ 2,132,819 \\ \hdashline 1,835,571 \\ 144,444 \\ 3,107,061 \end{bmatrix}.$$

If we assume that tough import quotas imposed in Western Europe and the US on this country's goods have reduced the final demand for output from the country's construction and manufacturing industries by 15 percent in the North, the impact on the output vector for the North region (as an example) is computed by first expressing the modified final demand vector as $\mathbf{f}^{new} = \hat{\mathbf{r}}\mathbf{f} = \begin{bmatrix} 1,453,353 \\ 111,595 \\ 1,858,274 \\ \hdashline 1,663,741 \\ 76,675 \\ 3,612,485 \end{bmatrix}$ where $\mathbf{r} = \begin{bmatrix} 1 & 1 & .85 & \vdots & 1 & 1 & 1 \end{bmatrix}$, which reflects the specified reduction in final demand for construction and from the North region. The corresponding impact on total outputs is found as

$$\mathbf{x}^{new} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{f}^{new} = \left[\begin{array}{ccc:ccc} 1.145 & 0.038 & 0.567 & 0.028 & 0.012 & 0.188 \\ 0.020 & 1.014 & 0.180 & 0.007 & 0.004 & 0.054 \\ 0.348 & 0.183 & 3.218 & 0.124 & 0.058 & 0.875 \\ \hdashline 0.033 & 0.016 & 0.219 & 1.111 & 0.024 & 0.365 \\ 0.011 & 0.006 & 0.075 & 0.014 & 1.012 & 0.135 \\ 0.215 & 0.112 & 1.500 & 0.284 & 0.147 & 2.868 \end{array}\right] \begin{bmatrix} 1,453,353 \\ 111,595 \\ 1,858,274 \\ \hdashline 1,663,741 \\ 76,675 \\ 3,612,485 \end{bmatrix} = \begin{bmatrix} 3,447,445 \\ 684,913 \\ 9,875,755 \\ \hdashline 3,625,440 \\ 742,265 \\ 13,957,983 \end{bmatrix}$$

and for the North region, in particular, $\mathbf{x}^{N,new} = \begin{bmatrix} 3,447,445 \\ 684,913 \\ 9,875,755 \end{bmatrix}$.

If, for comparison, we ignore interregional linkages by using the Leontief inverse for the North region only, i.e., using only $\mathbf{A}^{NN}$, we find $(\mathbf{I} - \mathbf{A}^{NN})^{-1} = \begin{bmatrix} 1.131 & 0.031 & 0.468 \\ 0.016 & 1.011 & 0.152 \\ 0.282 & 0.149 & 2.760 \end{bmatrix}$. In conjunction with $\mathbf{f}^{N,new} = \begin{bmatrix} 1{,}453{,}353 \\ 111{,}595 \\ 1{,}858{,}274 \end{bmatrix}$, we find $\mathbf{x}^{N,new} = (\mathbf{I} - \mathbf{A}^{NN})^{-1}\mathbf{f}^{N,new} = \begin{bmatrix} 2{,}517{,}159 \\ 417{,}336 \\ 5{,}554{,}462 \end{bmatrix}$.

Compared with the IRIO results we can conclude that interregional linkages are important in this economy since the outputs found for the three industries using the North region alone are 27, 39, and 44 percent below their corresponding values for each industry respectively using the full two-region IRIO model.

Computational Notes

We have already developed all the APL tools we need to solve this problem. We begin with specifying the IRIO matrix of transactions, `ZZ`, and the vector of total outputs, `XX`, and computing **A** and **L**.

```
ZZ←277757 3654 1710816 8293 26 179483
ZZ←ZZ,319 2412 598591 15 112 30921
ZZ←ZZ,342956 39593 6762703 45770 3499 1550298
ZZ←ZZ,7085 39 98386 255023 3821 1669107
ZZ←ZZ,177 92 15966 365 3766 669710
ZZ←ZZ,71798 7957 2017905 316256 36789 8386751
ZZ←6 6ρZZ
XX←3633382 743965 10931024 3697202 766751 14449941

L←LINV A←ZZ AMAT XX
```

```
A
     0.076     0.005     0.157     0.002     0.000     0.012
     0.000     0.003     0.055     0.000     0.000     0.002
     0.094     0.053     0.619     0.012     0.005     0.107
     0.002     0.000     0.009     0.069     0.005     0.116
     0.000     0.000     0.001     0.000     0.005     0.046
     0.020     0.011     0.185     0.086     0.048     0.580
L
     1.145     0.038     0.567     0.028     0.012     0.188
     0.020     1.014     0.180     0.007     0.004     0.054
     0.348     0.183     3.218     0.124     0.058     0.875
     0.033     0.016     0.219     1.111     0.024     0.365
     0.011     0.006     0.075     0.014     1.012     0.135
     0.215     0.112     1.500     0.284     0.147     2.868
```

The constrained vectors of total outputs, `xc`, and corresponding final demands, `fc`, are found by

```
xc←XX-0 0 0.05 0 0 0.1×XX
fc←((DIAG 6ρ1)-AA)+.×xc

   fc,             xc
1556842.1      3633382.0
 144616.6       743965.0
2132818.8     10384472.8
1835571.0      3697202.0
 144444.3       766751.0
3107061.2     13004946.9
```

For the new vector of final demands, `fnew`, we find the corresponding vector of total outputs, `xnew`, as

```
      fnew←f-0 0 0.15 0 0 0×f←XX-+/ZZ
      xnew←L+.×fnew

      f
1453353 111595 2186205 1663741 76675 3612485

      fnew,         xnew
       1453353.0      3447445.2
        111595.0       684913.3
       1858274.3      9875755.0
       1663741.0      3625440.4
         76675.0       742265.2
       3612485.0     13957983.0
```

For the North region only (ignoring interregional linkages) we can compute the matrices of technical coefficients and the Leontief inverse as

```
       AR←3 3↑AA
       R←LINV AR

AR
     0.076     0.005     0.157
     0.000     0.003     0.055
     0.094     0.053     0.619
LR
     1.131     0.031     0.468
     0.016     1.011     0.152
     0.282     0.149     2.760
```

The relative importance of interregional linkages is illustrated by

```
      xn←LR+.×3↑fnew
      DIF←100×((3↑xnew)-xn)÷3↑xnew
      xn
 2517158.7  417335.5 5554462.0
      DIF
 27.0      39.1      43.8
```

Problem 3.6: Exploring Interregional Linkages in Multiple Region IO Models

This problem illustrates some key features of the interregional linkage using data from a highly aggregated version of the 2000 China MRIO table.

Problem 3.6 Overview

Consider the following three-region, three-sector interregional transactions table:

China 2000		North Nat. Res.	North Manuf. & Const.	North Services	South Nat. Res.	South Manuf. & Const.	South Services	Rest of China Nat. Res.	Rest of China Manuf. & Const.	Rest of China Services
North	Natural Resources	1,724	6,312	406	188	1,206	86	14	49	4
	Manuf. & Const.	2,381	18,458	2,987	301	3,331	460	39	234	57
	Services	709	3,883	1,811	64	432	138	5	23	5
South	Natural Resources	149	656	42	3,564	8,828	806	103	178	15
	Manuf. & Const.	463	3,834	571	3,757	34,931	5,186	202	1,140	268
	Services	49	297	99	1,099	6,613	2,969	31	163	62
ROC	Natural Resources	9	51	3	33	254	18	1,581	3,154	293
	Manuf. & Const.	32	272	41	123	1,062	170	1,225	6,704	1,733
	Services	4	25	7	25	168	47	425	2,145	1,000
Total Output		16,651	49,563	15,011	27,866	81,253	23,667	11,661	21,107	8,910

If we denote the interindustry transactions matrix in this table by $\mathbf{Z}$ and the vector of total outputs by $\mathbf{x}$, the corresponding direct and total requirements matrices, are found by $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$ and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$, respectively. Suppose, however, for this economy all the inputs to the North region from the South region were replaced with corresponding industry production from the Rest of China (ROC) region. We would reflect this change in the transactions table by removing the transactions from the 3×3 matrix partition showing transactions from South to North (i.e., each element in that partition becomes zero) and add those transactions, element-by-element, to the partition showing transactions from ROC to the North (lower left 3×3 partition) with the rest of the table unchanged. This change corresponds to the situation where all inputs to the North from the South came instead from the Rest of China, and the resulting transactions table would be:

China 2000		North Nat. Res.	North Manuf. & Const.	North Services	South Nat. Res.	South Manuf. & Const.	South Services	Rest of China Nat. Res.	Rest of China Manuf. & Const.	Rest of China Services
North	Natural Resources	1,724	6,312	406	188	1,206	86	14	49	4
	Manuf. & Const.	2,381	18,458	2,987	301	3,331	460	39	234	57
	Services	709	3,883	1,811	64	432	138	5	23	5
South	Natural Resources	0	0	0	3,564	8,828	806	103	178	15
	Manuf. & Const.	0	0	0	3,757	34,931	5,186	202	1,140	268
	Services	0	0	0	1,099	6,613	2,969	31	163	62
ROC	Natural Resources	158	707	46	33	254	18	1,581	3,154	293
	Manuf. & Const.	494	4,106	613	123	1,062	170	1,225	6,704	1,733
	Services	53	321	105	25	168	47	425	2,145	1,000
Total Output		16,651	49,563	15,011	27,866	81,253	23,667	11,661	21,107	8,910

Note that we have not changed the vector of total outputs, $\mathbf{x}$. We can denote the revised transactions matrix as $\overline{\mathbf{Z}}$ and the revised direct and total requirements table then become $\overline{\mathbf{A}} = \overline{\mathbf{Z}}\hat{\mathbf{x}}^{-1}$ and $\overline{\mathbf{L}} = (\mathbf{I} - \overline{\mathbf{A}})^{-1}$ are the following:

Revised direct requirements:

China 2000		North Nat. Res.	North Manuf. & Const.	North Services	South Nat. Res.	South Manuf. & Const.	South Services	Rest of China Nat. Res.	Rest of China Manuf. & Const.	Rest of China Services
North	Natural Resources	0.1035	0.1273	0.0270	0.0067	0.0148	0.0036	0.0012	0.0023	0.0005
North	Manuf. & Const.	0.1430	0.3724	0.1990	0.0108	0.0410	0.0194	0.0034	0.0111	0.0064
North	Services	0.0426	0.0783	0.1206	0.0023	0.0053	0.0058	0.0004	0.0011	0.0006
South	Natural Resources	0.0000	0.0000	0.0000	0.1279	0.1087	0.0340	0.0089	0.0084	0.0017
South	Manuf. & Const.	0.0000	0.0000	0.0000	0.1348	0.4299	0.2191	0.0173	0.0540	0.0301
South	Services	0.0000	0.0000	0.0000	0.0394	0.0814	0.1255	0.0026	0.0077	0.0070
ROC	Natural Resources	0.0095	0.0143	0.0030	0.0012	0.0031	0.0008	0.1356	0.1494	0.0329
ROC	Manuf. & Const.	0.0297	0.0828	0.0408	0.0044	0.0131	0.0072	0.1050	0.3176	0.1945
ROC	Services	0.0032	0.0065	0.0070	0.0009	0.0021	0.0020	0.0364	0.1016	0.1122

Revised total requirements:

China 2003		North Nat. Res.	North Manuf. & Const.	North Services	South Nat. Res.	South Manuf. & Const.	South Services	Rest of China Nat. Res.	Rest of China Manuf. & Const.	Rest of China Services
North	Natural Resources	1.1603	0.2494	0.0929	0.0225	0.0575	0.0264	0.0064	0.0159	0.0084
North	Manuf. & Const.	0.2938	1.7104	0.3988	0.0530	0.1579	0.0840	0.0189	0.0523	0.0311
North	Services	0.0826	0.1651	1.1775	0.0114	0.0303	0.0200	0.0034	0.0092	0.0054
South	Natural Resources	0.0032	0.0077	0.0041	1.1897	0.2441	0.1081	0.0237	0.0438	0.0220
South	Manuf. & Const.	0.0137	0.0332	0.0180	0.3165	1.8924	0.4892	0.0710	0.1923	0.1136
South	Services	0.0026	0.0063	0.0034	0.0834	0.1879	1.1943	0.0137	0.0362	0.0244
ROC	Natural Resources	0.0365	0.0775	0.0367	0.0087	0.0236	0.0120	1.1966	0.2816	0.1075
ROC	Manuf. & Const.	0.1028	0.2545	0.1374	0.0258	0.0714	0.0399	0.2096	1.5757	0.3577
ROC	Services	0.0202	0.0471	0.0298	0.0060	0.0158	0.0098	0.0735	0.1930	1.1724

To illustrate the impact on total outputs of all regions and sectors for a final demand of ¥100,000 on export demand for manufactured goods produced in the North, we first specify the change in final demand as $(\Delta\mathbf{f}^N)' = \begin{bmatrix} 0 & 100 & 0 & \vdots & 0 & 0 & 0 & \vdots & 0 & 0 & 0 \end{bmatrix}$. The corresponding vector of total outputs is then $(\Delta\mathbf{x})' = (\bar{\mathbf{L}}\Delta\mathbf{f}^N)' = \begin{bmatrix} 24.9 & 171.0 & 16.5 & \vdots & 0.8 & 3.3 & 0.6 & \vdots & 7.7 & 25.4 & 4.7 \end{bmatrix}$. If we recast $\Delta\mathbf{x}$ in the format of Table 3.10 we have the following table which shows the changes in production by region and sector generated by the shift in the location of inputs to production from the South to the Rest of China:

	Produced in the North (original)			**Produced in the North (revised)**		
Sector	North	South	ROC	North	South	ROC
Nat. Res.	25.6	6.8	0.8	24.9	0.8	7.7
Mfg. &Const.	172.8	29.4	2.5	171	3.3	25.4
Services	16.9	4.5	0.5	16.5	0.6	4.7
Total	215.3	40.7	3.8	212.4	4.7	37.8

Computational Notes

We have all the APL tools needed to solve this problem. First, you can retrieve these basic MRIO tables from the online Appendix SD1 to the main text or enter it from the tables above. We presume the original MRIO transactions table is the variable `Z1` (and the vector of total outputs is `x1`) and the modified MRIO transactions table is the variable `Z2` (and the vector of total outputs is `x2`). We can compute matrices of technical coefficients and total requirements for each as

```
      L1←INV A1←Z1 AMAT x1
      L2←INV A2←Z2 AMAT x2
```

```
A1
   0.104   0.127   0.027   0.007   0.015   0.004   0.001   0.002   0.000
   0.143   0.372   0.199   0.011   0.041   0.019   0.003   0.011   0.006
   0.043   0.078   0.121   0.002   0.005   0.006   0.000   0.001   0.001
   0.009   0.013   0.003   0.128   0.109   0.034   0.009   0.008   0.002
   0.028   0.077   0.038   0.135   0.430   0.219   0.017   0.054   0.030
   0.003   0.006   0.007   0.039   0.081   0.125   0.003   0.008   0.007
   0.001   0.001   0.000   0.001   0.003   0.001   0.136   0.149   0.033
   0.002   0.005   0.003   0.004   0.013   0.007   0.105   0.318   0.194
   0.000   0.000   0.000   0.001   0.002   0.002   0.036   0.102   0.112
L1
   1.163   0.256   0.097   0.023   0.058   0.027   0.006   0.016   0.009
   0.301   1.727   0.408   0.054   0.160   0.085   0.019   0.053   0.031
   0.084   0.169   1.179   0.012   0.031   0.020   0.003   0.009   0.005
   0.033   0.068   0.032   1.192   0.250   0.111   0.024   0.046   0.023
   0.119   0.294   0.159   0.326   1.919   0.504   0.074   0.201   0.119
   0.019   0.045   0.028   0.085   0.192   1.196   0.014   0.037   0.025
   0.003   0.008   0.004   0.006   0.016   0.008   1.196   0.279   0.106
   0.010   0.024   0.013   0.018   0.048   0.027   0.207   1.568   0.353
   0.002   0.005   0.003   0.004   0.011   0.007   0.073   0.192   1.172
A2
   0.104   0.127   0.027   0.007   0.015   0.004   0.001   0.002   0.000
   0.143   0.372   0.199   0.011   0.041   0.019   0.003   0.011   0.006
   0.043   0.078   0.121   0.002   0.005   0.006   0.000   0.001   0.001
   0.000   0.000   0.000   0.128   0.109   0.034   0.009   0.008   0.002
   0.000   0.000   0.000   0.135   0.430   0.219   0.017   0.054   0.030
   0.000   0.000   0.000   0.039   0.081   0.125   0.003   0.008   0.007
   0.009   0.014   0.003   0.001   0.003   0.001   0.136   0.149   0.033
   0.030   0.083   0.041   0.004   0.013   0.007   0.105   0.318   0.194
   0.003   0.006   0.007   0.001   0.002   0.002   0.036   0.102   0.112
L2
   1.160   0.249   0.093   0.022   0.057   0.026   0.006   0.016   0.008
   0.294   1.710   0.399   0.053   0.158   0.084   0.019   0.052   0.031
   0.083   0.165   1.178   0.011   0.030   0.020   0.003   0.009   0.005
   0.003   0.008   0.004   1.190   0.244   0.108   0.024   0.044   0.022
   0.014   0.033   0.018   0.316   1.892   0.489   0.071   0.192   0.114
   0.003   0.006   0.003   0.083   0.188   1.194   0.014   0.036   0.024
   0.036   0.077   0.037   0.009   0.024   0.012   1.197   0.282   0.107
   0.103   0.254   0.137   0.026   0.071   0.040   0.210   1.576   0.358
   0.020   0.047   0.030   0.006   0.016   0.010   0.074   0.193   1.172
```

For the new vector of final demands, we can compute the vectors of total outputs for the two model configurations as

```
      Δy←0 100 0 0 0 0 0 0 0
      Δx1←L1+.×Δy
      Δx2←L2+.×Δy

    15 1⍕⍉3 9⍴Δy,Δx1,Δx2

        Δy            Δx1            Δx2
       0.0           25.6           24.9
     100.0          172.7          171.0
       0.0           16.9           16.5
       0.0            6.8            0.8
       0.0           29.4            3.3
       0.0            4.5            0.6
       0.0            0.8            7.7
       0.0            2.4           25.4
       0.0            0.5            4.7
```

Problem 3.7: Impact Analysis Using the US MRIO Model

This exercise problem illustrates application of a multiregional input-output (MRIO) model, using a three-region, five-sector version of the U.S. multiregional input-output economy (shown below and in Table A4.1-3 of Appendix S4.1 in the text).

Problem 3.7 Overview

Suppose that a new government military project is initiated in the western United States, stimulating new final demand in that region of (in millions of dollars) which we can express as $\Delta\mathbf{f}^W = \begin{bmatrix} 0 & 0 & 100 & 50 & 25 \end{bmatrix}'$. The impact on total production of all sectors in all three regions of the United States economy stimulated by this new final demand in the West, can be found by first defining the new final demand for the entire economy as

$$\left[\Delta\mathbf{f}\right]' = \left[(\Delta\mathbf{f}^E)' \;\vdots\; (\Delta\mathbf{f}^C)' \;\vdots\; (\Delta\mathbf{f}^W)'\right] = \left[0 \quad 0 \quad 0 \quad 0 \quad 0 \;\vdots\; 0 \quad 0 \quad 0 \quad 0 \quad 0 \;\vdots\; 0 \quad 0 \quad 100 \quad 50 \quad 25\right].$$

Finally, we find the vector of changes in total output as $\Delta\mathbf{x} = \bar{\mathbf{L}}\Delta\mathbf{f}$:

$$\left[\Delta\mathbf{x}\right]' = \left[0.75\ 1.125\ 23.3\ 13.2\ 8.225 \;\vdots\; 3.525\ 5.375\ 38.175\ 20.475\ 13.4 \;\vdots\; 4.825\ 4.65\ 96.45\ 68.125\ 25.6\right].$$

Five-Sector, Three-Region Multiregional Input–Output Tables for the United States (1963)

Regional Transactions (millions of dollars)

	Agric	Mining	Const & Manuf	Services	Transport & Utilities
East					
Agriculture	2,013	0	7,863	44	0
Mining	35	335	3,432	44	843
Const & Manuf	2,029	400	78,164	11,561	2,333
Services	1,289	294	19,699	26,574	2,301
Transport & Util	225	384	7,232	4,026	3,534
Central					
Agriculture	10,303	0	13,218	97	0
Mining	82	472	8,686	15	1,271
Const & Manuf	4,422	1,132	93,816	10,155	2,401
Services	4,952	2,378	21,974	22,358	2,473
Transport & Util	667	406	9,296	3,468	4,513
West					
Agriculture	2,915	0	3,452	65	0
Mining	4	292	2,503	0	353
Const & Manuf	1,214	466	27,681	4,925	1,015
Services	1,307	721	8,336	10,809	991
Transport & Util	338	160	2,936	1,659	1,576

Commodity Trade Flows and Total Outputs (millions of dollars)

	East	West	Central
Agriculture			
East	6,007	2,124	208
West	3,845	28,885	2,521
Central	403	2,922	7,028
Mining			
East	2,904	415	53
West	1,108	10,942	271
Central	71	772	3,996
Const & Manuf			
East	158,679	42,150	8,368
West	44,589	201,025	11,778
Central	4,702	6,726	61,385
Services			
East	146,336	16,116	2,955
West	9,328	121,079	3,185
Central	1,939	3,643	58,663
Transp & Util			
East	21,434	4,974	263
West	4,396	23,811	1,948
Central	1,009	1,334	9,635
Total Output			
Agriculture	10,259	33,939	9,753
Mining	4,084	12,129	4,319
Const & Manuf	207,948	249,840	81,512
Services	157,468	140,850	64,803
Transport & Util	26,847	30,130	11,841

Computational Notes

We presume the MRIO data (intraregional transactions `ZZ` and commodity flows `QQ`) are stored in three-dimensional arrays as earlier. For this case we have

```
      ZZ
 2013    0  7863    44    0
   35  335  3432    44  843
 2029  400 78164 11561 2333
 1289  294 19699 26574 2301
  225  384  7232  4026 3534

10303    0 13218    97    0
```

```
  82  472  8686    15 1271
4422 1132 93816 10155 2401
4952 2378 21974 22358 2473
 667  406  9296  3468 4513

2915    0  3452    65    0
   4  292  2503     0  353
1214  466 27681  4925 1015
1307  721  8336 10809  991
 338  160  2936  1659 1576

      QQ
  6007    2124    208
  3845   28885   2521
   403    2922   7028

  2904     415     53
  1108   10942    271
    71     772    399

158679   42150   8368
 44589  201025  11778
  4702    6726  61385

146336   16116   2955
  9328  121079   3185
  1939    3643  58663

 21434    4974    263
  4396   23811   1948
  1009    1334   9635
```

We presume the vector of total outputs is stored in a matrix the rows of which indicate industry sectors and the columns regions

```
      XX
 10259   33939   9753
  4084   12129   4319
207948  249840  81512
157468  140850  64803
 26847   30130  11841
```

Using the APL tools we developed above we can construct the corresponding **Z**, **C**, and **L** matrices as

```
      C←Q AMAT q←+⌿Q←GENQ QQ
      A←(Z←GENZ ZZ) AMAT x←,⍉XX
      L←(LINV C+.×A)+.×C

x
10259   4084 207948 157468  26847  33939  12129 249840 140850  30130   9753   4319  81512  64803  11841
```

```
q
10255   4083 207970 157603  26839  33931  12129 249901 140838  30119   9757    723  81531  64803  11846
C
 0.586 0.000 0.000 0.000 0.000 0.063 0.000 0.000 0.000 0.000 0.021 0.000 0.000 0.000 0.000
 0.000 0.711 0.000 0.000 0.000 0.000 0.034 0.000 0.000 0.000 0.000 0.073 0.000 0.000 0.000
 0.000 0.000 0.763 0.000 0.000 0.000 0.000 0.169 0.000 0.000 0.000 0.000 0.103 0.000 0.000
 0.000 0.000 0.000 0.929 0.000 0.000 0.000 0.000 0.114 0.000 0.000 0.000 0.000 0.046 0.000
 0.000 0.000 0.000 0.000 0.799 0.000 0.000 0.000 0.000 0.165 0.000 0.000 0.000 0.000 0.022
 0.375 0.000 0.000 0.000 0.000 0.851 0.000 0.000 0.000 0.000 0.258 0.000 0.000 0.000 0.000
 0.000 0.271 0.000 0.000 0.000 0.000 0.902 0.000 0.000 0.000 0.000 0.375 0.000 0.000 0.000
 0.000 0.000 0.214 0.000 0.000 0.000 0.000 0.804 0.000 0.000 0.000 0.000 0.144 0.000 0.000
 0.000 0.000 0.000 0.059 0.000 0.000 0.000 0.000 0.860 0.000 0.000 0.000 0.000 0.049 0.000
 0.000 0.000 0.000 0.000 0.164 0.000 0.000 0.000 0.000 0.791 0.000 0.000 0.000 0.000 0.164
 0.039 0.000 0.000 0.000 0.000 0.086 0.000 0.000 0.000 0.000 0.720 0.000 0.000 0.000 0.000
 0.000 0.017 0.000 0.000 0.000 0.000 0.064 0.000 0.000 0.000 0.000 0.552 0.000 0.000 0.000
 0.000 0.000 0.023 0.000 0.000 0.000 0.000 0.027 0.000 0.000 0.000 0.000 0.753 0.000 0.000
 0.000 0.000 0.000 0.012 0.000 0.000 0.000 0.000 0.026 0.000 0.000 0.000 0.000 0.905 0.000
 0.000 0.000 0.000 0.000 0.038 0.000 0.000 0.000 0.000 0.044 0.000 0.000 0.000 0.000 0.813
A
 0.196 0.000 0.038 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.003 0.082 0.017 0.000 0.031 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.198 0.098 0.376 0.073 0.087 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.126 0.072 0.095 0.169 0.086 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.022 0.094 0.035 0.026 0.132 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.304 0.000 0.053 0.001 0.000 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.002 0.039 0.035 0.000 0.042 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.130 0.093 0.376 0.072 0.080 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.146 0.196 0.088 0.159 0.082 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.020 0.033 0.037 0.025 0.150 0.000 0.000 0.000 0.000 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.299 0.000 0.042 0.001 0.000
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.068 0.031 0.000 0.030
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.124 0.108 0.340 0.076 0.086
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.134 0.167 0.102 0.167 0.084
 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.035 0.037 0.036 0.026 0.133
L
 0.681 0.003 0.032 0.003 0.003 0.101 0.002 0.016 0.002 0.002 0.043 0.002 0.010 0.001 0.001
 0.005 0.759 0.016 0.002 0.023 0.002 0.039 0.007 0.001 0.008 0.002 0.083 0.007 0.001 0.004
 0.197 0.112 1.142 0.098 0.108 0.094 0.049 0.369 0.042 0.056 0.067 0.047 0.238 0.028 0.033
 0.136 0.091 0.139 1.118 0.104 0.058 0.043 0.065 0.165 0.042 0.036 0.034 0.043 0.070 0.018
 0.028 0.073 0.045 0.030 0.907 0.014 0.015 0.025 0.011 0.215 0.009 0.014 0.014 0.004 0.036
 0.596 0.008 0.055 0.006 0.007 1.190 0.010 0.086 0.008 0.009 0.462 0.008 0.046 0.005 0.006
 0.010 0.305 0.023 0.003 0.021 0.013 0.944 0.046 0.005 0.043 0.009 0.411 0.027 0.003 0.023
 0.175 0.085 0.469 0.048 0.068 0.226 0.135 1.238 0.099 0.113 0.129 0.086 0.330 0.037 0.056
 0.116 0.077 0.066 0.089 0.036 0.202 0.203 0.138 1.011 0.097 0.093 0.099 0.053 0.074 0.035
 0.026 0.030 0.029 0.011 0.215 0.037 0.040 0.052 0.028 0.912 0.025 0.026 0.025 0.010 0.215
 0.079 0.001 0.009 0.001 0.001 0.153 0.002 0.013 0.001 0.002 0.940 0.004 0.045 0.005 0.004
 0.001 0.021 0.003 0.000 0.002 0.002 0.069 0.004 0.001 0.004 0.004 0.577 0.020 0.002 0.018
 0.025 0.010 0.054 0.007 0.011 0.034 0.017 0.063 0.008 0.013 0.140 0.079 1.041 0.088 0.092
 0.020 0.009 0.012 0.019 0.009 0.034 0.022 0.016 0.038 0.011 0.157 0.118 0.129 1.080 0.097
 0.006 0.006 0.006 0.003 0.050 0.009 0.006 0.007 0.003 0.059 0.039 0.027 0.041 0.029 0.920
fnew
     0     0     0     0     0     0     0     0     0     0     0     0   100    50    25
xnew
   1.0   0.8  26.0   8.2   2.5   5.0   3.4  36.3   9.9   8.4   4.8   2.6 110.8  69.3  28.5
```

Problem 3.8: Impact Analysis Using the Japanese IRIO Model

This problem explores the use of an interregional input-output (IRIO) model for impact analysis using the three-region, five-sector version of an interregional input-output economy of Japan for 1965 given in Table A4.1-1 of Appendix S4.1.

Problem 3.8 Overview

We begin with the vector of changes in total final demand

$$[\Delta\mathbf{f}]' = \left[(\Delta\mathbf{f}^{C})' \;\vdots\; (\Delta\mathbf{f}^{N})' \;\vdots\; (\Delta\mathbf{f}^{S})'\right] = \left[0\;0\;0\;0\;0 \;\vdots\; 0\;0\;0\;0\;0 \;\vdots\; 0\;0\;\;100\;\;50\;\;25\right]$$

This is the same vector as that used in Problem 3.6, but used in this case for the Japanese IRIO economy where the regions are Central, North, and South. Using $\Delta\mathbf{f}$, we find the corresponding vector of total outputs, $\Delta\mathbf{x} = \mathbf{L}\Delta\mathbf{f}$, for this very interconnected interregional economy:

$$[\Delta\mathbf{x}]' = [.386\;\;.024\;\;13.061\;\;0.892\;\;3.024\;\;\vdots\;.145\;\;.021\;\;3.669\;\;1.376\;\;.339\;\;\vdots\;3.634\;\;.475\;\;181.630\;\;56.029\;\;42.904]$$

Computational Notes

We presume basic data for this IRIO Japanese economy are saved in global variable `A` for the matrix of interregional technical coefficients. We can now compute

```
L←INV A
x2←L+.×f2←(10⍴0),0 0 100 50 2
```

```
    6 3⍕A
0.053 0.000 0.009 0.011 0.009 0.001 0.000 0.007 0.000 0.001 0.001 0.000 0.001 0.000 0.000
0.000 0.001 0.001 0.001 0.002 0.000 0.000 0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.428 0.723 0.250 0.240 0.180 0.012 0.004 0.052 0.001 0.013 0.017 0.005 0.044 0.000 0.014
0.000 0.001 0.010 0.090 0.012 0.000 0.000 0.002 0.015 0.001 0.000 0.000 0.001 0.007 0.001
0.012 0.029 0.042 0.117 0.125 0.000 0.001 0.015 0.001 0.010 0.000 0.000 0.007 0.001 0.014
0.004 0.000 0.000 0.000 0.000 0.089 0.001 0.017 0.039 0.021 0.002 0.000 0.000 0.000 0.000
0.000 0.000 0.000 0.000 0.000 0.002 0.005 0.002 0.007 0.011 0.000 0.000 0.000 0.000 0.000
0.068 0.041 0.020 0.000 0.002 0.362 0.521 0.160 0.233 0.129 0.034 0.028 0.012 0.000 0.001
0.000 0.002 0.000 0.014 0.000 0.000 0.008 0.010 0.025 0.011 0.000 0.000 0.000 0.023 0.000
0.003 0.034 0.001 0.000 0.001 0.010 0.033 0.027 0.095 0.103 0.002 0.008 0.000 0.000 0.001
0.002 0.000 0.002 0.000 0.000 0.002 0.000 0.006 0.000 0.000 0.072 0.000 0.011 0.016 0.010
0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.001 0.000 0.000 0.001 0.004 0.001 0.002 0.004
0.036 0.021 0.082 0.000 0.013 0.012 0.012 0.056 0.000 0.007 0.473 0.719 0.303 0.264 0.196
0.000 0.000 0.001 0.024 0.000 0.000 0.000 0.001 0.022 0.000 0.000 0.003 0.009 0.068 0.012
0.001 0.005 0.006 0.000 0.003 0.000 0.001 0.009 0.000 0.003 0.012 0.050 0.037 0.112 0.110
      6 3⍕L
1.064 0.012 0.014 0.019 0.014 0.006 0.006 0.011 0.003 0.003 0.003 0.003 0.003 0.001 0.001
0.001 1.002 0.002 0.002 0.003 0.001 0.001 0.001 0.000 0.000 0.000 0.000 0.000 0.000 0.000
0.639 1.016 1.379 0.413 0.300 0.064 0.066 0.109 0.042 0.044 0.081 0.084 0.098 0.040 0.050
0.008 0.014 0.016 1.106 0.019 0.002 0.003 0.005 0.019 0.003 0.002 0.003 0.003 0.010 0.003
0.050 0.088 0.071 0.169 1.161 0.013 0.018 0.029 0.014 0.019 0.012 0.016 0.019 0.011 0.024
0.007 0.003 0.001 0.001 0.000 1.108 0.015 0.024 0.053 0.030 0.004 0.001 0.001 0.002 0.000
0.000 0.001 0.000 0.000 0.000 0.004 1.007 0.003 0.009 0.013 0.000 0.000 0.000 0.000 0.000
0.109 0.086 0.038 0.019 0.013 0.489 0.648 1.218 0.336 0.200 0.060 0.055 0.025 0.018 0.009
0.001 0.004 0.001 0.017 0.000 0.005 0.016 0.013 1.031 0.015 0.001 0.001 0.001 0.026 0.001
0.008 0.042 0.003 0.003 0.002 0.028 0.059 0.039 0.120 1.123 0.005 0.011 0.001 0.004 0.002
0.006 0.005 0.006 0.003 0.002 0.007 0.006 0.010 0.004 0.002 1.088 0.015 0.019 0.026 0.017
0.000 0.000 0.000 0.000 0.000 0.001 0.001 0.001 0.000 0.000 0.002 1.006 0.002 0.003 0.005
0.149 0.170 0.178 0.071 0.063 0.075 0.088 0.128 0.050 0.038 0.772 1.098 1.488 0.481 0.352
0.003 0.004 0.004 0.030 0.002 0.002 0.002 0.003 0.026 0.001 0.008 0.015 0.015 1.081 0.018
0.013 0.021 0.018 0.010 0.009 0.009 0.012 0.019 0.010 0.008 0.049 0.106 0.065 0.157 1.142
```

```
       f2           x2
     0.000        0.386
     0.000        0.024
     0.000       13.061
     0.000        0.892
     0.000        3.024
     0.000        0.145
     0.000        0.021
     0.000        3.669
     0.000        1.376
     0.000        0.339
```

```
  0.000          3.634
  0.000          0.475
100.000        181.630
 50.000         56.029
 25.000         42.904
```

Problem 3.9: Analysis with 4-Region (China, Japan, US, & Other Asia) IRIO Model

This problem explores IRIO analysis using the 4-region, three sector IRIO model for China, Japan, the United States, and an aggregation of other Asian nations including Indonesia, Malaysia, the Philippines, Singapore, and Thailand for the year 2000.

Problem 3.9 Overview

The interindustry transactions and total outputs are specified in the following table.

2000		United States			Japan			China			Rest of Asia		
		Nat Res	Manuf & Const	Services	Nat Res	Manuf & Const	Services	Nat Res	Manuf & Const	Services	Nat Res	Manuf & Const	Services
US	Nat Res	75,382	296,016	17,829	351	4,764	473	174	403	17	103	2,740	83
	Manuf & Const	68,424	1,667,042	960,671	160	21,902	3,775	587	8,863	1,710	383	45,066	4,391
	Services	95,115	1,148,999	3,094,357	118	6,695	807	160	1,466	296	197	7,393	953
Japan	Nat Res	7	52	53	8,721	78,936	11,206	13	66	2	14	180	27
	Manuf & Const	859	41,484	11,337	28,088	1,414,078	484,802	764	20,145	2,809	462	72,258	4,108
	Services	97	4,390	1,424	24,901	662,488	1,001,832	107	2,763	335	270	7,816	1,189
China	Nat Res	72	343	147	50	2,316	229	49,496	183,509	15,138	102	2,430	99
	Manuf & Const	331	15,657	6,442	93	10,199	1,989	89,384	892,227	181,932	157	15,093	1,237
	Services	38	2,218	1,099	17	1,780	280	25,391	210,469	136,961	23	2,078	132
ROA	Nat Res	322	1,068	203	64	11,906	266	64	1,475	14	12,153	92,647	6,402
	Manuf & Const	503	56,287	18,129	278	35,418	3,562	1,141	41,496	4,685	23,022	566,274	144,417
	Services	152	4,578	1,921	41	3,982	447	138	3,669	422	15,163	213,470	239,053
Total Output		468,403	5,866,935	11,609,307	140,622	3,883,455	4,658,191	408,153	2,000,741	702,248	173,080	1,727,367	1,225,460

The table of direct requirements, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$, is the following:

2000		United States			Japan			China			Rest of Asia		
		Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services
U.S.	Nat. Res.	0.1609	0.0505	0.0015	0.0025	0.0012	0.0001	0.0004	0.0002	0.0000	0.0006	0.0016	0.0001
	Manuf. & Const.	0.1461	0.2841	0.0828	0.0011	0.0056	0.0008	0.0014	0.0044	0.0024	0.0022	0.0261	0.0036
	Services	0.2031	0.1958	0.2665	0.0008	0.0017	0.0002	0.0004	0.0007	0.0004	0.0011	0.0043	0.0008
Japan	Nat. Res.	0.0000	0.0000	0.0000	0.0620	0.0203	0.0024	0.0000	0.0000	0.0000	0.0001	0.0001	0.0000
	Manuf. & Const.	0.0018	0.0071	0.0010	0.1997	0.3641	0.1041	0.0019	0.0101	0.0040	0.0027	0.0418	0.0034
	Services	0.0002	0.0007	0.0001	0.1771	0.1706	0.2151	0.0003	0.0014	0.0005	0.0016	0.0045	0.0010
China	Nat. Res.	0.0002	0.0001	0.0000	0.0004	0.0006	0.0000	0.1213	0.0917	0.0216	0.0006	0.0014	0.0001
	Manuf. & Const.	0.0007	0.0027	0.0006	0.0007	0.0026	0.0004	0.2190	0.4459	0.2591	0.0009	0.0087	0.0010
	Services	0.0001	0.0004	0.0001	0.0001	0.0005	0.0001	0.0622	0.1052	0.1950	0.0001	0.0012	0.0001
ROA	Nat. Res.	0.0007	0.0002	0.0000	0.0005	0.0031	0.0001	0.0002	0.0007	0.0000	0.0702	0.0536	0.0052
	Manuf. & Const.	0.0011	0.0096	0.0016	0.0020	0.0091	0.0008	0.0028	0.0207	0.0067	0.1330	0.3278	0.1178
	Services	0.0003	0.0008	0.0002	0.0003	0.0010	0.0001	0.0003	0.0018	0.0006	0.0876	0.1236	0.1951

The table of total requirements is $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$:

2000		United States			Japan			China			Rest of Asia		
		Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services	Nat. Res.	Manuf. & Const.	Services
U.S.	Nat. Res.	1.2103	0.0889	0.0126	0.0043	0.0036	0.0008	0.0013	0.0019	0.0010	0.0022	0.0071	0.0016
	Manuf. & Const.	0.2953	1.4643	0.1660	0.0071	0.0164	0.0039	0.0082	0.0183	0.0113	0.0147	0.0641	0.0163
	Services	0.4140	0.4158	1.4113	0.0054	0.0096	0.0021	0.0040	0.0083	0.0051	0.0079	0.0289	0.0076
Japan	Nat. Res.	0.0002	0.0005	0.0001	1.0755	0.0366	0.0082	0.0004	0.0010	0.0006	0.0007	0.0027	0.0006
	Manuf. & Const.	0.0085	0.0201	0.0048	0.3924	1.6463	0.2196	0.0161	0.0420	0.0233	0.0235	0.1117	0.0238
	Services	0.0026	0.0061	0.0015	0.3280	0.3663	1.3236	0.0052	0.0136	0.0074	0.0090	0.0345	0.0083
China	Nat. Res.	0.0007	0.0012	0.0004	0.0013	0.0024	0.0005	1.1997	0.2184	0.1025	0.0020	0.0061	0.0013
	Manuf. & Const.	0.0043	0.0096	0.0028	0.0046	0.0108	0.0027	0.5518	2.0243	0.6666	0.0077	0.0317	0.0075
	Services	0.0009	0.0021	0.0007	0.0011	0.0026	0.0006	0.1649	0.2816	1.3374	0.0018	0.0071	0.0016
ROA	Nat. Res.	0.0015	0.0018	0.0005	0.0024	0.0070	0.0011	0.0022	0.0060	0.0028	1.0906	0.0914	0.0205
	Manuf. & Const.	0.0081	0.0239	0.0062	0.0101	0.0261	0.0051	0.0255	0.0711	0.0368	0.2440	1.5530	0.2293
	Services	0.0023	0.0055	0.0015	0.0028	0.0070	0.0014	0.0061	0.0166	0.0086	0.1562	0.2487	1.2798

If we assume that annual final demand growth in China is 8 percent, growth in the U.S. and Japan is 4 percent, and that of other Asian nations is 3 percent, we can compute the original and projected final demand vectors as the following. The original vector of final demands is computed by $\mathbf{f} = \mathbf{x} - \mathbf{Zi}$, so

$$\mathbf{f}' = \left[70{,}067 \;\; 3{,}083{,}962 \;\; 7{,}252{,}751 \;\vdots\; 41{,}344 \;\; 1{,}802{,}261 \;\; 2{,}950{,}579 \;\vdots\; 154{,}222 \;\; 786{,}002 \;\; 321{,}762 \;\vdots\; 46{,}495 \;\; 832{,}154 \;\; 742{,}423 \right]$$

For a level of growth in China at 8 percent, in the U.S. and Japan at 4 percent, and in the rest of Asia at 3 percent, the final demand in the next year is found by multiplying the first three elements of f (U.S. final demand) by 1.04, the next three (Japanese final demand) by 1.04, the next three (Chinese final demand) by 1.08, and the last three (final demand for the other nations in Asia) by 1.03, to yield

$$(\mathbf{f}^{new})' = \left[72{,}869 \;\; 3{,}207{,}321 \;\; 7{,}542{,}861 \;\vdots\; 42{,}998 \;\; 1{,}874{,}352 \;\; 3{,}068{,}602 \;\vdots\; 166{,}560 \;\; 848{,}883 \;\; 347{,}503 \;\vdots\; 47{,}890 \;\; 857{,}119 \;\; 764{,}696 \right]$$

The corresponding vector of total outputs is then found as $\mathbf{x}^{new} = \mathbf{L}\mathbf{f}^{new}$:

$$(\mathbf{x}^{new})' = \left[487{,}149 \;\; 6{,}101{,}723 \;\; 12{,}073{,}729 \;\vdots\; 146{,}262 \;\; 4{,}039{,}397 \;\; 4{,}844{,}723 \;\vdots\; 440{,}002 \;\; 2{,}156{,}077 \;\; 757{,}348 \;\vdots\; 178{,}822 \;\; 1{,}784{,}590 \;\; 1{,}263{,}502 \right]$$

Finally, the vector of the percentage growth in total output for each and all regions and sectors is then found as

$$100 \times \frac{(\mathbf{x}^{new} - \mathbf{x})}{\mathbf{x}} = \left[4.002 \;\; 4.002 \;\; 4.000 \;\vdots\; 4.011 \;\; 4.016 \;\; 4.004 \;\vdots\; 7.803 \;\; 7.764 \;\; 7.846 \;\vdots\; 3.317 \;\; 3.313 \;\; 3.104 \right]$$

Computational Notes

We presume basic data for this Asian IRIO table are saved in global variable Z for the interregional interindustry transactions and x for total outputs.

```
L←INV A←Z AMAT x
f←x-+/Z
f2←f×1.04 1.04 1.04 1.04 1.04 1.04 1.08 1.08 1.08 1.03 1.03 1.03
x2←L+.×f2
Δxp←100×(x2-x)÷x
```

```
      8 0⍕x
468403 586693511609307  140622 3883455 4658191  408153 2000741  702248  173080 1727367 1225460
      8 0⍕Z
```

```
  75382  296016   17829     351    4764     473     174     403      17     103    2740      83
  68424 1667042  960671     160   21902    3775     587    8863    1710     383   45066    4391
  95115 1148999 3094357     118    6695     807     160    1466     296     197    7393     953
      7      52      53    8721   78936   11206      13      66       2      14     180      27
    859   41484   11337   28088 1414078  484802     764   20145    2809     462   72258    4108
     97    4390    1424   24901  662488 1001832     107    2763     335     270    7816    1189
     72     343     147      50    2316     229   49496  183509   15138     102    2430      99
    331   15657    6442      93   10199    1989   89384  892227  181932     157   15093    1237
     38    2218    1099      17    1780     280   25391  210469  136961      23    2078     132
    322    1068     203      64   11906     266      64    1475      14   12153   92647    6402
    503   56287   18129     278   35418    3562    1141   41496    4685   23022  566274  144417
    152    4578    1921      41    3982     447     138    3669     422   15163  213470  239053
      8 0⍕f
  70067 3083962 7252751   41344 1802261 2950579  154222  786002  321762   46495  832154  742423


      8 4⍕A
 0.1609  0.0505  0.0015  0.0025  0.0012  0.0001  0.0004  0.0002  0.0000  0.0006  0.0016  0.0001
 0.1461  0.2841  0.0828  0.0011  0.0056  0.0008  0.0014  0.0044  0.0024  0.0022  0.0261  0.0036
 0.2031  0.1958  0.2665  0.0008  0.0017  0.0002  0.0004  0.0007  0.0004  0.0011  0.0043  0.0008
 0.0000  0.0000  0.0000  0.0620  0.0203  0.0024  0.0000  0.0000  0.0000  0.0001  0.0001  0.0000
 0.0018  0.0071  0.0010  0.1997  0.3641  0.1041  0.0019  0.0101  0.0040  0.0027  0.0418  0.0034
 0.0002  0.0007  0.0001  0.1771  0.1706  0.2151  0.0003  0.0014  0.0005  0.0016  0.0045  0.0010
 0.0002  0.0001  0.0000  0.0004  0.0006  0.0000  0.1213  0.0917  0.0216  0.0006  0.0014  0.0001
 0.0007  0.0027  0.0006  0.0007  0.0026  0.0004  0.2190  0.4459  0.2591  0.0009  0.0087  0.0010
 0.0001  0.0004  0.0001  0.0001  0.0005  0.0001  0.0622  0.1052  0.1950  0.0001  0.0012  0.0001
 0.0007  0.0002  0.0000  0.0005  0.0031  0.0001  0.0002  0.0007  0.0000  0.0702  0.0536  0.0052
 0.0011  0.0096  0.0016  0.0020  0.0091  0.0008  0.0028  0.0207  0.0067  0.1330  0.3278  0.1178
 0.0003  0.0008  0.0002  0.0003  0.0010  0.0001  0.0003  0.0018  0.0006  0.0876  0.1236  0.1951
      8 4⍕L
 1.2103  0.0889  0.0126  0.0043  0.0036  0.0008  0.0013  0.0019  0.0010  0.0022  0.0071  0.0016
 0.2953  1.4643  0.1660  0.0071  0.0164  0.0039  0.0082  0.0183  0.0113  0.0147  0.0641  0.0163
 0.4140  0.4158  1.4113  0.0054  0.0096  0.0021  0.0040  0.0083  0.0051  0.0079  0.0289  0.0076
 0.0002  0.0005  0.0001  1.0755  0.0366  0.0082  0.0004  0.0010  0.0006  0.0007  0.0027  0.0006
 0.0085  0.0201  0.0048  0.3924  1.6463  0.2196  0.0161  0.0420  0.0233  0.0235  0.1117  0.0238
 0.0026  0.0061  0.0015  0.3280  0.3663  1.3236  0.0052  0.0136  0.0074  0.0090  0.0345  0.0083
 0.0007  0.0012  0.0004  0.0013  0.0024  0.0005  1.1997  0.2184  0.1025  0.0020  0.0061  0.0013
 0.0043  0.0096  0.0028  0.0046  0.0108  0.0027  0.5518  2.0243  0.6666  0.0077  0.0317  0.0075
 0.0009  0.0021  0.0007  0.0011  0.0026  0.0006  0.1649  0.2816  1.3374  0.0018  0.0071  0.0016
 0.0015  0.0018  0.0005  0.0024  0.0070  0.0011  0.0022  0.0060  0.0028  1.0906  0.0914  0.0205
 0.0081  0.0239  0.0062  0.0101  0.0261  0.0051  0.0255  0.0711  0.0368  0.2440  1.5530  0.2293
 0.0023  0.0055  0.0015  0.0028  0.0070  0.0014  0.0061  0.0166  0.0086  0.1562  0.2487  1.2798

      8 0⍕f
72869 3207321 7542861   42998 1874352 3068602  166560  848883  347503   47890  857119  764696
      8 0⍕x2
487149 610172312073729  146262 4039397 4844723  440002 2156077  757348  178822 1784590 1263502
       8 3⍕Δxp
4.002   4.002   4.000   4.011   4.016   4.004   7.803   7.764   7.846   3.317   3.313   3.104
```

Problem 3.10: Exploring the Partitioned Leontief Inverse

This problem illustrates recursive use expressing the Leontief inverse of a matrix in terms of partitions of the original matrix, sometimes necessary for very large matrices (thousands of sectors).

Problem 3.10 Overview

Assume that a limited computer that can directly determine the inverse of matrices no larger than of dimension 2×2 (in practice this might be more like $5{,}000 \times 5{,}000$). For

$$\mathbf{A}=\begin{bmatrix} 0 & 0.1 & 0.3 & 0.2 & 0.2 \\ 0.1 & 0.1 & 0.1 & 0 & 0 \\ 0.2 & 0 & 0.1 & 0.3 & 0.1 \\ 0.3 & 0 & 0 & 0.1 & 0.3 \\ 0.3 & 0.2 & 0.1 & 0.1 & 0.2 \end{bmatrix}$$, we first partition the matrix $(\mathbf{I}-\mathbf{A})$ as

$$(\mathbf{I}-\mathbf{A})=\left[\begin{array}{cc:ccc} 1.0 & -0.1 & -0.3 & -0.2 & -0.2 \\ -0.1 & 0.9 & -0.1 & 0.0 & 0.0 \\ \hdashline -0.2 & 0.0 & 0.9 & -0.3 & -0.1 \\ -0.3 & 0.0 & 0.0 & 0.9 & -0.3 \\ -0.3 & -0.2 & -0.1 & -0.1 & 0.8 \end{array}\right]=\left[\begin{array}{c:c} \mathbf{E} & \mathbf{F} \\ \hdashline \mathbf{G} & \mathbf{H} \end{array}\right]$$ and then further partition the matrix $\mathbf{H}$ by

$$\mathbf{H}=\left[\begin{array}{cc:c} 0.9 & -0.3 & -0.1 \\ 0 & 0.9 & -0.3 \\ \hdashline -0.1 & -0.1 & 0.8 \end{array}\right]=\left[\begin{array}{c:c} \mathbf{H}_1 & \mathbf{H}_2 \\ \hdashline \mathbf{H}_3 & \mathbf{H}_4 \end{array}\right]$$. We then can define $(\mathbf{I}-\mathbf{A})^{-1}=\left[\begin{array}{c:c} \mathbf{S} & \mathbf{T} \\ \hdashline \mathbf{U} & \mathbf{V} \end{array}\right]$ where partitions in similar positions in $(\mathbf{I}-\mathbf{A})$ and $(\mathbf{I}-\mathbf{A})^{-1}$ have the same dimensions. From the results on the inverse of a partitioned inverse (Appendix A), we find that we need $\mathbf{E}^{-1}$ and $\mathbf{H}^{-1}$, the inverses of a 2×2 and a 3×3 matrix. Therefore, to find $\mathbf{H}^{-1}$ we again use the results on the inverse of a partitioned matrix, where $\mathbf{H}$ is partitioned as above. This requires that $\mathbf{H}^{-1}$ and $\mathbf{H}_4^{-1}$ be found; since these are 2×2 and 1×1 matrices, respectively, this is easily accomplished. Hence, we have

$$\mathbf{H}^{-1}=\left[\begin{array}{cc:c} 1.144 & 0.415 & 0.299 \\ 0.050 & 1.177 & 0.448 \\ \hdashline 0.149 & 0.199 & 1.343 \end{array}\right]$$. This in conjunction with $\mathbf{E}^{-1}$, $\mathbf{F}$ and $\mathbf{G}$ allows us to find $\mathbf{S}$, $\mathbf{T}$, $\mathbf{U}$ and $\mathbf{V}$ which comprise $(\mathbf{I}-\mathbf{A})^{-1}$: $\mathbf{S}=\begin{bmatrix} 1.566 & 0.332 \\ 0.253 & 1.172 \end{bmatrix}$, $\mathbf{T}=\begin{bmatrix} 0.638 & 0.640 & 0.711 \\ 0.231 & 0.150 & 0.148 \end{bmatrix}$,

$\mathbf{U}=\begin{bmatrix} 0.708 & 0.217 \\ 0.802 & 0.270 \\ 0.839 & 0.478 \end{bmatrix}$, and $\mathbf{V}=\begin{bmatrix} 1.441 & 0.707 & 0.622 \\ 0.388 & 1.509 & 0.815 \\ 0.525 & 0.554 & 1.733 \end{bmatrix}$. Therefore

$$(\mathbf{I}-\mathbf{A})^{-1}=\left[\begin{array}{cc:ccc} 1.566 & 0.332 & 0.638 & 0.640 & 0.711 \\ 0.253 & 1.172 & 0.231 & 0.150 & 0.148 \\ \hdashline 0.708 & 0.217 & 1.441 & 0.707 & 0.622 \\ 0.802 & 0.270 & 0.388 & 1.509 & 0.815 \\ 0.839 & 0.478 & 0.525 & 0.554 & 1.733 \end{array}\right]$$. In this problem we used the method of partitioning repeatedly (sometimes called recursive application of the method) on sub-partitions of the original four partitions of $(\mathbf{I}-\mathbf{A})$. We can in theory invert an infinitely large matrix by recursively partitioning it into smaller and smaller submatrices.

Computational Notes

In order to solve this problem, we can either manually work through successive application of the method of matrix inversion using partitioned matrices or develop a more generalized function for applying the method successively. Included in the appendix to this volume is an APL dyadic function `PINV` which takes the matrix to be inverted, `M`, as the right argument. The left argument is a scalar defining the number of rows (or columns) of the upper left partition of the matrix to be inverted, which in turn determines the numbers of rows and columns of the other matrices involved in computing the inverse by means of partitioning. For this problem we have

```
      A←5 5ρ0.1×0 1 3 2 2 1 1 1 0 0 2 0 1 3 1 3 0 0 1 3 3 2 1 1 2
      IA←(DIAG 5ρ1)-A

      R1←2 PINV ¯3 ¯3↑IA
      R2←2 PINV IA
```

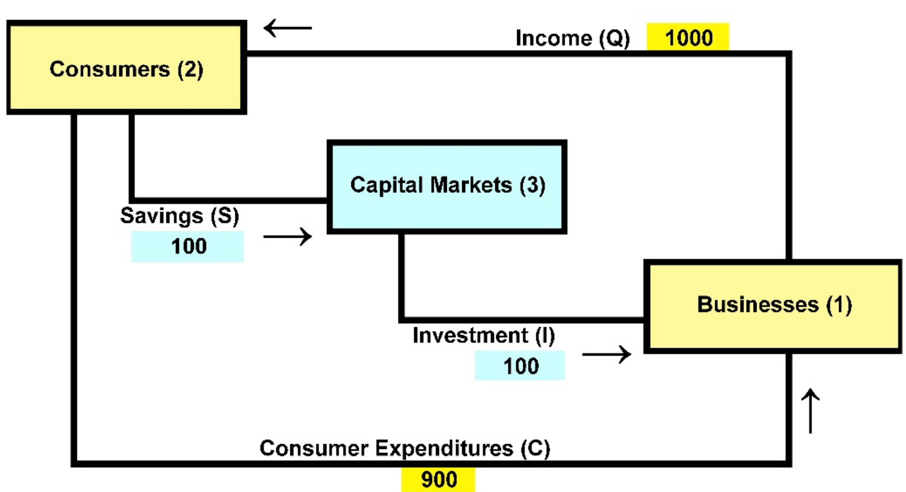
```
A
   0.0   0.1   0.3   0.2   0.2
   0.1   0.1   0.1   0.0   0.0
   0.2   0.0   0.1   0.3   0.1
   0.3   0.0   0.0   0.1   0.3
   0.3   0.2   0.1   0.1   0.2
I-A
   1.0  ¯0.1  ¯0.3  ¯0.2  ¯0.2
  ¯0.1   0.9  ¯0.1   0.0   0.0
  ¯0.2   0.0   0.9  ¯0.3  ¯0.1
  ¯0.3   0.0   0.0   0.9  ¯0.3
  ¯0.3  ¯0.2  ¯0.1  ¯0.1   0.8
PARTITIONED MATRIX:
     0.900    ¯0.300 |    ¯0.100
     0.000     0.900 |    ¯0.300
--------------------------------
    ¯0.100    ¯0.100 |     0.800
PARTITIONED INVERSE R1 IS:
     1.144     0.415 |     0.299
     0.050     1.177 |     0.448
--------------------------------
     0.149     0.199 |     1.343
PARTITIONED MATRIX:
     1.000    ¯0.100 |    ¯0.300    ¯0.200    ¯0.200
    ¯0.100     0.900 |    ¯0.100     0.000     0.000
----------------------------------------------------
    ¯0.200     0.000 |     0.900    ¯0.300    ¯0.100
    ¯0.300     0.000 |     0.000     0.900    ¯0.300
    ¯0.300    ¯0.200 |    ¯0.100    ¯0.100     0.800
PARTITIONED INVERSE R2 IS:
     1.566     0.332 |     0.638     0.640     0.711
     0.253     1.172 |     0.231     0.150     0.148
----------------------------------------------------
     0.708     0.217 |     1.441     0.707     0.622
     0.802     0.270 |     0.388     1.509     0.815
     0.839     0.478 |     0.525     0.554     1.733
```

Chapter 4, Organization of Basic Data for Input–Output Models

Chapter 4 deals with the construction of input–output tables from standardized conventions of national economic accounts, such as the widely used System of National Accounts (SNA) promoted by the United Nations, including a basic introduction to the so-called commodity-by-industry or supply-use input–output framework developed in additional detail in Chapter 5. A simplified SNA is derived from fundamental economic concepts of the circular flow of income and expenditure, that, as additional sectoral details are defined for businesses, households, government, foreign trade, and capital formation, ultimately result in the basic commodity-by-industry formulation of input–output accounts. The process is illustrated with the US input–output model and some of the key traditional conventions widely applied for such considerations as secondary production (multiple products or commodities produced by a business), competitive imports (commodities that are also produced domestically) versus non-competitive imports (commodities not produced domestically), trade and transportation margins on interindustry transactions, or the treatment of scrap and secondhand goods. The exercise problems for this chapter illustrate the key features of the SNA and relationships with input-output accounts and models.

Problem 4.1: Basic Concepts of the Circular Flow of Income and Expenditure

This problem illustrates the basic concepts of the circular flow of income and expenditure in a simple macroeconomy and corresponding set of national accounts. Consider a macroeconomy show below where transactions are measured in millions of dollars.



The balance equations, found by equating the sum of all flows into an account with the sum of all flows leaving the account, are the following:

$$Q = 1,000 = C + I = 900 + 100 = 1,000$$
$$C + S = 900 + 90 = Q + D = 1000 - 10 = 990$$
$$I + D = 100 - 10 = S = 90$$

The corresponding "T" account tables are the following:

Production (Domestic Product Account)			
Debits		**Credits**	
Income (Q)	1000	Sales of consumption goods (C)	900
		Sales of capital goods (I)	100
Total	1000	Total	1000

Consumption (Income and Outlay Account)			
Debits		**Credits**	
Purchases of consumption goods (C)	900	Income (Q)	1000
Savings (S)	90	Depreciation (D)	-10
Total	990	Total	990

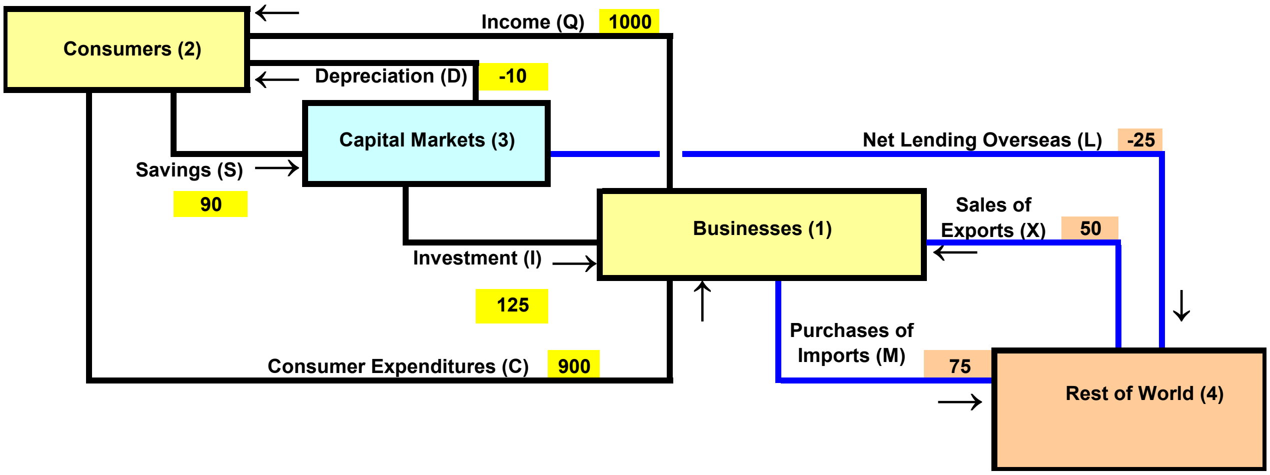
Accumulation (Capital Transactions Account)			
Debits		**Credits**	
Purchase of capital goods (I)	100	Savings (S)	90
Depreciation (D)	-10		
Total	90	Total	90

Problem 4.2: Adding Depreciation and Rest-of-World Accounts

This problem illustrates adding depreciation and rest-of-world accounts to the macroeconomy from Problem 4.1. We presume new transactions added are a capital consumption allowance to account for depreciation of capital investments of 10 percent of total investment (I), international trade allowances with a. "rest of world" account to accommodate purchases of imports of $75 million (M), sales of exports of $50 million (X), and savings made available to capital markets from overseas lenders of $25 million (L), resulting in a new total amount of capital available for businesses of $125 million. The modified balance equations for the businesses, consumers, capital, and trade accounts are:

$$Q + M = 1,000 + 75 = C + I + X = 900 + 125 + 50 = 1075$$
$$C + S = 900 + 90 = Q + D = 1,000 - 10 = 990$$
$$I + D + L = 125 - 10 - 25 = S = 90$$
$$X = W - L = 75 - 25 = 50$$

and the corresponding set of "T" accounts are the following:



Production (Domestic Product Account)			
Debits		**Credits**	
Income (Q)	1000	Sales of consumption goods (C)	900
Imports (W)	75	Sales of capital goods (I)	125
		Exports (X)	50
Total	1075	Total	1075

Consumption (Income and Outlay Account)			
Debits		**Credits**	
Purchases of consumption goods (C)	900	Income (Q)	1000
Savings (S)	90	Depreciation (D)	-10
Total	990	Total	990

Accumulation (Capital Transactions Account)			
Debits		**Credits**	
Purchase of capital goods (I)	125	Savings (S)	90
Depreciation (D)	-10		
Net Lending Overseas (L)	-25		
Total	90	Total	90

Rest of World (Balance of Payments Account)			
Debits		**Credits**	
Sales of exports (X)	50	Purchases of Imports (W)	75
		Net Overseas Lending (L)	-25
Total	50	Total	50

Problem 4.3: National Economic Balance Sheet

This problem illustrates expressing a national economic balance sheet for an economy as a collection of balance equations and as a matrix representation of the consolidated national accounts. Consider a national economic balance sheet for an economy is given by the following:

Debits					Economic Transaction	Credits				
Prod.	Cons.	Capital Accum.	Govt	Rest of World		Prod.	Cons.	Capital Accum.	Govt	Rest of World
	475				Consumption Goods (C)	475				
		54			Capital Goods (I)	54				
				46	Exports (X)	46				
46					Imports (M)					46
554					Income (Q)		554			
		-29			Depreciation (D)		-29			
	30				Savings (S)			30		
			25		Govt. Expenditures (G)	25				
	20				Taxes (T)				20	
		5			Govt Deficit Spending (B)				5	
600	525	30	25	46	Totals	600	525	30	25	46

The corresponding balance equations are:

Domestic Product Account: $Q+M=C+I+X+G$

Income and Outlay Account: $C+S+T=Q+D$

Capital Transactions Account: $I+D+B=S$

Balance of Payments Account: $X=M$

Government Account: $G=T+B$

The corresponding matrix representation of the consolidated national accounts is the following:

	Prod.	Cons.	Cap.	ROW	Govt.	Total
Production		475	54	46	25	600
Consumption	554		-29			525
Capital Accum.		30				30
Rest of World	46					46
Govt.		20	5			25
Total	600	525	30	46	25	

Problem 4.4: Double Deflation Adjustments to Interindustry Transactions

This problem illustrates the application of double deflation to adjusting interindustry transactions according the changes in relative prices.

Problem 4.4 Overview

Consider the following 4-sector input-output transactions table for the year 2015 along with industry prices for 2015 and 2020.

	Industry Transactions 1	2	3	4	Total Output	Price Year 2000	Price Year 2005
1	24	86	56	64	398	2	5
2	32	15	78	78	314	3	6
3	104	49	62	94	469	5	9
4	14	16	63	78	454	7	12

To compute the matrices of interindustry transactions and technical coefficients as well as the vector of total outputs deflated to year 2015 value terms, first, the vector of price indices is $\mathbf{p} = \begin{bmatrix} 2/5 & 3/6 & 5/9 & 7/12 \end{bmatrix} = \begin{bmatrix} 0.400 & 0.500 & 0.556 & 0.583 \end{bmatrix}$. This vector is comprised of the ratios of the year 2000 prices to the year 2005 prices. Hence, $\mathbf{Z}^{2000}$, $\mathbf{A}^{2000}$ and $\mathbf{x}^{2000}$ can be computed as

$$\mathbf{Z}^{2000} = \hat{\mathbf{p}}\mathbf{Z}^{2005} = \begin{bmatrix} .4 & 0 & 0 & 0 \\ 0 & .5 & 0 & 0 \\ 0 & 0 & .556 & 0 \\ 0 & 0 & 0 & .583 \end{bmatrix} \begin{bmatrix} 24 & 86 & 56 & 64 \\ 32 & 15 & 78 & 78 \\ 104 & 49 & 62 & 94 \\ 14 & 16 & 63 & 78 \end{bmatrix} = \begin{bmatrix} 9.6 & 34.4 & 22.4 & 25.6 \\ 16 & 7.5 & 39 & 39 \\ 57.78 & 27.22 & 34.44 & 52.22 \\ 8.17 & 9.33 & 36.75 & 45.5 \end{bmatrix},$$

$$\mathbf{A}^{2000} = \mathbf{Z}^{2000}(\hat{\mathbf{x}}^{2000})^{-1} = \hat{\mathbf{p}}\mathbf{Z}^{2005}(\hat{\mathbf{p}}\hat{\mathbf{x}}^{2005}) = \hat{\mathbf{p}}\mathbf{Z}^{2005}(\hat{\mathbf{x}}^{2005})^{-1}\hat{\mathbf{p}}^{-1} = \hat{\mathbf{p}}\mathbf{A}^{2005}\hat{\mathbf{p}}^{-1} =$$

$$\begin{bmatrix} .0603 & .2191 & .0860 & .0967 \\ .1005 & .0478 & .1497 & .1473 \\ .3629 & .1734 & .1322 & .1972 \\ .0513 & .0594 & .1410 & .1718 \end{bmatrix}, \text{ and } \mathbf{x}^{2000} = \hat{\mathbf{p}}\mathbf{x}^{2005} = \begin{bmatrix} 159.1 \\ 157 \\ 260.56 \\ 264.83 \end{bmatrix}.$$

Computational Notes

We have developed all the APL tools needed to solve this problem.

```
ph←DIAG p←2 3 5 7÷5 6 9 12
Z←4 4⍴24 86 56 64 32 15 78 78 104 49 62 94 14 16 63 78
A←Z AMAT x←398 314 469 454
Zb←ph+.×Z
Ab←ph+.×Z+.×(DIAG ÷x)+.×DIAG ÷p
xb←ph+.×x
```

```
p
     0.400      0.500      0.556      0.583
Z                                 x
        24         86         56         64
        32         15         78         78
       104         49         62         94
        14         16         63         78
x
       398        314        469        454
A
     0.060      0.274      0.119      0.141
     0.080      0.048      0.166      0.172
     0.261      0.156      0.132      0.207
```

```
      0.035      0.051      0.134      0.172
Zb
        9.6       34.4       22.4       25.6
       16.0        7.5       39.0       39.0
       57.8       27.2       34.4       52.2
        8.2        9.3       36.8       45.5
Ab
      0.060      0.219      0.086      0.097
      0.101      0.048      0.150      0.147
      0.363      0.173      0.132      0.197
      0.051      0.059      0.141      0.172
xb
      159.2      157.0      260.6      264.8
```

Problem 4.5: Sectoral Aggregation

This problem illustrates the impact of sector aggregation on the accounting for production of total outputs in an input-output economy.

Problem 4.5 Overview

Recall the interindustry transactions data given in Problem 2.7:

$$\mathbf{Z} = \begin{bmatrix} 8,565 & 8,069 & 8,843 & 3,045 & 1,124 & 276 & 230 & 3,464 \\ 1,505 & 6,996 & 6,895 & 3,530 & 3,383 & 365 & 219 & 2,946 \\ 98 & 39 & 5 & 429 & 5,694 & 7 & 376 & 327 \\ 999 & 1,048 & 120 & 9,143 & 4,460 & 228 & 210 & 2,226 \\ 4,373 & 4,488 & 8,325 & 2,729 & 2,9671 & 1,733 & 5,757 & 14,756 \\ 2,150 & 36 & 640 & 1,234 & 165 & 821 & 90 & 6,717 \\ 506 & 7 & 180 & 0 & 2,352 & 0 & 18,091 & 26,529 \\ 5,315 & 1,895 & 2,993 & 1,071 & 13,941 & 434 & 6,096 & 46,338 \end{bmatrix}$$

One way of illustrating the effects of aggregation is as follows. Using a final-demand vector of all 1's, determine the effect on total of total outputs throughout the entire economy (i.e., summed over all the sectors) by successively aggregating transactions from 8 to 7 to 6 sectors and so on (also aggregating the corresponding final-demand vector) and evaluating the relative impact on vectors of total outputs and the total of total outputs. Consider the following sequence of aggregations:

- Case 1 (8×8) No sectoral aggregation
- Case 2 (7×7) Combine sector 6 with sector 2
- Case 3 (6×6) Also combine sector 5 with sector 1
- Case 4 (5×5) Also combine sector 8 with sector 3
- Case 5 (4×4) Also combine sector 7 with previously combined 6 and 2
- Case 6 (3×3) Also combine sector 4 with previously combined 5 and 1

The impact of the sum of total outputs is indicated in the following table at each level of aggregation:

Aggregation Level	x'i	Aggregated Sector Total Output							
		1	2	3	4	5	6	7	8
8	16.26	2.31	1.84	1.12	1.6	2.88	1.43	2.26	2.82
7	16.56	2.48	3.33	1.13	1.61	2.87	2.28	2.87	
6	15.64	4.85	3.14	1.15	1.58	2.2	2.62		
5	15.62	4.78	3.11	3.86	1.58	2.29			
4	15.72	4.73	5.51	3.91	1.57				
3	15.53	6.15	5.44	3.94					

Computational Notes

To solve this problem, it is helpful to have a function that specifies an aggregation matrix from a table defining how sectors are to be aggregated. To do this efficiently, several new APL features are important. First is the distinction between numeric and textual data. So far, we have only used numeric data. Textual data in APL is contained in objects called *character arrays*. For example, to identify a string of data as text, we enclose the text in single quote characters. For example

```
      Text←'The World will continue despite the circumstances'
      Text
The World will continue despite the circumstances
```

Character arrays have many similarities with numeric arrays, but there are important differences as well. We can query the shape of the array but obviously we cannot perform numeric operations with it, as in

```
      ⍴Text
49

      100×Text
DOMAIN ERROR
      100×Text
         ^
```

Some characters can be stored as either numeric or text data, but it must be in numeric form to perform numeric operations with it. For example,

```
      11×'99'
DOMAIN ERROR
      11×'99'
          ^
      11×99
1089
```

For this problem it is helpful to use the simplest form of the powerful APL primitive operator named *execute,* denoted by the character `⍎.` In its simplest form, the execute operator converts character strings to numeric vector. For example,

```
      a←'99'
      b←⍎a
```

```
      11×b
1089
```

Of particular importance to this problem, the execute operator recognizes spaces between numeric data stored as text as delimiters separating the elements of a numeric vector, as in

```
      r←'1 3 5 9 11'
      r
1 3 5 9 11
      ⍴r
10
      s←⍎r
      s
1 3 5 9 11
      ⍴s
5
```

As a relevant aside, it is useful to note that the format operator (⍕), illustrated above in its dyadic form for formatting numeric data, does the reverse of the execute operator in its monadic form. That is, it converts numeric strings into character strings, as in

```
      a←100 200
      ⍴a
2
      b←⍕a
      ⍴b
7
```

The execute operator has many other important uses in APL, some of which will appear later in this handbook, but for the present we can use the feature just described to help with creating the aggregation matrix. Consider the following 3×3 matrix M, for which we seek to combine the first and third rows and columns in assembling an aggregated 2×2 matrix.

```
      M←3 3⍴14 75 46 53 22 5 68 68 93
      M
14 75 46
53 22  5
68 68 93
```

We create a character array with two rows corresponding to the sectors in the aggregated matrix. Each row will contain the indices (as characters) for the sectors of the unaggregated matrix to be combined in the aggregated matrix. Character arrays like this can be created in a variety of ways depending upon the APL implementation, but very simply for this illustration, it could be

```
      CODE←2 4⍴'1 3 2     '
      CODE
1 3
2
      ⍴CODE
2 4
```

With such a feature and using the execute operator, we can write a monadic APL function `SCREATE` to take the character matrix just described and create the aggregation matrix necessary to combine the rows and columns of the unaggregated matrix.

```
[0] S←SCREATE C;i;n;m
[1] n←(i←1)↑ρC
[2] m←ρ⍎,C,' '
[3] S←(n,m)ρ0
[4] L1:S[i;⍎C[i;]]←1
[5] →(n≥i←i+1)/L1
```

We can now use `SCREATE` to generate the aggregation matrix **S** and use it (pre-multiplying **M** by **S** and post-multiplying it by the transpose of **S**) to produce the aggregated matrix according to the specified code:

```
      S←SCREATE CODE
      S
1 0 1
0 1 0
      S+.×M+.×⍉S
221 143
 58  22
```

For this problem we can use `SCREATE` to successively aggregate **Z** according the alternative aggregation codes specified and tally the impact of aggregation bias on the sum of total outputs along the way in a six-element vector **T** with the unaggregated total as the first element.

```
 Z←8565 8069 8843 3045 1124 276 230 3464 1505 6996 6895 3530 3383 365 219
 Z←Z,2946 98 39 5 429 5694 7 376 327 999 1048 120 9143 4460 228 210 2226
 Z←Z,4373 4488 8325 2729 29671 1733 5757 14756 2150 36 640 1234 165 821 90
 Z←Z,6717 506 7 180 0 2352 0 18091 26529 5315 1895 2993 1071 13941 434
 Z←Z,6096 46338
 Z←8 8ρZ
 x←37610 45108 46323 41059 209403 11200 55992 161079
 T←6ρ0

 L←LINV A←Z AMAT x
 T[1]←+/ΔX1←L+.×ΔY1←8ρ1

 S←S1←SCREATE C1←7 3ρ'1  2 6 3  4  5  7  8     '
 ZZ←S+.×Z+.×⍉S
 XX←S+.×X
 YY←S+.×ΔY1
 T[2]←+/ΔX2←(LINV ZZ AMAT XX)+.×YY

 S←S2←SCREATE C2←6 3ρ'1 52 63  4  7  8  '
 ZZ←S+.×Z+.×⍉S
 XX←S+.×X
 YY←S+.×ΔY1
 T[3]←+/ΔX3←(LINV ZZ AMAT XX)+.×YY
```

```
S←S3←SCREATE C3←5 3ρ'1 52 63 84  7    '
ZZ←S+.×Z+.×⍉S
XX←S+.×X
YY←S+.×ΔY1
T[4]←+/ΔX4←(LINV ZZ AMAT XX)+.×YY

S←S4←SCREATE C4←4 5ρ'1 5  2 6 73 8  4      '
ZZ←S+.×Z+.×⍉S
XX←S+.×X
YY←S+.×ΔY1
T[5]←+/ΔX5←(LINV ZZ AMAT XX)+.×YY

S←S5←SCREATE C5←3 5ρ'1 5 42 6 73 8    '
ZZ←S+.×Z+.×⍉S
XX←S+.×X
YY←S+.×ΔY1
T[6]←+/ΔX6←(LINV ZZ AMAT XX)+.×YY


T
16.262447 16.56122 15.54133 15.624984 15.720623 15.533131
```

For this now growing list of steps, it might be more efficient to combine them all into an APL function, as in

```
[ 0] T←MB3Prob_04_05;ZZ;YY;XX;C1;C2;C3;C4;C5;A;L;S;f
[ 1] Z←8565 8069 8843 3045 1124 276 230 3464 1505 6996 6895 3530 3383 365 219
[ 2] Z←Z,2946 98 39 5 429 5694 7 376 327 999 1048 120 9143 4460 228 210 2226
[ 3] Z←Z,4373 4488 8325 2729 29671 1733 5757 14756 2150 36 640 1234 165 821 90
[ 4] Z←Z,6717 506 7 180 0 2352 0 18091 26529 5315 1895 2993 1071 13941 434
[ 5] Z←Z,6096 46338
[ 6]  Z←8 8ρZ
[ 7]  x←37610 45108 46323 41059 209403 11200 55992 161079
[ 8]  T←6ρ0
[ 9]  C1←7 3ρ'1  2 6 3  4  5  7  8     '
[10]  C2←6 3ρ'1 52 63  4  7  8  '
[11]  C3←5 3ρ'1 52 63 84  7   '
[12]  C4←4 5ρ'1 5  2 6 73 8  4      '
[13]  C5←3 5ρ'1 5 42 6 73 8   '
[14]  L←LINV A←Z AMAT x
[15]  T[i←1]←+/L+.×f←8ρ1
[16] L1:⍎'S←SCREATE C',⍕i
[17]  ZZ←S+.×Z+.×⍉S
[18]  XX←S+.×x
[19]  YY←S+.×f
[20]  T[i+1]←+/(LINV ZZ AMAT XX)+.×YY
[21]  →(5≥i←i+1)/L1

        MB3Prob_04_05
16.262447 16.56122 15.54133 15.624984 15.720623 15.533131
```

Note, however, that this function makes use of one of the additional features of the execute operator (in line `[16]`). The index number i increases with each cycle in the loop beginning with line `[16]` (labelled `L1`) for the 5 different aggregation schemes. The current index number i is converted to a text format and catenated with the text string `'S←SCREATE C'` to form the relevant string, for the first cycle, as `S←SCREATE C1`; for the second cycle, as `S←SCREATE C2`, etc. Then the execute operator "executes" the string as an APL statement. This is actually quite consistent with the earlier use where a string of character data is converted to numeric data as if it had been entered by the keyboard, as in

```
      a←'10 20 30 40'
      a
10 20 30 40
      ⍴a
11
      b←⍎a
      b
10 20 30 40
      ⍴b
4
```

Finally, as our APL statements accumulate in user defined functions it is important keep track of what is going on and to use some other features for organizing the function. First, *comments* can be inserted in a function, delimited with the character `⍝` placed before the comment which when encountered by the APL interpreter will be ignored. Another useful feature is to combine several statements on one line by delimiting them with the character `⋄`. Note, however, that statements delimited in this way are executed in the order left to right, unlike the order of execution within an individual state which is right to left. With these two conventions we could show the new function somewhat more transparently as

```
[ 0] T←MB3Prob_04_05;ZZ;YY;XX;C1;C2;C3;C4;C5;A;L;S;f
[ 1] Z←8565 8069 8843 3045 1124 276 230 3464 1505 6996 6895 3530 3383 365 219
[ 2] Z←Z,2946 98 39 5 429 5694 7 376 327 999 1048 120 9143 4460 228 210 2226
[ 3] Z←Z,4373 4488 8325 2729 29671 1733 5757 14756 2150 36 640 1234 165 821 90
[ 4] Z←Z,6717 506 7 180 0 2352 0 18091 26529 5315 1895 2993 1071 13941 434
[ 5] Z←Z,6096 46338
[ 6] Z←8 8⍴Z
[ 7] x←37610 45108 46323 41059 209403 11200 55992 161079
[ 8] ⍝Alternative aggregation codes
[ 9] C1←7 3⍴'1  2 6 3  4  5  7  8     '
[10] C2←6 3⍴'1 52 63  4  7  8  '
[11] C3←5 3⍴'1 52 63 84  7   '
[12] C4←4 5⍴'1 5  2 6 73 8  4      '
[13] C5←3 5⍴'1 5 42 6 73 8   '
[14] ⍝Accumulate total outputs for aggregations
[15] T←6⍴0
[16] L←LINV A←Z AMAT x
[17] ⍝---Unaggregated
[18] T[i←1]←+/L+.×f←8⍴1
[19] ⍝---The 5 alternative aggregation schemes
[20] L1:⍎'S←SCREATE C',⍕i
[21] ZZ←S+.×Z+.×⍉S ⋄ XX←S+.×x ⋄ YY←S+.×f
```

```
[22]  T[i+1]←+/(LINV ZZ AMAT XX)+.×YY
[23]  →(5≥i←i+1)/L1
```

Problem 4.6: Measuring Aggregation Bias

This problem illustrates the computation of first order and total aggregation bias.

Problem 4.6 Overview

The seven-sector input-output table of technical coefficients for the U.S. economy (1972) is given in Appendix SD1 (located on the supplemental resources website). Consider the following vector of final demands: $\Delta\mathbf{f} = \begin{bmatrix} 100 & 100 & 100 & 100 & 100 & 100 & 100 \end{bmatrix}'$. To compute the first order and total aggregation bias associated with, as an example, combining agriculture with mining, construction with manufacturing, and transportation-utilities with services and other sectors to yield a new three-sector model we first compute the interindustry transactions,

$$\mathbf{Z} = \mathbf{A}\hat{\mathbf{x}} = \begin{bmatrix} 26{,}370 & 9 & 465 & 41{,}257 & 377 & 2{,}768 & 193 \\ 160 & 1{,}647 & 1{,}511 & 22{,}531 & 6{,}038 & 104 & 322 \\ 579 & 857 & 50 & 3{,}273 & 5{,}887 & 13{,}734 & 2{,}676 \\ 12{,}056 & 2{,}865 & 58{,}464 & 287{,}046 & 15{,}360 & 46{,}582 & 1{,}257 \\ 5{,}172 & 1{,}462 & 17{,}314 & 59{,}830 & 36{,}984 & 23{,}082 & 3{,}256 \\ 7{,}262 & 4{,}470 & 11{,}387 & 44{,}987 & 43{,}664 & 84{,}651 & 1{,}693 \\ 193 & 191 & 697 & 8{,}906 & 4{,}453 & 5{,}013 & 532 \end{bmatrix}$$

. The aggregation matrix for the specified sectoral aggregation is $\mathbf{S} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$ so we compute the aggregated transactions and total outputs as $\mathbf{Z}^* = \mathbf{SZS}' = \begin{bmatrix} 28{,}186 & 65{,}763 & 9{,}804 \\ 16{,}358 & 348{,}834 & 85{,}496 \\ 18{,}750 & 143{,}121 & 203{,}328 \end{bmatrix}$,

$\mathbf{x}^* = \mathbf{Sx} = \begin{bmatrix} 114{,}341 \\ 927{,}192 \\ 1{,}060{,}811 \end{bmatrix}$ and $\mathbf{A}^* = \mathbf{Z}^*(\mathbf{x}^*)^{-1} = \begin{bmatrix} .247 & .071 & .009 \\ .143 & .376 & .081 \\ .164 & .154 & .192 \end{bmatrix}$, respectively. We subsequently

compute $\mathbf{L}^* = (\mathbf{I} - \mathbf{A}^*)^{-1} = \begin{bmatrix} 1.365 & 0.163 & 0.032 \\ 0.358 & 1.686 & 0.172 \\ 0.345 & 0.355 & 1.276 \end{bmatrix}$; $\tilde{\mathbf{x}}^* = \begin{bmatrix} 315.2 \\ 460.5 \\ 523.0 \end{bmatrix}$ and $\tilde{\mathbf{x}} = (\mathbf{I} - \mathbf{A})^{-1}\Delta\mathbf{f} = \begin{bmatrix} 174.5 \\ 120.9 \\ 116.5 \\ 329.7 \\ 184.6 \\ 218.3 \\ 110.1 \end{bmatrix}$.

The vector of first order bias for individual sectors is found by $\boldsymbol{\varphi} = (\mathbf{A}^{*}\mathbf{S} - \mathbf{SA})\Delta\mathbf{f} = \begin{bmatrix} 17.069 \\ 7.726 \\ 5.790 \end{bmatrix}$ and the total first order bias is $\mathbf{i}'\boldsymbol{\varphi} = 30.58$. The vector of the total aggregation bias for individual sectors is found by $\boldsymbol{\tau} = \tilde{\mathbf{x}}^{*} - \mathbf{S}\tilde{\mathbf{x}} = \begin{bmatrix} 19.751 \\ 14.326 \\ 10.025 \end{bmatrix}$ and the overall total aggregation bias is $\mathbf{i}'\boldsymbol{\tau} = 44.1$.

Computational Notes

We first retrieve the matrix of interindustry transactions **Z** and the vector of total outputs x from the textbook data Appendix SD1 and generate **A** in the usual manner, i.e., `A←Z AMAT x`.

```
Z
      26369        11       466     41271       375      2758       201
        159      1648      1506     22509      6035       120       315
        583       858        47      3245      5875     13715      2672
      12056      2866     58469    287075     15372     46564      1259
       5170      1460     17316     59839     36998     23092      3249
       7262      4471     11387     44991     43658     84626      1694
        195       190       700      8871      4460      5018       531
x
      83955     30386    165998    761194    377389    522215    161207
A
      0.314     0.000     0.003     0.054     0.001     0.005     0.001
      0.002     0.054     0.009     0.030     0.016     0.000     0.002
      0.007     0.028     0.000     0.004     0.016     0.026     0.017
      0.144     0.094     0.352     0.377     0.041     0.089     0.008
      0.062     0.048     0.104     0.079     0.098     0.044     0.020
      0.086     0.147     0.069     0.059     0.116     0.162     0.011
      0.002     0.006     0.004     0.012     0.012     0.010     0.003
```

We then use `SCREATE` to fashion the appropriate aggregation matrix **S**.

```
      S←SCREATE 3 5⍴'1 2  3 4  5 6 7'
      S
1 1 0 0 0 0 0
0 0 1 1 0 0 0
0 0 0 0 1 1 1
```

Then we can easily compute the aggregated matrix of transactions `ZS`, vector of total outputs `xs`, and matrices of technical requirements `AS` and the Leontief inverse `LS`.

```
     ZS←S+.×Z+.×⍉S
     LS←INV AS←ZS AMAT xs←S+.×x

ZS
      28186     65752      9804
      16363    348835     85457
      18748    143104    203327
```

```
AS
      0.247     0.071     0.009
      0.143     0.376     0.081
      0.164     0.154     0.192
LS
      1.365     0.163     0.032
      0.358     1.686     0.172
      0.345     0.355     1.276
```

If we denote the new test vector of final demands (all 7 elements equal to 100) as `f2`, we can compute the corresponding total outputs `x2`, and their respective aggregated values, `fs2` and `xs2`, as

```
      x2←(L←INV A)+.×f2
      xs2←LS+.×fs2←S+.×f2←7ρ100

f2
      100.0     100.0     100.0     100.0     100.0     100.0     100.0
x2
      174.5     120.9     116.5     329.7     184.6     218.2     110.1
fs2
      200.0     200.0     300.0
xs2
      315.2     460.5     523.0
```

And finally, the first order bias `FOB` and total aggregation bias `TAB`, can be computed as

```
      TAB←xs2-(S+.×x2)
      FOB←((AS+.×S)-(S+.×A))+.×f2

      FOB
17.065403 7.7211936 5.789844
      +/FOB
30.576441
      TAB
19.739274 14.316627 10.020306
      +/TAB
44.076206
```

Problem 4.7: Table Consolidated National Accounts

This problem illustrates construction of a table of consolidated national accounts in matrix form from a set of national accounting equations. Consider the following national accounting equations:

(1) $Q + M = C + I + X + G$

(2) $C + S + T = Q + D$

(3) $L + I + D + B = S$

(4) $X = M + L$

(5) $G = T + B$

where $Q =$ total consumer income payments; $M =$ purchases of imports; $C =$ total sales of consumption goods; $S =$ total consumer savings; $T =$ total taxes paid to government; $I =$ total purchases of capital goods; $D =$ total capital consumption allowance (depreciation); $L =$ net lending from overseas; $B =$ total government deficit spending; $X =$ total sales of exports; $G =$ total government purchases and the following are known: $Q = -500$, $M = 75$, $S = 60$, $T = 20$, $D = 10$, $L = -20$, and $B = 10$.

First, note that there are missing quantities C, I, X and G that are necessary to complete the table, which can be found with equations (2), (3), (4), and (5), respectively as: $C = Q + D - S - T = 410$; $I = -D + S - L - B = 80$; $X = L + M = 55$; and $G = T + B = 30$. The resulting consolidated table of national accounts represented in matrix form is the following:

	Prod.	Cons.	Cap.	ROW	Govt.	Total
Prod.		**C=410**	**I=80**	**X=55**	**G=30**	575
Cons.	Q=500		D=-10			490
Cap.		S=60				60
ROW	M=75		L=-20			55
Govt.		T=20	B=10			30
Total	575	490	60	55	30	

Problem 4.8: Supply and Use IO Tables Derived from Table of National Accounts

This problem illustrates conversion of table of national accounts to a consolidated table of supply and use input-output accounts. Consider the following table of national accounts (generated in Problem 4.7).

	Prod.	Cons.	Cap.	ROW	Govt.	Total
Prod.		410	80	55	30	575
Cons.	500		-10			490
Cap.		60				60
ROW	75		-20			55
Govt.		20	10			30
Total	575	490	60	55	30	

Suppose the following tables become available providing the interindustry supply and use detail for this economy.

Use of commodities by industries:

		Industry			Total Intermed. Output
		Nat. Res.	Manuf.	Serv.	
Commodity	Agriculture	20	12	18	50
	Mining	5	30	12	47
	Manufacturing	10	13	11	34
	Services	12	17	40	69

Final uses of commodity production:

	Households	Government	Investment	Exports
Agriculture	30	6	16	5
Mining	60	9	16	17
Manufacturing	50	3	40	22
Services	70	12	8	11
Totals	210	30	80	55

Supply of commodities by industries:

		Commodity				Total Industry Output
		Agric.	Mining	Manuf.	Services	
Industry	Natural Resources	99			10	109
	Manufacturing	8	143	137	10	298
	Services		6	12	150	168
	Total Commodity Output	107	149	149	170	575

The corresponding consolidated set of supply and use accounts including the sector detail for interindustry transactions becomes the following:

		Commodities				Industries			Final Demand	Total Output
		Agric.	Mining	Manuf.	Serv.	Nat. Res.	Manuf.	Serv.		
Comm.	Agriculture					20	12	18	57	107
	Mining					5	30	12	102	149
	Manufacturing					10	13	11	115	149
	Services					12	17	40	101	170
Ind.	Natural Resources	99			10					109
	Manufacturing	8	143	137	10					298
	Services		6	12	150					168
	Value Added					62	226	87	375	
	Total Output	107	149	149	170	109	298	168		575

Problem 4.9: Generating Domestic Transactions Tables from "US-Style" IO Tables

This problem illustrates a process of "scrubbing" a U.S. style input-output transactions table of competitive imports to yield a domestic transactions table.

Problem 4.9 Overview

We define an input-output economy with $\mathbf{Z} = \begin{bmatrix} 500 & 0 & 0 \\ 50 & 300 & 150 \\ 200 & 150 & 550 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 1{,}000 \\ 750 \\ 1{,}000 \end{bmatrix}$. We also know the vector of the total value of competitive imports, $\mathbf{m} = \begin{bmatrix} 150 \\ 105 \\ 210 \end{bmatrix}$. Knowing **m**, we can define

$\mathbf{g} = \mathbf{f} + \mathbf{m} = \begin{bmatrix} 650 & 355 & 310 \end{bmatrix}'$, which is the vector of total final demands, including imports. In some cases, the accounting is such that **m** is recorded as a negative final demand so that the vector of total outputs, $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$, reflects total domestic output, which is the convention used for the US input-output tables. Using the assumption of import similarity, we can compute the domestic transactions matrix where competitive imports are removed from interindustry transactions by the following steps. First, we compute the vectors of total final demand and intermediate outputs, $\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 500 \\ 250 \\ 100 \end{bmatrix}$ and $\mathbf{u} = \begin{bmatrix} 500 \\ 500 \\ 900 \end{bmatrix}$, respectively. The import similarity scaling factors are found for each commodity as the ratio of the value of total interindustry imports of that commodity divided by the total output (including imports), $r_i = \dfrac{m_i}{u_i + f_i}$, or

$\mathbf{r} = \begin{bmatrix} .15 & .24 & .21 \end{bmatrix}'$.

We can then compute the scaled quantities $\overline{\mathbf{D}} = \mathbf{Z} - \hat{\mathbf{r}}\mathbf{Z} = \begin{bmatrix} 425 & 0 & 0 \\ 43 & 258 & 129 \\ 158 & 118.5 & 434.5 \end{bmatrix}$,

$\overline{\mathbf{M}} = \hat{\mathbf{r}}\mathbf{Z} = \begin{bmatrix} 75 & 0 & 0 \\ 7 & 42 & 21 \\ 42 & 31.5 & 115.5 \end{bmatrix}$, $\mathbf{h} = \hat{\mathbf{r}}\mathbf{f} = \begin{bmatrix} 75 \\ 35 \\ 21 \end{bmatrix}$, and $\overline{\mathbf{g}} = \mathbf{g} - \hat{\mathbf{r}}\mathbf{f} = \begin{bmatrix} 575 \\ 320 \\ 289 \end{bmatrix}$. Note that the identity $\mathbf{x} = \overline{\mathbf{D}}\mathbf{i} + \overline{\mathbf{g}}$ (analogous to $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$) still holds, but this balance equation now accounts for only domestic transactions with interindustry imports reassigned to total value added. The new total value-added vector is $\overline{\mathbf{v}}' = \mathbf{x}' - \mathbf{i}'\overline{\mathbf{D}} = \begin{bmatrix} 374 & 373.5 & 436.5 \end{bmatrix}$, which inflates the original vector of total valued added, $\mathbf{v}' = \begin{bmatrix} 250 & 300 & 300 \end{bmatrix}$ by interindustry imports to each industry, i.e., $\bar{\mathbf{m}}' = \begin{bmatrix} 124 & 73.5 & 136.5 \end{bmatrix}$, excluding the value of imports consumed directly in final demand.

If we subsequently learn that $\mathbf{M} = \begin{bmatrix} 150 & 0 & 0 \\ 25 & 50 & 30 \\ 35 & 75 & 100 \end{bmatrix}$, i.e., we know precisely the competitive imports associated with interindustry transactions, we can compute the domestic transactions matrix (rather than approximate it with import similarity scaling factors) by $\mathbf{D} = \mathbf{Z} - \mathbf{M} = \begin{bmatrix} 350 & 0 & 0 \\ 25 & 250 & 120 \\ 165 & 75 & 450 \end{bmatrix}$ and compute $\mathbf{m} = \mathbf{Mi} = \begin{bmatrix} 150 \\ 105 \\ 210 \end{bmatrix}$, and $\mathbf{g} = \mathbf{f} + \mathbf{m} = \begin{bmatrix} 650 \\ 355 \\ 310 \end{bmatrix}$ where $\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 500 \\ 250 \\ 100 \end{bmatrix}$. Note that the balance equation $\mathbf{x} = \mathbf{Di} + \mathbf{g}$ holds here as well. Then the new total value added vector, $\tilde{\mathbf{v}}' = \mathbf{x}' - \mathbf{i}'\mathbf{D} = \begin{bmatrix} 460 & 425 & 430 \end{bmatrix}$, inflates the original vector of total valued added, $\mathbf{v}' = \mathbf{x}' - \mathbf{i}'\mathbf{Z} = \begin{bmatrix} 250 & 300 & 300 \end{bmatrix}$, by the total value of all imports to each industry, $\tilde{\mathbf{m}}' = \begin{bmatrix} 210 & 125 & 130 \end{bmatrix}$, i.e. $\tilde{\mathbf{m}}' = \mathbf{i}'\mathbf{M} = \tilde{\mathbf{v}}' - \mathbf{v}'$.

We compute the Leontief inverse for the first case as $\mathbf{L}^I = (\mathbf{I} - \mathbf{A}^I)^{-1} = \begin{bmatrix} 1.739 & 0 & 0 \\ .222 & 1.613 & .368 \\ .548 & .451 & 1.871 \end{bmatrix}$ for $\mathbf{A}^I = \bar{\mathbf{D}}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .425 & 0 & 0 \\ .043 & .344 & .129 \\ .158 & .158 & .435 \end{bmatrix}$ and for the second case as $\mathbf{L}^{II} = (\mathbf{I} - \mathbf{A}^{II})^{-1} = \begin{bmatrix} 1.538 & 0 & 0 \\ .146 & 1.551 & .338 \\ .488 & .282 & 1.88 \end{bmatrix}$ for $\mathbf{A}^{II} = \mathbf{D}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .35 & 0 & 0 \\ .025 & .333 & .12 \\ .165 & .1 & .45 \end{bmatrix}$.

The mean absolute deviation (*mad*) between $\mathbf{A}^I$ and $\mathbf{A}^{II}$ is $mad(\mathbf{A}) = (1/9)\sum_{i=1}^{3}\sum_{j=1}^{3}\left|a_{ij}^I - a_{ij}^{II}\right| = .0215$ and the *mad* between $\mathbf{L}^I$ and $\mathbf{L}^{II}$ is found to be $mad(\mathbf{L}) = (1/9)\sum_{i=1}^{3}\sum_{j=1}^{3}\left|l_{ij}^I - l_{ij}^{II}\right| = .0673$.

Computational Notes

We first create **Z** and **x**, from which we can create the vectors of final demands **f**, intermediate outputs **u**, and value-added inputs **v**.

```
Z←3 3⍴500 0 0 50 300 150 200 150 550
x←1000 750 1000
```

```
      f←x-u←+/Z
      v←x-+⌿Z

Z
       500         0         0
        50       300       150
       200       150       550
x
      1000       750      1000
v
       250       300       300
u
       500       500       900
f
       500       250       100
```

Then, knowing the vector of domestic imports m, we can compute the vector of domestic scaling factors r, and the associated domestic data.

```
      m←150 105 210
      g←f+m
      r←m÷(u+f)
      DB←Z-(rh←DIAG r)+.×Z
      MB←rh+.×Z
      h←rh+.×f
      gb←g-rh+.×f
      vb←x-+⌿DB
      mb←+⌿MB

m
150 105 210
g
650 355 310
r
0.15 0.14 0.21
DB
     425.0       0.0       0.0
      43.0     258.0     129.0
     158.0     118.5     434.5
MB
      75.0       0.0       0.0
       7.0      42.0      21.0
      42.0      31.5     115.5
h
      75.0      35.0      21.0
gb
     575.0     320.0     289.0
vb
     374.0     373.5     436.5
mb
     124.0      73.5     136.5
```

Next, we presume we know the matrix of imports **M** and compute the "real" domestic transactions matrix **D**, the associated vectors of value-added inputs `vt` and of the total value of imports `mt`.

```
      M←3 3⍴150 0 0 25 50 30 35 75 100
      D←Z-M
      vt←x-+⌿D
      mt←vt-v

 M
        150         0         0
         25        50        30
         35        75       100
D
      350.0       0.0       0.0
       25.0     250.0     120.0
      165.0      75.0     450.0
vt
      460.0     425.0     430.0
mt
      210.0     125.0     130.0
```

Finally, we compute the matrices of technical coefficients using the estimated and actual matrices of transactions, `DB` and `D`, as `A1` and `A2`, respectively, and the associated Leontief inverses, `L1` and `L2`, as well as the `MAD` measures between them.

```
      L1←LINV A1←DB AMAT x
      L2←LINV A2←D AMAT x
      MAD←((+/+/|A1-A2)÷9),(+/+/|L1-L2)÷9
A1
      0.425     0.000     0.000
      0.043     0.344     0.129
      0.158     0.158     0.435
L1
      1.739     0.000     0.000
      0.222     1.613     0.368
      0.548     0.451     1.871
A2
      0.350     0.000     0.000
      0.025     0.333     0.120
      0.165     0.100     0.450
L3
      1.538     0.000     0.000
      0.146     1.551     0.338
      0.488     0.282     1.880
MAD
0.021462963  0.067318963
```

Problem 4.10: Spatial Aggregation

This problem illustrates the calculation of spatial aggregation bias.

Problem 4.10 Overview

Recall the three-region, three-sector Chinese interregional model (for the year 2000) specified in Problem 3.6. Using that table as a point of departure, we aggregate regions 1 and 2 and leave region 3 unaggregated to yield a two-region model.

To aggregate the North and South regions and leave the Rest of China region unaggregated, we construct the aggregation matrix $\mathbf{S}=\left[\begin{array}{ccc|ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$ and compute the aggregated matrix of interindustry transactions, $\mathbf{Z}^{(a)}=\mathbf{SZS}'$, and the aggregated vector of total outputs, $\mathbf{x}^{(a)}=\mathbf{Sx}$, which are shown in the following table:

China 2000		North and South: Nat. Res.	North and South: Manuf & Const.	North and South: Services	Rest of China: Nat. Res.	Rest of China: Manuf & Const.	Rest of China: Services
North & South	Nat. Resources	5,625	17,001	1,339	117	227	19
	Manuf. & Const.	6,902	60,554	9,203	241	1,374	325
	Services	1,920	11,225	5,017	35	186	68
Rest of China	Nat. Resources	43	305	22	1,581	3,154	293
	Manuf. & Const.	155	1,334	212	1,225	6,704	1,733
	Services	29	193	53	425	2,145	1,000
Total Chinese Output		44,517	130,816	38,678	11,661	21,107	8,910

The corresponding technical coefficients matrix and Leontief inverse, respectively, are

$$\mathbf{A}^{(a)}=\mathbf{Z}^{(a)}(\hat{\mathbf{x}}^{(a)})^{-1}=\left[\begin{array}{ccc|ccc} .126 & .13 & .035 & .01 & .011 & .002 \\ .155 & .463 & .238 & .021 & .065 & .037 \\ .043 & .086 & .13 & .003 & .009 & .008 \\ \hline .001 & .002 & .001 & .136 & .149 & .033 \\ .003 & .01 & .005 & .105 & .318 & .194 \\ .001 & .001 & .001 & .036 & .102 & .112 \end{array}\right] \text{ and}$$

$$\mathbf{L}^{(a)} = (\mathbf{I} - \mathbf{A}^{(a)})^{-1} = \left[\begin{array}{ccc|ccc} 1.207 & .315 & .135 & .031 & .062 & .032 \\ .395 & 2.055 & .580 & .093 & .252 & .149 \\ .099 & .219 & 1.213 & .018 & .046 & .03 \\ \hline .005 & .013 & .006 & 1.196 & .279 & .106 \\ .015 & .039 & .022 & .207 & 1.567 & .353 \\ .004 & .009 & .006 & .073 & .191 & 1.171 \end{array}\right].$$

To calculate the aggregation bias measured as a percent of gross outputs with a reference vector of final demands given by $\tilde{\mathbf{f}} = [100 \quad 100 \quad \cdots \quad 100]'$ for the unaggregated model, we can specify $\tilde{\mathbf{f}} = [100 \quad 100 \quad 100 \mid 100 \quad 100 \quad 100 \mid 100 \quad 100 \quad 100]'$ for the unaggregated case and we can write $\tilde{\mathbf{f}}^{(a)} = \mathbf{S}\tilde{\mathbf{f}} = [200 \quad 200 \quad 200 \mid 100 \quad 100 \quad 100]'$ for the aggregated case. We can now compute $\tilde{\mathbf{x}} = \mathbf{L}\tilde{\mathbf{f}} = [165 \quad 284 \quad 151 \mid 178 \quad 371 \quad 164 \mid 163 \quad 227 \quad 147]'$ and $\tilde{\mathbf{x}}^{(a)} = \mathbf{L}^{(a)}\tilde{\mathbf{f}}^{(a)} = [344 \quad 655 \quad 316 \mid 163 \quad 228 \quad 147]'$ from which we can compute the aggregation bias as $100 \times \dfrac{|\mathbf{S}\tilde{\mathbf{x}} - \tilde{\mathbf{x}}^{(a)}|\mathbf{i}}{\mathbf{S}\tilde{\mathbf{x}}\mathbf{i}} = 100 \times (2.115 / 1{,}850.718) = 0.114$ percent.

Computational Notes

We can retrieve the data from the textbook data appendix and presume that **Z** and **x** are defined as variables in the APL workspace. We can then compute **A** and **L** in the usual manner.

```
   L←INV A←Z AMAT x
Z
    1723.8    6311.6     405.6     187.8    1205.7      85.8      13.9      48.8       4.1
    2380.7   18457.7    2986.8     301.1    3331.1     459.6      39.1     234.1      57.4
     708.6    3882.7    1810.7      63.9     432.3     138.0       4.6      23.1       5.3
     148.7     656.1      42.2    3564.4    8828.1     805.7     103.5     177.9      15.4
     462.7    3834.0     571.4    3757.5   34931.1    5185.6     202.1    1140.1     267.9
      48.6     296.8      98.7    1098.6    6612.9    2969.1      30.8     163.1      62.2
       9.4      50.7       3.4      33.5     254.0      18.3    1581.4    3154.0     292.8
      31.7     271.6      41.3     123.4    1062.3     170.3    1224.8    6704.1    1732.8
       3.9      24.6       6.6      24.9     168.0      46.7     424.9    2145.0     999.8
x
   16651.1   49563.3   15011.4   27866.2   81252.9   23666.8   11660.8   21107.3    8910.2
A
    0.1035    0.1273    0.0270    0.0067    0.0148    0.0036    0.0012    0.0023    0.0005
    0.1430    0.3724    0.1990    0.0108    0.0410    0.0194    0.0034    0.0111    0.0064
    0.0426    0.0783    0.1206    0.0023    0.0053    0.0058    0.0004    0.0011    0.0006
    0.0089    0.0132    0.0028    0.1279    0.1087    0.0340    0.0089    0.0084    0.0017
    0.0278    0.0774    0.0381    0.1348    0.4299    0.2191    0.0173    0.0540    0.0301
    0.0029    0.0060    0.0066    0.0394    0.0814    0.1255    0.0026    0.0077    0.0070
    0.0006    0.0010    0.0002    0.0012    0.0031    0.0008    0.1356    0.1494    0.0329
    0.0019    0.0055    0.0027    0.0044    0.0131    0.0072    0.1050    0.3176    0.1945
    0.0002    0.0005    0.0004    0.0009    0.0021    0.0020    0.0364    0.1016    0.1122
L
    1.1631    0.2561    0.0965    0.0227    0.0582    0.0268    0.0064    0.0161    0.0085
    0.3008    1.7275    0.4080    0.0537    0.1596    0.0849    0.0191    0.0529    0.0314
    0.0840    0.1686    1.1794    0.0115    0.0306    0.0202    0.0035    0.0093    0.0054
    0.0325    0.0681    0.0321    1.1919    0.2504    0.1114    0.0245    0.0459    0.0232
```

```
    0.1194    0.2943    0.1588    0.3258    1.9193    0.5036    0.0742    0.2010    0.1187
    0.0193    0.0447    0.0284    0.0848    0.1920    1.1965    0.0142    0.0375    0.0252
    0.0034    0.0079    0.0039    0.0062    0.0164    0.0082    1.1958    0.2793    0.1061
    0.0098    0.0245    0.0133    0.0176    0.0478    0.0272    0.2068    1.5681    0.3532
    0.0021    0.0051    0.0030    0.0045    0.0114    0.0075    0.0730    0.1916    1.1716
```

Since we are aggregating regions rather than sectors, we can streamline creating the aggregation matrix by the following

```
    I←3 3ρ1,3ρ0
    O←3 3ρ0
    S←(I,I,O),[1]O,O,I

    S
1 0 0 1 0 0 0 0 0
0 1 0 0 1 0 0 0 0
0 0 1 0 0 1 0 0 0
0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1
```

Now we can easily compute the aggregated matrices (and vectors), `ZA`, `AA`, `LA`, and `xa` as well as the aggregate the test final demand vector `ft` (a 9-element vector with all elements equal to 100) to `fta` (and the associated vectors of total outputs `xt` and `xta`; and then compute the total aggregation bias `TAB`.

```
    ZA←S+.×Z+.×⍉S
    xa←S+.×x
    LA←INV AA←ZA AMAT xa
    ft←9ρ100
    fta←S+.×ft
    xt←L+.×ft
    xta←LA+.×fta
    xts←S+.×xt
    TAB←100×((+/|xts-xta)÷+/xts)

ZA
    5624.6   17001.5    1339.3     117.3     226.7      19.5
    6902.0   60553.9    9203.3     241.1    1374.2     325.3
    1919.8   11224.8    5016.5      35.3     186.2      67.5
      42.9     304.7      21.7    1581.4    3154.0     292.8
     155.2    1333.8     211.5    1224.8    6704.1    1732.8
      28.8     192.6      53.3     424.9    2145.0     999.8
xa
       6.0
AA
     0.126     0.130     0.035     0.010     0.011     0.002
     0.155     0.463     0.238     0.021     0.065     0.037
     0.043     0.086     0.130     0.003     0.009     0.008
     0.001     0.002     0.001     0.136     0.149     0.033
     0.003     0.010     0.005     0.105     0.318     0.194
     0.001     0.001     0.001     0.036     0.102     0.112
LA
     1.207     0.315     0.135     0.031     0.062     0.032
     0.395     2.055     0.580     0.093     0.252     0.149
```

```
     0.099      0.219      1.213      0.018      0.046      0.030
     0.005      0.013      0.006      1.196      0.279      0.106
     0.015      0.039      0.022      0.207      1.567      0.353
     0.004      0.009      0.006      0.073      0.191      1.171
ft
     100.0      100.0      100.0      100.0      100.0      100.0      100.0
100.0     100.0
xt
     165.4      283.8      151.3      178.0      371.5      164.2      162.7
226.8     147.0
fta
     200.0      200.0      200.0      100.0      100.0      100.0
xta
     344.0      655.4      315.6      163.0      227.7      147.2
xts
     343.4      655.3      315.5      162.7      226.8      147.0
TAB
0.11429872
```

Problem 4.11: Removing Imports from a 6-Sector Input-Output Table for Nepal

This exercise applies the same procedure for removing competitive imports from the interindustry transactions table utilized in Problem 4.9, but this time applied to real-world data.

Problem 4.11 Overview

Consider a six-sector input-output table for Nepal for the year 2013, defined by the matrix of interindustry transactions, **Z**, and vector of total outputs, **x**, in the following:

Interindustry Transactions	Agric.	Mining	Manuf.	Const.	Utilities	Services	Total Output
Agriculture	774	0	1,149	45	0	719	9,766
Mining	0	0	87	0	119	1	252
Manufacturing	1,037	22	2,029	166	1,654	1,743	13,015
Construction	47	5	109	47	79	376	834
Utilities	11	1	13	6	2	394	3,963
Services	780	22	781	201	377	3,443	20,446

This table includes both domestic transactions, **D**, and competitive imports, **M**, such that $\mathbf{Z} = \mathbf{D} + \mathbf{M}$. However, for the present, we presume we do not know the detailed transactions reported as **M** and, instead, know only $\mathbf{m} = \mathbf{Mi} = \begin{bmatrix} 68 & 48 & 3,227 & 65 & 1 & 457 \end{bmatrix}'$, the value of all imports of each commodity to the economy. To estimate the interindustry import transactions we assume import similarity, i.e., the imports as a fraction of interindustry activity are the same as that of the entire economy. To do this, first, we compute the vectors of intermediate outputs and total final demand, respectively, as $\mathbf{u} = \mathbf{Zi} = \begin{bmatrix} 2{,}686 & 206 & 6{,}651 & 663 & 427 & 5{,}603 \end{bmatrix}'$ and $\mathbf{f} = \mathbf{x} - \mathbf{u} = \begin{bmatrix} 7{,}080 & 45 & 6{,}363 & 171 & 3{,}536 & 14{,}843 \end{bmatrix}'$.

Knowing **m**, we can define $\mathbf{g} = \mathbf{f} + \mathbf{m} = \begin{bmatrix} 7{,}147 & 94 & 9{,}591 & 236 & 3{,}537 & 15{,}300 \end{bmatrix}'$, which is the vector of total final demands, including imports. Note that in some cases the accounting is such that **m** is recorded as a negative final demand so that $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$ reflects total domestic output, which is the convention used for the US input-output tables. Using the assumption of import similarity, we can estimate the domestic transactions matrix where competitive imports are removed from interindustry transactions by the following steps.

We can develop import similarity scaling factors for each commodity as the ratio of the value of total interindustry imports of that commodity divided by the total output (including imports), $r_i = \dfrac{m_i}{u_i + f_i}$, or $\mathbf{r} = \begin{bmatrix} 0.007 & 0.193 & 0.248 & 0.078 & 0.000 & 0.022 \end{bmatrix}'$. We can then compute the scaled quantities for imports, domestic transactions, and final demands as

$$\bar{\mathbf{M}} = \hat{\mathbf{r}}\mathbf{Z} = \begin{bmatrix} 5 & 0 & 8 & 0 & 0 & 5 \\ 0 & 0 & 17 & 0 & 23 & 0 \\ 257 & 6 & 503 & 41 & 410 & 432 \\ 4 & 0 & 8 & 4 & 6 & 29 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 17 & 0 & 17 & 4 & 8 & 77 \end{bmatrix},$$

$$\bar{\mathbf{D}} = \mathbf{Z} - \bar{\mathbf{M}} = \begin{bmatrix} 769 & 0 & 1{,}141 & 45 & 0 & 714 \\ 0 & 0 & 70 & 0 & 96 & 0 \\ 780 & 17 & 1{,}526 & 125 & 1{,}244 & 1{,}311 \\ 43 & 5 & 100 & 43 & 73 & 347 \\ 11 & 1 & 13 & 6 & 2 & 394 \\ 762 & 22 & 763 & 196 & 368 & 3{,}366 \end{bmatrix},$$

$\mathbf{h} = \hat{\mathbf{r}}\mathbf{f} = \begin{bmatrix} 49 & 9 & 1{,}578 & 13 & 1 & 332 \end{bmatrix}'$, and $\bar{\mathbf{g}} = \mathbf{g} - \mathbf{h} = \begin{bmatrix} 7{,}098 & 85 & 8{,}013 & 223 & 3{,}536 & 14{,}968 \end{bmatrix}'$.

Note that with these scaled quantities the identity $\mathbf{x} = \bar{\mathbf{D}}\mathbf{i} + \bar{\mathbf{g}}$ (analogous to $\mathbf{x} = \mathbf{Zi} + \mathbf{f}$) still holds, but this balance equation now accounts for only domestic transactions with interindustry imports reassigned to total value added. The new total value-added vector is $\bar{\mathbf{v}}' = \mathbf{x}' - \mathbf{i}'\bar{\mathbf{D}} = \begin{bmatrix} 7{,}401 & 208 & 9{,}402 & 419 & 2{,}180 & 14{,}314 \end{bmatrix}$, which inflates the original vector of total valued added, $\mathbf{v}' = \begin{bmatrix} 7{,}118 & 201 & 8{,}848 & 369 & 1{,}732 & 13{,}770 \end{bmatrix}$ by interindustry imports to each industry, i.e., $\bar{\mathbf{m}}' = \mathbf{i}'\bar{\mathbf{M}} = \begin{bmatrix} 283 & 6 & 554 & 50 & 448 & 544 \end{bmatrix}$, excluding the value of imports consumed directly in final demand.

If we subsequently learn that $\mathbf{M} = \begin{bmatrix} 18 & 0 & 30 & 1 & 0 & 18 \\ 0 & 0 & 20 & 0 & 28 & 0 \\ 406 & 16 & 793 & 124 & 672 & 1,216 \\ 5 & 0 & 12 & 5 & 10 & 33 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 19 & 2 & 54 & 10 & 22 & 350 \end{bmatrix}$, i.e., we know precisely the competitive imports associated with interindustry transactions, we can compute the domestic transactions matrix by $\mathbf{D} = \mathbf{Z} - \mathbf{M} = \begin{bmatrix} 756 & 0 & 1,118 & 44 & 0 & 701 \\ 0 & 0 & 67 & 0 & 91 & 0 \\ 631 & 7 & 1,236 & 42 & 982 & 527 \\ 42 & 5 & 97 & 42 & 69 & 344 \\ 11 & 1 & 13 & 6 & 2 & 393 \\ 760 & 20 & 727 & 190 & 355 & 3,093 \end{bmatrix}$ (rather than approximate it with import similarity scaling factors) and compute $\mathbf{m} = \mathbf{Mi} = \begin{bmatrix} 68 & 48 & 3,227 & 65 & 1 & 457 \end{bmatrix}'$ and $\mathbf{g} = \mathbf{f} - \mathbf{m} = \begin{bmatrix} 7,012 & -3 & 3,136 & 106 & 3,535 & 14,386 \end{bmatrix}'$ where $\mathbf{f} = \mathbf{x} - \mathbf{u} = \begin{bmatrix} 7,080 & 45 & 6,363 & 171 & 3,536 & 14,843 \end{bmatrix}'$, as earlier. Note that the balance equation $\mathbf{x} = \mathbf{Di} + \mathbf{g}$ holds here as well. Then the new total value-added vector, $\tilde{\mathbf{v}}' = \mathbf{x}' \text{-} \mathbf{i}'\mathbf{D} = \begin{bmatrix} 7,565 & 219 & 9,757 & 510 & 2,465 & 15,388 \end{bmatrix}$, inflates the original vector of total valued added, $\mathbf{v}' = \mathbf{x}' - \mathbf{i}'\mathbf{Z} = \begin{bmatrix} 7,118 & 201 & 8,848 & 369 & 1,732 & 13,770 \end{bmatrix}$, by the total value of all imports to each industry, $\tilde{\mathbf{m}}' = \begin{bmatrix} 447 & 18 & 909 & 141 & 733 & 1,618 \end{bmatrix}$, i.e., $\tilde{\mathbf{m}}' = \mathbf{i}'\mathbf{M} = \tilde{\mathbf{v}}' - \mathbf{v}'$.

We compute the Leontief inverse for the first case as

$$\mathbf{L}^I = (\mathbf{I} - \mathbf{A}^I)^{-1} = \begin{bmatrix} 1.1007 & 0.0145 & 0.1141 & 0.0947 & 0.0433 & 0.0577 \\ 0.0007 & 1.0006 & 0.0063 & 0.0015 & 0.0264 & 0.0012 \\ 0.1105 & 0.0909 & 1.1537 & 0.2180 & 0.3784 & 0.1063 \\ 0.0081 & 0.0234 & 0.0118 & 1.0632 & 0.0260 & 0.0234 \\ 0.0036 & 0.0047 & 0.0033 & 0.0146 & 1.0041 & 0.0238 \\ 0.1134 & 0.1175 & 0.0960 & 0.3253 & 0.1524 & 1.2193 \end{bmatrix}$$ for

$$\mathbf{A}^I = \bar{\mathbf{D}}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} 0.0787 & 0.0001 & 0.0876 & 0.0534 & 0.0000 & 0.0349 \\ 0.0000 & 0.0000 & 0.0054 & 0.0000 & 0.0243 & 0.0000 \\ 0.0798 & 0.0665 & 0.1172 & 0.1499 & 0.3139 & 0.0641 \\ 0.0044 & 0.0193 & 0.0077 & 0.0520 & 0.0184 & 0.0170 \\ 0.0011 & 0.0021 & 0.0010 & 0.0075 & 0.0005 & 0.0193 \\ 0.0780 & 0.0857 & 0.0587 & 0.2352 & 0.0930 & 0.1646 \end{bmatrix}$$ and for the second case

as $$\mathbf{L}^{II} = (\mathbf{I} - \mathbf{A}^{II}) = \begin{bmatrix} 1.0961 & 0.0084 & 0.1079 & 0.0786 & 0.0328 & 0.0498 \\ 0.0005 & 1.0003 & 0.0058 & 0.0007 & 0.0245 & 0.0008 \\ 0.0828 & 0.0346 & 1.1167 & 0.0771 & 0.2829 & 0.0452 \\ 0.0076 & 0.0228 & 0.0110 & 1.0590 & 0.0236 & 0.0221 \\ 0.0035 & 0.0042 & 0.0030 & 0.0137 & 1.0036 & 0.0233 \\ 0.1085 & 0.1051 & 0.0872 & 0.2984 & 0.1362 & 1.1943 \end{bmatrix}$$ for

$$\mathbf{A}^{II} = \mathbf{D}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} 0.0774 & 0.0000 & 0.0859 & 0.0526 & 0.0000 & 0.0343 \\ 0.0000 & 0.0000 & 0.0051 & 0.0000 & 0.0230 & 0.0000 \\ 0.0646 & 0.0259 & 0.0949 & 0.0506 & 0.2477 & 0.0258 \\ 0.0043 & 0.0195 & 0.0075 & 0.0499 & 0.0174 & 0.0168 \\ 0.0011 & 0.0020 & 0.0010 & 0.0074 & 0.0004 & 0.0192 \\ 0.0779 & 0.0810 & 0.0559 & 0.2281 & 0.0896 & 0.1513 \end{bmatrix}.$$

The mean absolute deviations between $\mathbf{L}^{I}$ and $\mathbf{L}^{II}$ and between $\mathbf{A}^{I}$ and $\mathbf{A}^{II}$ are found to be $mad(\mathbf{L}) = (1/36)\sum_{i=1}^{6}\sum_{j=1}^{6}\left|l_{ij}^{I} - l_{ij}^{II}\right| = .016$ and $mad(\mathbf{A}) = (1/36)\sum_{i=1}^{6}\sum_{j=1}^{6}\left|a_{ij}^{I} - a_{ij}^{II}\right| = .009$, respectively.

Computational Notes

First, we define **Z** and **x**. Then compute **u**, **f**, and **v** as in Problem 4.9.

```
x←9766 252 13015 834 3963 20446
Z←774 0 1149 45 0 719 0 0 87 0 119 1
Z←Z,1037 22 2029 166 1654 1743 47 5 109 47 79 376
Z←Z,11 1 13 6 2 394 780 22 781 201 377 3443
Z←6 6⍴Z
f←x-u←+/Z
v←x-+⌿Z

Z
      774          0       1149         45          0        719
        0          0         87          0        119          1
     1037         22       2029        166       1654       1743
       47          5        109         47         79        376
       11          1         13          6          2        394
      780         22        781        201        377       3443
x
     9766        252      13015        834       3963      20446
v
     7117        202       8847        369       1732      13770
u
     2687        207       6651        663        427       5604
f
     7079         45       6364        171       3536      14842
```

Then, knowing the vector of domestic imports **m**, we can compute the vector of domestic scaling factors **r**, and the associated domestic data.

```
m←68 48 3227 65 1 457
g←f+m
r←m÷(u+f)
DB←Z-(rh←DIAG r)+.×Z
MB←rh+.×Z
h←rh+.×f
gb←g-rh+.×f
vb←x-+⌿DB
mb←+⌿MB
```

```
m
68 48 3227 65 1 457
g
7147 93 9591 236 3537 15299
r
0.0069629326 0.19047619 0.24794468 0.07793765 0.00025233409 0.02235156
DB
     768.6       0.0    1141.0      44.7       0.0     714.0
       0.0       0.0      70.4       0.0      96.3       0.8
     779.9      16.5    1525.9     124.8    1243.9    1310.8
      43.3       4.6     100.5      43.3      72.8     346.7
      11.0       1.0      13.0       6.0       2.0     393.9
     762.6      21.5     763.5     196.5     368.6    3366.0
MB
       5.4       0.0       8.0       0.3       0.0       5.0
       0.0       0.0      16.6       0.0      22.7       0.2
     257.1       5.5     503.1      41.2     410.1     432.2
       3.7       0.4       8.5       3.7       6.2      29.3
       0.0       0.0       0.0       0.0       0.0       0.1
      17.4       0.5      17.5       4.5       8.4      77.0
h
      49.3       8.6    1577.9      13.3       0.9     331.7
gb
    7097.7      84.4    8013.1     222.7    3536.1   14967.3
vb
    7400.6     208.3    9400.6     418.6    2179.4   14313.7
mb
     283.6       6.3     553.6      49.6     447.4     543.7
```

Next, we presume we know the matrix of imports **M** and compute the “real” domestic transactions matrix **D**, the associated vectors of value added `vt` and of the total value of imports `mt`.

```
     M←18 0 30 1 0 18 0 0 20 0 28 0 406 16 793 124 672 1216
     M←6 6⍴M,5 0 12 5 10 33 0 0 0 0 0 0 19 2 54 10 22 350
     D←Z-M
     vt←x-+⌿D
     mt←vt-v
M
        18         0        30         1         0        18
         0         0        20         0        28         0
       406        16       793       124       672      1216
         5         0        12         5        10        33
         0         0         0         0         0         0
        19         2        54        10        22       350
D
     756.0       0.0    1119.0      44.0       0.0     701.0
       0.0       0.0      67.0       0.0      91.0       1.0
     631.0       6.0    1236.0      42.0     982.0     527.0
      42.0       5.0      97.0      42.0      69.0     343.0
      11.0       1.0      13.0       6.0       2.0     394.0
     761.0      20.0     727.0     191.0     355.0    3093.0
```

```
vt
    7565.0      220.0     9756.0      509.0     2464.0   15387.0
mt
     448.0       18.0      909.0      140.0      732.0    1617.0
```

Finally, we compute the matrices of technical coefficients using the estimated and actual matrices of transactions, DB and D, as A1 and A2, respectively, and the associated Leontief inverses, L1 and L2, as well as the MAD measures between them.

```
     L1←LINV A1←DB AMAT x
     L2←LINV A2←D AMAT x
     MAD←((+/+/|A1-A2)÷36),(+/+/|L1-L2)÷36

A1
     0.079      0.000      0.088      0.054      0.000      0.035
     0.000      0.000      0.005      0.000      0.024      0.000
     0.080      0.066      0.117      0.150      0.314      0.064
     0.004      0.018      0.008      0.052      0.018      0.017
     0.001      0.004      0.001      0.007      0.001      0.019
     0.078      0.085      0.059      0.236      0.093      0.165
L1
     1.101      0.014      0.114      0.095      0.043      0.058
     0.001      1.001      0.006      0.002      0.026      0.001
     0.111      0.090      1.154      0.218      0.378      0.106
     0.008      0.022      0.012      1.063      0.026      0.023
     0.004      0.006      0.003      0.014      1.004      0.024
     0.113      0.117      0.096      0.326      0.152      1.219
A2
     0.077      0.000      0.086      0.053      0.000      0.034
     0.000      0.000      0.005      0.000      0.023      0.000
     0.065      0.024      0.095      0.050      0.248      0.026
     0.004      0.020      0.007      0.050      0.017      0.017
     0.001      0.004      0.001      0.007      0.001      0.019
     0.078      0.079      0.056      0.229      0.090      0.151
L2
     1.096      0.008      0.108      0.079      0.033      0.050
     0.001      1.000      0.006      0.001      0.025      0.001
     0.083      0.033      1.117      0.077      0.283      0.045
     0.008      0.023      0.011      1.060      0.024      0.022
     0.003      0.006      0.003      0.014      1.004      0.023
     0.109      0.103      0.087      0.300      0.136      1.194
MAD
0.0090619248 0.016152259
```

Chapter 5, The Commodity-by-Industry Approach in Input–Output Models

Chapter 5 explores variations to the commodity-by-industry input–output framework introduced in Chapter 4, expanding the basic input–output framework to include distinguishing between commodities and industries, i.e., the supply of specific commodities in the economy and the use of those commodities by collections of businesses defined as industries. The chapter introduces the fundamental commodity-by-industry accounting relationships and how they relate to the basic input–output framework. Alternative assumptions are defined for handling the common accounting issue of secondary production, and economic interpretations of those alternative assumptions are presented. The formulations of commodity-driven and industry-driven models are also presented along with illustrations of variants on combining alternative assumptions for secondary production. Finally, the chapter illustrates the problems encountered with commodity-by-industry models, such as nonsquare commodity–industry systems, mixed technology options or the interpretation of negative elements. The exercise problems for this chapter illustrate key features of commodity-by-industry accounts and their applications.

Problem 5.1: Basic Configuration of Commodity-by-Industry Models

This problem illustrates the basic configuration a commodity by industry model using make and use matrices.

Problem 5.1 Overview

For a system of commodity-by-industry accounts, suppose we have defined three commodities and two industries.

The use matrix, **U**, and the make matrix, **V**, and are the following:

$$\mathbf{U}=\begin{bmatrix}3 & 5\\ 2 & 7\\ 2 & 3\end{bmatrix},\quad \mathbf{V}=\begin{bmatrix}15 & 5 & 10\\ 5 & 25 & 0\end{bmatrix}$$

From these matrices we can compute the vector of commodity final demands, **e**, the vector of industry value added inputs, $\mathbf{v}'$, the vector of total commodity outputs, **q**, and the vector of total industry outputs, **x**, as the following:

$$\mathbf{x}=\mathbf{Vi}=\begin{bmatrix}30\\ 30\end{bmatrix},\quad \mathbf{q}=(\mathbf{i}'\mathbf{V})'=\begin{bmatrix}20\\ 30\\ 10\end{bmatrix},\quad \mathbf{v}'=\mathbf{x}'-\mathbf{i}'\mathbf{U}=\begin{bmatrix}23 & 15\end{bmatrix},\ \text{and}\ \ \mathbf{e}=\mathbf{x}-\mathbf{Ui}=\begin{bmatrix}12\\ 21\\ 5\end{bmatrix}$$

The matrix of commodity-by-industry direct requirements is $\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .1 & .167 \\ .067 & .233 \\ .067 & .1 \end{bmatrix}$ and the matrix of commodity output proportions is $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1} = \begin{bmatrix} .75 & 0.167 & 1 \\ .25 & 0.833 & 0 \end{bmatrix}$.

Among the various configurations for total requirements matrices, as an example, if we assume a fixed commodity sales structure, the industry-by-industry total requirements matrix is found by $(\mathbf{I}-\mathbf{DB})^{-1}\mathbf{D} = \mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1} = \begin{bmatrix} 1.021 & .555 & 1.22 \\ .435 & 1.149 & 0.129 \end{bmatrix}$.

Computational Notes

First, we define **U** and **V** from which we can compute the vector of total commodity outputs **q**, the vector of total industry outputs **x**, the vector of commodity final demands **e**, and the vector of value-added inputs **v**, as the following:

```
     U←3 2ρ3 5 2 7 2 3
     V←2 3ρ15 5 10 5 25 0
     x←+/V
     q←+⌿V
     e←q-+/U
     v←x-+⌿U
U
3 5
2 7
2 3
V
15  5 10
 5 25  0
x
30 30
q
20 30 10
v
23 15
e
12 21 5
```

Finally, we can compute **B**, **D**, and the industry-by-industry, industry-based technology, industry-demand-driven, commodity-by-industry total requirements matrix as `T1` or equivalently `T2`.

```
 B←U+.×DIAG÷x
 D←V+.×DIAG÷q
 T1←D+.×INV B+.×D
 T2←(INV D+.×B)+.×D

B
     0.100     0.167
```

```
      0.067      0.233
      0.067      0.100
D
      0.750      0.167      1.000
      0.250      0.833      0.000
T1
      1.021      0.555      1.220
      0.435      1.149      0.129
T2
      1.021      0.555      1.220
      0.435      1.149      0.129
```

Problem 5.2: Commodity-by-Industry Total Requirements Matrices

This problem illustrates commodity-by-industry total requirements matrices under alternative assumptions of industry-based and commodity-based technology.

Problem 5.2 Overview

Consider the following system of commodity and industry accounts for a region:

		Commodities		Industries		Final Demand	Total Output
		1	2	1	2		
Commodities	1			1	2	7	10
	2			3	4	3	10
Industries	1	10	2				12
	2	0	8				8
Value Added				8	2	10	
Total Inputs		10	10	12	8		

From this table, the use matrix is $\mathbf{U} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$; the make matrix is $\mathbf{V} = \begin{bmatrix} 10 & 2 \\ 0 & 8 \end{bmatrix}$; the vector of commodity final demands is $\mathbf{e} = \begin{bmatrix} 7 \\ 3 \end{bmatrix}$; the vector of total commodity output is $\mathbf{q} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$; the vector of total industry output is $\mathbf{x} = \begin{bmatrix} 12 \\ 8 \end{bmatrix}$; and the vector of industry value added is $\mathbf{v}' = \begin{bmatrix} 8 & 2 \end{bmatrix}$. We these definitions we can compute the commodity-by-industry matrix of direct requirements as $\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .083 & .25 \\ .25 & .5 \end{bmatrix}$; the commodity output proportions matrix as $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1} = \begin{bmatrix} 1.0 & .2 \\ 0 & .8 \end{bmatrix}$; and the industry output proportions matrix as $\mathbf{C} = \mathbf{V}'\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .833 & 0 \\ .167 & 1 \end{bmatrix}$.

As one common variant, the industry-based technology, industry-demand-driven, commodity-by-industry total requirements matrix is $(\mathbf{I} - \mathbf{DB})^{-1}\mathbf{D} = \begin{bmatrix} 1.333 & 0.889 \\ .444 & 1.63 \end{bmatrix}$. As another variant, the commodity-based technology, industry-demand drive, commodity-by-industry total requirements matrix is $(\mathbf{I}-\mathbf{C}^{-1}\mathbf{B})^{-1}\mathbf{C}^{-1}=\begin{bmatrix}1.412 & .706\\ .235 & 2.118\end{bmatrix}$ under the fixed industry sale structure assumption, $\Delta\mathbf{x}=\begin{bmatrix}14.11\\ 10.95\end{bmatrix}$, and under a commodity-based technology assumption, $\Delta\mathbf{x}=\begin{bmatrix}12\\ 12\end{bmatrix}$. These are different because the accounting of secondary production is different in the two assumptions.

Computational Notes

First, we define **U** and **V** from which we can compute the vector of total commodity outputs **q**, the vector of total industry outputs **x**, and the vector of commodity final demands **e**, as the following:

```
      U←2 2ρ1 2 3 4
      V←2 2ρ10 2 0 8
      x←+/V
      q←+⌿V
      e←x-+/U

U
1 2
3 4
V
10 2
 0 8
x
12 8
q
10 10
e
9 1
```

Next, we can compute **B**, **D**, **C**, `CI` (the inverse of **C**), and the total requirements variants, industry-based technology, industry-demand-driven, commodity-by-industry total requirements matrix as `T1`, and the commodity-based technology, industry-demand drive, commodity-by-industry total requirements matrix as `T2`.

```
      B←U+.×DIAG ÷x
      D←V+.×DIAG ÷q
      C←(VT←⍉V)+.×DIAG ÷x
      CI←⌹C
      T1←D+.×(IBI←INV B)+.×D
      T2←(INV CI+.×B)+.×CI

B
      0.083     0.250
      0.250     0.500
D
      1.000     0.200
      0.000     0.800
```

```
C
      0.833     0.000
      0.167     1.000
T1
      1.389     1.154
      0.505     1.583
CI
      1.200     0.000
     ¯0.200     1.000
T2
      1.412     0.706
      0.235     2.118
```

Problem 5.3: Mixed Technology Assumption in Commodity-by-Industry Models

This problem illustrates the adoption of mixed technology assumptions in construction commodity-by-industry models.

Problem 5.3 Overview

Consider again the system of accounts given in Problem 5.1: $\mathbf{U} = \begin{bmatrix} 3 & 5 \\ 2 & 7 \\ 2 & 3 \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} 15 & 5 & 10 \\ 5 & 25 & 0 \end{bmatrix}$ so that $\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 30 \\ 30 \end{bmatrix}$ v and we can compute the industry input requirements matrix as

$$\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} 3 & 5 \\ 2 & 7 \\ 2 & 3 \end{bmatrix} \begin{bmatrix} 1/30 & 0 \\ 0 & 1/30 \end{bmatrix} = \begin{bmatrix} .1 & .167 \\ .067 & .233 \\ .067 & .1 \end{bmatrix}.$$

Suppose we can split the make matrix, $\mathbf{V} = \begin{bmatrix} 15 & 5 & 10 \\ 5 & 25 & 0 \end{bmatrix}$ into two components, $\mathbf{V}_1 = \begin{bmatrix} 5 & 5 & 5 \\ 5 & 5 & 0 \end{bmatrix}$ and $\mathbf{V}_2 = \begin{bmatrix} 10 & 0 & 5 \\ 0 & 20 & 0 \end{bmatrix}$ such that $\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2$. This means that $\mathbf{x}_1 = \mathbf{V}_1\mathbf{i} = \begin{bmatrix} 15 \\ 10 \end{bmatrix}$ and $\mathbf{q}_1 = \mathbf{i}\mathbf{V} = \begin{bmatrix} 10 \\ 10 \\ 5 \end{bmatrix}$.

We might be interested in comparing the two "mixed technology" assumptions that were covered in Sections 5.7.1 and 5.7.2 in computing the industry-by-commodity total requirements matrix for this system of accounts. However, since $\mathbf{V}$ is nonsquare, the matrix of industry output proportions, $\mathbf{C}$, will be nonsquare and hence no unique $\mathbf{C}^{-1}$ exists.

Since no unique $\mathbf{C}^{-1}$ exists we cannot use the mixed-technology assumption requiring computation of $\mathbf{C}^{-1}$; that is, we cannot determine either $(\mathbf{I}-\mathbf{C}^{-1}\mathbf{B})^{-1}\mathbf{C}^{-1}$ or $\mathbf{R}(\mathbf{I}-\mathbf{BR})^{-1}$ where $\mathbf{R}=[\mathbf{C}_1^{-1}(\mathbf{I}-\langle\mathbf{D}_2'\rangle)+\mathbf{D}_2]$. Nonetheless, we can use the industry-based technology assumption with $\mathbf{T}=[(\mathbf{I}+\mathbf{D}_1\mathbf{C}_2-\langle\mathbf{i}'\mathbf{C}_2\rangle)^{-1}\mathbf{D}_1]$ where $\mathbf{D}_1=\mathbf{V}_1\hat{\mathbf{q}}_1^{-1}=\begin{bmatrix} .5 & .5 & 1 \\ .5 & .5 & 0 \end{bmatrix}$ and $\mathbf{C}_2=\mathbf{V}_2'\hat{\mathbf{x}}^{-1}=\begin{bmatrix} .333 & 0 \\ 0 & .667 \\ .137 & 0 \end{bmatrix}$, which in this case is $\mathbf{T}=\begin{bmatrix} .333 & .333 & 1.333 \\ .667 & .667 & -.333 \end{bmatrix}$ and then we compute the matrix of total requirements as $\mathbf{T}(\mathbf{I}-\mathbf{BT})^{-1}=\begin{bmatrix} .685 & .685 & 1.476 \\ .949 & .949 & -.264 \end{bmatrix}$. Note the negative element in this matrix of total requirements, the implications of which are discussed in Section 5.5 of the text.

Computational Notes.

First, as in Problem 5.1, we define **U** and **V** and compute **x**, **q**, **B**, and **D** but in addition we decompose **V** in `V1` and `V2`.

```
      U←3 2ρ3 5 2 7 2 3
      V←2 3ρ15 5 10 5 25 ⋄ x←+/V ⋄ q←+⌿V
      B←U+.×DIAG ÷x
      D←V+.×DIAG ÷q
      V1←2 3ρ5 5 5 5 5 0
      V2←2 3ρ10 0 5 0 20 0
U
         3         5
         2         7
         2         3
V
        15         5        10
         5        25         0
x
        30        30
q
        20        30        10
B
    0.1000    0.1667
    0.0667    0.2333
    0.0667    0.1000
D
     0.750     0.167     1.000
     0.250     0.833     0.000

V1
         5         5         5
         5         5         0
```

```
V2
        10         0          5
         0        20          0
```

We compute the two total requirements matrices using industry-based technology assumptions as `T` and `T4`, which requires the additional computation of `x1`, `q1`, `D1`, and `C2`.

```
x1←+/V1
D1←V1+.×DIAG ÷q1←+⌿V1
C2←(V2T←⍉V2)+.×DIAG ÷x
T←(⌹I2+(D1+.×C2)-(DIAG+⌿C2))+.×D1
T4←T+.×LINV B+.×T
```

```
x1
        15        10
q1
        10        10          5
D1
     0.500     0.500      1.000
     0.500     0.500      0.000
C2
     0.333     0.000
     0.000     0.667
     0.167     0.000
T
     0.333     0.333      1.333
     0.667     0.667     ¯0.333
T4
     0.685     0.685      1.476
     0.949     0.949     ¯0.264
```

Problem 5.4: Additional Considerations in Mixed Technology Assumptions

This problem explores further the use mixed technology assumptions in deriving industry-by-commodity total requirements matrices.

Problem 5.3 Overview

Recall the system of accounts given in Problem 5.2. In this case the make matrix, **V**, is a square matrix so it is possible to compute the inverse of matrix of industry output proportions, **C**. First, we split $\mathbf{V}=\begin{bmatrix}10 & 2\\ 0 & 8\end{bmatrix}$ into two components, $\mathbf{V}_1=\begin{bmatrix}10 & 0\\ 0 & 2\end{bmatrix}$ and $\mathbf{V}_2=\begin{bmatrix}0 & 2\\ 0 & 6\end{bmatrix}$ such that $\mathbf{V}=\mathbf{V}_1+\mathbf{V}_2$. The vectors of industry and commodity outputs are then found by $\mathbf{x}_1=\mathbf{V}_1\mathbf{i}=\begin{bmatrix}10\\ 2\end{bmatrix}$, $\mathbf{q}_1=\mathbf{i}'\mathbf{V}_1=\begin{bmatrix}10\\ 2\end{bmatrix}$, respectively.

With this configuration, we can compute $\mathbf{C}_1 = \mathbf{V}_1'(\hat{\mathbf{x}}_1)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\mathbf{D}_2 = \mathbf{V}_2(\hat{\mathbf{q}})^{-1} = \begin{bmatrix} 0 & .2 \\ 0 & .6 \end{bmatrix}$, and, subsequently, $\mathbf{C}_1^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. From **V**, we can compute $\mathbf{q} = (\mathbf{i}'\mathbf{V})' = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$ and $\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 12 \\ 8 \end{bmatrix}$, and, subsequently, one variant of direct requirements using mixed technology assumptions as $\mathbf{R} = \mathbf{C}_1^{-1}(\mathbf{I} - \langle \mathbf{D}_2'\mathbf{i} \rangle) + \mathbf{D}_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 0.8 \end{bmatrix}$ and $\mathbf{R}(\mathbf{I} - \mathbf{B}\mathbf{R})^{-1} = \begin{bmatrix} 1.333 & .889 \\ .444 & 1.630 \end{bmatrix}$. As another variant of direct requirements using mixed technology assumptions, **T**, we first compute $\mathbf{D}_1 = \mathbf{V}_1(\hat{\mathbf{q}}_1)^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{C}_2 = \mathbf{V}_2'(\hat{\mathbf{x}})^{-1} = \begin{bmatrix} 0 & 0 \\ 0.167 & 0.750 \end{bmatrix}$, so we can compute $\mathbf{T} = (\mathbf{I} + \mathbf{D}_1\mathbf{C}_2 - \langle \mathbf{C}_2'\mathbf{i} \rangle)^{-1}\mathbf{D}_1 = \begin{bmatrix} 1.2 & 0 \\ -0.2 & 1 \end{bmatrix}$ and $\mathbf{T}(\mathbf{I} - \mathbf{B}\mathbf{T})^{-1} = \begin{bmatrix} 1.412 & .706 \\ .235 & 2.118 \end{bmatrix}$.

Of course, there are alternative partitions of the **V** matrix into its $\mathbf{V}_1$ and $\mathbf{V}_2$ components with the requirement that $\mathbf{V} = \mathbf{V}_1 + \mathbf{V}_2$, depending upon the suitable assumptions for industries and commodities in the economy.

Computational Notes

We begin with recalling the data for Problem 5.2, but in addition decomposing **V** into `V1` and `V2` so we define the associated total industry and commodity outputs for `V1` as `x1` and `q1`, respectively, and for convenience we define a two-element identity vector as `i` and matrix as `I`.

```
U←2 2⍴1 2 3 4
V←2 2⍴10 2 0 8 ⋄ x←+/V ⋄ q←+⌿V
B←U+.×DIAG ÷x
D←V+.×DIAG ÷q
C←(⍉V)+.×DIAG ÷x
V1←2 2⍴10 0 0 2
V2←2 2⍴0 2 0 6
q1←+⌿V1
x1←+/V1
I←2 2⍴1,2⍴0
i←1 1
```

```
U
        1         2
        3         4
V
       10         2
        0         8
x
       12         8
q
       10        10
```

```
B
      0.083     0.250
      0.250     0.500
D
      1.000     0.200
      0.000     0.800
C
      0.833     0.000
      0.167     1.000
V1
         10         0
          0         2
V2
          0         2
          0         6
x1
         10         2
q1
         10         2
```

Now invoking the alternative **R** and **T** mixed technology assumptions, we need to compute `C1`, `D2`, `D1`, and `C1` to compute the total requirements matrices for **R** and **T** which we define as `T3` and `T4`, respectively.

```
      C1←(⍉V1)+.×DIAG ÷x1
      D2←V2+.×DIAG ÷q
      R←D2+(⌹C1)+.×I-DIAG (⍉D2)+.×i

      T3←R+.×LINV B+.×R

      D1←V1+.×DIAG ÷q1
      C2←(⍉V2)+.×DIAG ÷x
      T←(⌹I+(D1+.×C2)-DIAG (⍉C2)+.×i)+.×D1

      T4←T+.×LINV B+.×T
```

```
C1
      1.000     0.000
      0.000     1.000
D2
      0.000     0.200
      0.000     0.600
C2
      0.000     0.000
      0.167     0.750
D1
      1.000     0.000
      0.000     1.000


R
      1.000     0.200
      0.000     0.800
```

```
T
     1.200      0.000
    ¯0.200      1.000
T3
     1.333      0.889
     0.444      1.630
T4
     1.412      0.706
     0.235      2.118
```

Problem 5.5: Mixed Technology and Nonsquare Commodity-by-Industry Models

In this problem we explore further the characteristics technology assumptions of commodity-by-industry models.

Problem 5.5 Overview

In a system of commodity-by-industry accounts, suppose we have defined four commodities and three industries. The make matrix, **V**, and the use matrix, **U**, are given as $\mathbf{U} = \begin{bmatrix} 20 & 12 & 18 \\ 5 & 30 & 12 \\ 10 & 13 & 11 \\ 12 & 17 & 40 \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} 99 & 0 & 0 & 10 \\ 8 & 143 & 137 & 10 \\ 0 & 6 & 12 & 150 \end{bmatrix}$. We can compute vectors of total commodity outputs and total industry outputs, respectively, as $\mathbf{q} = \mathbf{V}'\mathbf{i} = \begin{bmatrix} 107 \\ 149 \\ 149 \\ 170 \end{bmatrix}$ and $\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 109 \\ 298 \\ 168 \end{bmatrix}$.

Recall that the commodity-by-industry total requirements matrix with the assumption of industry-based technology is $\mathbf{D}^{-1}(\mathbf{I} - \mathbf{BD})^{-1}$. In this case since there are more commodities than industries, the matrix **D** is non-square, hence, $\mathbf{D}^{-1}$ does not exist so it is impossible to compute $\mathbf{D}^{-1}(\mathbf{I} - \mathbf{BD})^{-1}$.

For industry-by-commodity total requirements using the assumption of industry-based technology for commodity-driven final demand, we can compute:

$$\mathbf{D}(\mathbf{I} - \mathbf{BD})^{-1} = \begin{bmatrix} 1.164 & .077 & .082 & .25 \\ .321 & 1.159 & 1.122 & .321 \\ .182 & .148 & .197 & .1.187 \end{bmatrix}.$$

To illustrate mixed technology assumptions, we aggregate the first two commodities to one in both the make and use matrices. Hence, we have $\mathbf{U} = \begin{bmatrix} 25 & 42 & 30 \\ 10 & 13 & 11 \\ 12 & 17 & 40 \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} 99 & 0 & 10 \\ 151 & 137 & 10 \\ 6 & 12 & 150 \end{bmatrix}$. We can compute vectors of total commodity outputs and total industry outputs, respectively, as $\mathbf{q} = \mathbf{V}'\mathbf{i} = \begin{bmatrix} 256 \\ 149 \\ 170 \end{bmatrix}$ and $\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 109 \\ 298 \\ 168 \end{bmatrix}$. We assume that $\mathbf{V}$ can be decomposed into $\mathbf{V}_1$ and $\mathbf{V}_2$ where $\mathbf{V}_1 = \begin{bmatrix} 99 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 30 \end{bmatrix}$, so we can compute $\mathbf{V}_2 = \mathbf{V} - \mathbf{V}_1 = \begin{bmatrix} 0 & 0 & 10 \\ 151 & 127 & 10 \\ 6 & 12 & 120 \end{bmatrix}$. With these definitions we can compute the commodity-by-industry matrix of direct requirements as $\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .229 & .141 & .179 \\ .092 & .044 & .066 \\ .110 & .057 & .238 \end{bmatrix}$ and the commodity output proportions matrix as $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1} = \begin{bmatrix} .387 & 0 & .059 \\ .590 & .919 & .059 \\ .023 & .081 & .882 \end{bmatrix}$. If we assume a commodity-based technology for $\mathbf{V}_1$ and an industry-based technology for $\mathbf{V}_2$, the four total requirements matrices (i.e., commodity-by-commodity, industry-by-commodity, commodity-by-industry and industry-by-industry) to be used with commodity-driven demand calculations are found by first computing $\mathbf{x}_1 = \mathbf{V}_1\mathbf{i} = \begin{bmatrix} 99 \\ 10 \\ 30 \end{bmatrix}$. Also, in this case, since $\mathbf{V}_1$ is diagonal, we can easily find

$$\mathbf{x}_1 = \mathbf{V}_1\mathbf{i} = \mathbf{q}_1 = \mathbf{V}_1'\mathbf{i} = \begin{bmatrix} 99 \\ 10 \\ 30 \end{bmatrix}.$$

From these quantities we can compute $\mathbf{C}_1 = \mathbf{V}_1'\hat{\mathbf{x}}_1^{-1} = \mathbf{C}_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ and, subsequently,

$\mathbf{D}_2 = \mathbf{V}_2(\hat{\mathbf{q}})^{-1} = \begin{bmatrix} 0 & 0 & .059 \\ .59 & .852 & .059 \\ .023 & .081 & .706 \end{bmatrix}$ so that $\mathbf{R} = \mathbf{C}_1^{-1}\left[\mathbf{I} - \langle \mathbf{D}_2'\mathbf{i} \rangle\right] + \mathbf{D}_2 = \begin{bmatrix} .387 & 0 & .059 \\ .59 & .925 & .059 \\ .023 & .081 & .882 \end{bmatrix}$ and

$\mathbf{R}^{-1} = \begin{bmatrix} 2.573 & .015 & -.173 \\ -1.656 & 1.084 & .038 \\ .083 & -.099 & 1.134 \end{bmatrix}$. (Note the negative elements in $\mathbf{R}^{-1}$). We can now compute the family of total requirements matrices as

$$(\mathbf{I} - \mathbf{BR})^{-1} = \begin{bmatrix} 1.260 & .213 & .308 \\ .093 & 1.070 & .111 \\ .141 & .121 & 1.324 \end{bmatrix}, \ (\mathbf{I} - \mathbf{BR})^{-1}\mathbf{R}^{-1} = \begin{bmatrix} 2.916 & .22 & .14 \\ -1.524 & 1.15 & .15 \\ .272 & .001 & 1.483 \end{bmatrix},$$

$$\mathbf{R}(\mathbf{I} - \mathbf{BR})^{-1} = \begin{bmatrix} .496 & .09 & .197 \\ .837 & 1.117 & .362 \\ .161 & .198 & 1.185 \end{bmatrix} \text{ and } (\mathbf{I} - \mathbf{RB})^{-1} = \begin{bmatrix} 1.144 & .085 & .141 \\ .334 & 1.187 & .309 \\ .186 & .099 & 1.324 \end{bmatrix}.$$

Computational Notes

We first define the basic Use and Make matrices as **U** and **V**, respectively, and compute the associated vectors of total industry and commodity outputs as **x** and **q**, respectively, and the matrix of commodity input proportions **B**.

```
U←3 3⍴25 42 30 10 13 11 12 17 40
V←3 3⍴99 0 10 151 137 10 6 12 150
x←+/V
q←+⌿V
B←U+.×DIAG ÷x
```

```
U
        25         42         30
        10         13         11
        12         17         40
V
        99          0         10
       151        137         10
         6         12        150
x
       109        298        168
q
       256        149        170
```

```
B
    0.2294    0.1409    0.1786
    0.0917    0.0436    0.0655
    0.1101    0.0570    0.2381
```

Now we decompose **V** into `V1` (commodity-based technology) and `V2` (industry-based technology) and compute the associated `x1`, `q2`, `C1`, and `D2` in order to ultimately compute **R** and the family of total requirements matrices for commodity-by-commodity (`TCCIC`), commodity-by-industry (`TCCCC`), industry-by-commodity (`TICIC`), and industry-by-industry (`TICCC`) configurations under mixed technology assumptions.

```
      V2←V-V1←3 3ρ99 0 0 0 10 0 0 0 30
      x1←+/V1
      q2←+⌿V2
      I←3 3ρ1,3ρ0
      iii←3ρ1
      D2←V2+.×DIAG ÷q
      C1←(⍉V1)+.×DIAG ÷x1
      R←((CI←⌹C1)+.×((I-DIAG (⍉D2)+.×iii)))+D2
      TCCIC←INV BR←B+.×R
      TCCCC←TCCIC+.×RI←⌹R
      TICIC←R+.×TCCIC
      TICCC←INV R+.×B
```

```
V1
        99         0         0
         0        10         0
         0         0        30
V2
         0         0        10
       151       127        10
         6        12       120
D2
    0.0000    0.0000    0.0588
    0.5898    0.8523    0.0588
    0.0234    0.0805    0.7059
C1
    1.0000    0.0000    0.0000
    0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000
R
    0.3867    0.0000    0.0588
    0.5898    0.9195    0.0588
    0.0234    0.0805    0.8824


TCCIC
    1.2604    0.2132    0.3077
    0.0925    1.0699    0.1114
    0.1407    0.1206    1.3244
TCCCC
    2.9158    0.2196    0.1397
   ¯1.5244    1.1503    0.1512
```

```
    0.2719    0.0013    1.4828
TICIC
    0.4957    0.0895    0.1969
    0.8368    1.1165    0.3619
    0.1611    0.1976    1.1848
TICCC
    1.1436    0.0850    0.1413
    0.3342    1.1873    0.3087
    0.1855    0.0989    1.3238
```

Problem 5.6: Additional Properties of Mixed Technology Assumptions

This problem explores further the properties of mixed technology assumptions for commodity-by-industry total requirements matrices. Recall first from the numerical results in Section 5.7.3 for the examples provided that the column sums of both the mixed technology direct requirements matrices, **R** and **T**, are one, i.e., $\mathbf{i'R} = \mathbf{i'T}$. We can show that such is generally the case for **C**, **D**, **R**, and **T**.

We start with the matrix of industry output proportions, $\mathbf{C} = \mathbf{V}'\hat{\mathbf{x}}^{-1}$. The column sums of **C** are found by premultiplying **C** by $\mathbf{i}'$, so $\mathbf{i}'\mathbf{C} = \mathbf{i}'\mathbf{V}'\hat{\mathbf{x}}^{-1}$. Since for any pair of matrices, **A** and **B**, $(\mathbf{AB})' = \mathbf{B}'\mathbf{A}'$, we rewrite this as $\mathbf{i}'\mathbf{C} = \mathbf{i}'\mathbf{V}'\hat{\mathbf{x}}^{-1} = (\mathbf{Vi})'\hat{\mathbf{x}}^{-1}$, and substituting $\mathbf{x} = \mathbf{Vi}$ yields $\mathbf{i}'\mathbf{C} = \mathbf{x}'\hat{\mathbf{x}}^{-1} = \mathbf{i}'$ which proves the case generally for **C**.

Similarly, if we start with the matrix of commodity output proportions, $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1}$ the column sums of **D** are found with $\mathbf{i}'\mathbf{D} = \mathbf{i}'\mathbf{V}'\hat{\mathbf{q}}^{-1}$, using the same property of the transpose of a product of matrices as in the previous case, yields $\mathbf{i}'\mathbf{D} = (\mathbf{Vi})'\hat{\mathbf{q}}^{-1}$ and substituting $\mathbf{q} = \mathbf{Vi}$ yields $\mathbf{i}'\mathbf{D} = \mathbf{q}'\hat{\mathbf{q}}^{-1} = \mathbf{i}'$ which proves the case generally for **D**.

One variant of mixed technology began with the identity, $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 = \mathbf{C}_1^{-1}\mathbf{q}_1 + \mathbf{D}_2\mathbf{q}$ developed in Section 5.7.1 but expressing $\mathbf{q}_1$ as a function of **q** to yield $\mathbf{x} = \mathbf{Rq} = [\mathbf{C}_1^{-1}(\mathbf{I} - \langle \mathbf{D}_2'\mathbf{i} \rangle) + \mathbf{D}_2]\mathbf{q}$. Another variant, derived in Section 5.7.2, again expressing **x** as a function of **q** was $\mathbf{x} = \mathbf{Tq} = [(\mathbf{I} + \mathbf{D}_1\mathbf{C}_2 - \langle \mathbf{i'C}_2 \rangle)^{-1}\mathbf{D}_1]\mathbf{q}$. Applying property of the transpose of a product of matrices once again, this time on the term, $\langle \mathbf{D}_2'\mathbf{i} \rangle$, yields $\mathbf{i}'\mathbf{R} = \mathbf{i}'\mathbf{C}_1^{-1} - \mathbf{i}'\mathbf{C}_1^{-1}\mathbf{i}'\mathbf{D}_2 + \mathbf{i}'\mathbf{D}_2$. Finally, substituting $\mathbf{C}_1 = (\mathbf{V}_1')(\hat{\mathbf{x}}_1)^{-1}$ or $\mathbf{C}_1^{-1} = \hat{\mathbf{x}}_1(\mathbf{V}_1')^{-1}$ and $\mathbf{D}_2 = \mathbf{V}_2\hat{\mathbf{q}}^{-1}$ yields $\mathbf{i}'\mathbf{R} = \mathbf{i}'\hat{\mathbf{x}}_1(\mathbf{V}_1')^{-1} - \mathbf{i}'\hat{\mathbf{x}}_1(\mathbf{V}_1')^{-1}\mathbf{i}'\mathbf{V}_2\hat{\mathbf{q}}_2^{-1} + \mathbf{i}'\mathbf{V}_2\hat{\mathbf{q}}_2^{-1}$ and it follows directly that $\mathbf{i}'\mathbf{C}_1 = \mathbf{i}'$, $\mathbf{i}'\mathbf{R} = \mathbf{i}'$, $\mathbf{i}'\mathbf{D}_1 = \mathbf{i}'$, and $\mathbf{i}'\mathbf{T} = \mathbf{i}'$.

Problem 5.7: Industry-by-Commodity Model Impact Analysis

This problem explores the use of total industry-by-commodity requirements under an assumption of industry-based technology for impact analysis.

Problem 5.7 Overview

Consider the following make and use matrices.

$$\mathbf{U} = \begin{bmatrix} 20 & 12 & 18 \\ 5 & 30 & 12 \\ 10 & 13 & 11 \\ 12 & 17 & 40 \end{bmatrix} \text{ and } \mathbf{V} = \begin{bmatrix} 99 & 0 & 0 & 10 \\ 8 & 143 & 137 & 10 \\ 0 & 6 & 12 & 150 \end{bmatrix}. \text{ We compute } \mathbf{q} = \mathbf{V}'\mathbf{i} = \begin{bmatrix} 107 \\ 149 \\ 149 \\ 170 \end{bmatrix} \text{ and}$$

$$\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 109 \\ 298 \\ 168 \end{bmatrix} \text{ so } \mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .183 & .040 & .107 \\ .046 & .101 & .071 \\ .092 & .044 & .065 \\ .110 & .057 & .238 \end{bmatrix} \text{ and } \mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1} = \begin{bmatrix} .925 & 0 & 0 & .059 \\ .075 & .960 & .919 & .059 \\ 0 & .040 & .081 & .882 \end{bmatrix}.$$

We assume that the three industries are: Agriculture, Oil Production, and Manufacturing and the four commodities are Agricultural Products, Crude Oil, Natural Gas, and Manufactured Products. We can interpret this as meaning in this case that natural gas is considered a secondary product of the oil industry.

To compute the levels of oil and natural gas industry production necessary to support a final demand of 100 manufactured products, first generate the total industry-by-commodity requirements using an industry-based technology: $\mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1} = \begin{bmatrix} 1.164 & .078 & .082 & .25 \\ .321 & 1.160 & 1.122 & .321 \\ .182 & .148 & .197 & 1.187 \end{bmatrix}$.

For the final demand of 100 for manufactured products, $\Delta\mathbf{f} = \begin{bmatrix} 0 & 0 & 0 & 100 \end{bmatrix}'$, we have

$$\Delta\mathbf{x} = \mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1}\Delta\mathbf{f} = \begin{bmatrix} 25.02 \\ 32.08 \\ 118.65 \end{bmatrix}.$$

Computational Notes

We begin by defining **U** and **V** and computing the familiar commodity-by-industry quantities **x**, **q**, **B**, **D**, and **C**.

```
U←4 3⍴20 12 18 5 30 12 10 13 11 12 17 40
V←3 4⍴99 0 0 10 8 143 137 10 0 6 12 150
x←+/V
q←+⌿V
B←U+.×DIAG ÷x
D←V+.×DIAG ÷q
C←(VT←⍉V)+.×DIAG ÷x
```

```
U
      20          12          18
       5          30          12
```

```
         10          13          11
         12          17          40
V
         99           0           0          10
          8         143         137          10
          0           6          12         150
x
        109         298         168
q
        107         149         149         170
B
      0.183       0.040       0.107
      0.046       0.101       0.071
      0.092       0.044       0.065
      0.110       0.057       0.238
D
      0.925       0.000       0.000       0.059
      0.075       0.960       0.919       0.059
      0.000       0.040       0.081       0.882
C
      0.908       0.027       0.000
      0.000       0.480       0.036
      0.000       0.460       0.071
      0.092       0.034       0.893
```

In this problem we compute the total industry-by-commodity requirements using an industry-based technology, assigned to the matrix `TICIC`, and compute the vector of total outputs `Δx` needed to supply final demand `Δy`.

```
      TICIC←D+.×LINV B+.×D
      Δx←TICIC+.×Δy←0 0 0 100
TICIC
1.164042   0.077481726 0.082421628 0.25017804
0.32113128 1.1594591   1.1220161   0.32078062
0.18217735 0.1479298   0.19733828  1.1864668

Δy
       0.00        0.00        0.00      100.00
Δx
      25.02       32.08      118.65
```

Problem 5.8: "Purifying" Commodity-by-Commodity Models

This problem explores the issues of using commodity-by-commodity models with commodity-based technology. Consider the following make and use matrices: $\mathbf{U} = \begin{bmatrix} 20 & 15 & 18 \\ 5 & 30 & 12 \\ 10 & 16 & 11 \end{bmatrix}$ and

$\mathbf{V} = \begin{bmatrix} 30 & 0 & 0 \\ 10 & 50 & 35 \\ 0 & 25 & 150 \end{bmatrix}$. First, we compute $\mathbf{q} = \mathbf{i}'\mathbf{V} = \begin{bmatrix} 40 \\ 75 \\ 185 \end{bmatrix}$ and then

$$\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1} = \begin{bmatrix} .75 & 0 & 0 \\ .25 & .667 & .189 \\ 0 & .333 & .811 \end{bmatrix}.$$

A standard calculation for producing the commodity-by-commodity transactions matrix with commodity-base technology begins with the matrix of technical requirements, $\underset{(c\times c)}{\mathbf{A}_C} = \mathbf{BC}^{-1} = [\mathbf{U}\hat{\mathbf{x}}^{-1}][\mathbf{V}'\hat{\mathbf{x}}^{-1}]^{-1} = [\mathbf{U}\hat{\mathbf{x}}^{-1}][\hat{\mathbf{x}}(\mathbf{V}')^{-1}] = \mathbf{U}(\mathbf{V}')^{-1}$. To express in terms of inter-commodity transactions, we postmultiply through by $\hat{\mathbf{q}}$ to obtain $\mathbf{Z}_C = \mathbf{A}_C\hat{\mathbf{q}} = \mathbf{U}[(\mathbf{V}')^{-1}\hat{\mathbf{q}}]$.

Recall that the definition of **D** is $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1}$. Since the transpose of a product of matrices is the reverse product of the transposes of each matrix and that the transpose of a diagonal matrix is the original matrix itself, we can rewrite this definition as $\hat{\mathbf{q}}^{-1}\mathbf{V}' = \mathbf{D}'$. Further, since the inverse of a product of matrices is the reverse product of the inverses of each matrix, this equation becomes $(\mathbf{V}')^{-1}\hat{\mathbf{q}} = (\mathbf{D}')^{-1}$ which we can substitute in the equation defining $\mathbf{Z}_C$ above so that $\mathbf{Z}_C = \mathbf{A}_C\hat{\mathbf{q}} = \mathbf{U}[(\mathbf{V}')^{-1}\hat{\mathbf{q}}] = \mathbf{U}(\mathbf{D}')^{-1}$.

For this case, the commodity-by-commodity matrix of interindustry transactions is $\mathbf{Z}_C = \mathbf{U}(\mathbf{D}')^{-1} = \begin{bmatrix} 26.667 & 7.019 & 19.315 \\ 6.667 & 43.359 & -3.025 \\ 13.333 & 17.151 & 6.516 \end{bmatrix}$. Note that there is a negative element. So, we can apply the Almon purifying algorithm (Appendix 5.2 in the text) which iteratively distributes negative elements across positive elements to remove them while preserving the essential accounting identities. The result is a "purified" non-negative transactions matrix:

$$\tilde{\mathbf{Z}}_C = \mathbf{U}(\mathbf{D}')^{-1} = \begin{bmatrix} 26.667 & 7.019 & 19.315 \\ 6.667 & 40.334 & 0 \\ 13.333 & 17.151 & 6.516 \end{bmatrix}.$$

Problem 5.9: Commodity-by-Industry US Model

This problem explores application of an industry-based technology commodity-by-industry model using the use and make matrices for highly aggregated U.S. input-output tables for 2003.

Problem 5.9 Overview

The following are the use and make tables:

US Use Table for 2003	1	2	3	4	5	6	7
1. Agriculture	61,946	1	1,270	147,559	231	18,453	2,093
2. Mining	441	33,299	6,927	174,235	89,246	1,058	11,507
3. Construction	942	47	1,278	8,128	10,047	65,053	48,460
4. Manufacturing	47,511	22,931	265,115	1,249,629	132,673	516,730	226,689
5. Trade, Transport & Utils	24,325	13,211	100,510	382,630	190,185	297,537	123,523
6. Services	25,765	42,276	147,876	509,084	490,982	2,587,543	442,674
7. Other	239	1,349	2,039	48,835	35,110	83,322	36,277

US Make Table for 2003	1	2	3	4	5	6	7
1. Agriculture	273,244	-	-	67	-	1,748	-
2. Mining	-	232,387	-	10,843	-	-	-
3. Construction	-	-	1,063,285	-	-	-	-
4. Manufacturing	-	-	-	3,856,583	-	30,555	3,278
5. Trade, Transport & Utils	-	570	-	-	2,855,126	41	957
6. Services	-	475	-	-	133	9,136,001	3,278
7. Other	3,359	896	-	3,936	104,957	323,996	1,827,119

Below is a table providing the detail of the components of total commodity final demand. Note that the total final demand entry for mining is negative due to a negative trade balance, i.e., the value of net exports (exports minus imports) is negative and is sufficiently large to offset other components of final demand to render total final demand negative.

Commodity Final Demands for U.S. 2003 Input-Output Tables

Commodity\Final Demand	Personal consumption expenditures	Private fixed investment	Change in private inventories	Exports of goods and services	Imports of goods and services	Government consumption expenditures and gross investment	Total Final Demand
Agriculture	47,922	-	175	24,859	(26,769)	(1,136)	45,050
Mining	72	35,698	1,912	4,739	(125,508)	702	(82,384)
Construction	-	704,792	-	71	-	224,468	929,331
Manufacturing	1,301,616	573,197	8,983	506,780	(1,075,128)	94,705	1,410,152
Trade, Transportation & Utili	1,549,792	125,271	2,994	131,884	8,065	10,289	1,828,294
Services	4,780,516	303,426	461	175,546	(44,060)	30,256	5,246,145
Other	80,963	(75,404)	(15,748)	98,989	(177,578)	1,716,238	1,627,459
Total	7,760,881	1,666,980	(1,224)	942,868	(1,440,979)	2,075,522	11,004,047

Suppose that the value for total imports of manufactured goods is projected to increase by \$1 trillion from its 2003 value with, for simplicity, all other elements of total final demand remaining identical to those for 2003. To compute the impact on gross national product and on total output of all sectors of the economy, we first observe that if net exports are reduced by a rise in imports of \$1 trillion, then final demand is reduced by the same amount and, all other values remaining constant, so GDP is also reduced by the same amount.

To estimate the vector total outputs, we must first determine the commodity-by-industry input matrix, **B**, and the commodity output proportions matrix, **D**, to specify the industry-based technology, commodity-by-industry total requirements matrix, $\mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1}$:

$$\mathbf{B}=\begin{bmatrix} 0.225 & 0.000 & 0.001 & 0.038 & 0.000 & 0.002 & 0.001 \\ 0.002 & 0.137 & 0.007 & 0.045 & 0.031 & 0.000 & 0.005 \\ 0.003 & 0.000 & 0.001 & 0.002 & 0.004 & 0.007 & 0.021 \\ 0.173 & 0.094 & 0.249 & 0.321 & 0.046 & 0.057 & 0.100 \\ 0.088 & 0.054 & 0.095 & 0.098 & 0.067 & 0.033 & 0.055 \\ 0.094 & 0.174 & 0.139 & 0.131 & 0.172 & 0.283 & 0.196 \\ 0.001 & 0.006 & 0.002 & 0.013 & 0.012 & 0.009 & 0.016 \end{bmatrix}$$

$$\mathbf{D}=\begin{bmatrix} 0.988 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.992 & 0.000 & 0.003 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.996 & 0.000 & 0.003 & 0.002 \\ 0.000 & 0.002 & 0.000 & 0.000 & 0.964 & 0.000 & 0.001 \\ 0.000 & 0.002 & 0.000 & 0.000 & 0.000 & 0.962 & 0.002 \\ 0.012 & 0.004 & 0.000 & 0.001 & 0.035 & 0.034 & 0.996 \end{bmatrix}$$

$$\mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1}=\begin{bmatrix} 1.290 & 0.011 & 0.023 & 0.076 & 0.007 & 0.011 & 0.012 \\ 0.029 & 1.163 & 0.036 & 0.092 & 0.045 & 0.011 & 0.021 \\ 0.009 & 0.004 & 1.006 & 0.008 & 0.008 & 0.012 & 0.025 \\ 0.377 & 0.207 & 0.421 & 1.551 & 0.118 & 0.145 & 0.206 \\ 0.170 & 0.104 & 0.157 & 0.185 & 1.060 & 0.068 & 0.096 \\ 0.284 & 0.341 & 0.313 & 0.355 & 0.292 & 1.387 & 0.339 \\ 0.044 & 0.035 & 0.030 & 0.048 & 0.068 & 0.068 & 1.035 \end{bmatrix}$$

The revised vector of total final demands is specified by simply reducing the value for total imports of manufactured goods by $1 trillion, so computing the corresponding change in total outputs is found by

$$\Delta\mathbf{x}=\mathbf{D}(\mathbf{I}-\mathbf{BD})^{-1}\Delta\mathbf{f}=-\begin{bmatrix} 75{,}840 & 91{,}880 & 7{,}742 & 1{,}550{,}525 & 184{,}877 & 355{,}257 & 48{,}461 \end{bmatrix}'$$

Computational Notes

We presume that the US 2003 Use and Make matrices are saved as **U** and **V** in the APL workspace the compute the familiar commodity-by-industry commodities **x**, **q**, **B** and **D**.

```
q←+⌿V
x←+/V
B←U AMAT x
D←V AMAT q
```

```
U
   61946.0        1.0     1270.0  147559.0      231.0    18453.0     2093.0
     441.0    33299.0     6927.0  174235.0    89246.0     1058.0    11507.0
     942.0       47.0     1278.0    8128.0    10047.0    65053.0    48460.0
   47511.0    22931.0   265115.0 1249629.0   132673.0   516730.0   226689.0
   24325.0    13211.0   100510.0  382630.0   190185.0   297537.0   123523.0
   25765.0    42276.0   147876.0  509084.0   490982.0  2587543.0   442674.0
     239.0     1349.0     2039.0   48835.0    35110.0    83322.0    36277.0
V
  273244.0        0.0        0.0      67.0        0.0     1748.0        0.0
       0.0   232387.0        0.0   10843.0        0.0        0.0        0.0
       0.0        0.0  1063285.0       0.0        0.0        0.0        0.0
       0.0        0.0        0.0 3856583.0        0.0    30555.0     3278.0
       0.0      570.0        0.0       0.0  2855126.0       41.0      957.0
       0.0      475.0        0.0       0.0      133.0  9136001.0     3278.0
    3359.0      896.0        0.0    3936.0   104957.0   323996.0  1827119.0
x
  275059.0   243230.0  1063285.0 3890416.0  2856694.0  9139887.0  2264263.0
q
  276603.0   234328.0  1063285.0 3871429.0  2960216.0  9492341.0  1834632.0
B
     0.225      0.000      0.001     0.038      0.000      0.002      0.001
     0.002      0.137      0.007     0.045      0.031      0.000      0.005
     0.003      0.000      0.001     0.002      0.004      0.007      0.021
     0.173      0.094      0.249     0.321      0.046      0.057      0.100
     0.088      0.054      0.095     0.098      0.067      0.033      0.055
     0.094      0.174      0.139     0.131      0.172      0.283      0.196
     0.001      0.006      0.002     0.013      0.012      0.009      0.016
D
     0.988      0.000      0.000     0.000      0.000      0.000      0.000
     0.000      0.992      0.000     0.003      0.000      0.000      0.000
     0.000      0.000      1.000     0.000      0.000      0.000      0.000
     0.000      0.000      0.000     0.996      0.000      0.003      0.002
     0.000      0.002      0.000     0.000      0.964      0.000      0.001
     0.000      0.002      0.000     0.000      0.000      0.962      0.002
     0.012      0.004      0.000     0.001      0.035      0.034      0.996
```

Now we can compute the industry-based technology, commodity-by-industry total requirements matrix **T** and use it to evaluate the change in final demand `Δf` in terms of the change in total outputs `Δx`.

```
     T←D+.×INV B+.×D
     e03←45050 ¯82384 929331 1410152 1828294 5246145 1627459
     e04←45050 ¯82384 929331 410152 1828294 5246145 1627459
     Δx←T+.×Δf←e04-e

e03
     45050     ¯82384     929331    1410152    1828294    5246145    1627459
e04
     45050     ¯82384     929331     410152    1828294    5246145    1627459
Δf
         0          0          0   ¯1000000          0          0          0
```

```
Δx
    ¯75840     ¯91880      ¯7742  ¯1550525   ¯184877   ¯355257    ¯48461
```

Chapter 6, Multipliers in the Input–Output Model

Chapter 6 examines key summary analytical measures known as multipliers that can be derived from input–output models to estimate the effects of exogenous changes on (1) new outputs of economic sectors, (2) income earned by households resulting from new outputs, and (3) employment generated from new outputs or (4) value-added generated by production or (5) energy and environmental effects.

The chapter develops the general structure of multiplier analysis and special considerations associated with regional, IRIO, and MRIO models. Extensions to capture the effects of income generation for various household groups are then explored, as well as additional multiplier variants. Chapter appendices expand on mathematical formulations of household and income multipliers.

The exercise problems for this chapter illustrate various types of input-output multipliers and their applications.

Problem 6.1: Total Output Multipliers

This problem explores the use of total output multipliers as an indicator of relative importance to the economy using the input-output tables utilized in the exercise Problems 2.1 through 2.10.

Problem 6.1 Overview

For exercise Problems 2.1 through 2.10, the output multipliers, the column sums of the Leontief inverse in each case (with the largest multiplier in each case highlighted in boldface), are the following:

Problem	Output Multipliers							
2.1	6.444	**6.944**						
2.2	2.970	**4.167**	3.611					
2.3	6.444	**6.944**						
2.4	2.006	**2.428**	1.307					
2.5	1.412	**1.588**						
2.6	**1.839**	1.437						
2.7	2.301	2.031	2.209	2.035	1.551	1.616	2.156	**2.364**
2.8	1.716	**1.814**						
2.9	1.919	1.605	1.722	**1.925**	1.487	1.608	1.599	
2.10	4.000	**5.000**	1.000					

Computational Notes

The APL expression `mult←+⌿L←LINV A` applied to the tables used in the ten problems yields the following values of **L** followed by the values for `mult` for each problem.

```
2-1
        4.074 3.241
        2.370 3.704
        6.444 6.944
```

2-2

1.538	0.000	0.000
0.449	2.500	0.833
0.983	1.667	2.778
2.970	4.167	3.611

2-3

4.074	3.241
2.370	3.704
6.444	6.944

2-4

1.092	0.269	0.040
0.084	1.154	0.036
0.830	1.005	1.230
2.006	2.428	1.307

2-5

1.147	0.353
0.265	1.235
1.412	1.588

2-6

1.494	0.230
0.345	1.207
1.839	1.437

2-7

1.339	0.296	0.312	0.172	0.034	0.058	0.030	0.067
0.089	1.214	0.209	0.153	0.038	0.057	0.025	0.051
0.013	0.009	1.011	0.019	0.034	0.008	0.018	0.013
0.065	0.056	0.034	1.306	0.038	0.041	0.021	0.041
0.265	0.215	0.320	0.174	1.207	0.230	0.229	0.240
0.100	0.029	0.045	0.059	0.011	1.089	0.018	0.074
0.109	0.049	0.068	0.035	0.054	0.030	1.547	0.372
0.321	0.162	0.210	0.117	0.135	0.103	0.269	1.506
2.301	2.031	2.209	2.035	1.551	1.616	2.156	2.364

2-8

1.257	0.437
0.459	1.377
1.716	1.814

2-9

1.262	0.006	0.013	0.057	0.004	0.007	0.007
0.009	1.075	0.012	0.034	0.019	0.003	0.007
0.008	0.003	1.005	0.006	0.007	0.011	0.025
0.229	0.119	0.262	1.342	0.069	0.086	0.126
0.149	0.085	0.137	0.156	1.089	0.060	0.085
0.238	0.293	0.270	0.292	0.271	1.412	0.314
0.024	0.024	0.023	0.037	0.028	0.030	1.034
1.919	1.605	1.722	1.925	1.487	1.608	1.599

```
2-10
          2.500 2.500 0.000
          0.500 1.500 0.000
          1.000 1.000 1.000
          4.000 5.000 1.000
```

Problem 6.2: Using Output Multipliers to Compute Impact Analysis Total Outputs

This problem explores the use of output multipliers to derive the total value of output (across all sectors) associated with the new final demands, again using the exercise problems in Chapter 2.

Problem 6.2 Overview

Using the already-calculated the multipliers in Problem 6.1 in conjunction with the new final demands in the problems in Chapter 2, we can derive the total value of output (across all sectors) associated with the new final demands.

Using Problem 2.2 as an example, the row vector of output multipliers is $\mathbf{m}(o) = [2.970 \quad 4.167 \quad 3.611]$. In conjunction with the final-demand vector used in that problem, namely $\mathbf{f}^{t+1} = \begin{bmatrix} 1{,}300 \\ 100 \\ 200 \end{bmatrix}$, we find $\mathbf{m}(o)^{t+1} = 5{,}000$. In the solution to Problem 2.2, we found that $\mathbf{x}^{t+1} = \begin{bmatrix} 2{,}000 \\ 1{,}000 \\ 2{,}000 \end{bmatrix}$, and the sum of these elements is 5,000; that is, $\mathbf{i}'\mathbf{x}^{t+1} = 5{,}000$. In matrix notation, this is comparing $\mathbf{m}(o)\Delta\mathbf{f}$ with $\mathbf{i}'\Delta\mathbf{x} = \mathbf{i}'\mathbf{L}\Delta\mathbf{f}$; we know that they must be equal, since output multipliers are the column sums of the Leontief inverse— $\mathbf{m}(o) = \mathbf{i}'\mathbf{L}$.

Computational Notes

The values of **Z** and **x** are specified and the corresponding values of **A** and **L** are computed

```
      Z←3 3⍴350 0 0 50 250 150 200 150 550
      x←1000 500 1000
      L←LINV A←Z AMAT x

Z
        350          0          0
         50        250        150
        200        150        550
x
       1000        500       1000
A
      0.350      0.000      0.000
      0.050      0.500      0.150
      0.200      0.300      0.550
```

```
L
     1.538     0.000     0.000
     0.449     2.500     0.833
     0.983     1.667     2.778
```

For the new vector of final demands, defined as `f2`, the corresponding vector of total outputs `x2` is computed with **L**, along with the total of all outputs `xt2` and the output multipliers mo.

```
   x2←L+.×f2←1300 100 200
   mo←+⌿L
   xt2←+/mo×f2

f2
    1300.0     100.0     200.0
x2
    2000.0    1000.0    2000.0
mo
     2.970     4.167     3.611
xt2
    5000.0
```

Problem 6.3: Type I and Type II Income Multipliers

This problem explores type I and type II income multipliers in addition to total output multipliers.

Problem 6.3 Overview

Using the data in Problem 2.3 of a model closed to households, which included the matrix of interindustry transactions, $\mathbf{Z}^c = \begin{bmatrix} 500 & 350 & 90 \\ 320 & 360 & 50 \\ 100 & 60 & 40 \end{bmatrix}$, and vector of total outputs, $\mathbf{x}^c = \begin{bmatrix} 1{,}000 \\ 800 \\ 300 \end{bmatrix}$, we could the matrices of direct and total requirements, respectively, as

$$\mathbf{A}^c = \mathbf{Z}^c(\hat{\mathbf{x}}^c)^{-1} = \begin{bmatrix} .5 & .438 & .3 \\ .32 & .45 & .167 \\ .1 & .075 & .133 \end{bmatrix} \text{ and } \mathbf{L}^c = (\mathbf{I} - \mathbf{A}^c)^{-1} = \begin{bmatrix} 5.820 & 5.036 & 2.983 \\ 3.686 & 5.057 & 2.248 \\ 0.990 & 1.019 & 1.693 \end{bmatrix}$$

The output multipliers for the three-sector model, closed with respect to households, are $\mathbf{m}(o) = [10.496 \quad 11.112 \quad 6.924]$. The type I income multipliers require that we have the labor-input coefficients, which are $a_{31} = 0.100$ and $a_{32} = 0.075$, along with the Leontief inverse of the model that is open with respect to households (from Problem 2.3), $(\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 4.074 & 3.241 \\ 2.370 & 3.704 \end{bmatrix}$.

Then $m(h)_1 =$ (0.1)(4.074)+(0.075)(2.370) = 0.5852 and $m(h)_2 =$ (0.1)(3.241) + (0.075)(3.704) = 0.6019; $m(h)_1^I = 0.5852/0.1 = 5.852$ and $m(h)_2^I = 0.6019/0.075 = 8.025$.

The total household income multipliers can be found as the first two elements in the bottom row of the Leontief inverse of the model closed with respect to households, $\mathbf{L}^c = (\mathbf{I} - \mathbf{A}^c)^{-1}$, which are $m(h)_1 =$ 0.990 and $m(h)_2 =$ 1.019, so the type II income multipliers are therefore $m(h)_1^{II} = 0.990/0.1 = 9.90$ and $m(h)_2^{II} = 1.019/0.075 = 13.59$. Note that, for both sectors, the ratio of the type II to the type I income multiplier is 1.69.

Computational Notes

The three sector versions of the matrix of interindustry transactions `Z3` and vector of total outputs `x3` are used to compute `A3` and `L3`.

```
     Z3←3 3ρ500 350 90 320 360 50 100 60 40
     x3←1000 800 300
     L3←LINV A3←Z3 AMAT x3

Z3
       500       350        90
       320       360        50
       100        60        40
x3
      1000       800       300
A3
     0.500     0.438     0.300
     0.320     0.450     0.167
     0.100     0.075     0.133
L3
     5.820     5.036     2.983
     3.686     5.057     2.248
     0.990     1.019     1.693
```

The two sector versions of the matrix of interindustry transactions `Z2` and vector of total outputs `x2` are used to compute `A2` and `L2`.

```
     Z2←2 2ρ500 350 320 360
     x2←1000 800
     f2←x2-+/Z2
     L2←LINV A2←Z2 AMAT x2

Z2
       500       350
       320       360
f2
       150       120


x2
      1000       800
A2
     0.500     0.438
     0.320     0.450
L2
     4.074     3.241
     2.370     3.704
```

For the three-sector and two-vectors of final demands, `f3` and `f32`, respectively, the corresponding total outputs are found by

```
      x3←L3+.×f3←200 100 0
      x32←L2+.×f32←200 100

f3
200 100 0
x3
  1667.537  1242.921   299.969
f32
   200.000   100.000
x32
  1138.889   844.444
```

The total output multipliers for the two- and three-sectors models, `mo2` and `mo3`, respectively are found as

```
      mo2←+⌿L2
      mo3←+⌿L3

mo2
     6.444     6.944
mo3
    10.496    11.112     6.924
```

The type I and type II income multipliers and the ratio of type II to type I income multipliers are computed as

```
      l2←A3[3;1 2]
      mh←l2+.×L2
      mhI←mh÷l2
      mbh←L3[3;1 2]
      mhII←mbh÷l2
      Ratio←mhII÷mhI

l2
     0.100     0.075
mh
     0.585     0.602
mhI
     5.852     8.025
mbh
     0.990     1.019
mhII
     9.905    13.583
Ratio
     1.693     1.693
```

Problem 6.4: Policy Analysis with Input-Output Multipliers

This problem configures a typical policy question that can be addressed with input-output multipliers.

Problem 6.4 Overview

Suppose we assemble the following facts about the two sectors that make up the economy of a small country under study where the available data pertain to the most recent quarter. Total interindustry inputs were \$50 and \$100, respectively, for Sectors 1 and 2. Sector 1's sales to final demand were \$60 and Sector 1's total output was \$100. Sector 2's sales to Sector 1 were \$30 and this represented 10 percent of Sector 2's total output. If we define the matrix of interindustry transactions as $\mathbf{Z} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \end{bmatrix}$ and the vectors of final demands, interindustry inputs, and total outputs, respectively, as $\mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}$, and $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, we can summarize the known values as $z_{21} = 30$, $f_1 = 10$, $v_1 = 50$, $v_2 = 100$, and $x_1 = 100$. The value of x_2 is easily found from $x_2 = z_{21} / 0.1 = 120$ and the unavailable data for z_{11}, z_{12}, z_{22}, and f_2 can be computed from the basic accounting identities, $\mathbf{Zi} + \mathbf{f} = \mathbf{x}$ and $\mathbf{i'Z} + \mathbf{v} = \mathbf{x}$ so that **Z**, **v**, **f**, and **x** are, respectively, $\mathbf{Z} = \begin{bmatrix} 20 & 20 \\ 30 & 80 \end{bmatrix}$, $\mathbf{v} = \begin{bmatrix} 50 & 100 \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} 60 \\ 10 \end{bmatrix}$. and $\mathbf{x} = \begin{bmatrix} 100 \\ 120 \end{bmatrix}$, from which we can then calculate the direct requirements and total requirements matrices, respectively, as $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .2 & .067 \\ .3 & .267 \end{bmatrix}$ and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.294 & .118 \\ .529 & 1.412 \end{bmatrix}$.

If we project that, after national elections are held, it may turn out that different government policy will be forthcoming during the first quarter of the coming year. For example, if there is an increase of \$100 in government purchases of sector 1's output, we specify the projected change in total final demand as $\Delta\mathbf{f}^1 = \begin{bmatrix} 160 \\ 190 \end{bmatrix}$, while if the same increase is of sector 2's output, we specify the change in final demand as $\Delta\mathbf{f}^2 = \begin{bmatrix} 60 \\ 290 \end{bmatrix}$.

We can compare the stimulative effect of the two scenarios by calculating the sum of total outputs for each that would be necessary to support the changed final demands, i.e., $\Delta\mathbf{x}^1 = \mathbf{i'L}\Delta\mathbf{f}^1 = 582.35$ and $\Delta\mathbf{x}^2 = \mathbf{i'L}\Delta\mathbf{f}^2 = 552.94$. The first option generates the larger stimulative effect by $\Delta\mathbf{x}^1 - \Delta\mathbf{x}^2 = 582.35 - 552.94 = 29.41$.

Computational Notes

We can summarize the known values in APL statements with the following:

```
x←100,30÷0.1
f←60 0
u←50 100
Z←2 2ρ0
Z[2;1]←30
Z[1;1]←u[1]-Z[2;1]
```

```
      Z[1;2]←x[1]-(Z[1;1]+f[1])
      Z[2;2]←u[2]-Z[1;2]
      f[2]←x[2]-(Z[2;1]+Z[2;2])

      x
100 300
      f
60 190
      u
50 100
      Z
20 20
30 80
```

We can then compute the corresponding **A** and **L**; then for the alternative final demand vectors defined as `f1` and `f2` the corresponding sums of the vectors total outputs defined as `Δx1` and `Δx2`, respectively, and the absolute difference between them, defined as `ΔxΔ`, are computed as

```
       L←LINV A←Z AMAT x
       Δx1←+/L+.×f1←f+100 0
       Δx2←+/L+.×f2←f+0 100
       ΔxΔ←|Δx1-Δx2

       A
      0.200     0.067
      0.300     0.267
       L
      1.294     0.118
      0.529     1.412

      f1
160 190
      Δx1
582.35294
      f2
60 290
      Δx2
552.94118
      ΔxΔ
29.411765
```

Problem 6.5: Economic Planning Considerations

This problem explores a typical economic planning question addressed with input-output analysis.

Problem 6.5 Overview

Consider an input output economy defined by $\mathbf{Z} = \begin{bmatrix} 140 & 350 \\ 800 & 50 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 1{,}000 \\ 1{,}000 \end{bmatrix}$. Suppose economic planners are asked to design an advertising campaign to stimulate export sales of one of the goods produced in the country and need to determine which of the two sectors on which to concentrate their efforts or perhaps if some combination would be more effective. The answer rests on the relative size of the output multipliers, which will indicate the relative stimulative effect of focusing on one sector or the other (or a combination).

The output multipliers are found by first computing the technical coefficients matrix, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .14 & .35 \\ .8 & .05 \end{bmatrix}$, and the total requirements matrix, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.769 & .652 \\ 1.490 & 1.601 \end{bmatrix}$, so the vector of output multipliers is $\mathbf{m}(o) = \mathbf{i}'\mathbf{L} = \begin{bmatrix} 3.259 & 2.253 \end{bmatrix}$. So, in terms of relative stimulative effect, it is more effective to concentrate on stimulating export demand for the product of sector 1; since it has a considerably larger output multiplier.

If we determine labor income coefficients for the two sectors in the region to be $a_{31} = 0.1$ and $a_{32} = 0.18$, and the focus is on job creation, it is possible the priorities may change. In this case, knowing $a_{31} = 0.1$ and $a_{32} = 0.18$, we can find $H_1 = 0.4451$ and $H_2 = 0.3534$ by $\mathbf{H} = \mathbf{lL} = \begin{bmatrix} .4451 & .3534 \end{bmatrix}$ where the vector of labor coefficients is $\mathbf{l} = \begin{bmatrix} .1 & .18 \end{bmatrix}$. Thus, converting output effects to income earned per dollar of new final demand for each of the sectors does not change the ranking, so, in this case, stimulation of export demand for the output of sector 1 is still more beneficial.

Computational Notes

Define **Z** and **x** and compute **A**, **L**, and the total output multipliers m.

```
      Z←2 2⍴140 350 800 50
      x←1000 1000
      m←+⌿L←LINV A←Z AMAT x

      Z

        140       350
        800        50
      x
       1000      1000
      A
      0.140     0.350
      0.800     0.050
      L
      1.769     0.652
      1.490     1.601
      m
      3.259     2.253
```

Use the labor coefficients **l** to calculate the labor input multipliers **mh**.

```
      l←0.1 0.18
      mh←l+.×L

      l
0.1 0.18
      mh
0.44506518 0.35344507
```

Problem 6.6: Interregional Input-Output Multipliers in IRIO Models

This problem explores interregional input-output multipliers.

Problem 6.6 Overview

Recall the elements in the two-region interregional Leontief inverse from Problem 3.2:

$$\mathbf{L} = (\mathbf{I}-\mathbf{A})^{-1} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{L}_{12} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} = \left[\begin{array}{cc|cc} 1.205 & .202 & .115 & .123 \\ .263 & 1.116 & .189 & .131 \\ \hline .273 & .262 & 1.177 & .2 \\ .33 & .289 & .179 & 1.156 \end{array}\right].$$

We can first calculate the vectors of simple intraregional output multipliers for sectors 1 and 2 as $\mathbf{m}(o)^{rr} = \mathbf{i}'\mathbf{L}_{11} = \begin{bmatrix}1.468 & 1.318\end{bmatrix}$ and $\mathbf{m}(o)^{ss} = \mathbf{i}'[\mathbf{L}_{22}] = [1.356 \quad 1.356]$. The vectors of simple national (total) output multipliers for sectors 1 and 2 are

$\mathbf{m}(o)^{r} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{11} \\ \mathbf{L}_{21}\end{bmatrix} = \begin{bmatrix}2.070 & 1.869\end{bmatrix}$ and $\mathbf{m}(o)^{s} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{12} \\ \mathbf{L}_{22}\end{bmatrix} = \begin{bmatrix}1.660 & 1.610\end{bmatrix}$.

Finally, the sector-specific simple national output multipliers for sectors 1 and 2 in regions *r* and *s*. show the impact on sector *i* throughout the entire country, because of a dollar's worth of final demand for sector *j* in either region. In this case it means finding the four multipliers for each region as:

$\mathbf{m}(o)^{\bullet r} = \begin{bmatrix} m(o)^{\bullet r}_{11} & m(o)^{\bullet r}_{21} & m(o)^{\bullet r}_{12} & m(o)^{\bullet r}_{22}\end{bmatrix} = \begin{bmatrix}1.478 & 0.593 & 0.464 & 1.405\end{bmatrix}$ and

$\mathbf{m}(o)^{\bullet s} = \begin{bmatrix} m(o)^{\bullet s}_{11} & m(o)^{\bullet s}_{21} & m(o)^{\bullet s}_{12} & m(o)^{\bullet s}_{22}\end{bmatrix} = \begin{bmatrix}1.292 & 0.368 & 0.323 & 1.287\end{bmatrix}$.

Problem 6.7: Additional Characteristics of IRIO multipliers

This problem further explores the characteristics of interregional input-output multipliers. Using the results of Problem 6.6, to determine which sector's output increases the most for an arbitrary new final demand in the two regions, we can first simply compare the intraregional multipliers for each sector in each region, $\mathbf{m}(o)^{rr} = \mathbf{i}'[\mathbf{L}_{11}] = [1.468 \quad 1.318]$ and $\mathbf{m}(o)^{ss} = \mathbf{i}'[\mathbf{L}_{22}] = [1.356 \quad 1.356]$. In region *r* sector 1's multiplier is larger than sector 2's (1.468>1.318) and in region *s* the multipliers are equal for the two sectors.

To determine which sector in which region produces the largest national (two-region) impact for an arbitrary increase in final demand we compare the sector-specific simple national output multipliers:

$\mathbf{m}(o)^{\bullet r} = \begin{bmatrix} m(o)^{\bullet r}_{11} & m(o)^{\bullet r}_{21} & m(o)^{\bullet r}_{12} & m(o)^{\bullet r}_{22}\end{bmatrix} = \begin{bmatrix}1.478 & 0.593 & 0.464 & 1.405\end{bmatrix}$ and

$\mathbf{m}(o)^{\bullet s} = \begin{bmatrix} m(o)^{\bullet s}_{11} & m(o)^{\bullet s}_{21} & m(o)^{\bullet s}_{12} & m(o)^{\bullet s}_{22}\end{bmatrix} = \begin{bmatrix}1.292 & 0.368 & 0.323 & 1.287\end{bmatrix}$, the largest of which is 1.478 for sector 1 in region *r*.

To determine whether it would be better to institute policies that would increase household demand in region *r* or in region *s*, increasing the output of sector 1 nationally (i.e., in both regions), we compare the total interregional output multipliers

$\mathbf{m}(o)^{r} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{11}\\ \mathbf{L}_{21}\end{bmatrix} = \begin{bmatrix}2.070 & 1.869\end{bmatrix}$ and $\mathbf{m}(o)^{s} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{12}\\ \mathbf{L}_{22}\end{bmatrix} = \begin{bmatrix}1.660 & 1.610\end{bmatrix}$. The multiplier for sector 1 in region r is larger than the corresponding multiplier in region s (2.07>1.66) so increasing household demand in region r is more beneficial. The same is true for sector 2 (1.869>1.610).

Problem 6.8: MRIO Multipliers

This problem explores the basic characteristics of multiregional input-output (MRIO) multipliers.

Problem 6.8 Overview

Recall the MRIO model defined in Problem 3.3. The elements in $(\mathbf{I}-\mathbf{CA})^{-1}\mathbf{C}$ from that problem are: $(\mathbf{I}-\mathbf{CA})^{-1}\mathbf{C} = \left[\begin{array}{cc|cc} 0.971 & 0.556 & 1.024 & 0.524 \\ 0.882 & 1.197 & 0.889 & 1.251 \\ \hline 1.297 & 0.714 & 1.264 & 0.677 \\ 0.663 & 1.010 & 0.673 & 0.854 \end{array}\right]$. To determine which sector's output increases the most for an arbitrary new final demand in the two regions, we simply compare the intraregional multipliers for each sector in each region, $\mathbf{m}(o)^{rr} = \mathbf{i}'[\mathbf{L}_{11}] = [1.853 \quad 1.753]$ and $\mathbf{m}(o)^{ss} = \mathbf{i}'[\mathbf{L}_{22}] = [1.937 \quad 1.530]$. In region r sector 1's multiplier is greater than sector 2's (1.853>1.753) and in region s the same is true (1.937>1.530).

To determine which sector in which region produces the largest national (two-region) impact for an arbitrary increase in final demand we compare the sector-specific simple national output multipliers:
$\mathbf{m}(o)^{\bullet r} = \begin{bmatrix} m(o)^{\bullet r}_{11} & m(o)^{\bullet r}_{21} & m(o)^{\bullet r}_{12} & m(o)^{\bullet r}_{22}\end{bmatrix} = \begin{bmatrix}2.269 & 1.545 & 1.270 & 2.207\end{bmatrix}$ and
$\mathbf{m}(o)^{\bullet s} = \begin{bmatrix} m(o)^{\bullet s}_{11} & m(o)^{\bullet s}_{21} & m(o)^{\bullet s}_{12} & m(o)^{\bullet s}_{22}\end{bmatrix} = \begin{bmatrix}2.288 & 1.562 & 1.201 & 2.105\end{bmatrix}$, the largest of which is 2.288 for sector 1 in region s.

To determine whether it would be better to institute policies that would increase household demand in region r or in region s so as to increase the output of sector 1 nationally (i.e., in both regions), we compare the total interregional output multipliers
$\mathbf{m}(o)^{r} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{11}\\ \mathbf{L}_{21}\end{bmatrix} = \begin{bmatrix}3.813 & 3.477\end{bmatrix}$ and $\mathbf{m}(o)^{s} = \mathbf{i}'\begin{bmatrix}\mathbf{L}_{12}\\ \mathbf{L}_{22}\end{bmatrix} = \begin{bmatrix}3.849 & 3.306\end{bmatrix}$. The multiplier for sector 1 in region s is larger than the corresponding multiplier in region r (3.849>3.813) so increasing household demand in region s is more beneficial. The opposite is true for sector 2, i.e., The multiplier for sector 1 in region r is larger than the corresponding multiplier in region s (3.477>3.306) so increasing household demand in region r is more beneficial.

Computational Notes

First generate the total requirements coefficients matrix from the MRIO data specified in Problem 3.3.

```
      ZZ←4 4⍴Z←2 2 2⍴40 50 60 10 30 45 70 45
      QQ←4 2⍴Q←2 2 2⍴50 60 70 70 50 80 50 50
      L←(INV C+.×A←Z GENA x←+/Q)+.×C←GENC Q
```

```
L
      0.971     0.556     1.024     0.524
      0.882     1.197     0.889     1.251
      1.297     0.714     1.264     0.677
      0.663     1.010     0.673     0.854
```

Compute the intraregional output multipliers for regions r and s, as `mrr` and `mss`, respectively.

```
      mrr←+⌿L[1 2;1 2]
      mss←+⌿L[3 4;3 4]
```

```
mrr
      1.853     1.753
mss
      1.937     1.530
```

Compute and compare the total interregional output multipliers for region r as `mr` and region s as `ms`.

```
      mr←+⌿L[;1 2]
      ms←+⌿L[;3 4]
```

```
mr
      3.813     3.477
ms
      3.849     3.305
```

Compute and compare the sector-specific simple national output multipliers for region r as `mdr` and region s as `mds`.

```
      mdr←,⍉L[1 2;1 2]+L[3 4;1 2]
      mds←,⍉L[1 2;3 4]+L[3 4;3 4]
```

```
mdr
      2.269     1.545     1.270     2.207
mds
      2.288     1.562     1.201     2.105
```

Problem 6.9: Policy Analysis with Regional Output Multipliers

This problem explores the use of regional output multipliers in analysis of a typical policy problem.

Problem 6.9 Overview

Using the basic data introduced in Problem 3.4, suppose the government is interested in starting an overseas advertising and promotion campaign aimed at increasing export sales of the products of the country. There is specialization of production in the regions of the country; in particular, the products are shown in the table below:

	Region A	Region B	Region C
Manufacturing	Scissors	Cloth	Pottery
Agriculture	Oranges	Walnuts	None

To determine the product for which an increase in export sales would produce the greatest stimulation of the national economy, we calculate the total regional output multipliers for each region:

$$\mathbf{m}(o)^A = \mathbf{i}'(\mathbf{I}-\mathbf{A}^A)^{-1} = \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}1.714 & 0.857\\ 0.429 & 1.714\end{bmatrix} = \begin{bmatrix}2.143 & 2.571\end{bmatrix}$$

$$\mathbf{m}(o)^B = \mathbf{i}'(\mathbf{I}-\mathbf{A}^B)^{-1} = \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}2.857 & 2.286\\ 0.333 & 1.667\end{bmatrix} = \begin{bmatrix}3.190 & 3.952\end{bmatrix}$$

$$\mathbf{m}(o)^C = \mathbf{i}'(\mathbf{I}-\mathbf{A}^C)^{-1} = \begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}2.0 & 0\\ 0.5 & 1.0\end{bmatrix} = \begin{bmatrix}2.5 & 1.0\end{bmatrix}$$

The largest total output multiplier is associated with sector 2 in region B (3.952); that is, with walnuts (this of course ignores and interregional multiplier effects that might be found with an IRIO or MRIO model).

Computational Notes

Retrieve the basic data from the example from Section 6.2.1 in the text for the matrices of interindustry transactions as `Z1`, `Z2`, and `Z3`, and for the corresponding vectors of total outputs as `x1`, `x2`, and `x3`. Then compute the corresponding matrices of technical coefficients `A1`, `A2`, and `A3` and of total requirements `L1`, `L2`, and `L3`. Finally compute the corresponding output multipliers of each as `m1`, `m2`, and `m3`.

```
Z1←2 2⍴200 100 100 100
Z2←2 2⍴700 400 100 200
Z3←2 2⍴100 0 50 0
x1←600 300
x2←1200 700
x3←200 0

m1←+⌿L1←LINV A1←Z1 AMAT x1
m2←+⌿L2←LINV A2←Z2 AMAT x2
A3←Z3 AMAT x3
A3[;2]←0
m3←+⌿L3←LINV A3
```

Note an important point about the calculation of `A3`. The actual result of executing the APL statement `A3←Z3 AMAT x3` is

```
      Z3 AMAT x3
0.5  1
0.25 1
```

This is incorrect since the technical coefficients in the second column should both be zero but while the general convention in APL is that division by 0 results in a `DOMAIN ERROR` , an exception is that the expression `0÷0` results in unity, at least by default. The default behavior in most APL implementations can be changed since such a convention may be inappropriate (as it is here) but in this particular case it is simpler to just refine the offending technical coefficients as zero (as in the next line).

```
A1, A2, A3
     0.333     0.333     0.583     0.571     0.500     0.000
     0.167     0.333     0.083     0.286     0.250     0.000
L1, L2, L3
     1.714     0.857     2.857     2.286     2.000     0.000
     0.429     1.714     0.333     1.667     0.500     1.000
m1, m2, m3
     2.143     2.571     3.190     3.952     2.500     1.000
```

Problem 6.10: The Relationships Between Type I and Type II Income Multipliers

This problem explores the relationships between Type I and Type II income multipliers.

Problem 6.10 Overview

We use the example provided in Section 6.2.1 (revisited from Section 2.5), which began with the matrix of interindustry transactions, $\mathbf{Z} = \begin{bmatrix} 150 & 500 \\ 200 & 100 \end{bmatrix}$ and the vector of total outputs,

$\mathbf{x} = \begin{bmatrix} 1{,}000 \\ 2{,}000 \end{bmatrix}$ from which we can derive the matrix of technical coefficients,

$\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .15 & .25 \\ .20 & .05 \end{bmatrix}$ and corresponding matrix of total requirements, $\mathbf{L} = \begin{bmatrix} 1.254 & .330 \\ .264 & 1.122 \end{bmatrix}$.

We developed this model closed to households as $\overline{\mathbf{A}} = \left[\begin{array}{cc:c} .15 & .25 & .05 \\ .20 & .05 & .40 \\ \hdashline .30 & .25 & .05 \end{array}\right]$ with the corresponding

matrix of total requirements computed as $\overline{\mathbf{L}} = (\mathbf{I} - \overline{\mathbf{A}})^{-1} = \left[\begin{array}{cc:c} 1.365 & .425 & .251 \\ .527 & 1.348 & .595 \\ \hdashline .570 & .489 & 1.289 \end{array}\right]$.

Here, $|(\mathbf{I} - \mathbf{A})| = 0.7575$ and $|(\mathbf{I} - \overline{\mathbf{A}})| = 0.587875$, giving $|(\mathbf{I} - \mathbf{A})| / |(\mathbf{I} - \overline{\mathbf{A}})| = 1.289$, which is the same as $\bar{l}_{33}$ in $\overline{\mathbf{L}}$.

In (A6.2.2) from Appendix 6.2 we showed in general for $\overline{\mathbf{L}}$, partitioned as $\overline{\mathbf{L}} = (\mathbf{I} - \overline{\mathbf{A}})^{-1} = \begin{bmatrix} \overline{\mathbf{L}}_{11} & \overline{\mathbf{L}}_{12} \\ \overline{\mathbf{L}}_{21} & \overline{\mathbf{L}}_{22} \end{bmatrix}$, that $\overline{\mathbf{L}}_{21} = -\overline{\mathbf{L}}_{21}(\mathbf{GE}^{-1}) = -\overline{\mathbf{L}}_{22}(\mathbf{GL})$ and $\mathbf{h}_c'\mathbf{L} = -\mathbf{GL}$ where **E**, **F**, and

G are defined by $(\mathbf{I}-\bar{\mathbf{A}})=\begin{bmatrix}\mathbf{E} & \mathbf{F}\\ \mathbf{G} & \mathbf{H}\end{bmatrix}$ and $\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}$. Finally, $\mathbf{h}'_c=\mathbf{h}'\hat{\mathbf{x}}^{-1}$, where **h** is household employment in units such as person-years, **h**. Then it was shown that the row vector of the ratios of the Type II to Type I income multipliers, $R_j = m(h)^{II}_j / m(h)^{I}_j$, is $\underset{(1\times 1)}{\mathbf{R}=-\bar{\mathbf{L}}_{22}}\underset{(1\times n)}{(\mathbf{GL})}\underset{(n\times n)}{[\langle -\mathbf{GL}\rangle]^{-1}}=\underset{(1\times 1)}{\bar{\mathbf{L}}_{22}}\underset{(1\times n)}{[1,\cdots,1]}=\bar{\mathbf{L}}_{22}\mathbf{i}'$; that is, the ratios are all the same and are equal to the element in the lower-right of the closed model inverse. Here $\bar{\mathbf{L}}_{22}=\bar{l}_{33}=1.289$.

Computational Notes

Retrieve the matrices of technical coefficients for the example provided in Section 6.2.1 (revisited from Section 2.5) for the open model (defined above as **A)** by the variable `A` and the closed model (defined above as $\bar{\mathbf{A}}$) by the variable `A2`. For convenience, generate 2nd and 3rd order identity matrices `I2` and `I3`, respectively and compute the determinants. $|(\mathbf{I}-\mathbf{A})|$ and $|(\mathbf{I}-\bar{\mathbf{A}})|$, as `D1` and `D2`, respectively along with the ratio of `D1` to `D2` as `R`.

```
A←2 2ρ0.15 0.25 0.2 0.05
A2←3 3ρ0.15 0.25 0.05 0.2 0.05 0.4 0.3 0.25 0.05
I2←2 2ρ1,2ρ0
I3←3 3ρ1,3ρ0

D1←DETER I2-A
D2←DETER I3-A2
R←D1÷D2
```

```
A
    0.1500    0.2500
    0.2000    0.0500


A2
    0.1500    0.2500    0.0500
    0.2000    0.0500    0.4000
    0.3000    0.2500    0.0500
D1
    0.7575
D2
    0.5879
R
    1.2885
```

Chapter 7, Supply-Side Models, Linkages, and Important Coefficients

Chapter 7 presents the supply side input–output model. It is discussed both as a quantity model (the early interpretation) and as a price model (the more modern interpretation). Relationships to the standard Leontief quantity and price models are also explored. In addition, the fast-growing literature on quantification of economic linkages and analysis of the overall structure of economies using input–output data is examined. Finally, approaches for identifying key or important coefficients in input–output models and alternative measures of coefficient importance are presented.

The exercise problems for this chapter illustrate the configuration of supply side input-models and measures of forward and backward economic linkages in both demand and supply models.

Problem 7.1: The Output Inverse in Supply-Side Input-Output Models

This problem explores the basic properties of the output inverse in a supply side input output model.

Problem 7.1 Overview

Consider the centrally planned economy of Czaria, which is involved in its planning for the next fiscal year. The matrix of technical coefficients, **A**, and vector of total industry outputs, **x**, for Czaria are given as the following:

	1	2	3	4	Total Output
1. Agriculture	0.168	0.155	0.213	0.212	12,000
2. Mining	0.194	0.193	0.168	0.115	15,000
3. Military Manufacturing	0.105	0.025	0.126	0.124	12,000
4. Civilian Manufacturing	0.178	0.101	0.219	0.186	16,000

From the table we define $\mathbf{B} = \hat{\mathbf{x}}^{-1}\mathbf{Z} = \begin{bmatrix} .168 & .194 & .213 & .283 \\ .155 & .193 & .134 & .123 \\ .105 & .031 & .126 & .165 \\ .134 & .095 & .164 & .186 \end{bmatrix}$ for $\mathbf{Z} = \mathbf{A}\hat{\mathbf{x}}$. The output inverse for this economy is then $\mathbf{G} = (\mathbf{I} - \mathbf{B})^{-1} = \begin{bmatrix} 1.468 & .455 & .558 & .692 \\ .376 & 1.393 & .384 & .418 \\ .253 & .155 & 1.300 & .375 \\ .336 & .268 & .399 & 1.466 \end{bmatrix}$. The next year's value-added inputs for agriculture, mining, military manufacturing products, and civilian manufacturing in Czaria are projected to be \$4,558 million, \$5,665 million, \$2,050 million and \$5,079 million, respectively. The nation's projected *GDP*, since it is the sum of either all final demands or value added, i.e., $GDP = \mathbf{i}'\mathbf{f} = \mathbf{v}'\mathbf{i}$, can be computed very simply for the projected new final demands, $(\mathbf{v}^{new})' = [4{,}558 \quad 5{,}665 \quad 2{,}050 \quad 5{,}079]$, as $GDP^{new} = (\mathbf{v}^{new})'\mathbf{i} = 17{,}352$, the sum of all new value-added inputs, compared with the current value $GDP = \mathbf{v}'\mathbf{i} = 21{,}246$. The corresponding vector of new total gross production is

$\mathbf{x}^{new} = [13{,}928.5 \quad 12{,}518.4 \quad 6{,}606.6 \quad 11{,}313.2]'$, found by $\mathbf{x}^{new} = (\mathbf{I} - \mathbf{B}')^{-1}\mathbf{v}^{new}$, the supply side model. Note that this is the "old view" of the Ghosh model as described in Section 7.1.1.

Computational Notes

First define from the problem statement the values of **A** and **x** as well as the new value-added inputs defined as the variable `vnew`.

```
      A←0.168 0.155 0.213 0.212 0.194 0.193 0.168 0.115
      A←A,0.105 0.025 0.126 0.124 0.178 0.101 0.219 0.186
      A←4 4⍴A
      x←12000 15000 12000 16000
      vnew←4558 5665 2050 5079
A
    0.1680    0.1550    0.2130    0.2120
    0.1940    0.1930    0.1680    0.1150
    0.1050    0.0250    0.1260    0.1240
    0.1780    0.1010    0.2190    0.1860
x
     12000     15000     12000     16000
vnew
      4558      5665      2050      5079
```

From these data we can compute the corresponding matrix of interindustry transactions **Z**, the current vectors of value-added inputs **v** and of final demands **f** along with the gross domestic product `GDP`.

```
      Z←A+.×DIAG x
      v←x-+⌿Z
      f←x-+/Z
      GDP←+/v

Z
      2016      2325      2556      3392
      2328      2895      2016      1840
      1260       375      1512      1984
      2136      1515      2628      2976
f
      1711      5921      6869      6745
x
     12000     15000     12000     16000
v
    4260.0    7890.0    3288.0    5808.0
GDP
21246
```

Finally, we can compute the supply model matrices **B** and **G**, as well as the new vector of total outputs `xnew` corresponding to `vnew` and the new value of gross domestic product `GDPN`.

```
     G←INV B←(DIAG÷x)+.×Z
     xnew←G+.×vnew
     GDPN←+/vnew
B
    0.1680    0.1938    0.2130    0.2827
    0.1552    0.1930    0.1344    0.1227
    0.1050    0.0313    0.1260    0.1653
    0.1335    0.0947    0.1643    0.1860
G
    1.4682    0.4553    0.5578    0.6918
    0.3756    1.3933    0.3844    0.4185
    0.2533    0.1552    1.3003    0.3755
    0.3356    0.2681    0.3986    1.4664
xnew
     13928     12518      6607     11313

GDPN
17352
```

Problem 7.2: Comparing Results with Mean Absolute Percentage Differences

This problem illustrates the use of mean absolute percentage difference (MAPD) as a measure for comparing output coefficients in supply-side input-output models.

Problem 7.2 Overview

Consider a case where $\mathbf{Z} = \begin{bmatrix} 13 & 75 & 45 \\ 53 & 21 & 48 \\ 67 & 68 & 93 \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} 130 \\ 150 \\ 220 \end{bmatrix}$ for a base year. If final demands for the next year are projected to be $\mathbf{f}^{new} = \begin{bmatrix} 200 \\ 300 \\ 500 \end{bmatrix}$ and the change in interindustry transactions is expected to be $\Delta\mathbf{Z} = \begin{bmatrix} 0 & 5 & 0 \\ 10 & 0 & 0 \\ 0 & 0 & 15 \end{bmatrix}$ the MAPD between the direct output coefficients for the base year and next year is found by first computing the output coefficients for the two years. First, $\mathbf{B} = \hat{\mathbf{x}}^{-1}\mathbf{Z} = \begin{bmatrix} .049 & .285 & .171 \\ .195 & .077 & .176 \\ .150 & .152 & .208 \end{bmatrix}$, and, since $\mathbf{Z}^{new} = \mathbf{Z} + \Delta\mathbf{Z} = \begin{bmatrix} 13 & 80 & 45 \\ 63 & 21 & 48 \\ 67 & 68 & 108 \end{bmatrix}$ and $\mathbf{x}^{new} = \mathbf{f}^{new} + \mathbf{Z}^{new}\mathbf{i} = \begin{bmatrix} 338 \\ 432 \\ 743 \end{bmatrix}$, $\mathbf{B}^{new}$ is found as $\mathbf{B}^{new} = (\hat{\mathbf{x}}^{new})^{-1}\mathbf{Z}^{new} = \begin{bmatrix} .038 & .123 & .133 \\ .146 & .049 & .111 \\ .090 & .092 & .145 \end{bmatrix}$.

The MAPD between $\mathbf{B}$ and $\mathbf{B}^{new}$ is found by $MAPD = (1/n^2)\sum_{i=1}^{n}\sum_{j=1}^{n}[\left|b_{ij} - b_{ij}^{new}\right| / b_{ij}] \times 100 = 29.1$. For the total output coefficients or output inverses, $\mathbf{G} = (\mathbf{I} - \mathbf{B})^{-1} = \begin{bmatrix} 1.195 & .427 & .353 \\ .307 & 1.235 & .341 \\ .284 & .317 & 1.394 \end{bmatrix}$ and $\mathbf{G}^{new} = (\mathbf{I} - \mathbf{B}^{new})^{-1} = \begin{bmatrix} 1.104 & .295 & .210 \\ .185 & 1.114 & .174 \\ .136 & .150 & 1.211 \end{bmatrix}$,

the MAPD between $\mathbf{G}$ and $\mathbf{G}^{new}$ is 32.806

Computational Notes

First define the matrix of transactions **Z** and the vector of final demands **f** as well as the new vector of final demands `f2` and the matrix of changes in transactions `ΔZ`.

```
      Z
13 75 45
53 21 48
67 68 93
      f
130 150 220
      f2
200 300 500
      ΔZ
 0 5  0
10 0  0
 0 0 15
```

We now compute the vector of total outputs **x** so we can compute the matrix of technical coefficients **A** and the Leontief inverse **L** as well as the direct and total supply coefficients **B** and **G**.

```
      x←f++/Z
      L←INV A←Z AMAT x
      G←LINV B←(DIAG ÷x)+.×Z
      x
        263        272        448
      A,L
      0.049      0.276      0.100      1.195      0.413      0.207
      0.202      0.077      0.107      0.317      1.235      0.207
      0.255      0.250      0.208      0.484      0.522      1.394


         B,G
      0.049      0.285      0.171      1.195      0.427      0.353
      0.195      0.077      0.176      0.307      1.235      0.341
      0.150      0.152      0.208      0.284      0.317      1.394
```

Next, we compute the new interindustry transactions `Z2` and new vector of total outputs `x2` corresponding to the new vector of final demands `f2`. From these we can compute the new direct and total supply coefficients B2 and `G2`, respectively.

```
      Z2←Z+ΔZ
      x2←f2++/Z2
      G2←LINV B2←(DIAG÷x2)+.×Z2
```

```
Z2
    13        80        45
    63        21        48
    67        68       108
x2
338.0     432.0     743.0
B2,G2
0.038     0.237     0.133     1.104     0.295     0.210
0.146     0.049     0.111     0.185     1.114     0.174
0.090     0.092     0.145     0.136     0.150     1.211
```

Finally, we compute the mean absolute percentage errors between **B** and `B2` as `MAPB` and between **G** and `G2` as `MAPG`

```
     MAPB←(÷9)×+/+/100×(|B-B2)÷B
     MAPDG←(÷9)×+/+/100×(|G-G2)÷G

     MAPB
29.999834
          MAPDG
32.806272
```

Problem 7.3: Relative Price Changes Using the Ghosh Price Model

This problem explores the calculation of relative price changes using the Ghosh price model.

Problem 7.3 Overview

For an input-output transactions matrix of $\mathbf{Z} = \begin{bmatrix} 384 & 520 & 831 \\ 35 & 54 & 530 \\ 672 & 8 & 380 \end{bmatrix}$ and total outputs of $\mathbf{x} = \begin{bmatrix} 2,500 \\ 1,200 \\ 3,000 \end{bmatrix}$ given for a base year, if additional growth in value added for the next year is projected to result in $\mathbf{v}^{new} = \begin{bmatrix} 2,000 \\ 1,000 \\ 1,500 \end{bmatrix}$, the corresponding price changes of output for the three industries for the new year relative to the base year are found by first calculating $\mathbf{B} = (\hat{\mathbf{x}})^{-1}\mathbf{Z} = \begin{bmatrix} .154 & .208 & .332 \\ .029 & .045 & .442 \\ .224 & .003 & .127 \end{bmatrix}$ and $\mathbf{G} = (\mathbf{I} - \mathbf{B})^{-1} = \begin{bmatrix} 1.37 & .3 & .673 \\ .205 & 1.093 & .631 \\ .352 & .080 & 1.320 \end{bmatrix}$. Using the Ghosh price model, the new total outputs are found as $\mathbf{x}^{new} = \mathbf{G}'\mathbf{v}^{new} = \begin{bmatrix} 3,472.7 \\ 1,814.5 \\ 3,956.9 \end{bmatrix}$ and the relative price changes between the two years are $\boldsymbol{\pi} = \hat{\mathbf{x}}^{-1}\mathbf{x}^{new} = \begin{bmatrix} 1.389 & 1.512 & 1.319 \end{bmatrix}$.

Computational Notes

First define the matrix of transactions **Z** and the vector of total outputs **x**, along with the vector of new value-added inputs `vnew`.

```
      Z←3 3⍴384 520 831 35 54 530 672 8 380
      x←2500 1200 3000
      vnew←2000 1000 1500

      Z
384 520 831
 35  54 530
672   8 380
      x
2500 1200 3000
      vnew
    2000.0    1000.0    1500.0
```

Now we can compute the matrices of direct and total supply coefficients **B** and **G** respectively and for convenience save the transpose of **G** as the variable `GT`.

```
      GT←⍉G←INV B←(DIAG ÷x)+.×Z

      B
      0.154     0.208     0.332
      0.029     0.045     0.442
      0.224     0.003     0.127
      G
      1.370     0.300     0.673
      0.205     1.093     0.631
      0.352     0.080     1.320
```

Finally we can compute the new vector of total outputs `xnew` corresponding to `vnew` and the relative prices changes between the two periods `p`.

```
      xnew←GT+.×vnew
      p←(DIAG ÷x)+.×GT+.×vnew

      xnew
    3472.7    1814.5    3956.9
      p
      1.389     1.512     1.319
```

Problem 7.4: The Leontief Price Model and The Ghosh Price Model

This problem illustrates that the Ghosh and Leontief price models produce the same result.

Problem 7.4 Overview

Using the basic data in Problem 7.3, first recall that $\mathbf{Z} = \begin{bmatrix} 384 & 520 & 831 \\ 35 & 54 & 530 \\ 672 & 8 & 380 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 2{,}500 \\ 1{,}200 \\ 3{,}000 \end{bmatrix}$ produces $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .154 & .433 & .277 \\ .014 & .045 & .177 \\ .269 & .007 & .127 \end{bmatrix}$ and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.37 & .626 & .561 \\ .098 & 1.093 & .252 \\ .422 & .201 & 1.32 \end{bmatrix}$. Also, with the new year's value added defined in Problem 7.3 as $\mathbf{v}^{new} = \begin{bmatrix} 2{,}000 \\ 1{,}000 \\ 1{,}500 \end{bmatrix}$, we calculate the vector of the new value added as a fraction of the base year total outputs, $\mathbf{v}_c^{new} = \hat{\mathbf{x}}^{-1}\mathbf{v}^{new} = \begin{bmatrix} .8 & .833 & .5 \end{bmatrix}'$.

The vector of relative price changes using the Leontief price model as $\tilde{\mathbf{p}} = (\mathbf{I} - \mathbf{A}')^{-1}\mathbf{v}_c^{new} = \mathbf{L}'\mathbf{v}_c^{new} = \begin{bmatrix} 1.389 & 1.512 & 1.319 \end{bmatrix}'$ which are identical to the Ghosh model price changes, i.e., $\tilde{\mathbf{p}} = \boldsymbol{\pi}$ from Problem 7.3.

Computational Notes

Recalling **Z**, **x**, and `vnew` from Problem 7.3, we can compute the corresponding **A** and **L** as well as the value-added coefficients `vcnew` from `vnew` and **x**.

```
L←LINV A←Z AMAT x
vcnew←vnew÷x
L
     1.370     0.626     0.561
     0.098     1.093     0.252
     0.422     0.201     1.320
vcnew
     0.800     0.833     0.500
```

Using the transpose of **L** and `vcnew` we can compute the relative prices `pl` with the Leontief price model as

```
pl←(⍉L)+.×vcnew

pl
1.3890738 1.5121032 1.3189828
```

Note that these values are identical to the relative prices generated in Problem 7.3.

Problem 7.5: Forward and Backward Linkages in Input-Output Models

This problem explores the basic concepts of forward and backward linkages in input-output models.

Problem 7.5 Overview

Consider the case of a matrix of transactions, $\mathbf{Z} = \begin{bmatrix} 418 & 687 & 589 & 931 \\ 847 & 527 & 92 & 654 \\ 416 & 702 & 911 & 763 \\ 263 & 48 & 737 & 329 \end{bmatrix}$, and vector of total final demands, $\mathbf{f} = \begin{bmatrix} 2,000 \\ 3,000 \\ 2,500 \\ 1,500 \end{bmatrix}$. With $\mathbf{x} = \mathbf{f} + \mathbf{Zi} = \begin{bmatrix} 4,625 \\ 5,120 \\ 5,292 \\ 2,877 \end{bmatrix}$ we compute the matrix of direct requirements, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .090 & .134 & .111 & .324 \\ .183 & .103 & .017 & .227 \\ .090 & .137 & .172 & .265 \\ .057 & .009 & .139 & .114 \end{bmatrix}$, and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.207 & .227 & .264 & .579 \\ .280 & 1.182 & .138 & .447 \\ .214 & .241 & 1.332 & .539 \\ .114 & .065 & .228 & 1.256 \end{bmatrix}$, for the demand-driven model. We next compute $\mathbf{B} = \hat{\mathbf{x}}^{-1}\mathbf{Z} = \begin{bmatrix} .090 & .149 & .127 & .201 \\ .165 & .103 & .018 & .128 \\ .079 & .133 & .172 & .144 \\ .091 & .017 & .256 & .114 \end{bmatrix}$, the matrix of direct requirements, and $\mathbf{G} = (\mathbf{I} - \mathbf{B})^{-1} = \begin{bmatrix} 1.207 & .251 & .303 & .360 \\ .253 & 1.182 & .142 & .251 \\ .187 & .233 & 1.332 & .293 \\ .183 & .116 & .419 & 1.256 \end{bmatrix}$, the matrix of total requirements for supply driven input-output models.

The vectors of direct and total backward linkages are found as $\mathbf{i}'\mathbf{A} = \begin{bmatrix} .420 & .384 & .440 & .930 \end{bmatrix}$ and $\mathbf{i}'\mathbf{L} = \begin{bmatrix} 1.815 & 1.716 & 1.962 & 2.820 \end{bmatrix}$, respectively, from the demand-driven model. The vectors of direct and total forward linkages are found as $\mathbf{Bi} = \begin{bmatrix} 0.568 & 0.414 & 0.528 & 0.479 \end{bmatrix}'$ and $\mathbf{Gi} = \begin{bmatrix} 2.121 & 1.828 & 2.046 & 1.974 \end{bmatrix}'$, respectively, from the supply-driven model.

Computational Notes

We define the matrix of interindustry transactions **Z** and vector of total final demands **f**, from which we can compute the vector of total outputs **x**.

```
      Z←418 687 589 931 847 527 92 654
      Z←4 4⍴Z,416 702 911 763 263 48 737 329
      f←2000 3000 2500 1500
      x←f++/Z

      Z
418 687 589 931
847 527  92 654
416 702 911 763
263  48 737 329
      f
2000 3000 2500 1500
      x
4625 5120 5292 2877
```

Now we can compute the matrix of technical coefficients **A** and the Leontief inverse **L** along with the corresponding vectors of direct backward linkages `BLD` and total backward linkages `BLT`.

```
      A
      0.090     0.134     0.111     0.324
      0.183     0.103     0.017     0.227
      0.090     0.137     0.172     0.265
      0.057     0.009     0.139     0.114
      L
      1.207     0.227     0.264     0.579
      0.280     1.182     0.138     0.447
      0.214     0.241     1.332     0.539
      0.114     0.065     0.228     1.256
      BLD
      0.420     0.384     0.440     0.930
      BLT
      1.815     1.716     1.962     2.820
```

Finally, we can also compute the direct supply coefficients **B** and the supply invers **G**, along with the corresponding vectors of direct forward linkages `FLD` and total forward linkages `FLT`.

```
      G←LINV B←(DIAG ÷x)+.×Z
      FLD←+/B
      FLT←+/G

      B
      0.090     0.149     0.127     0.201
      0.165     0.103     0.018     0.128
      0.079     0.133     0.172     0.144
      0.091     0.017     0.256     0.114
```

```
G
1.207     0.251     0.303     0.360
0.253     1.182     0.142     0.251
0.187     0.233     1.332     0.293
0.183     0.116     0.419     1.256
FLD
0.568     0.414     0.528     0.479
FLT
2.121     1.828     2.046     1.974
```

Problem 7.6: Forward and Backward Linkages in IRIO Models

This problem explores spatial forward and backward linkages in an interregional input-output (IRIO) model.

Problem 7.6 Overview

Using the three region IRIO table for Japan given in problem in Table A4.1.1 of Appendix S4.1, first define $B(d)^{rr} = (1/n)\mathbf{i}'\mathbf{A}^{rr}\mathbf{i}$ for regions, $r = 1,\ 2,\text{ and } 3$ designating the average direct spatial linkage of a region to itself as the average of the intraregional technical coefficients. Also designate $B(d)^{sr} = (1/n)\mathbf{i}'\mathbf{A}^{sr}\mathbf{i}$ for regions $r = 1,\ 2,\text{ and } 3$ and $s = 1,\ 2,\text{ and } 3$ but for $r \neq s$ to designate the average direct interregional spatial linkage of a region to other regions as the average of the interregional technical coefficients relating the regions. Similarly, $B(t)^{rr} = (1/n)\mathbf{i}'\mathbf{L}^{rr}\mathbf{i}$ and $B(t)^{sr} = (1/n)\mathbf{i}'\mathbf{L}^{sr}\mathbf{i}$ designate the total spatial linkage of a region to itself and the total spatial linkage between regions, respectively.

For the Japanese IRIO model provided, $n = 5$ industry sectors, and $r = c,\ n,\text{ and } s$ for the Japanese central, north, and south regions, the specific direct backward linkage measures are $B(d)^{c} = B(d)^{cc} + B(d)^{cn} + B(d)^{cs}$, $B(d)^{n} = B(d)^{nn} + B(d)^{nc} + B(d)^{ns}$ and $B(d)^{s} = B(d)^{ss} + B(d)^{sn} + B(d)^{sc}$. These are the direct backward linkages for the central, north, and south regions, respectively. Analogous notation applies for the total backward linkages and the direct and total forward linkages which use **B** and **G** instead of **A** and **L**.

The results of these calculations are the following:

	$r =$ Central	$r =$ North	$r =$ South
$b(d)^{r}$	.865	.741	.939
$b(t)^{r}$	3.177	2.731	3.434
$f(d)^{r}$	.579	.453	.597
$f(t)^{r}$	2.615	2.483	2.595

From the table of results, we can observe that the North region is both the least backward-linked and forward-linked among the 3 regions.

Computational Notes

We retrieve 15 sector (3 region, 5 sector) matrix of technical coefficients matrix **A** and the corresponding vector of total outputs **x**, from which we can calculate the interindustry transactions matrix **Z**, direct supply coefficients **B**, total supply coefficients **G**, and the Leontief inverse **L**. Each of these matrices can be partitioned into 9 blocks, each of shape 5x5 with the intraregional matrices along the block diagonal and interregional matrices describing the activity from supply regions along the rows to demand regions along the columns. Which we denote for **A** by $\left[\begin{array}{c:c:c} \mathbf{A11} & \mathbf{A12} & \mathbf{A13} \\ \hdashline \mathbf{A21} & \mathbf{A22} & \mathbf{A23} \\ \hdashline \mathbf{A31} & \mathbf{A32} & \mathbf{A33} \end{array}\right]$ where each partition is defined by a corresponding variable. We replace the **A** with **B**, **G**, and **L** for the other matrices. We compute the forward and backward direct and total linkages `bd`, `bt`, `fd`, and `ft` with the following APL function.

```
[  0]  A LINKAGES x;n;m;i;j
[  1]  Z←A+.×DIAG x ⋄ B←(DIAG÷x)+.×Z ⋄ G←LINV⍉B ⋄ L←LINV A ⋄ n←5 ⋄ m←3
[  2] ⍝--linkage blocks (backward)
[  3]  A11←A[ιn;ιn] ⋄ A12←A[ιn;n+ιn] ⋄ A13←A[ιn;(2×n)+ιn]
[  4]  A21←A[n+ιn;ιn] ⋄ A22←A[n+ιn;n+ιn] ⋄ A23←A[n+ιn;(2×n)+ιn]
[  5]  A31←A[(2×n)+ιn;ιn] ⋄ A32←A[(2×n)+ιn;n+ιn] ⋄ A33←A[(2×n)+ιn;(2×n)+ιn]
[  6]  L11←L[ιn;ιn] ⋄ L12←L[ιn;n+ιn] ⋄ L13←L[ιn;(2×n)+ιn]
[  7]  L21←L[n+ιn;ιn] ⋄ L22←L[n+ιn;n+ιn] ⋄ L23←L[n+ιn;(2×n)+ιn]
[  8]  L31←L[(2×n)+ιn;ιn] ⋄ L32←L[(2×n)+ιn;n+ιn] ⋄ L33←L[(2×n)+ιn;(2×n)+ιn]
[  9] ⍝--linkage blocks (forward)
[ 10]  B11←B[ιn;ιn] ⋄ B12←B[ιn;n+ιn] ⋄ B13←B[ιn;(2×n)+ιn]
[ 11]  B21←B[n+ιn;ιn] ⋄ B22←B[n+ιn;n+ιn] ⋄ B23←B[n+ιn;(2×n)+ιn]
[ 12]  B31←B[(2×n)+ιn;ιn] ⋄ B32←B[(2×n)+ιn;n+ιn] ⋄ B33←B[(2×n)+ιn;(2×n)+ιn]
[ 13]  G11←G[ιn;ιn] ⋄ G12←G[ιn;n+ιn] ⋄ G13←G[ιn;(2×n)+ιn]
[ 14]  G21←G[n+ιn;ιn] ⋄ G22←G[n+ιn;n+ιn] ⋄ G23←G[n+ιn;(2×n)+ιn]
[ 15]  G31←G[(2×n)+ιn;ιn] ⋄ G32←G[(2×n)+ιn;n+ιn] ⋄ G33←G[(2×n)+ιn;(2×n)+ιn]
[ 16] ⍝--aggregate measures
[ 17] ⍝--spatial-sectoral linkage
[ 18]  BDIJ←TDIJ←FDIJ←FTIJ←(2ρm)ρ0 ⋄ i←j←1
[ 19] l1:⍎'BDIJ[i;j]←(÷m)×+/+/A',(1 0⍕i,j) ⋄ ⍎'TDIJ[i;j]←(÷m)×+/+/L',(1 0⍕i,j)
[ 20]  ⍎'FDIJ[i;j]←(÷m)×+/+/B',(1 0⍕i,j) ⋄ ⍎'FTIJ[i;j]←(÷m)×+/+/G',(1 0⍕i,j)
[ 21]  →(m≥j←j+1)/l1
[ 22]  →(m≥i←i+(j←1))/l1
[ 23] ⍝--spatial linkage
[ 24]  bd←+/BDIJ ⋄ bt←+/TDIJ ⋄ fd←+⌿FDIJ ⋄ ft←+⌿FTIJ
```

```
A
  0.053  0.000  0.009  0.011  0.009  0.001  0.000  0.007  0.000  0.001  0.001  0.000  0.001  0.000  0.000
  0.000  0.001  0.001  0.001  0.002  0.000  0.000  0.001  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.428  0.723  0.250  0.240  0.180  0.012  0.004  0.052  0.001  0.013  0.017  0.005  0.044  0.000  0.014
  0.000  0.001  0.010  0.090  0.012  0.000  0.000  0.002  0.015  0.001  0.000  0.000  0.001  0.007  0.001
  0.012  0.029  0.042  0.117  0.125  0.000  0.001  0.015  0.001  0.010  0.000  0.000  0.007  0.001  0.014
  0.004  0.000  0.000  0.000  0.000  0.089  0.001  0.017  0.039  0.021  0.002  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.002  0.005  0.002  0.007  0.011  0.000  0.000  0.000  0.000  0.000
  0.068  0.041  0.020  0.000  0.002  0.362  0.521  0.160  0.233  0.129  0.034  0.028  0.012  0.000  0.001
  0.000  0.002  0.000  0.014  0.000  0.000  0.008  0.010  0.025  0.011  0.000  0.000  0.000  0.023  0.000
  0.003  0.034  0.001  0.000  0.001  0.010  0.033  0.027  0.095  0.103  0.002  0.008  0.000  0.000  0.001
  0.002  0.000  0.002  0.000  0.000  0.002  0.000  0.006  0.000  0.000  0.072  0.000  0.011  0.016  0.010
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.001  0.000  0.000  0.001  0.004  0.001  0.002  0.004
  0.036  0.021  0.082  0.000  0.013  0.012  0.012  0.056  0.000  0.007  0.473  0.719  0.303  0.264  0.196
  0.000  0.000  0.001  0.024  0.000  0.000  0.000  0.001  0.022  0.000  0.000  0.003  0.009  0.068  0.012
  0.001  0.005  0.006  0.000  0.003  0.000  0.001  0.009  0.000  0.003  0.012  0.050  0.037  0.112  0.110
x
 1307.0  123.016400.0 1342.0 8591.0 1308.0  201.0 4167.0  394.0 2759.0 2131.0  267.022053.0 1546.0 9968.0
Z
   69.3    0.0  147.6   14.8   77.3    1.3    0.0   29.2    0.0    2.8    2.1    0.0   22.1    0.0    0.0
    0.0    0.1   16.4    1.3   17.2    0.0    0.0    4.2    0.0    0.0    0.0    0.0    0.0    0.0    0.0
  559.4   88.9 4100.0  322.1 1546.4   15.7    0.8  216.7    0.4   35.9   36.2    1.3  970.3    0.0  139.6
    0.0    0.1  164.0  120.8  103.1    0.0    0.0    8.3    5.9    2.8    0.0    0.0   22.1   10.8   10.0
   15.7    3.6  688.8  157.0 1073.9    0.0    0.2   62.5    0.4   27.6    0.0    0.0  154.4    1.5  139.6
```

```
     5.2     0.0     0.0     0.0     0.0   116.4     0.2    70.8    15.4    57.9     4.3     0.0     0.0     0.0     0.0
     0.0     0.0     0.0     0.0     0.0     2.6     1.0     8.3     2.8    30.3     0.0     0.0     0.0     0.0     0.0
    88.9     5.0   328.0     0.0    17.2   473.5   104.7   666.7    91.8   355.9    72.5     7.5   264.6     0.0    10.0
     0.0     0.2     0.0    18.8     0.0     0.0     1.6    41.7     9.9    30.3     0.0     0.0     0.0    35.6     0.0
     3.9     4.2    16.4     0.0     8.6    13.1     6.6   112.5    37.4   284.2     4.3     2.1     0.0     0.0    10.0
     2.6     0.0    32.8     0.0     0.0     2.6     0.0    25.0     0.0     0.0   153.4     0.0   242.6    24.7    99.7
     0.0     0.0     0.0     0.0     0.0     0.0     0.0     4.2     0.0     0.0     2.1     1.1    22.1     3.1    39.9
    47.1     2.6  1344.8     0.0   111.7    15.7     2.4   233.4     0.0    19.3  1008.0   192.0  6682.1   408.1  1953.7
     0.0     0.0    16.4    32.2     0.0     0.0     0.0     4.2     8.7     0.0     0.0     0.8   198.5   105.1   119.6
     1.3     0.6    98.4     0.0    25.8     0.0     0.2    37.5     0.0     8.3    25.6    13.4   816.0   173.2  1096.5
B
   0.053   0.000   0.113   0.011   0.059   0.001   0.000   0.022   0.000   0.002   0.002   0.000   0.017   0.000   0.000
   0.000   0.001   0.133   0.011   0.140   0.000   0.000   0.034   0.000   0.000   0.000   0.000   0.000   0.000   0.000
   0.034   0.005   0.250   0.020   0.094   0.001   0.000   0.013   0.000   0.002   0.002   0.000   0.059   0.000   0.009
   0.000   0.000   0.122   0.090   0.077   0.000   0.000   0.006   0.004   0.002   0.000   0.000   0.016   0.008   0.007
   0.002   0.000   0.080   0.018   0.125   0.000   0.000   0.007   0.000   0.003   0.000   0.000   0.018   0.000   0.016
   0.004   0.000   0.000   0.000   0.000   0.089   0.000   0.054   0.012   0.044   0.003   0.000   0.000   0.000   0.000
   0.000   0.000   0.000   0.000   0.000   0.013   0.005   0.041   0.014   0.151   0.000   0.000   0.000   0.000   0.000
   0.021   0.001   0.079   0.000   0.004   0.114   0.025   0.160   0.022   0.085   0.017   0.002   0.064   0.000   0.002
   0.000   0.001   0.000   0.048   0.000   0.000   0.004   0.106   0.025   0.077   0.000   0.000   0.000   0.090   0.000
   0.001   0.002   0.006   0.000   0.003   0.005   0.002   0.041   0.014   0.103   0.002   0.001   0.000   0.000   0.004
   0.001   0.000   0.015   0.000   0.000   0.001   0.000   0.012   0.000   0.000   0.072   0.000   0.114   0.012   0.047
   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.016   0.000   0.000   0.008   0.004   0.083   0.012   0.149
   0.002   0.000   0.061   0.000   0.005   0.001   0.000   0.011   0.000   0.001   0.046   0.009   0.303   0.019   0.089
   0.000   0.000   0.011   0.021   0.000   0.000   0.000   0.003   0.006   0.000   0.000   0.001   0.128   0.068   0.077
   0.000   0.000   0.010   0.000   0.003   0.000   0.000   0.004   0.000   0.001   0.003   0.001   0.082   0.017   0.110
G
   1.064   0.009   0.051   0.008   0.008   0.007   0.002   0.034   0.005   0.004   0.004   0.002   0.009   0.002   0.002
   0.001   1.002   0.008   0.001   0.001   0.000   0.000   0.003   0.001   0.002   0.000   0.000   0.001   0.000   0.000
   0.181   0.210   1.379   0.201   0.136   0.011   0.010   0.150   0.032   0.018   0.043   0.019   0.132   0.042   0.030
   0.019   0.020   0.034   1.106   0.026   0.001   0.001   0.006   0.057   0.002   0.002   0.001   0.004   0.026   0.001
   0.094   0.186   0.157   0.120   1.161   0.002   0.002   0.026   0.010   0.007   0.007   0.004   0.025   0.009   0.008
   0.006   0.006   0.005   0.002   0.002   1.108   0.023   0.153   0.018   0.013   0.004   0.003   0.004   0.001   0.001
   0.001   0.001   0.001   0.000   0.000   0.002   1.007   0.031   0.008   0.004   0.001   0.001   0.001   0.000   0.000
   0.034   0.047   0.028   0.015   0.014   0.077   0.063   1.218   0.139   0.059   0.019   0.023   0.024   0.009   0.008
   0.001   0.001   0.001   0.006   0.001   0.016   0.018   0.032   1.031   0.017   0.001   0.001   0.001   0.007   0.000
   0.007   0.006   0.007   0.006   0.006   0.064   0.178   0.132   0.104   1.123   0.003   0.003   0.005   0.002   0.002
   0.005   0.003   0.011   0.003   0.003   0.006   0.002   0.031   0.005   0.004   1.088   0.017   0.075   0.011   0.011
   0.001   0.000   0.001   0.001   0.000   0.000   0.000   0.004   0.001   0.001   0.002   1.006   0.013   0.003   0.003
   0.049   0.029   0.132   0.053   0.048   0.010   0.008   0.135   0.038   0.009   0.196   0.151   1.488   0.220   0.144
   0.002   0.001   0.004   0.012   0.002   0.002   0.002   0.007   0.101   0.002   0.019   0.019   0.034   1.081   0.024
   0.009   0.009   0.031   0.020   0.028   0.002   0.002   0.022   0.015   0.006   0.079   0.187   0.159   0.118   1.142
L
   1.064   0.012   0.014   0.019   0.014   0.006   0.006   0.011   0.003   0.003   0.003   0.003   0.003   0.001   0.001
   0.001   1.002   0.002   0.002   0.003   0.001   0.001   0.001   0.000   0.000   0.000   0.000   0.000   0.000   0.000
   0.639   1.016   1.379   0.413   0.300   0.064   0.066   0.109   0.042   0.044   0.081   0.084   0.098   0.040   0.050
   0.008   0.014   0.016   1.106   0.019   0.002   0.003   0.005   0.019   0.003   0.002   0.003   0.003   0.010   0.003
   0.050   0.088   0.071   0.169   1.161   0.013   0.018   0.029   0.014   0.019   0.012   0.016   0.019   0.011   0.024
   0.007   0.003   0.001   0.001   0.000   1.108   0.015   0.024   0.053   0.030   0.004   0.001   0.001   0.002   0.000
   0.000   0.001   0.000   0.000   0.000   0.004   1.007   0.003   0.009   0.013   0.000   0.000   0.000   0.000   0.000
   0.109   0.086   0.038   0.019   0.013   0.489   0.648   1.218   0.336   0.200   0.060   0.055   0.025   0.018   0.009
   0.001   0.004   0.001   0.017   0.000   0.005   0.016   0.013   1.031   0.015   0.001   0.001   0.001   0.026   0.001
   0.008   0.042   0.003   0.003   0.002   0.028   0.059   0.039   0.120   1.123   0.005   0.011   0.001   0.004   0.002
   0.006   0.005   0.006   0.003   0.002   0.007   0.006   0.010   0.004   0.002   1.088   0.015   0.019   0.026   0.017
   0.000   0.000   0.000   0.000   0.000   0.001   0.001   0.001   0.000   0.000   0.002   1.006   0.002   0.003   0.005
   0.149   0.170   0.178   0.071   0.063   0.075   0.088   0.128   0.050   0.038   0.772   1.098   1.488   0.481   0.352
   0.003   0.004   0.004   0.030   0.002   0.002   0.002   0.003   0.026   0.001   0.008   0.015   0.015   1.081   0.018
   0.013   0.021   0.018   0.010   0.009   0.009   0.012   0.019   0.010   0.008   0.049   0.106   0.065   0.157   1.142

bd    0.865    0.741    0.939
bt    3.177    2.731    3.434
fd    0.579    0.453    0.597
ft    2.615    2.483    2.595
```

Problem 7.7: Hypothetical Extraction

This problem explores the concept of hypothetically extracting an industry sector from the economy and calculating the decrease in total output of the economy resulting from the hypothetical extraction.

Problem 7.7 Overview

The following is a highly aggregated version of the 2005 U.S. input-output table:

US Technical Coefficients 2005	1	2	3	4	5	6	7	Tot. Output
1 Agriculture	0.2258	0.0000	0.0015	0.0384	0.0001	0.0017	0.0007	312,754
2 Mining	0.0027	0.1432	0.0075	0.0675	0.0367	0.0004	0.0070	396,563
3 Construction	0.0051	0.0002	0.0010	0.0018	0.0037	0.0071	0.0215	1,302,388
4 Manufacturing	0.1955	0.0877	0.2591	0.3222	0.0547	0.0566	0.1010	4,485,529
5 Trade, Transport & Utilities	0.0819	0.0422	0.1011	0.0994	0.0704	0.0334	0.0487	3,355,944
6 Services	0.0843	0.1276	0.1225	0.1172	0.1760	0.2783	0.2026	10,477,640
7 Other	0.0099	0.0095	0.0093	0.0219	0.0215	0.0188	0.0240	2,526,325

To hypothetically extract the agriculture sector (sector 1), we set the first row and first column of the matrix **A** to zero, the result of which we define as $\mathbf{A}^{(1)}$ and set the first element of the vector **f** to zero which we define as $\mathbf{f}^{(1)}$. Then we compute $\mathbf{L}^{(1)} = (\mathbf{I} - \mathbf{A}^{(1)})^{-1}$ and subsequently $t_1 = \mathbf{i}'\mathbf{x} - \mathbf{i}'\mathbf{L}^{(1)}\mathbf{f}^{(1)} = 54,744,946$, which would be the reduction in total output of the economy if the agriculture sector were extracted.

If we now define $p_i = 100 \times (\mathbf{i}'\mathbf{x} - \mathbf{i}'\mathbf{L}^{(i)}\mathbf{f}^{(i)}) / \mathbf{i}'\mathbf{x}$ as the percentage reduction in total output by extracting industry *i*, we can compute the vector of all the seven p_i's for this economy as $\mathbf{p} = \begin{bmatrix} 2.4 & 2.6 & 11.5 & 29.8 & 22.0 & 54.8 & 18.8 \end{bmatrix}'$, which indicates that the services sector (sector 6) would yield the highest reduction in output from a hypothetical extraction with a 54.8 percent reduction in total output.

Computational Notes

We retrieve **A** and **x** and compute **f**. Then we set the first row and first column of **A** to zero in a variable labeled `Aj` and the first element of **f** to zero in a variable labeled `fj`. We can then compute the corresponding `Aj` and `Lj` and use `Lj` to compute the vector of total outputs `xj` associated with `fj`. The total outputs of the original economy summed and divided by the sum of all total outputs for the economy with sector one extracted in percentage terms is t. These steps are included in the dyadic function `HEXTRACT1` which takes as the right argument an n×1 by n matrix, the first n rows of which comprise the technical coefficients matrix **A** and the last row is the vector of total outputs **x**. The left argument is the index of the vector to be hypothetically extracted.

```
[  0]  R←i HEXTRACT1 Ax;n;A;x;f;fj;Aj;Lj;xt;t;xj
[  1] ⍝Hypotherical Extraction
[  2] ⍝Ax is (n+1)×n A,[1]x
[  3]  n←¯1↑⍴Ax ⋄ A←(2⍴n)↑Ax ⋄ x←,Ax[n+1;]
[  4]  Aj←A ⋄ fj←f←x-A+.×x
[  5]  fj[1]←0 ⋄ Aj[1;]←0 ⋄ Aj[;1]←0
[  6]  xj←(Lj←LINV Aj)+.×fj
[  7]  R←t,100×(t←xt-+/xj)÷xt←+/x
```

We can repeat this process for each sector in the economy and save the results, or perhaps more efficiently modify `HEXTRACT1` to form `HEXTRACT2 that` take a vector defining a selection of sectors **S** to be hypothetically extracted (instead of just one sector j), as in

```
R←S HEXTRACT2 Ax;n;A;x;f;fj;Aj;Lj;xt;t;xj;S;r;k;m
⍝Hypothetical Extraction
⍝Ax is (n+1)×n A,[1]x; S is vector of sectors
 m←⍴S ⋄ n←¯1↑⍴Ax ⋄ A←(2⍴n)↑Ax ⋄ x←,Ax[n+1;]
 r←(2,m)⍴0 ⋄ k←1
L1:Aj←A ⋄ fj←f←x-A+.×x
 fj[k]←0 ⋄ Aj[k;]←0 ⋄ Aj[;k]←0
 xj←(Lj←LINV Aj)+.×fj
 r[;k]←t,100×(t←xt-+/xj)÷xt←+/x
 →(m≥k←k+1)/L1
 R←(3,m)⍴(⍳m),,r
```

We can use `HEXTRACT2` to hypothetically extract all the sectors. The second line of the results below (computed as the explicit result) show the total outputs associated with the hypothetical extraction of the sectors indicated by the column and the third line shows those results in percentage terms.

```
      (⍳7) HEXTRACT2 Ax
         1           2           3           4           5           6           7
547449.46   591737.58   2632898.3   6814684.2   5036387.3    12526318   4307032.5
 2.3950914   2.5888519    11.51893   29.814244   22.034195   54.802643   18.843267
```

Problem 7.8: Identifying "Inverse-Important" Coefficients

This problem explores the concept of "inverse important" coefficients in a Leontief model.

Problem 7.8 Overview

Consider an economy with $\mathbf{Z}=\begin{bmatrix} 8 & 64 & 89 \\ 28 & 44 & 77 \\ 48 & 24 & 28 \end{bmatrix}$ and $\mathbf{x}=\begin{bmatrix} 300 \\ 250 \\ 200 \end{bmatrix}$. Using the element a_{13} as an example, we define the criteria for a sector designated as "inverse important" as the following.

We define parameters $\alpha = 30$ and $\beta = 5$ as specifying that a 30 percent change in a_{13} generates a 5 percent change in one or more elements in the associated Leontief inverse. We can explore the sensitivity of the results to the values of α and β as a relative indication of inverse importance. First, compute the matrix of technical coefficients,

$\mathbf{A}=\mathbf{Z}\hat{\mathbf{x}}^{-1}=\begin{bmatrix} 0.0267 & 0.2560 & 0.4450 \\ 0.0933 & 0.1760 & 0.3850 \\ 0.1600 & 0.0960 & 0.1400 \end{bmatrix}$, and the matrix of total requirements,

$\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}=\begin{bmatrix} 1.2107 & 0.4738 & 0.8386 \\ 0.2557 & 1.3805 & 0.7503 \\ 0.2538 & 0.2423 & 1.4026 \end{bmatrix}$. If a_{13} is increased to 0.5785 $(\alpha = 30)$ we find

$\mathbf{L}^{*}_{(13)}=\begin{bmatrix} 1.2531 & 0.5144 & 1.0732 \\ 0.2647 & 1.3890 & 0.7999 \\ 0.2627 & 0.2507 & 1.4517 \end{bmatrix}$ and consequently the element by element normalized percentage difference between elements in $\mathbf{L}^*_{(13)}$ and **L** which is found by

$$\mathbf{P}_{(13)} = 100\{[\mathbf{L}^*_{(13)} - \mathbf{L}] \oslash \mathbf{L}\} = \begin{bmatrix} 3.5069 & 8.5531 & 27.9806 \\ 3.5069 & 0.6201 & 6.6051 \\ 3.5069 & 3.5069 & 3.5069 \end{bmatrix}$$

. In this case, a_{13} is identified as inverse important because a 30 percent change in its value causes a greater than 5 percent change in three inverse elements—l_{12}, l_{13}, and l_{23}. Notice that, as expected, the largest impact of a change in a_{13} is on the corresponding element in **L**, namely l_{13}.

If we change the parameters to $\alpha = 20$ and $\beta = 10$ then,

$$\mathbf{L}^*_{(13)} = \begin{bmatrix} 1.2387 & 0.5005 & 0.9932 \\ 0.2616 & 1.3861 & 0.7830 \\ 0.2597 & 0.2478 & 1.4350 \end{bmatrix} \text{ and } \mathbf{P}_{(13)} = \begin{bmatrix} 2.3109 & 5.6362 & 18.4382 \\ 2.3109 & 0.4086 & 4.3525 \\ 2.3109 & 2.3109 & 2.3109 \end{bmatrix}$$

, so a_{13} would still be classified as inverse important, since there is (now only) one element, l_{13}, that is changed by more than $\beta = 10$ percent. Finally, as another illustration, with $\alpha = 10$ and $\beta = 10$, we find

$$\mathbf{L}^*_{(13)} = \begin{bmatrix} 1.2245 & 0.4870 & 0.9150 \\ 0.2586 & 1.3832 & 0.7664 \\ 0.2567 & 0.2450 & 1.4186 \end{bmatrix} \text{ and } \mathbf{P}_{(13)} = \begin{bmatrix} 1.1423 & 2.7859 & 9.1138 \\ 1.1423 & 0.2020 & 2.1514 \\ 1.1423 & 1.1423 & 1.1423 \end{bmatrix}$$

. In this case, a_{13} would not be labeled inverse important since the largest percentage change in an element of the Leontief inverse is less than the threshold of $\beta = 10$ percent.

Computational Notes

Define **Z** and **x** and computer **A** and **L**. Then, sequentially test the three cases of inverse importance in the APL function `INVIMP`.

```
[  0]  INVIMP
[  1]  Z←3 3⍴8 64 89 28 44 77 48 24 28 ⋄ x←300 250 200
[  2]  L←INV A←Z AMAT x
[  3] ⍝--7.8a
[  4]  A1←A ⋄ A1[1;3]←1.3×A[1;3]
[  5]  L113←L1←LINV A1
[  6]  P113←100×(|L1-L)÷L
[  7]  P1135←P113≥5
[  8] ⍝--7.8b
[  9]  A2←A ⋄ A2[1;3]←1.2×A[1;3]
[ 10]  L213←L2←LINV A2
[ 11]  P213←100×(|L2-L)÷L
[ 12]  P21310←P213≥10
[ 13] ⍝--7.8c
[ 14]  A3←A ⋄ A3[1;3]←1.1×A[1;3]
[ 15]  L313←L3←LINV A3
[ 16]  P313←100×(|L3-L)÷L
[ 17]  P31310←P313≥10
```

```
Z
 8 64 89
28 44 77
48 24 28
x
300 250 200
A,L
    0.0267    0.2560    0.4450    1.2107    0.4738    0.8386
    0.0933    0.1760    0.3850    0.2557    1.3805    0.7503
    0.1600    0.0960    0.1400    0.2538    0.2423    1.4026
A1,L1
    0.0267    0.2560    0.5785    1.2531    0.5144    1.0732
    0.0933    0.1760    0.3850    0.2647    1.3890    0.7999
    0.1600    0.0960    0.1400    0.2627    0.2507    1.4517
P113
    3.5069    8.5531   27.9806
    3.5069    0.6201    6.6051
    3.5069    3.5069    3.5069
P13≥5
         0         1         1
         0         0         1
         0         0         0
A2,L2
    0.0267    0.2560    0.5340    1.2387    0.5005    0.9932
    0.0933    0.1760    0.3850    0.2616    1.3861    0.7830
    0.1600    0.0960    0.1400    0.2597    0.2478    1.4350
P213
    2.3109    5.6362   18.4382
    2.3109    0.4086    4.3525
    2.3109    2.3109    2.3109
P213≥10
         0         0         1
         0         0         0
         0         0         0
A3,L3
    0.0267    0.2560    0.4895    1.2245    0.4870    0.9150
    0.0933    0.1760    0.3850    0.2586    1.3832    0.7664
    0.1600    0.0960    0.1400    0.2567    0.2450    1.4186
P313
    1.1423    2.7859    9.1138
    1.1423    0.2020    2.1514
    1.1423    1.1423    1.1423
P313≥5
         0         0         0
         0         0         0
         0         0         0
```

Problem 7.9: Analysis of Supply Interruption with Supply Driven IO Models

This problem explores the use of a supply-driven model to determine the sensitivity of an economy to an interruption in availability of a scarce-factor input

Problem 7.9 Overview

Consider a labor strike in one of the sectors using the U.S. economy for 2005 (using the data presented in Problem 7.7). For this economy, specified are **A** and **x** from which we can compute the interindustry transaction matrix,

$$\mathbf{Z} = \mathbf{A}\hat{\mathbf{x}} = \begin{bmatrix} 70,629 & 10 & 1,973 & 172,428 & 435 & 18,296 & 1,739 \\ 832 & 56,798 & 9,707 & 302,783 & 123,117 & 4,273 & 17,745 \\ 1,597 & 74 & 1,329 & 7,886 & 12,449 & 74,678 & 54,282 \\ 61,158 & 34,779 & 3,37,412 & 1,445,451 & 183,602 & 593,372 & 255,282 \\ 25,620 & 16,748 & 131,675 & 445,685 & 236,309 & 350,316 & 123,084 \\ 26352 & 50611 & 159600 & 525827 & 590537 & 2915594 & 511919 \\ 3091 & 3773 & 12087 & 98416 & 72256 & 197062 & 60628 \end{bmatrix}.$$

With **Z** specified we can now find the direct and total supply coefficients, $\mathbf{B} = \hat{\mathbf{x}}^{-1}\mathbf{Z}$ and $\mathbf{G} = (\mathbf{I} - \mathbf{B})^{-1}$, respectively as:

$$\mathbf{B} = \begin{bmatrix} 0.2258 & 0.0000 & 0.0063 & 0.5513 & 0.0014 & 0.0585 & 0.0056 \\ 0.0021 & 0.1432 & 0.0245 & 0.7635 & 0.3105 & 0.0108 & 0.0447 \\ 0.0012 & 0.0001 & 0.0010 & 0.0061 & 0.0096 & 0.0573 & 0.0417 \\ 0.0136 & 0.0078 & 0.0752 & 0.3222 & 0.0409 & 0.1323 & 0.0569 \\ 0.0076 & 0.0050 & 0.0392 & 0.1328 & 0.0704 & 0.1044 & 0.0367 \\ 0.0025 & 0.0048 & 0.0152 & 0.0502 & 0.0564 & 0.2783 & 0.0489 \\ 0.0012 & 0.0015 & 0.0048 & 0.0390 & 0.0286 & 0.0780 & 0.0240 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1.3139 & 0.0129 & 0.1027 & 1.1313 & 0.0811 & 0.3446 & 0.0987 \\ 0.0365 & 1.1863 & 0.1691 & 1.5050 & 0.4944 & 0.4018 & 0.1883 \\ 0.0026 & 0.0010 & 1.0054 & 0.0257 & 0.0190 & 0.0930 & 0.0499 \\ 0.0301 & 0.0169 & 0.1284 & 1.5707 & 0.0997 & 0.3280 & 0.1182 \\ 0.0165 & 0.0102 & 0.0674 & 0.2631 & 1.1072 & 0.2229 & 0.0716 \\ 0.0085 & 0.0102 & 0.0379 & 0.1502 & 0.1005 & 1.4409 & 0.0868 \\ 0.0041 & 0.0036 & 0.0154 & 0.0863 & 0.0454 & 0.1363 & 1.0390 \end{bmatrix}$$

We can also determine the corresponding vector of total value added as:

$$\mathbf{v}' = \mathbf{x}' - \mathbf{i}'\mathbf{Z} = [123,475 \;\; 233,770 \;\; 648,605 \;\; 1,487,054 \;\; 2,137,239 \;\; 6,324,050 \;\; 1,501,645]$$

As an example, a 10 percent reduction in construction primary inputs (sector 3) would reduce v_3 to 583,745 from 648,605. If we define the new **v** incorporating the reduced manufacturing labor input as $\overline{\mathbf{v}}$, then we can compute

$$\overline{\mathbf{x}}' = \overline{\mathbf{v}}'\mathbf{G} = [312,584 \;\; 396,496 \;\; 1,237,177 \;\; 4,483,860 \;\; 3,354,712 \;\; 10,471,611 \;\; 2,523,091],$$

which represents a 0.33 percent reduction in total output ($\mathbf{i}'\overline{\mathbf{x}}$ compared with $\mathbf{i}'\mathbf{x}$). For comparison, a 10 percent reduction in the services sector (sector 6) generates a 5.08 percent reduction in total output.

Computational Notes

Retrieve **A** and **x** and compute the associated **Z**, **B**, and **G**, as well as the vector of total value-added inputs **v**.

```
      G←LINV B←(DIAG ÷x)+.×Z←A+.×DIAG x
      v←x-+⌿Z
A
    0.2258    0.0000    0.0015    0.0384    0.0001    0.0017    0.0007
    0.0027    0.1432    0.0075    0.0675    0.0367    0.0004    0.0070
    0.0051    0.0002    0.0010    0.0018    0.0037    0.0071    0.0215
    0.1955    0.0877    0.2591    0.3222    0.0547    0.0566    0.1010
    0.0819    0.0422    0.1011    0.0994    0.0704    0.0334    0.0487
    0.0843    0.1276    0.1225    0.1172    0.1760    0.2783    0.2026
    0.0099    0.0095    0.0093    0.0219    0.0215    0.0188    0.0240

x
    312753.9    396562.7    1302388.3    4485529.1    3355943.7   10477640.1

B
    0.2258    0.0000    0.0063    0.5513    0.0014    0.0585    0.0056
    0.0021    0.1432    0.0245    0.7635    0.3105    0.0108    0.0447
    0.0012    0.0001    0.0010    0.0061    0.0096    0.0573    0.0417
    0.0136    0.0078    0.0752    0.3222    0.0409    0.1323    0.0569
    0.0076    0.0050    0.0392    0.1328    0.0704    0.1044    0.0367
    0.0025    0.0048    0.0152    0.0502    0.0564    0.2783    0.0489
    0.0012    0.0015    0.0048    0.0390    0.0286    0.0780    0.0240
G
    1.3139    0.0129    0.1027    1.1313    0.0811    0.3446    0.0987
    0.0365    1.1863    0.1691    1.5050    0.4944    0.4018    0.1883
    0.0026    0.0010    1.0054    0.0257    0.0190    0.0930    0.0499
    0.0301    0.0169    0.1284    1.5707    0.0997    0.3280    0.1182
    0.0165    0.0102    0.0674    0.2631    1.1072    0.2229    0.0716
    0.0085    0.0102    0.0379    0.1502    0.1005    1.4409    0.0868
    0.0041    0.0036    0.0154    0.0863    0.0454    0.1363    1.0390

v
  123474.5  233769.6  648605.2 1487053.6 2137238.6 6324050.5 1501645.3
```

We define the new vector of total value-added inputs reflecting the reduction in construction as `v1` and the resulting vector of differences in value added from **v** as `dv` and the sum of all differences as `tdv`. The corresponding differences in total output using the total supply coefficients are then `x1`, `dx`, and `tdx`. We specify the percentage changes in **x** and **v** and their totals as `pdx`, `ptdx`, `pdv`, and `ptdv`.

```
      tdv←+/dv←v-v1←v×1 1 0.9 1 1 1 1
      tdx←+/dx←x-x1←,(1 7ρv1)+.×G
      pdx←100×dx÷x
      ptdx←100×tdx÷tx
      pdv←100×dv÷v
      ptdv←100×tdv÷tv
v1
  123474.5  233769.6  583744.7 1487053.6 2137238.6 6324050.5 1501645.3
dv
       0.0       0.0   64860.5       0.0       0.0       0.0       0.0
tdv
64860.52
x1
    312584.2    396496.4   1237177.4   4483860.2   3354712.2  10471611.0
2523090.8
dx
     169.7      66.3   65210.9    1668.9    1231.5    6029.1    3234.1
tdx
77610.436
pdx
0.054258429 0.016706818 5.007025 0.037206196 0.036694852 0.057542365
0.12801772
ptdx
0.33954566
pdv
0 0 10 0 0 0 0
ptdv
0.52072389
```

We define new vector of value-added inputs reflecting a reduction in services instead of construction as `v12` so

```
      tdv2←+/dv2←v-v12←v×1 1 1 1 1 0.9 1
      tdx2←+/dx2←x-x12←,(1 7ρv12)+.×G
      pdx2←100×dx÷x
      ptdx2←100×tdx2÷tx
      pdv2←100×dv÷v
      ptdv2←100×tdv2÷tv


v1
  123474.5  233769.6  648605.2 1487053.6 2137238.6 5691645.4 1501645.3
dv
       0.0       0.0       0.0       0.0       0.0  632405.0       0.0
tdv
632405.05
x1
    307352.4    390098.5   1278392.7   4390562.8   3292412.5   9566439.1
2471433.6
dx
    5401.5    6464.2   23995.6   94966.3   63531.2  911201.0   54891.3
tdx
1160451
```

```
pdx
0.054258429 0.016706818 5.007025 0.037206196 0.036694852 0.057542365
0.12801772
ptdx
5.076973
pdv
0 0 10 0 0 0 0
ptdv
5.0771782
```

Problem 7.10: Direct and Indirect Linkages of the US Economy

This problem explores the direct and the direct and indirect forward and backward linkages for the sectors in the U.S. economy and examine how these linkages have changed over time, using the seven-sector input-output data for the United States presented in the Supplemental Resources to this text (Appendix SD1, described in Appendix B).

Overview of Problem 7.10

The following shows, for the seven industry sectors for the years 1919-2018, the backward direct linkages, $\mathbf{b}(d) = \mathbf{i}'\mathbf{A}$ [denoted as B(d) in the table, found as the column sums of **A** for each year along with the average across the 7 sectors, b(d)], the backward total linkages, $\mathbf{b}(t) = \mathbf{i}'\mathbf{L}$ [denoted as B(t) in the table found by the column sums of **L** along with the average across the 7 sectors; b(t)];the forward direct linkages $\mathbf{f}(d) = \mathbf{Bi}$ [denoted by F(d) in the table, found as the row sums of **B** along with the average across the 7 sectors, f(d)]; and the forward total linkages, $\mathbf{f}(t) = \mathbf{Gi}$ [denoted by F(t) in the table, found by the row sums of **G**, along with the average across the 7 sectors, f(t)].

B(d)	1	2	3	4	5	6	7	b(d)
1919	0.556	0.748	0.729	0.722	0.57	0.546	0.524	0.628
1929	0.57	0.653	0.59	0.706	0.53	0.638	0.444	0.59
1938	0.624	0.724	0.517	0.807	0.639	0.449	0.626	0.627
1947	0.38	0.467	0.58	0.657	0.348	0.358	0.161	0.422
1958	0.462	0.472	0.609	0.633	0.35	0.333	0.266	0.446
1963	0.528	0.459	0.586	0.62	0.346	0.326	0.238	0.443
1967	0.548	0.492	0.553	0.611	0.334	0.335	0.265	0.448
1972	0.541	0.488	0.596	0.619	0.302	0.335	0.239	0.446
1977	0.572	0.479	0.559	0.656	0.357	0.332	0.263	0.46
1982	0.581	0.447	0.557	0.665	0.379	0.325	0.302	0.465
1987	0.547	0.454	0.573	0.636	0.347	0.351	0.308	0.459
1992	0.546	0.494	0.544	0.629	0.344	0.347	0.294	0.457
1997	0.579	0.463	0.521	0.645	0.351	0.372	0.312	0.463
2002	0.604	0.425	0.491	0.63	0.36	0.374	0.334	0.46
2007	0.585	0.34	0.466	0.659	0.395	0.393	0.357	0.456
2012	0.603	0.415	0.485	0.666	0.412	0.374	0.363	0.474
2018	0.627	0.444	0.478	0.627	0.415	0.389	0.358	0.477

B(t)	1	2	3	4	5	6	7	b(t)
1919	2.366	2.922	3.01	2.879	2.547	2.354	2.47	2.65
1929	2.359	2.5	2.475	2.701	2.292	2.415	2.141	2.412
1938	2.915	3.245	2.702	3.525	2.988	2.401	2.978	2.965
1947	1.742	1.916	2.219	2.359	1.648	1.703	1.296	1.84
1958	1.899	1.913	2.262	2.304	1.636	1.634	1.528	1.882
1963	2.066	1.843	2.219	2.291	1.623	1.6	1.467	1.873
1967	2.1	1.92	2.133	2.259	1.596	1.613	1.509	1.876
1972	2.071	1.886	2.2	2.263	1.524	1.602	1.454	1.857
1977	2.218	1.944	2.215	2.42	1.662	1.622	1.522	1.943
1982	2.23	1.862	2.179	2.41	1.707	1.604	1.595	1.941
1987	2.088	1.838	2.142	2.302	1.616	1.63	1.59	1.887
1992	2.078	1.892	2.071	2.266	1.602	1.607	1.543	1.865
1997	2.178	1.872	2.068	2.374	1.617	1.658	1.599	1.909
2002	2.211	1.772	1.974	2.294	1.622	1.649	1.62	1.877
2007	2.214	1.634	1.967	2.38	1.716	1.707	1.692	1.901
2012	2.271	1.797	2.011	2.415	1.752	1.667	1.704	1.945
2018	2.299	1.837	1.964	2.297	1.739	1.687	1.684	1.929

F(d)	1	2	3	4	5	6	7	f(d)
1919	0.821	0.806	0.671	0.531	0.679	0.307	0.631	0.635
1929	0.743	0.835	0.624	0.554	0.691	0.412	0.52	0.625
1938	0.717	0.871	0.732	0.571	0.959	0.18	0.879	0.701
1947	0.866	0.812	0.161	0.555	0.406	0.41	0.123	0.476
1958	0.808	0.917	0.151	0.598	0.448	0.39	0.137	0.493
1963	0.847	0.947	0.114	0.607	0.412	0.373	0.138	0.491
1967	0.844	0.915	0.139	0.6	0.416	0.386	0.128	0.49
1972	0.846	0.976	0.162	0.601	0.42	0.383	0.12	0.501
1977	0.793	0.252	0.133	0.619	0.442	0.378	0.128	0.535
1982	0.786	0.957	0.122	0.62	0.448	0.384	0.144	0.494
1987	0.888	0.096	0.146	0.619	0.417	0.404	0.151	0.532
1992	0.8	0.115	0.173	0.6	0.412	0.39	0.163	0.522
1997	0.812	0.14	0.127	0.627	0.4	0.418	0.171	0.528
2002	0.812	0.244	0.134	0.628	0.391	0.425	0.16	0.542
2007	0.811	0.294	0.155	0.657	0.403	0.43	0.159	0.558
2012	0.848	0.197	0.218	0.643	0.404	0.416	0.159	0.555
2018	0.84	0.879	0.179	0.659	0.403	0.43	0.164	0.508

F(t)	1	2	3	4	5	6	7	f(t)
1919	3.357	3.081	2.725	2.36	2.902	1.8	2.571	2.685
1929	2.941	3.038	2.444	2.316	2.75	1.947	2.269	2.529
1938	3.153	4.006	3.379	2.789	3.981	1.579	3.702	3.227
1947	2.885	2.744	1.271	2.122	1.813	1.782	1.208	1.975
1958	2.802	3.016	1.251	2.194	1.895	1.747	1.237	2.02
1963	2.97	3.041	1.185	2.22	1.81	1.693	1.252	2.024
1967	2.989	2.949	1.229	2.197	1.809	1.712	1.226	2.016
1972	2.974	3.089	1.258	2.184	1.812	1.691	1.216	2.032
1977	2.872	3.858	1.222	2.293	1.902	1.715	1.244	2.158
1982	2.851	3.125	1.199	2.266	1.905	1.72	1.271	2.048
1987	3.105	3.388	1.231	2.224	1.812	1.725	1.276	2.109
1992	2.819	3.355	1.27	2.168	1.785	1.686	1.29	2.053
1997	2.952	3.511	1.22	2.283	1.766	1.752	1.31	2.113
2002	2.917	3.675	1.228	2.234	1.724	1.748	1.285	2.116
2007	2.976	3.966	1.268	2.334	1.77	1.772	1.29	2.197
2012	3.073	3.784	1.369	2.309	1.783	1.74	1.29	2.193
2018	3.04	2.96	1.298	2.306	1.761	1.763	1.297	2.061

Computational Notes

We presume that the USIO tables for the years 1919, 1929, and 1938 are present in the workspace and saved under the variable names `Zii` and and `xii` where `ii` is replaced by the last two digits of the designated year, and for rest of the tables the Make and Use tables are named `Uii` and `Vii`, respectively, with the same naming convention for designated years.

For the first set of tables, for each designated year, we compute the corresponding **A**, **L**, **B**, and **G** matrices.

```
L←INV A←Z AMAT x
G←INV B←(DIAG ÷x)+.×Z
```

For the second group of tables, for each designated year, we first compute the total industry and commodity output vectors as **x** and **q**, respectively, and then compute the commodity-by-industry requirements matrix `BB` and the commodity output proportions matrix **D** in order to compute the industry-by-industry direct requirements matrix (under an industry-based technology assumption) `AI` and corresponding total requirements matrix `LI`. Then we derive the corresponding matrices of interindustry transactions **Z**, of direct supply coefficients **B**, and of total supply coefficients **G**.

```
x←+/V ⋄ q←+⌿V ⋄ D←V AMAT q ⋄ BB←U AMAT x
LI←LINV AI←D+.×BB
G←LINV B←(DIAG ÷x)+.×Z←AI+.×DIAG x
```

The APL function `USIOLINKAGE` listed below sequentially applies these APL expressions to generate the results table listed above as an explicit result `RES`.

```
RES←USIOLINKAGE
⍝--This version uses MB3 Data for USIO tables
⍝Compute Bd, Bt, Fd, Ft for all US 7 sector tables
 nall←1919 1929 1938 1947 1958 1963 1967 1972 1977
 n←ρnall←nall,1982 1987 1992 1997 2002 2007 2012 2018
 nall←(((k←1),n)ρnall),[1](3ρ1),14ρ2 ⋄ nw←54
 label←(6 0⍕⍳7)[3+⍳39],(3ρ' ')
 Bd←(1,nw)ρ' B(d) ',label,'  b(d)' ⋄ Bt←(1,nw)ρ' B(t) ',label,'  b(t)'
 Fd←(1,nw)ρ' F(d) ',label,'  f(d)' ⋄ Ft←(1,nw)ρ' F(t) ',label,'  f(t)'
LL:K←⍕nall[1;k] ⋄ nK←ρK
 ⍝For 1919-1938 use Zxx and Xxx
 ⍝For 1946-2018 use Uxx and Vxx
 ⍝where xx are last two digits of the designated year
 →(2=nall[2;k])/L2
 ⍎'Z←D',K,'_7_Z'
 ⍎'x←,D',K,'_7_x'
 L←INV A←Z AMAT x ⋄ G←LINV B←(DIAG ÷x)+.×Z
 Bd←Bd,[1](6 0⍕nall[1;k]),6 3⍕(+⌿A),(+/+⌿A)÷7
 Bt←Bt,[1](6 0⍕nall[1;k]),6 3⍕(+⌿L),(+/+⌿L)÷7
 Fd←Fd,[1](6 0⍕nall[1;k]),6 3⍕(+/B),(+/+/B)÷7
 Ft←Ft,[1](6 0⍕nall[1;k]),6 3⍕(+/G),(+/+/G)÷7
 →END
L2:⍎'U←D',K,'_7_U'
 ⍎'V←⍉D',K,'_7_S'
 x←+/V ⋄ q←+⌿V ⋄ D←V AMAT q ⋄ BB←U AMAT x
 LI←LINV AI←D+.×BB
⍝AI and LI for industry based tech; change to AC snd LC for comm based tech
 G←LINV B←(DIAG ÷x)+.×Z←AI+.×DIAG x
 Bd←Bd,[1](6 0⍕nall[1;k]),6 3⍕(+⌿AI),(+/+⌿AI)÷7
 Bd[;2++\8ρ6]←' '
 Bt←Bt,[1](6 0⍕nall[1;k]),6 3⍕(+⌿LI),(+/+⌿LI)÷7
 Fd←Fd,[1](6 0⍕nall[1;k]),6 3⍕(+/B),(+/+/B)÷7
 Fd[;2++\8ρ6]←' '
 Ft←Ft,[1](6 0⍕nall[1;k]),6 3⍕(+/G),(+/+/G)÷7
END:→(n≥k←k+1)/LL
 s←(1,nw)ρ' '
 RES←Bd,[1]s,[1]Bt,[1]s,[1]Fd,[1]s,[1]Ft
```

Chapter 8, Decomposition Approaches

Chapter 8 introduces and illustrates the basic concepts of structural decomposition analysis (SDA) within an input–output framework, in related additive and multiplicative formulations. The application of SDA to MRIO is developed to introduce a spatial context. Appendices to this chapter develop extended presentations of additional decomposition results as well as an overview of early applied studies and some further mathematical results. The exercise problems for this chapter illustrate various analytical features of SDA.

Problem 8.1: The Basic Principles of Structural Decomposition Analysis (SDA)

This problem explores the basic principles of SDA.

Overview of Problem 8.1

Consider an input-output economy specified at two points in time, t^0 and t^1 by matrices of interindustry transactions and final demand vectors:

$\mathbf{Z}^0 = \begin{bmatrix} 10 & 20 & 30 \\ 5 & 2 & 25 \\ 20 & 40 & 60 \end{bmatrix}$, $\mathbf{f}^0 = \begin{bmatrix} 60 \\ 40 \\ 55 \end{bmatrix}$, $\mathbf{Z}^1 = \begin{bmatrix} 15 & 25 & 40 \\ 12 & 7.5 & 30 \\ 10 & 30 & 40 \end{bmatrix}$, and $\mathbf{f}^1 = \begin{bmatrix} 75 \\ 55 \\ 40 \end{bmatrix}$. To measure how the economy has changed in structure over the period, we can compute for each sector the change in total output between the two years that was attributable to changing final demand or to changing technology by the following.

First, we compute $\mathbf{A}^0 = \mathbf{Z}^0(\hat{\mathbf{x}}^0)^{-1} = \begin{bmatrix} .083 & .260 & .176 \\ .042 & .026 & .147 \\ .167 & .519 & .353 \end{bmatrix}$ and

$\mathbf{A}^1 = \mathbf{Z}^1(\hat{\mathbf{x}}^1)^{-1} = \begin{bmatrix} .097 & .239 & .333 \\ .077 & .072 & .250 \\ .065 & .287 & .333 \end{bmatrix}$. Next, $\mathbf{L}^0 = (\mathbf{I}-\mathbf{A}^0)^{-1} = \begin{bmatrix} 1.199 & .562 & .455 \\ .111 & 1.221 & .308 \\ .398 & 1.125 & 1.910 \end{bmatrix}$ and

$\mathbf{L}^1 = (\mathbf{I}-\mathbf{A}^1)^{-1} = \begin{bmatrix} 1.214 & .566 & .82 \\ .15 & 1.289 & .558 \\ .182 & 1.61 & 1.82 \end{bmatrix}$. Then $\Delta\mathbf{f} = \mathbf{f}^1 - \mathbf{f}^0 = \begin{bmatrix} 15 \\ 10 \\ -10 \end{bmatrix}$ and

$\Delta\mathbf{L} = \mathbf{L}^1 - \mathbf{L}^0 = \begin{bmatrix} .015 & .004 & .365 \\ .039 & .068 & .251 \\ -.216 & -.515 & -.09 \end{bmatrix}$.

We can find $\mathbf{x}^0 = \mathbf{L}^0\mathbf{f}^0 = \begin{bmatrix} 120 \\ 77 \\ 170 \end{bmatrix}$ and $\mathbf{x}^1 = \mathbf{L}^1\mathbf{f}^1 = \begin{bmatrix} 155 \\ 104.5 \\ 120 \end{bmatrix}$ so $\Delta\mathbf{x} = \begin{bmatrix} 35 \\ 27.5 \\ -50 \end{bmatrix}$. Then, using the basic structural decomposition relationship, $\Delta\mathbf{x} = \underbrace{(1/2)(\Delta\mathbf{L})(\mathbf{f}^0 + \mathbf{f}^1)}_{\text{Technology change effect}} + \underbrace{(1/2)(\mathbf{L}^0 + \mathbf{L}^1)(\Delta\mathbf{f})}_{\text{Final-demand change effect}}$, we have the results in the following table (figures in parentheses are percentages of the total output change in each row).

	Output Change	Technology Change Contribution	Final-Demand Change Contribution
Sector 1	35	17.65 (50)	17.35 (50)
Sector 2	27.5	17.3 (63)	10.2 (37)
Sector 3	-50	-44.4 (89)	-5.6 (11)
Economy-wide Total	12.5	-9.45 (-75)	21.95 (175)

Computational Notes

We define the given matrices of transactions and vectors of final demands for the two years, respectively, as `Z0`, `Z2`, `f0`, and `f1` and compute the corresponding vector of total outputs and matrices of direct requirements and total requirements with the same naming convention.

```
      Z0←3 3ρ10 20 30 5 2 25 20 40 60 ⋄ f0←60 45 50
      Z1←3 3ρ15 25 40 12 7.5 30 10 30 40 ⋄ f1←75 55 40
      L1←INV A1←Z1 AMAT x1←f1++/Z1 ⋄ L0←INV A0←Z0 AMAT x0←f0++/Z0

10 0⍕Z0,⍉2 3ρf0,x0
        10        20        30        60       120
         5         2        25        45        77
        20        40        60        50       170
10 0⍕Z1,⍉2 3ρf1,x1

        15        25        40        75       155
        12         8        30        55       105
        10        30        40        40       120
A0,L0
    0.0833    0.2597    0.1765    1.1992    0.5624    0.4549
    0.0417    0.0260    0.1471    0.1114    1.2205    0.3078
    0.1667    0.5195    0.3529    0.3983    1.1247    1.9097
A1,L1
    0.0968    0.2392    0.3333    1.2142    0.5664    0.8195
    0.0774    0.0718    0.2500    0.1504    1.2888    0.5585
    0.0645    0.2871    0.3333    0.1823    0.6098    1.8198
```

We then compute the changes in the Leontief invers coefficients `ΔL`, the changes in final demands `Δf`, and the resulting changes in total requirements `Δx`, including the technology change effect `T` and the final demand change effect `F`.

```
      ΔL←L1-L0 ⋄ Δf←f1-f0
      Δx←0.5×(T←ΔL+.×f0+f1)+F←(L0+L1)+.×Δf
```

```
ΔL
    0.0151     0.0040     0.3647
    0.0389     0.0683     0.2507
   ¯0.2161    ¯0.5149    ¯0.0899
Δf
      15.0       10.0      ¯10.0
f0+f1
     135.0      100.0       90.0
L0+L1
    2.4134     1.1288     1.2744
    0.2618     2.5093     0.8663
    0.5806     1.7345     3.7295
Δx
  35.0  27.5 ¯50.0
T
  35.3  34.6 ¯88.8
F
  34.7  20.4 ¯11.2
```

Problem 8.2: Illustrating SDA with the US Economy

This problem illustrates the basic principles of SDA using input-output data for the U.S. economy for the years 1972 and 2002 aggregated to 7 industry sectors.

Problem 8.2 Overview

The technical requirements and total outputs tables for 2002 and 1972 are the following:

A and **x** for US, 2002	1	2	3	4	5	6	7	Tot. Output
1 Agriculture	0.2637	0.0020	0.0028	0.0374	0.0007	0.0008	0.0008	270,514
2 Mining	0.0032	0.0467	0.0097	0.0377	0.0226	0.0005	0.0040	184,516
3 Construction	0.0040	0.0336	0.0007	0.0030	0.0053	0.0078	0.0186	967,568
4 Manufacturing	0.1502	0.0942	0.2399	0.3464	0.0645	0.0464	0.0939	3,850,417
5 Trade, Transport & Utils	0.0868	0.0676	0.0960	0.0920	0.0816	0.0302	0.0475	2,811,865
6 Services	0.1310	0.2416	0.1436	0.1349	0.1813	0.2640	0.1954	8,948,582
7 Other	0.0098	0.0159	0.0083	0.0160	0.0276	0.0179	0.0203	2,146,282

A and **x** for US, 1972	1	2	3	4	5	6	7	Tot. Output
1 Agriculture	0.3141	0.0003	0.0028	0.0542	0.0010	0.0053	0.0012	83,955
2 Mining	0.0019	0.0542	0.0091	0.0296	0.0160	0.0002	0.0020	30,386
3 Construction	0.0069	0.0282	0.0003	0.0043	0.0156	0.0263	0.0166	165,998
4 Manufacturing	0.1436	0.0943	0.3522	0.3771	0.0407	0.0892	0.0078	761,194
5 Trade, Transport & Utils	0.0616	0.0481	0.1043	0.0786	0.0980	0.0442	0.0202	377,389
6 Services	0.0865	0.1471	0.0686	0.0591	0.1157	0.1621	0.0105	522,215
7 Other	0.0023	0.0063	0.0042	0.0117	0.0118	0.0096	0.0033	161,207

To compute the changes in total output between 1972 and 2002 for all sectors attributed to changes in final demand and to changes in technology, we employ the basic SDA relationship, $\Delta\mathbf{x} = \underbrace{(1/2)(\Delta\mathbf{L})(\mathbf{f}^0 + \mathbf{f}^1)}_{\text{Technology change effect}} + \underbrace{(1/2)(\mathbf{L}^0 + \mathbf{L}^1)(\Delta\mathbf{f})}_{\text{Final demand change effect}}$, to yield:

Sector	$\Delta\mathbf{x}$	Technology	Final Demand
1	186,559	-107,931	294,490
2	154,131	27,372	126,758
3	801,570	-73,551	875,121
4	3,089,223	-209,242	3,298,465
5	2,434,476	-3,750	2,438,225
6	8,426,367	1,224,317	7,202,050
7	1,985,075	109,282	1,875,793

Computational Notes

We presume the matrix of technical coefficients for 1972 resides in the workspace as `A0`, along with the corresponding vector of total outputs as `x0` and the same quantities for 2002 as `A1` and `x1`. We can compute the corresponding vectors of final demand, `f0` and `f1`, and matrices of total requirements, `L0` and `L1`.

```
     Z0←A0+.×DIAG x0 ⋄ f0←x0-+/Z0
     Z1←A1+.×DIAG x1 ⋄ f1←x1-+/Z1
     L1←LINV A1 ⋄ L0←LINV A0
A0
  0.3141  0.0003  0.0028  0.0542  0.0010  0.0053  0.0012
  0.0019  0.0542  0.0091  0.0296  0.0160  0.0002  0.0020
  0.0069  0.0282  0.0003  0.0043  0.0156  0.0263  0.0166
  0.1436  0.0943  0.3522  0.3771  0.0407  0.0892  0.0078
  0.0616  0.0481  0.1043  0.0786  0.0980  0.0442  0.0202
  0.0865  0.1471  0.0686  0.0591  0.1157  0.1621  0.0105
  0.0023  0.0063  0.0042  0.0117  0.0118  0.0096  0.0033
x0
   83955   30386  165998  761194  377389  522215  161207
f0
   12504   ¯1907  139002  337534  230264  324127  141242
L0
  1.4913  0.0204  0.0552  0.1353  0.0125  0.0262  0.0044
  0.0182  1.0665  0.0326  0.0563  0.0232  0.0087  0.0037
  0.0206  0.0387  1.0117  0.0172  0.0237  0.0351  0.0179
  0.3979  0.2261  0.6256  1.6905  0.1187  0.2087  0.0292
  0.1503  0.0936  0.1854  0.1704  1.1327  0.0850  0.0286
  0.2078  0.2216  0.1642  0.1683  0.1723  1.2272  0.0212
  0.0121  0.0128  0.0157  0.0241  0.0168  0.0155  1.0043
A1
  0.2637  0.0020  0.0028  0.0374  0.0007  0.0008  0.0008
  0.0032  0.0467  0.0097  0.0377  0.0226  0.0005  0.0040
  0.0040  0.0336  0.0007  0.0030  0.0053  0.0078  0.0186
  0.1502  0.0942  0.2399  0.3464  0.0645  0.0464  0.0939
  0.0868  0.0676  0.0960  0.0920  0.0816  0.0302  0.0475
  0.1310  0.2416  0.1436  0.1349  0.1813  0.2640  0.1954
```

```
  0.0098  0.0159  0.0083  0.0160  0.0276  0.0179  0.0203
x1
  270514  184516  967568 3850417 2811865 8948582 2146281
f1
   41201  ¯56147  824036 1428128 1727026 4918450 1789205
L1
  1.3779  0.0148  0.0260  0.0831  0.0093  0.0077  0.0116
  0.0242  1.0614  0.0314  0.0692  0.0330  0.0072  0.0147
  0.0120  0.0411  1.0071  0.0123  0.0108  0.0125  0.0234
  0.3743  0.2192  0.4243  1.6164  0.1508  0.1179  0.1951
  0.1850  0.1218  0.1651  0.1913  1.1213  0.0623  0.0889
  0.3784  0.4379  0.3371  0.3943  0.3293  1.4097  0.3434
  0.0326  0.0327  0.0271  0.0411  0.0408  0.0298  1.0333
```

We the compute the changes in the Leontief inverse coefficients `ΔL`, the changes in final demands `Δf`, and the resulting changes in total requirements `Δx`, including the technology change effect `T` and the final demand change effect `F`.

```
     ΔL←L1-L0 ⋄ Δf←f1-f0
     Δx←0.5×(T←ΔL+.×f0+f1)+F←(L0+L1)+.×Δf

ΔL
 ¯0.1134 ¯0.0057 ¯0.0292 ¯0.0522 ¯0.0032 ¯0.0186  0.0072
  0.0060 ¯0.0051 ¯0.0012  0.0130  0.0098 ¯0.0015  0.0110
 ¯0.0085  0.0024 ¯0.0046 ¯0.0049 ¯0.0128 ¯0.0227  0.0055
 ¯0.0236 ¯0.0069 ¯0.2013 ¯0.0741  0.0321 ¯0.0908  0.1659
  0.0346  0.0283 ¯0.0203  0.0210 ¯0.0114 ¯0.0227  0.0603
  0.1706  0.2164  0.1730  0.2260  0.1570  0.1825  0.3223
  0.0205  0.0199  0.0114  0.0170  0.0240  0.0142  0.0290
Δf
   28698  ¯54240  685034 1090594 1496763 4594323 1647963
f0+f1
   53705  ¯58054  963038 1765662 1957290 5242577 1930446
L0+L1
  2.8692  0.0352  0.0811  0.2184  0.0218  0.0339  0.0160
  0.0425  2.1279  0.0639  0.1255  0.0562  0.0158  0.0183
  0.0326  0.0798  2.0188  0.0295  0.0345  0.0476  0.0414
  0.7722  0.4454  1.0499  3.3068  0.2695  0.3266  0.2243
  0.3353  0.2154  0.3505  0.3617  2.2540  0.1473  0.1175
  0.5862  0.6595  0.5013  0.5626  0.5015  2.6368  0.3646
  0.0447  0.0455  0.0428  0.0653  0.0576  0.0453  2.0376

Δx
  186559  154131  801570 3089223 2434475 8426367 1985075
T
 ¯215862   54745 ¯147102 ¯418484   ¯7500 2448634  218564
.5×T
 ¯107931   27372  ¯73551 ¯209242   ¯3750 1224317  109282
F
  588981  253517 1750242 6596931 487645114404100 3751585
.5×F
  294490  126758  875121 3298465 2438225 7202050 1875793
```

Problem 8.3: Uniform Growth

This problem illustrates a special case of SDA, uniform growth.

Problem 8.3 Overview

Consider an input-output economy specified by transactions matrices and final demand vectors for periods 0 and 1.

$$\mathbf{Z}^0 = \begin{bmatrix} 10 & 20 & 25 \\ 15 & 5 & 30 \\ 30 & 40 & 5 \end{bmatrix}, \mathbf{f}^0 = \begin{bmatrix} 45 \\ 30 \\ 25 \end{bmatrix}, \mathbf{Z}^1 = \begin{bmatrix} 15 & 30 & 37.5 \\ 22.5 & 7.5 & 45 \\ 45 & 60 & 7.5 \end{bmatrix}, \text{ and } \mathbf{f}^1 = \begin{bmatrix} 67.5 \\ 45 \\ 37.5 \end{bmatrix}$$

If we apply the basic SDA formulation in this case, we find that the changes represent uniform growth, i.e., both transactions and final demand grow uniformly by 50 percent between periods 0 and 1 as will the resulting total outputs (perhaps obvious in retrospect).

Computational Notes

We define the matrices of transactions and corresponding vectors of total outputs for the two periods, respectively, as `Z0`, `f0`, `Z1`, and `f1`. Then compute the vectors of total outputs, `x0` and `x1`, and the matrices of direct requirements, `A0` and `A1`, and of total requirements, `L0` and `L1`.

```
    Z0←3 3ρ10 20 25 15 5 30 30 40 5 ⋄ f0←45 30 25
    Z1←3 3ρ15 30 37.5 22.5 7.5 45 45 60 7.5 ⋄ f1←67.5 45 37.5
    L1←LINV A1←Z1 AMAT x1←f1++/Z1 ⋄ L0←LINV A0←Z0 AMAT x0←f0++/Z0

10 0⍕Z0,⍉2 3ρf0,x0
        10        20        25        45       100
        15         5        30        30        80
        30        40         5        25       100
10 0⍕Z1,⍉2 3ρf1,x1
        15        30        38        68       150
        23         8        45        45       120
        45        60         8        38       150
A0,L0
    0.1000    0.2500    0.2500    1.4260    0.6980    0.5957
    0.1500    0.0625    0.3000    0.4477    1.5018    0.5921
    0.3000    0.5000    0.0500    0.6859    1.0108    1.5523
A1,L1
    0.1000    0.2500    0.2500    1.4260    0.6980    0.5957
    0.1500    0.0625    0.3000    0.4477    1.5018    0.5921
    0.3000    0.5000    0.0500    0.6859    1.0108    1.5523
```

We the compute the changes in the Leontief inverse coefficients `ΔL`, the changes in final demands `Δf`, and the resulting changes in total requirements Δx, including the technology change effect `T` and the final demand change effect `F`.

```
ΔL
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
```

```
Δf
      22.5      15.0      12.5
Δx
  50.0  40.0  50.0
.5×T
   0.0   0.0   0.0
.5×F
  50.0  40.0  50.0
```

Problem 8.4: More Complex Forms of Structural Decomposition

This problem explores a more complex form of SDA involving changes in technology and final demand as well as interactions between technology and final demand.

Problem 8.4 Overview

Consider two observations on an input-output economy specified by matrices of interindustry transactions and vectors of total outputs for two years, designate 0 and 1:

$\mathbf{Z}^0 = \begin{bmatrix} 12 & 15 & 35 \\ 24 & 11 & 30 \\ 36 & 50 & 8 \end{bmatrix}$, $\mathbf{f}^0 = \begin{bmatrix} 50 \\ 35 \\ 26 \end{bmatrix}$, $\mathbf{Z}^1 = \begin{bmatrix} 20 & 30 & 45 \\ 35 & 23 & 50 \\ 50 & 65 & 24 \end{bmatrix}$, $\mathbf{f}^1 = \begin{bmatrix} 55 \\ 50 \\ 60 \end{bmatrix}$. From these basic data we can compute the vectors of total outputs and the matrices of technical requirements and total requirements for both years: $\mathbf{x}^0 = \mathbf{f}^0 + \mathbf{Z}^0\mathbf{i} = \begin{bmatrix} 112 \\ 100 \\ 120 \end{bmatrix}$, $\mathbf{x}^1 = \mathbf{f}^1 + \mathbf{Z}^1\mathbf{i} = \begin{bmatrix} 150 \\ 158 \\ 199 \end{bmatrix}$,

$\mathbf{A}^0 = \mathbf{Z}^0(\hat{\mathbf{x}}^0)^{-1} = \begin{bmatrix} .107 & .150 & .292 \\ .214 & .110 & .25 \\ .321 & .5 & .067 \end{bmatrix}$, $\mathbf{A}^1 = \mathbf{Z}^1(\hat{\mathbf{x}}^1)^{-1} = \begin{bmatrix} .133 & .19 & .226 \\ .233 & .146 & .251 \\ .333 & .411 & .121 \end{bmatrix}$,

$\mathbf{L}^0 = (\mathbf{I}-\mathbf{A})^{-1} = \begin{bmatrix} 1.491 & .604 & .628 \\ .592 & 1.563 & .604 \\ .831 & 1.045 & 1.611 \end{bmatrix}$, and $\mathbf{L}^1 = (\mathbf{I}-\mathbf{A})^{-1} = \begin{bmatrix} 1.541 & .618 & .573 \\ .687 & 1.633 & .643 \\ .905 & .998 & 1.655 \end{bmatrix}$. The changes in total outputs, final demands, and elements of the total requirements matrices are the

$\Delta\mathbf{x} = \mathbf{x}^1 - \mathbf{x}^0 = \begin{bmatrix} 38 \\ 58 \\ 79 \end{bmatrix}$, $\Delta\mathbf{f} = \mathbf{f}^1 - \mathbf{f}^0 = \begin{bmatrix} 5 \\ 15 \\ 34 \end{bmatrix}$, and $\Delta\mathbf{L} = \mathbf{L}^1 - \mathbf{L}^0 = \begin{bmatrix} .05 & .014 & -.055 \\ .095 & .07 & .039 \\ .075 & -.047 & .044 \end{bmatrix}$,

respectively. Now we can compute a variety of alternative structural decompositions accounting for the interaction term as summarized in equations (8.3) through (8.7):

(8.3) $\Delta\mathbf{x} = \mathbf{L}^1(\mathbf{f}^0 + \Delta\mathbf{f}) - (\mathbf{L}^1 + \Delta\mathbf{L})\mathbf{f}^0 = (\Delta\mathbf{L})\mathbf{f}^0 + \mathbf{L}^1(\Delta\mathbf{f})$

(8.4) $\Delta\mathbf{x} = (\mathbf{L}^0 + \Delta\mathbf{L})\,\mathbf{f}^1 - \mathbf{L}^0(\mathbf{f}^1 - \Delta\mathbf{f}) = (\Delta\mathbf{L})\mathbf{f}^1 + \mathbf{L}^0(\Delta\mathbf{f})$

(8.5) $\Delta\mathbf{x} = (\mathbf{L}^0 + \Delta\mathbf{L})(\mathbf{f}^0 + \Delta\mathbf{f}) - \mathbf{L}^0\mathbf{f}^0 = (\Delta\mathbf{L})\mathbf{f}^0 + \mathbf{L}^0(\Delta\mathbf{f}) + (\Delta\mathbf{L})(\Delta\mathbf{f})$

(8.6) $\Delta\mathbf{x} = \mathbf{L}^1\mathbf{f}^1 - (\mathbf{L}^1 - \Delta\mathbf{L})(\mathbf{f}^1 - \Delta\mathbf{f}) = (\Delta\mathbf{L})\mathbf{f}^1 + \mathbf{L}^1(\Delta\mathbf{f}) - (\Delta\mathbf{L})(\Delta\mathbf{f})$

$$(8.7) \quad \mathbf{\Delta x} = (1/2)\underbrace{(\mathbf{\Delta L})(\mathbf{f}^0+\mathbf{f}^1)}_{\text{Technology change}} + (1/2)\underbrace{(\mathbf{L}^0+\mathbf{L}^1)(\mathbf{\Delta f})}_{\text{Final-demand change}}$$

The following are the results from applying these equations:

		Total Output Change		Technology Change Contribution		Final-Demand Change Contribution		Interaction Term	
		Output	Percent	Output	Percent	Output	Percent	Output	Percent
Equation (8.3)	Sector 1	38	100	1.55	4.09	36.45	95.91	0.00	0.00
	Sector 2	58	100	8.21	14.15	49.79	85.85	0.00	0.00
	Sector 3	79	100	3.23	4.09	75.77	95.91	0.00	0.00
	Total	175	100	12.99	7.42	162.01	92.58	0.00	0.00
Equation (8.4)	Sector 1	38	100	0.15	0.39	37.85	99.61	0.00	0.00
	Sector 2	58	100	11.08	19.10	46.92	80.90	0.00	0.00
	Sector 3	79	100	4.40	5.57	74.60	94.43	0.00	0.00
	Total	175	100	15.62	8.93	159.38	91.07	0.00	0.00
Equation (8.5)	Sector 1	38	100	1.55	4.09	37.85	99.61	-1.41	-3.70
	Sector 2	58	100	8.21	14.15	46.92	80.90	2.87	4.94
	Sector 3	79	100	3.23	4.09	74.60	94.43	1.17	1.48
	Total	175	100	12.99	7.42	159.38	91.07	2.63	1.50
Equation (8.6)	Sector 1	38	100	0.15	0.39	36.45	95.91	1.41	3.70
	Sector 2	58	100	11.08	19.10	49.79	85.85	-2.87	-4.94
	Sector 3	79	100	4.40	5.57	75.77	95.91	-1.17	-1.48
	Total	175	100	15.62	8.93	162.01	92.58	-2.63	-1.50
Equation (8.7)	Sector 1	38	100	0.85	2.24	37.15	97.76	0.00	0.00
	Sector 2	58	100	9.64	16.63	48.36	83.37	0.00	0.00
	Sector 3	79	100	3.81	4.83	75.19	95.17	0.00	0.00
	Total	175	100	14.31	8.17	160.69	91.83	0.00	0.00

Computational Notes

We define the transactions and final demand data for the two periods as `Z0`, `f0`, `Z1`, and `f1`, and compute the associated `x0`, `A0`, `L0`, `x1`, `A1`, and `L1`, and the important differences for SDA equations as `df`, `dx`, and `DL`.

```
Z0←3 3⍴12 15 35 24 11 30 36 50 8 ⋄ f0←50 35 26
Z1←3 3⍴20 30 45 35 23 50 50 65 24 ⋄ f1←55 50 60
L1←LINV A1←Z1 AMAT x1←f1++/Z1
L0←LINV A0←Z0 AMAT x0←f0++/Z0
df←f1-f0 ⋄ dx←x1-x0 ⋄ DL←L1-L0
```

We compute the components for each of the alternative SDA approaches and save the results as the data in the Table above in `RES`.

```
⍝eq 8.3
 DX←⍉3 3⍴(dx1a←DL+.×f0),(dx2a←L1+.×df),3⍴0
⍝eq 8.4
 DX←DX,[1]⍉3 3⍴(dx1b←DL+.×f1),(dx2b←L0+.×df),3⍴0
⍝eq 8.5
 DX←DX,[1]⍉3 3⍴(dx1c←DL+.×f0),(dx2c←L0+.×df),dx3c←DL+.×df
⍝eq 8.6
```

```
 DX←DX,[1]⍉3 3ρ(dx1d←DL+.×f1),(dx2d←L1+.×df),dx3d←-DL+.×df
⍝eq 8.7
 DX←DX,[1]⍉3 3ρ(dx1e←0.5×DL+.×f0+f1),(dx2e←0.5×(L1+L0)+.×df),3ρ0
 ⍝sector specific results
 DXS←5 3 4ρ(+/DX),DX ⋄ DXS←DXS,[2]+/[2]DXS
 DXSP←100×DXS÷1 3 2⍉5 4 4ρdx,+/dx
 RES←5 4 8ρ0
 RES[;;1 3 5 7]←DXS
 RES[;;2 4 6 8]←DXSP
 ((4ρ5 0),12ρ10 2 5 0)⍕RES

 RES
 38 100  1.5523912   4.0852399   36.447609 95.91476   0          0
 58 100  8.209741   14.154726    49.790259 85.845274  0          0
 79 100  3.2292861   4.0877039   75.770714 95.912296  0          0
175 100 12.991418    7.4236676  162.00858  92.576332  0          0

 38 100  0.14669727  0.38604544  37.853303 99.613955  0          0
 58 100 11.076367   19.097184    46.923633 80.902816  0          0
 79 100  4.397554    5.566524    74.602446 94.433476  0          0
175 100 15.620618    8.9260675  159.37938  91.073933  0          0

 38 100  1.5523912   4.0852399   37.853303 99.613955 ¯1.4056939 ¯3.6991944
 58 100  8.209741   14.154726    46.923633 80.902816  2.8666258  4.9424583
 79 100  3.2292861   4.0877039   74.602446 94.433476  1.1682679  1.4788201
175 100 12.991418    7.4236676  159.37938  91.073933  2.6291998  1.5023999

 38 100  0.14669727  0.38604544  36.447609 95.91476   1.4056939  3.6991944
 58 100 11.076367   19.097184    49.790259 85.845274 ¯2.8666258 ¯4.9424583
 79 100  4.397554    5.566524    75.770714 95.912296 ¯1.1682679 ¯1.4788201
175 100 15.620618    8.9260675  162.00858  92.576332 ¯2.6291998 ¯1.5023999

 38 100  0.84954421  2.2356427   37.150456 97.764357  0          0
 58 100  9.6430539  16.625955    48.356946 83.374045  0          0
 79 100  3.8134201   4.827114    75.18658  95.172886  0          0
175 100 14.306018    8.1748675  160.69398  91.825132  0          0
```

Problem 8.5: Final Demand Decomposition

This problem explores further sector-specific and economy-wide structural decomposition with additional details for sectoral technology and final-demand decomposition of level, mix, and distribution.

Problem 8.5 Overview

Using the input-output economy specified in Problem 8.4, we first we assume that the final demand vectors can be specified with two components: $\mathbf{F}^0 = \begin{bmatrix} \mathbf{f}_1^0 & \mathbf{f}_2^0 \end{bmatrix} = \begin{bmatrix} 20 & 30 \\ 15 & 20 \\ 12 & 14 \end{bmatrix}$ and

$\mathbf{F}^1 = \begin{bmatrix} \mathbf{f}_1^1 & \mathbf{f}_2^1 \end{bmatrix} = \begin{bmatrix} 25 & 30 \\ 30 & 20 \\ 35 & 25 \end{bmatrix}$ (in both cases $\mathbf{f} = \mathbf{F}\mathbf{i}$).

The quantities $\mathbf{x}^0$, $\mathbf{x}^1$, $\mathbf{A}^0, \mathbf{A}^1$, $\mathbf{L}^0$, $\mathbf{L}^1$, $\Delta\mathbf{f}$, $\Delta\mathbf{x}$ and $\Delta\mathbf{L}$ were computed in Problem 8.4. From **F** we can now compute distribution across final demand categories [from (8.13)]:

$$\mathbf{d}^0 = \begin{bmatrix} 47/111 \\ 64/111 \end{bmatrix} = \begin{bmatrix} 0.4234 \\ 0.5766 \end{bmatrix} \quad \text{and} \quad \mathbf{d}^1 = \begin{bmatrix} 58/111 \\ 53/111 \end{bmatrix} = \begin{bmatrix} 0.5455 \\ 0.4545 \end{bmatrix}$$

The bridge matrices [from (8.14)] are found as

$$\mathbf{B}^0 = \begin{bmatrix} 20 & 30 \\ 15 & 20 \\ 12 & 14 \end{bmatrix} \begin{bmatrix} 1/47 & 0 \\ 0 & 1/64 \end{bmatrix} = \begin{bmatrix} 0.4255 & 0.4545 \\ 0.3191 & 0.3636 \\ 0.2553 & 0.1818 \end{bmatrix} \text{ and}$$

$$\mathbf{B}^1 = \begin{bmatrix} 25 & 30 \\ 30 & 20 \\ 35 & 25 \end{bmatrix} \begin{bmatrix} 1/90 & 0 \\ 0 & 1/75 \end{bmatrix} = \begin{bmatrix} 0.2778 & 0.4000 \\ 0.3333 & 0.2667 \\ 0.3889 & 0.3333 \end{bmatrix}$$

And changes are computed as

$$\Delta\mathbf{d} = \begin{bmatrix} .1220 \\ -.1220 \end{bmatrix}, \quad \Delta\mathbf{B} = \begin{bmatrix} -.1478 & -.0687 \\ .0142 & -.0458 \\ .1336 & .1146 \end{bmatrix} \text{ and } \quad \Delta f = 54$$

Equation (8.31) defines, for $\Delta\mathbf{x}$, both the final-demand decomposition (including distribution across final-demand categories) and the technology change decomposition in the same expression, now including all six of the change components, i.e., the three-sector specific technology change components, the final demand level component, the final demand mix component, and the final demand distribution component:

$$\begin{aligned}
\Delta\mathbf{x} &= (1/2)(\Delta\mathbf{L})(\mathbf{f}^0 + \mathbf{f}^1) + (1/2)(\mathbf{L}^0 + \mathbf{L}^1)(\Delta\mathbf{f}) \\
&= \underbrace{(1/2)[\mathbf{L}^1(\Delta\mathbf{A}^{(1)})\mathbf{L}^0](\mathbf{f}^0 + \mathbf{f}^1)}_{\text{Effect of technology change in sector 1}} + \underbrace{(1/2)[\mathbf{L}^1(\Delta\mathbf{A}^{(2)})\mathbf{L}^0](\mathbf{f}^0 + \mathbf{f}^1)}_{\text{Effect of technology change in sector 2}} \\
&\quad + \underbrace{(1/2)[\mathbf{L}^1(\Delta\mathbf{A}^{(3)})\mathbf{L}^0](\mathbf{f}^0 + \mathbf{f}^1)}_{\text{Effect of technology change in sector 3}} + \underbrace{(1/4)(\mathbf{L}^0 + \mathbf{L}^1)(\Delta f)(\mathbf{B}^0\mathbf{d}^0 + \mathbf{B}^1\mathbf{d}^1)}_{\text{Effect of change in final-demand level}} \\
&\quad + \underbrace{(1/4)(\mathbf{L}^0 + \mathbf{L}^1)[f^0(\Delta\mathbf{B})\mathbf{d}^1 + f^1(\Delta\mathbf{P})\mathbf{d}^0]}_{\text{Effect of change in final-demand mix}} + \underbrace{(1/4)(\mathbf{L}^0 + \mathbf{L}^1)(f^0\mathbf{B}^0 + f^1\mathbf{B}^1)(\Delta\mathbf{d})}_{\text{Effect of change in final-demand distribution}}
\end{aligned}$$

Application of equation (8.31) yields the various change contributions given in the table below.

	Output Change		Technology Change Contribution Sector 1	Sector 2	Sector 3	Total	Final Demand Change Contribution Level	Mix	Dist.	Total
Sector 1	38	Output	7.72	4.03	-10.90	0.85	51.97	-13.43	-1.39	37.15
		Percentage	20	11	-29	2	137	-35	-4	98
Sector 2	58	Output	7.43	3.52	-1.30	9.64	50.27	-2.59	0.67	48.36
		Percentage	13	6	-2	17	87	-4	1	83
Sector 3	79	Output	8.17	-9.27	4.91	3.81	61.79	12.67	0.73	75.19
		Percentage	10	-12	6	5	78	16	1	95
Total	175	Output	23.32	-1.72	-7.30	14.31	164.02	-3.34	0.01	160.69
		Percentage	13	-1	-4	8	94	-2	0	92

Computational Notes

We retrieve the transactions and final demands for the two periods of economy in Problem 8.4 as `Z0`, `f0`, `Z1`, and `f1` along with the final demand each divided into two components columns of the matrices `F0` and `F1`.

```
      Z0←3 3ρ12 15 35 24 11 30 36 50 8 ⋄ f0←50 35 26
      Z1←3 3ρ20 30 45 35 23 50 50 65 24 ⋄ f1←55 50 60
      F0←⍉2 3ρ20 15 12 30 20 14 ⋄ F1←⍉2 3ρ25 30 35 30 20 25

      Z0
12 15 35
24 11 30
36 50  8
      f0
50 35 26
      Z1
20 30 45
35 23 50
50 65 24
      f1
55 50 60
```

We compute the associated x0, A0, L0, x1, A1, and L1.

```
     L1←LINV A1←Z1 AMAT x1←f1++/Z1 ⋄ L0←LINV A0←Z0 AMAT x0←f0++/Z0

      x0
       112       100       120
      10 3⍕A0,L0
     0.107     0.150     0.292     1.491     0.604     0.628
     0.214     0.110     0.250     0.592     1.563     0.604
     0.321     0.500     0.067     0.831     1.045     1.611
      x1
       150       158       199
      10 3⍕A1,L1
     0.133     0.190     0.226     1.541     0.618     0.573
```

```
     0.233     0.146     0.251     0.687     1.633     0.643
     0.333     0.411     0.121     0.905     0.998     1.655
```

From F0 and F1, we compute the changes as df, dx, DA, DL, and the vectors of distribution of total final demand d0 and d1. We compute sums of final demands for each period, ft0 and ft1, and the totals of final demand components (column totals F0 and F1) as y0 and y1.

```
       df←f1-f0 ⋄ dx←x1-x0 ⋄ DL←L1-L0 ⋄ DA←A1-A0
       ft1←+/f1 ⋄ ft0←+/f0 ⋄ y1←+⌿F1 ⋄ y0←+⌿F0
       d0←(+⌿F0)÷+/+/F0 ⋄ d1←(+⌿F1)÷+/+/F1

       df
5 15 34
       dx
38 58 79
       DA
0.026190476  0.039873418 ¯0.065536013
0.019047619  0.03556962   0.0012562814
0.011904762 ¯0.088607595  0.053936348
       d0
0.42342342 0.57657658
       d1
0.54545455 0.45454545
       y0
47 64
       y1
90 75
       ft0
111
       ft1
165
       DA
0.026190476  0.039873418 ¯0.065536013
0.019047619  0.03556962   0.0012562814
0.011904762 ¯0.088607595  0.053936348
       DL
0.049653707  0.014216275 ¯0.05491784
0.094590793  0.070114224  0.039469367
0.074525847 ¯0.046981021  0.044128057
```

We compute the bridge matrices B0 and B1 and compute the differences between d0 and d1 as dd; differences between B0 and B1 as DB; and between ft1 and ft0 as dfs. The sum of all final demands (both periods) is tfs.

```
       B0←F0+.×DIAG ÷+⌿F0 ⋄ B1←F1+.×DIAG ÷+⌿F1
       dd←d1-d0 ⋄ DB←B1-B0 ⋄ dfs←ft1-ft0 ⋄ tfs←ft1+ft0

       B0
0.42553191 0.46875
0.31914894 0.3125
0.25531915 0.21875
       B1
0.27777778 0.4
```

```
0.33333333 0.26666667
0.38888889 0.33333333
       DB
¯0.14775414  ¯0.06875
 0.014184397 ¯0.045833333
 0.13356974   0.11458333
       dd
0.12203112 ¯0.12203112
       dfs
54
       tfs
276
```

Finally, we compute the final demand decomposition, DF1, DF2, and DF3, and the total output decomposition, DX1, DX2, and DX3, and compile the results in the variable DATA and expressed as percentages in DATAP shown in the table of results above.

```
      ⍝Final demand decompoosition from equation 8.31
       DF1←0.25×(L0+L1)+.×dfs×((B0+.×d0)+(B1+.×d1))
       DF2←0.25×(L0+L1)+.×((ft0×DB+.×d1)+(ft1×DB+.×d0))
       DF3←0.25×(L0+L1)+.×((ft0×B0)+(ft1×B1))+.×dd
      ⍝total output decompostion (equation 8.28 & 8.31)
       DA1←DA2←DA3←3 3⍴0
       DA1[;1]←DA[;1] ⋄ DA2[;2]←DA[;2] ⋄ DA3[;3]←DA[;3]
       DX1←0.5×(L1+.×DA1+.×L0)+.×(f0+f1)
       DX2←0.5×(L1+.×DA2+.×L0)+.×(f0+f1)
       DX3←0.5×(L1+.×DA3+.×L0)+.×(f0+f1)
      ⍝results
       DXALL←(⍉(3 3⍴DX1,DX2,DX3))
       DXALL←DXALL,3 1⍴+/DXALL
       DFALL←(⍉(3 3⍴DF1,DF2,DF3))
       DFALL←DFALL,3 1⍴+/DFALL
       DATA←(3 1⍴dx),DXALL,DFALL
       DATA←DATA,[1]+⌿DATA
      ⍝Percentages
       DATAP←100×DATA÷⍉9 4⍴DATA[;1]
```

```
       DATA
 38  7.7167494  4.0326763 ¯10.899882   0.84954421  51.9655   ¯13.428276  ¯1.3867677    37.150456
 58  7.4294337  3.5159897  ¯1.3023694  9.6430539   50.271104  ¯2.587164   0.67300641   48.356946
 79  8.1730017 ¯9.2665518   4.9069702  3.8134201   61.785777  12.673491   0.72731184   75.18658
175 23.319185  ¯1.7178859  ¯7.2952807 14.306018   164.02238   ¯3.3419487  0.013550537 160.69398
       DATAP
100 20.307235  10.612306   ¯28.683899   2.2356427 136.75131  ¯35.337569  ¯3.6493887    97.764357
100 12.809368   6.0620511   ¯2.2454645 16.625955   86.674317  ¯4.4606275  1.1603559    83.374045
100 10.345572 ¯11.729812     6.2113547  4.827114   78.209844  16.042394   0.92064789   95.172886
100 13.325248  ¯0.98164907  ¯4.1687318  8.1748675  93.727074  ¯1.9096849  0.0077431638 91.825132
```

Chapter 9, Nonsurvey and Partial-Survey Methods: Fundamentals

Chapter 9 introduces approaches designed to deal with a major challenge in input–output analysis that the kinds of information-gathering surveys needed to collect input–output data for an economy can be expensive and very time consuming, resulting in tables of input–output coefficients that are outdated before they are produced. These techniques, known as partial survey and nonsurvey approaches to input–output table construction, are central to modern applications of input–output analysis.

The chapter begins by reviewing the basic factors contributing to the stability of input–output data over time, such as changing technology, prices, and the scale and scope of business enterprises. Several techniques for updating input–output data are developed and the economic implications of each described. The bulk of the chapter is concerned with the widely utilized biproportional scaling (or RAS) technique and some related "hybrid model" variants. The exercise problems for this chapter explore various nonsurvey approaches to assembling input-output tables and measures for the accuracy of such tables.

Problem 9.1: Input-Output Tables in Constant Value Terms

This exercise explores the adjustment of input-output tables to express input-output relationships in constant value terms in prices of another point in time.

Problem 9.1 Overview

Using highly aggregated U.S. input-output tables for 1997[4], 2003 and 2005. The following are the make and use tables for these years all expressed in current year dollars.

US Use Matrix 1997	1	2	3	4	5	6	7	Imports
1 Agriculture	74,938	15	1,121	150,341	2,752	13,400	11	(23,123)
2 Mining	370	19,461	4,281	112,513	53,778	5,189	30	(64,216)
3 Construction	1,122	29	832	7,499	11,758	50,631	27	-
4 Manufacturing	49,806	19,275	178,903	1,362,660	169,915	418,412	1,914	(765,454)
5 Trade, Transport & Utilities	21,650	11,125	76,056	380,272	199,004	224,271	612	6,337
6 Services	32,941	45,234	107,723	483,686	545,779	1,592,426	3,801	(16,942)
7 Other	63	781	422	33,905	19,771	26,730	-	(126,350)
US Make Matrix 1997	1	2	3	4	5	6	7	Ind. Output
1 Agriculture	284,511	-	65	356	455	1,152	-	286,539
2 Mining	-	158,239	109	9,752	295	258	-	168,653
3 Construction	-	-	670,210	-	-	-	-	670,210
4 Manufacturing	-	727	1,258	3,703,275	39,720	36,034	3,669	3,784,683
5 Trade, Transport & Utilities	556	381	21,393	15,239	2,201,532	141,674	-	2,380,776
6 Services	-	410	54,850	1,306	109,292	6,444,098	1,821	6,611,778
7 Other	-	-	6,206	-	-	7,010	947,023	960,238
Total Commodity Output	285,067	159,757	754,091	3,729,928	2,351,295	6,630,226	952,513	14,862,876

[4] These tables differ from those provided in the supplemental resources for the text (described in Appendix B in the text) in that they reflect data assembled "before redefinitions" as discussed in Chapter 4.

US Use Matrix 2003	1	2	3	4	5	6	7	Imports
1 Agriculture	61,946	1	1,270	147,559	231	18,453	2,093	(26,769)
2 Mining	441	33,299	6,927	174,235	89,246	1,058	11,507	(125,508)
3 Construction	942	47	1,278	8,128	10,047	65,053	48,460	-
4 Manufacturing	47,511	22,931	265,115	1,249,629	132,673	516,730	226,689	(1,075,128)
5 Trade, Transport & Utilities	24,325	13,211	100,510	382,630	190,185	297,537	123,523	8,065
6 Services	25,765	42,276	147,876	509,084	490,982	2,587,543	442,674	(44,060)
7 Other	239	1,349	2,039	48,835	35,110	83,322	36,277	(177,578)

US Make Matrix 2003	1	2	3	4	5	6	7	Ind. Output
1 Agriculture	273,244	-	-	67	-	1,748	-	275,058
2 Mining	-	232,387	-	10,843	-	-	-	243,231
3 Construction	-	-	1,063,285	-	-	-	-	1,063,285
4 Manufacturing	-	-	-	3,856,583	-	30,555	3,278	3,890,416
5 Trade, Transport & Utilities	-	570	-	-	2,855,126	41	957	2,856,693
6 Services	-	475	-	-	133	9,136,001	3,278	9,139,886
7 Other	3,359	896	-	3,936	104,957	323,996	1,827,119	2,264,263
Total Commodity Output	276,602	234,328	1,063,285	3,871,429	2,960,216	9,492,341	1,834,631	19,732,832

US Use Matrix 2005	1	2	3	4	5	6	7	Imports
1 Agriculture	71,682	1	1,969	174,897	335	18,047	1,671	(31,248)
2 Mining	524	57,042	8,045	297,601	123,095	1,290	16,570	(226,059)
3 Construction	1,597	74	1,329	7,886	12,449	74,678	54,282	-
4 Manufacturing	61,461	34,860	339,047	1,452,738	183,135	589,452	255,456	(1,372,424)
5 Trade, Transport & Utilities	26,501	17,197	136,193	460,348	244,153	362,324	127,266	6,790
6 Services	27,274	52,297	165,179	543,690	610,978	3,017,728	529,779	(50,588)
7 Other	240	1,323	2,021	61,316	44,561	90,071	39,656	(208,971)

US Make Matrix 2005	1	2	3	4	5	6	7	Ind. Output
1 Agriculture	310,868	-	-	65	-	1,821	-	312,754
2 Mining	-	373,811	-	22,752	-	-	-	396,563
3 Construction	-	-	1,302,388	-	-	-	-	1,302,388
4 Manufacturing	-	-	-	4,454,957	-	26,106	4,467	4,485,529
5 Trade, Transport & Utilities	-	808	-	-	3,354,043	47	1,046	3,355,944
6 Services	-	556	-	-	152	10,473,161	3,771	10,477,640
7 Other	4,657	1,410	-	4,111	115,428	339,582	2,061,136	2,526,325
Total Commodity Output	315,525	376,586	1,302,388	4,481,885	3,469,622	10,840,717	2,070,419	22,857,143

First, as one variant, we produce industry-by-industry transactions tables using the assumption of industry-based technology for these three years. That is, for each year from the corresponding table if the make and use matrices are **V** and **U**, respectively, and the total industry and commodity outputs are **x** and **g**, respectively, we construct transactions tables in current dollar terms by computing $\mathbf{Z} = \mathbf{D}\mathbf{B}\hat{\mathbf{x}}$ where $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1}$ and $\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1}$:

Z(1997)	1	2	3	4	5	6	7
1	74,807	27	1,170	150,337	2,897	13,738	12
2	501	19,330	4,722	115,074	53,759	6,331	35
3	997	26	739	6,665	10,450	44,999	24
4	49,998	19,663	179,517	1,362,631	175,369	428,076	1,932
5	21,358	11,509	74,280	372,728	199,152	247,197	663
6	33,123	44,541	108,369	489,159	540,798	1,562,040	3,725
7	106	825	540	34,282	20,330	28,677	4

Z(2003)	1	2	3	4	5	6	7
1	61,199	9	1,286	145,882	321	18,714	2,153
2	570	33,088	7,612	176,292	88,879	2,496	12,046
3	942	47	1,278	8,128	10,047	65,053	48,460
4	47,412	22,981	264,578	1,246,562	133,807	523,227	227,310
5	23,463	12,824	96,960	369,498	183,671	287,032	119,187
6	24,800	40,759	142,346	490,430	472,802	2,490,571	426,150
7	2,782	3,406	10,953	83,306	58,947	182,603	55,918

Z(2005)	1	2	3	4	5	6	7
1	70,629	10	1,973	172,428	435	18,296	1,739
2	832	56,798	9,707	302,783	123,117	4,273	17,745
3	1,597	74	1,329	7,886	12,449	74,678	54,282
4	61,158	34,779	337,412	1,445,451	183,602	593,372	255,282
5	25,620	16,748	131,675	445,685	236,309	350,316	123,084
6	26,352	50,611	159,600	525,827	590,537	2,915,594	511,919
7	3,091	3,773	12,087	98,416	72,256	197,062	60,628

Suppose historical price indices for these tables are given in the following table (price indices in percent relative to some arbitrary earlier year):

	1997	2003	2005
Agriculture	100	113.5	122.7
Mining	96.6	131.3	201
Construction	181.6	188.9	209.9
Manufactuirng	133.7	150.8	156.9
Trade, Transport & Utilities	200.4	205.7	217.1
Services	129.3	151.6	219.8
Other	140	144.7	161.4

To generate price indices relative to the year 2005, the elements in each row of the historical price indices are divided by the last element in that row to yield the following table of relative price indices:

	1997	2003	2005
1 Agriculture	0.815	0.925	1
2 Mining	0.481	0.653	1
3 Construction	0.865	0.900	1
4 Manufacturing	0.852	0.961	1
5 Trade, Transport & Utilities	0.923	0.947	1
6 Services	0.588	0.690	1
7 Other	0.867	0.897	1

The constant price transactions tables expressed relative to 2005 dollars are then found as $\mathbf{Z}(1997)^{(2005)} = \hat{\mathbf{p}}_{1997}^{(2005)}\mathbf{Z}(1997)$ where $\hat{\mathbf{p}}_{1997}^{(2005)}$ is a matrix with the first column of the relative price table placed along the diagonal and zeros elsewhere. The matrix $\mathbf{Z}(2003)^{(2003)}$ is computed in the same manner, i.e., $\mathbf{Z}(2003)^{(2005)} = \hat{\mathbf{p}}_{2003}^{(2005)}\mathbf{Z}(2003)$ where $\hat{\mathbf{p}}_{2003}^{(2005)}$ is matrix of price indices converting 2003 to 2005 year prices, but $\mathbf{Z}(2005)^{(2005)} = \hat{\mathbf{p}}_{2005}^{(2005)}\mathbf{Z}(2005)$ is, of course, identical to the $\mathbf{Z}(2005)$ since 2005 is the base year of the price indices, i.e., $\hat{\mathbf{p}}_{2005}^{(2005)} = \mathbf{I}$:

$\mathbf{Z}(1997)^{2005}$	1	2	3	4	5	6	7
1	60,967	22	953	122,524	2,361	11,197	10
2	241	9,290	2,269	55,304	25,837	3,043	17
3	863	22	640	5,766	9,041	38,932	21
4	42,605	16,755	152,973	1,161,145	149,438	364,778	1,646
5	19,715	10,624	68,566	344,057	183,832	228,182	612
6	19,485	26,202	63,749	287,754	318,131	918,889	2,191
7	92	715	469	29,737	17,635	24,874	4

$\mathbf{Z}(2003)^{2005}$	1	2	3	4	5	6	7
1	56,611	8	1,190	134,944	297	17,311	1,991
2	372	21,614	4,973	115,160	58,059	1,631	7,869
3	847	42	1,150	7,315	9,042	58,544	43,612
4	45,568	22,088	254,292	1,198,098	128,605	502,885	218,472
5	22,231	12,150	91,869	350,096	174,027	271,960	112,928
6	17,105	28,112	98,179	338,258	326,100	1,717,792	293,923
7	2,494	3,053	9,820	74,687	52,848	163,709	50,132

Computational Notes

First, we define the price table `PP` and create the table of normalized prices `PI`. Then we can construct the price index matrices for the three years as `P1`, `P2`, and `P3`.

```
PP←100 113.5 122.7 96.6 131.3 201 181.6 188.9 209.9 133.7 150.8 156.9
PP←7 3⍴PP,200.4 205.7 217.1 129.3 151.6 219.8 140 144.7 161.4
PI←PP÷⍉3 7⍴PP[;3] ⋄ P1←⍉7 7⍴PI[;1] ⋄ P2←⍉7 7⍴PI[;2] ⋄ P3←⍉7 7⍴PI[;3]
```

```
      PP
 100.0 113.5 122.7
  96.6 131.3 201.0
```

```
 181.6 188.9 209.9
 133.7 150.8 156.9
 200.4 205.7 217.1
 129.3 151.6 219.8
 140.0 144.7 161.4

     PI
 0.815 0.925 1.000
 0.481 0.653 1.000
 0.865 0.900 1.000
 0.852 0.961 1.000
 0.923 0.947 1.000
 0.588 0.690 1.000
 0.867 0.897 1.000
```

We retrieve the three pairs of use and make matrices, saved as `U1`, `V1`, `U2`, `V2`, `U3`, and V3. From each **V**, we calculate **x** and **q**. We then calculate **B** and **D**, from which we can compute an industry-by-industry based **A** and **L**.

```
D1←V1 AMAT q1 ⋄ B1←U1 AMAT x1 ⋄ A1←D1+.×B1 ⋄ L1←LINV A1 ⋄ Z1←A1+.×DIAG x1
D2←V2 AMAT q2 ⋄ B2←U2 AMAT x2 ⋄ A2←D2+.×B2 ⋄ L2←LINV A2 ⋄ Z2←A2+.×DIAG x2
D3←V3 AMAT q3 ⋄ B3←U3 AMAT x3 ⋄ A3←D3+.×B3 ⋄ L3←LINV A3 ⋄ Z3←A3+.×DIAG x3
```

Now we can apply the price index matrices and construct the constant price transactions matrices `Z1C`, `Z2C`, and `Z3C` and total outputs vectors `x1c`, `x2c`, and `x3c`. Then we can calculate the direct requirements matrices, `A1C`, `A2C`, and `A3C`, and the total requirements matrices `L1C`, `L2C`, and `L3C`.

```
 Z1C←P1×Z1 ⋄ Z2C←P2×Z2 ⋄ Z3C←P3×Z3
 L1C←LINV A1C←Z1C AMAT x1c←x1×PI[;1]
 L2C←LINV A2C←Z2C AMAT x2c←x2×PI[;2]
 L3C←LINV A3C←Z3C AMAT x3c←x3×PI[;3]
```

```
Z1
   74806.8      26.5    1169.8  150336.5    2897.2   13738.5      11.7
     500.8   19329.9    4722.0  115074.4   53759.4    6331.4      34.8
     997.5      25.8     739.3    6664.8   10449.8   44999.0      24.1
   49998.3   19662.7  179516.9 1362630.7  175368.9  428075.8    1931.8
   21357.6   11509.1   74280.1  372727.9  199151.8  247197.1     662.7
   33122.7   44541.3  108368.6  489159.4  540797.7 1562040.5    3725.0
     106.2     824.6     540.2   34282.2   20330.4   28676.5       4.2
x1
  286538.7  168653.1  670209.7 3784683.0 2380775.6 6611777.7  960238.4
Z1C
   60967.2      21.6     953.4  122523.7    2361.2   11196.8       9.6
     240.7    9289.9    2269.4   55304.4   25836.6    3042.9      16.7
     863.0      22.3     639.6    5766.2    9040.9   38932.0      20.8
   42605.3   16755.2  152972.7 1161145.5  149438.0  364778.4    1646.1
   19714.7   10623.8   68566.2  344056.5  183832.4  228181.9     611.7
   19484.9   26201.9   63749.1  287753.9  318130.8  918889.2    2191.3
      92.1     715.2     468.6   29736.7   17634.8   24874.3       3.7
x1c
  233527.9   81054.2  579847.9 3225061.3 2197639.0 3889457.9  832920.5
Z2
   61199.4       8.7    1286.4  145881.9     320.9   18714.4    2152.7
```

```
    569.9   33087.5    7612.2  176291.6   88878.9    2496.1   12046.4
    941.6      46.8    1277.5    8128.1   10047.2   65052.7   48460.3
  47411.7   22981.3  264578.4 1246562.5  133807.0  523226.8  227309.7
  23462.6   12823.8   96960.1  369498.4  183671.2  287031.7  119186.9
  24799.9   40758.9  142346.5  490430.0  472802.5 2490571.3  426150.2
   2782.3    3405.7   10953.0   83306.4   58946.9  182602.5   55917.9
x2
 275057.9  243230.7 1063284.8 3890415.8 2856693.4 9139886.4 2264263.2
Z2C
  56610.7       8.0    1189.9  134943.8     296.8   17311.2    1991.3
    372.3   21613.9    4972.5  115159.6   58058.7    1630.6    7869.1
    847.4      42.1    1149.7    7314.9    9042.0   58544.3   43612.0
  45568.4   22087.8  254292.0 1198098.3  128604.8  502884.6  218472.3
  22230.5   12150.4   91868.7  350095.9  174026.5  271959.6  112928.4
  17104.9   28112.2   98178.9  338258.4  326100.3 1717791.6  293923.4
   2494.4    3053.3    9819.7   74686.7   52847.7  163708.7   50132.1
x2c
 254434.2  158886.5  956905.7 3739163.2 2706687.4 6303943.5 2029980.7

Z3
  70629.2       9.9    1973.0  172427.6     435.4   18295.9    1739.0
    831.9   56798.4    9706.8  302782.7  123117.3    4272.6   17744.7
   1596.9      74.3    1328.7    7885.9   12448.9   74678.0   54281.8
  61158.3   34779.1  337412.2 1445450.9  183601.9  593371.7  255282.2
  25619.8   16747.6  131674.9  445685.3  236308.7  350315.5  123083.9
  26351.8   50610.8  159600.1  525827.5  590536.8 2915593.7  511919.4
   3091.4    3773.0   12087.4   98415.7   72256.1  197062.2   60628.5
x3
 312753.9  396562.7 1302388.3 4485529.1 3355943.710477640.1 2526324.9
Z3C
  70629.2       9.9    1973.0  172427.6     435.4   18295.9    1739.0
    831.9   56798.4    9706.8  302782.7  123117.3    4272.6   17744.7
   1596.9      74.3    1328.7    7885.9   12448.9   74678.0   54281.8
  61158.3   34779.1  337412.2 1445450.9  183601.9  593371.7  255282.2
  25619.8   16747.6  131674.9  445685.3  236308.7  350315.5  123083.9
  26351.8   50610.8  159600.1  525827.5  590536.8 2915593.7  511919.4
   3091.4    3773.0   12087.4   98415.7   72256.1  197062.2   60628.5
x3c
 312753.9  396562.7 1302388.3 4485529.1 3355943.710477640.1 2526324.9

A2C,L2C
 0.261 0.000 0.002 0.038 0.001 0.003 0.000 1.378 0.029 0.029 0.087 0.011 0.017 0.000
 0.001 0.115 0.004 0.017 0.012 0.001 0.000 0.013 1.143 0.017 0.036 0.019 0.007 0.000
 0.004 0.000 0.001 0.002 0.004 0.010 0.000 0.009 0.008 1.005 0.007 0.007 0.015 0.000
 0.182 0.207 0.264 0.360 0.068 0.094 0.002 0.459 0.496 0.487 1.664 0.168 0.226 0.004
 0.084 0.131 0.118 0.107 0.084 0.059 0.001 0.199 0.263 0.208 0.225 1.131 0.118 0.002
 0.083 0.323 0.110 0.089 0.145 0.236 0.003 0.249 0.596 0.251 0.263 0.244 1.365 0.004
 0.000 0.009 0.001 0.009 0.008 0.006 0.000 0.008 0.021 0.009 0.019 0.012 0.012 1.000
A2C,L2C
 0.222 0.000 0.001 0.036 0.000 0.003 0.001 1.305 0.016 0.023 0.073 0.006 0.014 0.012
 0.001 0.136 0.005 0.031 0.021 0.000 0.004 0.021 1.172 0.026 0.060 0.032 0.010 0.015
 0.003 0.000 0.001 0.002 0.003 0.009 0.021 0.008 0.006 1.006 0.007 0.007 0.015 0.026
 0.179 0.139 0.266 0.320 0.048 0.080 0.108 0.393 0.305 0.449 1.556 0.117 0.193 0.218
 0.087 0.076 0.096 0.094 0.064 0.043 0.056 0.175 0.147 0.165 0.183 1.096 0.091 0.101
 0.067 0.177 0.103 0.090 0.120 0.272 0.145 0.211 0.358 0.240 0.255 0.211 1.428 0.259
 0.010 0.019 0.010 0.020 0.020 0.026 0.025 0.031 0.042 0.030 0.044 0.031 0.044 1.039
```

```
A3C,L3C
 0.226 0.000 0.002 0.038 0.000 0.002 0.001 1.314 0.010 0.025 0.079 0.008 0.010 0.012
 0.003 0.143 0.007 0.068 0.037 0.000 0.007 0.046 1.186 0.051 0.133 0.058 0.015 0.030
 0.005 0.000 0.001 0.002 0.004 0.007 0.021 0.011 0.003 1.005 0.007 0.007 0.012 0.026
 0.196 0.088 0.259 0.322 0.055 0.057 0.101 0.432 0.191 0.442 1.571 0.133 0.140 0.210
 0.082 0.042 0.101 0.099 0.070 0.033 0.049 0.177 0.086 0.174 0.197 1.107 0.071 0.095
 0.084 0.128 0.123 0.117 0.176 0.278 0.203 0.286 0.270 0.305 0.351 0.314 1.441 0.360
 0.010 0.010 0.009 0.022 0.022 0.019 0.024 0.033 0.023 0.030 0.049 0.034 0.033 1.039
```

Problem 9.2: Measuring Year-to-Year Changes in Technical Coefficients

This exercise explores measurement of year-to-year changes in technical coefficients of an input-output model as the average of the absolute value of differences between the column sums of **A** for the same industry sectors in two different years.

Problem 9.2 Overview

Using the series of transactions tables developed in exercise Problem 9.1, we arbitrarily pick the years 1997 and 2005 with 2005 assumed to be the base year. First, we must compute the technical coefficients matrices for 2005 and 1997 expressed in current year prices, $\mathbf{A}(2005) = \mathbf{Z}(2005)\hat{\mathbf{x}}(2005)^{-1}$ and $\mathbf{A}(1997) = \mathbf{Z}(1997)\hat{\mathbf{x}}(1997)^{-1}$, as well as the technical coefficient matrix for 1997 expressed in 2005 (the base year) prices: $\mathbf{A}(1997)^{(2005)} = \mathbf{Z}(1997)^{(2005)}\left[\hat{\mathbf{x}}(1997)^{(2005)}\right]^{-1}$:

A (1997)	1	2	3	4	5	6	7
1	0.2611	0.0002	0.0017	0.0397	0.0012	0.0021	0.0000
2	0.0017	0.1146	0.0070	0.0304	0.0226	0.0010	0.0000
3	0.0035	0.0002	0.0011	0.0018	0.0044	0.0068	0.0000
4	0.1745	0.1166	0.2679	0.3600	0.0737	0.0647	0.0020
5	0.0745	0.0682	0.1108	0.0985	0.0836	0.0374	0.0007
6	0.1156	0.2641	0.1617	0.1292	0.2272	0.2363	0.0039
7	0.0004	0.0049	0.0008	0.0091	0.0085	0.0043	0.0000

A(2005)	1	2	3	4	5	6	7
1	0.2258	0.0000	0.0015	0.0384	0.0001	0.0017	0.0007
2	0.0027	0.1432	0.0075	0.0675	0.0367	0.0004	0.0070
3	0.0051	0.0002	0.0010	0.0018	0.0037	0.0071	0.0215
4	0.1955	0.0877	0.2591	0.3222	0.0547	0.0566	0.1010
5	0.0819	0.0422	0.1011	0.0994	0.0704	0.0334	0.0487
6	0.0843	0.1276	0.1225	0.1172	0.1760	0.2783	0.2026
7	0.0099	0.0095	0.0093	0.0219	0.0215	0.0188	0.0240

$A(1997)^{(2005)}$	1	2	3	4	5	6	7
1	0.2611	0.0003	0.0016	0.0380	0.0011	0.0029	0.0000
2	0.0010	0.1146	0.0039	0.0171	0.0118	0.0008	0.0000
3	0.0037	0.0003	0.0011	0.0018	0.0041	0.0100	0.0000
4	0.1824	0.2067	0.2638	0.3600	0.0680	0.0938	0.0020
5	0.0844	0.1311	0.1182	0.1067	0.0836	0.0587	0.0007
6	0.0834	0.3233	0.1099	0.0892	0.1448	0.2363	0.0026
7	0.0004	0.0088	0.0008	0.0092	0.0080	0.0064	0.0000

Then, using the constant price technical coefficient tables, i.e., $\mathbf{A}(2005)$ and $\mathbf{A}(1997)^{(2005)}$, we compute the average of the absolute value of differences between the column sums of $\mathbf{A}$ for each industry:

$$\frac{1}{7}[\mathbf{i}'|\mathbf{A}(2005)-\mathbf{A}(1997)^{(2005)}|]=\begin{bmatrix}.009 & .062 & .007 & .02 & .014 & .017 & .057\end{bmatrix}.$$

The most changed sectors in decreasing order are 2, 7 and 4. If we, instead, compare the current price tables, $\mathbf{A}(2005)$ and $\mathbf{A}(1997)$, these values are:

$$\frac{1}{7}[\mathbf{i}'|\mathbf{A}(2005)-\mathbf{A}(1997)|]=\begin{bmatrix}.015 & .032 & .01 & .015 & .016 & .01 & .057\end{bmatrix}.$$

In this case, the most changed sectors in decreasing order are 7, 2 and 5. The differences in rates of inflation explain the difference between the comparisons in constant and current year prices.

Computational Notes

For this problem, using the tables compiled in Problem 9.1, we compute the mean absolute differences between the original and constant price technical coefficients `mad` and `madc`.

```
      madc←(+⌿|A3C-A1C)÷7
      mad←(+⌿|A3-A1)÷7

      madc
0.0091716325 0.061875254 0.006673803 0.019524133 0.013931034 0.017315044 0.057171175
      mad
0.015295436 0.03211081 0.0095505127 0.014564504 0.016033482 0.0099650962 0.056991671
```

Problem 9.3: Marginal Technical Coefficients

This exercise problem illustrates the computation of marginal technical coefficients.

Problem 9.3 Overview

Using the current price transactions tables developed in Problem 9.1 between the years 1997 and 2005, the matrix of marginal input coefficients is computed as

$$[\mathbf{Z}(2005)-\mathbf{Z}(1997)][\langle\mathbf{x}(2005)-\mathbf{x}(1997)\rangle]^{-1}.$$

	1	2	3	4	5	6	7
1	-0.1594	-0.0001	0.0013	0.0315	-0.0025	0.0012	0.0011
2	0.0126	0.1644	0.0079	0.2678	0.0711	-0.0005	0.0113
3	0.0229	0.0002	0.0009	0.0017	0.0020	0.0077	0.0346
4	0.4257	0.0663	0.2498	0.1182	0.0084	0.0428	0.1618
5	0.1626	0.0230	0.0908	0.1041	0.0381	0.0267	0.0782
6	-0.2583	0.0266	0.0810	0.0523	0.0510	0.3501	0.3245
7	0.1139	0.0129	0.0183	0.0915	0.0532	0.0436	0.0387

Note that these marginal coefficients deal with changes, so negative entries can appear and do in this case in industries 1, 2, 5, and 6.

Computational Notes

Using the Problem 9.1 data again, we compute marginal input coefficients `AM` based on the difference in interindustry transactions `Z31` and in total outputs `x31`.

```
      AM←(Z31←Z3-Z1) AMAT x31←x3-x1

      Z3-Z1
¯4177.6     ¯16.6      803.2   22091.1    ¯2461.7     4557.4    1727.3
  331.2   37468.5     4984.8  187708.3    69357.9    ¯2058.8   17709.9
  599.4      48.5      589.4    1221.1     1999.1    29679.0   54257.7
11160.1   15116.5   157895.3   82820.2     8233.1   165295.9  253350.4
 4262.2    5238.5    57394.8   72957.4    37156.9   103118.5  122421.2
¯6770.9    6069.5    51231.5   36668.1    49739.0  1353553.2  508194.5
 2985.2    2948.4    11547.2   64133.5    51925.6   168385.7   60624.3
      x3-x1
26215.2  227909.6   632178.6  700846.1   975168.1  3865862.4 1566086.5
      AM
 ¯0.159     0.000      0.001     0.032     ¯0.003      0.001     0.001
  0.013     0.164      0.008     0.268      0.071     ¯0.001     0.011
  0.023     0.000      0.001     0.002      0.002      0.008     0.035
  0.426     0.066      0.250     0.118      0.008      0.043     0.162
  0.163     0.023      0.091     0.104      0.038      0.027     0.078
 ¯0.258     0.027      0.081     0.052      0.051      0.350     0.324
  0.114     0.013      0.018     0.092      0.053      0.044     0.039
```

Problem 9.4: Biproportional Scaling (The RAS Technique)

This exercise problem illustrates use of the so-called RAS technique of biproportional scaling to generate an estimate of a future technical coefficients table for an economy based on a previous year's table and future estimates of the vectors for total final demand, total value-added, and total output.

Problem 9.4 Overview

Consider the following interindustry transactions and total outputs two-sector input-output economy for the year 2020:

2020	A	B	Total Output
A	1	2	10
B	3	4	10

Estimates for the year 2030 for the vectors of total final demand, total value-added, and total output are the following:

2030	Final Demand	Value Added	Total Output
A	12	10	25
B	6	8	20

To use the 2020 table as a base and the 2030 projections for final demand, value-added and total output in computing an estimate of the 2030 technical coefficients table first, from the matrix of interindustry transactions, $\mathbf{Z}(0)=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$, and vector of total outputs, $\mathbf{x}(0)=\begin{bmatrix}10\\ 10\end{bmatrix}$, we can compute $\mathbf{A}(0)=\mathbf{Z}(0)\hat{\mathbf{x}}(0)^{-1}=\begin{bmatrix}.1 & .2\\ .3 & .4\end{bmatrix}$. Finally, we can compute the vectors of intermediate outputs and inputs, respectively, as $\mathbf{u}(1)=\mathbf{x}(1)-\mathbf{f}(1)=\begin{bmatrix}25\\ 20\end{bmatrix}-\begin{bmatrix}12\\ 6\end{bmatrix}=\begin{bmatrix}13\\ 14\end{bmatrix}$ and $\mathbf{v}(1)=\mathbf{x}(1)-\mathbf{va}(1)=\begin{bmatrix}25\\ 20\end{bmatrix}-\begin{bmatrix}10\\ 8\end{bmatrix}=\begin{bmatrix}15\\ 12\end{bmatrix}$. [Here we use **va**(1) for the value added vector in 2030 to differentiate it from **v**(1), the total intermediate inputs vector in 2030.] Performing the RAS procedure using **A**(0), **u**(1), **v**(1) and **x**(1) converges to the 2030 matrix in 6 iterations. That is, the result is $\tilde{\mathbf{A}}(1)=\begin{bmatrix}.262 & .323\\ .338 & .277\end{bmatrix}$, such that $\mathbf{u}(1)=\tilde{\mathbf{A}}(1)\mathbf{x}(1)$ and $\mathbf{v}(1)=\mathbf{i}'\tilde{\mathbf{A}}(1)\hat{\mathbf{x}}(1)$ within .0001 for each element of the intermediate inputs and outputs vectors, **u**(1) and **v**(1), respectively.

Computational Notes

We define the base matrix of transactions `Z0` and vector of total outputs `x0` from which we can calculate the base matrix of technical coefficients `A0`. We also define the target vectors of final demands `f1`, of value-added inputs `w1`, and total outputs `x1` from which we can calculate the target vectors of intermediate outputs `u1` and of intermediate inputs `v1`.

```
Z0←2 2ρ1 2 3 4 ⋄ x0←10 10
A0←Z0 AMAT x0

f1←12 6 ⋄ w1←10 8 ⋄ x1←25 20
u1←x1-f1 ⋄ v1←x1-w1
```

```
Z0
         1         2
         3         4
x0
        10        10
A0
     0.100     0.200
     0.300     0.400
f1
        12         6
w1
        10         8
x1
        25        20
u1
        13        14
v1
        15        12
```

For this problem we need an APL function `RAS` to implement the RAS algorithm.

```
[  0]  A←A0 RAS UVX;tol;lim;n;nn;k;XD;u1;v1;x;r;s
[  1] ⍝Basic function for RAS biproportional scaling
[  2] ⍝INPUT: A0 and rows of UVX are u1 v1 x
[  3]  nn←2⍴n←1↑⍴A0 ⋄ tol←0.001 ⋄ lim←100
[  4]  A←A0 ⋄ u1←UVX[1;] ⋄ v1←UVX[2;] ⋄ x←UVX[3;] ⋄ XD←(nn⍴x)×nn⍴1,n⍴0
[  5]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←0
[  6]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[  7] ⍝----ROW ADJUSTMENT
[  8] LOOP:A←A×⍉nn⍴r←u1÷u
[  9]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←k+1
[ 10]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[ 11] ⍝----COL ADJUSTMENT
[ 12]  A←A×nn⍴s←v1÷v
[ 13]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←k+1
[ 14]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[ 15]  →(lim>k)/LOOP
[ 16]  →0,0⍴⎕←'**** STOPPED: ',(⍕k),' ITERATIONS ****'
[ 17] CON:→0
```

Note several new APL features in this function. First, assigning values to the character ⎕ (called Quad) in and APL statement enables delivery of immediate output of an array from the function as it is executed, such as an intermediate result. For example,

```
⎕←3 4 5
3 4 5
      ⎕←'This is the end'
This is the end

     (⎕←1 2 3)+⎕←4 5 6
4 5 6
1 2 3
5 7 9
```

In the `RAS` function it is used to issue a message indicating that the biproportional scaling iterations did not converge within a specified number of iterations if that condition materializes.

Also new is the dyadic logical function *and* (denoted with the symbol `∧`) which returns the value 1 as the explicit result if two logical statements as arguments are both true. As an aside, the function *or* (denoted by `∨`) returns the value 1 as the explicit result if either of the two logical statements as arguments is true (or both are true). For example,

```
      (2=5)∧(1=1)
0
      (5=5)∧(1=1)
1
```

In the RAS function, the and function is used to determine if in the current iteration of biproportionately scaling the technical coefficients matrix produces values for the differences between the current values of all intermediate inputs and outputs are within a specified tolerance level.

We use the `RAS` function by providing `A0` as the left argument and a matrix where `u1` is the first row, `v1` is the second, and `x1` is the third with the explicit result as the biproportionately scaled **A** within a default tolerance of .001 and if the iterative process is unable to meet the specified tolerance within 100 iterations the process is halted and the last value of the matrix **A** is returned as the explicit result. For the problem,

```
      A←A0 RAS 3 2⍴u1,v1,x1

      A   (RAS estimate)
      0.262     0.323
      0.338     0.277
```

In this case the RAS algorithm converges in 6 iterations.

<u>Problem 9.5: Measurement of Error in Nonsurvey Estimation</u>

This exercise explores measurement of error between an RAS-estimated table of technical coefficients and a "real" table with the mean absolute percentage error (MAPE) of the element-by-element comparison of the two tables as the error metric.

Problem 9.5 Overview

Recall the 1997 input-output table expressed in 1997 dollars constructed in Problem 9.1 and the vectors of intermediate inputs, intermediate outputs, and total outputs from the corresponding input-output table for 2005. The 1997 input-output table, $\mathbf{A}(1997) = \mathbf{Z}(1997)\mathbf{x}(1997)^{-1}$ was computed in exercise Problem 9.2. We can retrieve the year 2005 total outputs, $\mathbf{x}(2005)$, from exercise Problem 9.1 and compute the year 2005 intermediate outputs, $\mathbf{u}(2005) = \mathbf{Z}(2005)\mathbf{i}$, and intermediate inputs, $\mathbf{v}(2005) = \mathbf{i}'\mathbf{Z}(2005)$, all given in the following table:

	1	2	3	4	5	6	7
u(2005)'	265,510	515,254	152,295	2,911,056	1,329,436	4,780,440	447,314
v(2005)'	189,279	162,793	653,783	2,998,476	1,218,705	4,153,590	1,024,680
x(2005)	312,754	396,563	1,302,388	4,485,529	3,355,944	10,477,640	2,526,325

Performing the RAS procedure using **A**(1997), **u**(2005), **v**(2005) and **x**(2005), yields the RAS-estimate of **A**(2005), which we designate as $\tilde{\mathbf{A}}(2005)$, given in the following table:

$\tilde{\mathbf{A}}(2005)$	1	2	3	4	5	6	7
1	0.2448	0.0001	0.0015	0.0357	0.0009	0.0021	0.0007
2	0.0037	0.1423	0.0140	0.0622	0.0373	0.0022	0.0048
3	0.0052	0.0001	0.0015	0.0025	0.0050	0.0109	0.0023
4	0.1592	0.0618	0.2274	0.3148	0.0519	0.0638	0.1129
5	0.0737	0.0392	0.1020	0.0933	0.0639	0.0400	0.0420
6	0.1172	0.1557	0.1525	0.1256	0.1780	0.2588	0.2419
7	0.0015	0.0112	0.0030	0.0343	0.0261	0.0185	0.0011

For the 2005 "real" input-output table, $\mathbf{A}(2005) = \mathbf{Z}(2005)\mathbf{x}(2005)^{-1}$ (also derived in exercise Problem 9.2), since there are a total of $7 \times 7 = 49$ elements to compare, the MAPE is computed as $(\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7} 100 \times \left[\frac{\left|\tilde{a}_{ij}(2005) - a_{ij}(2005)\right|}{a_{ij}(2005)}\right] = 49.028$ [for $a_{ij}(2005) \neq 0$ and 0 otherwise].

Note that in this case the RAS estimate is very weak since the average error is nearly 50 percent.

Computational Notes

We retrieve the technical coefficients matrix `A1` from Problem 9.1 along with the target table's `u3`, `v3`, and `x3`.

```
u3
   265510.0  515254.5  152294.5 2911056.4 1329435.7 4780440.1  447314.2
v3
   189279.4  162793.1  653783.1 2998475.5 1218705.1 4153589.6 1024679.6
x3
   312753.9  396562.7 1302388.3 4485529.1 3355943.710477640.1 2526324.9
A1
      0.261     0.000     0.002     0.040     0.001     0.002     0.000
      0.002     0.115     0.007     0.030     0.023     0.001     0.000
      0.003     0.000     0.001     0.002     0.004     0.007     0.000
      0.174     0.117     0.268     0.360     0.074     0.065     0.002
      0.075     0.068     0.111     0.098     0.084     0.037     0.001
      0.116     0.264     0.162     0.129     0.227     0.236     0.004
      0.000     0.005     0.001     0.009     0.009     0.004     0.000
```

We now use the function `RAS` with `A1` as the left argument and a matrix with `u3`, `v3`, and `x3` as the rows to yield `AR` and calculate the mean average percentage error, mape, between `AR` and `A3` which we also retrieve from Problem 9.1.

```
AR←A1 RAS 3 7ρu3,v3,x3
mape←(÷49)×+/+/100×(|A3-AR)÷A3
```

```
AR
      0.245      0.000      0.002      0.036      0.001      0.002      0.001
      0.004      0.142      0.014      0.062      0.037      0.002      0.005
      0.005      0.000      0.002      0.003      0.005      0.011      0.002
      0.159      0.062      0.227      0.315      0.052      0.064      0.113
      0.074      0.039      0.102      0.093      0.064      0.040      0.042
      0.117      0.156      0.153      0.126      0.178      0.259      0.242
      0.001      0.011      0.003      0.034      0.026      0.019      0.001
mape
49.027628
```

Problem 9.6: RAS Estimation of Interindustry Transactions

This exercise demonstrates an example of the equivalence of performing an RAS-estimate using either interindustry transactions or technical coefficients.

Problem 9.6 Overview

Suppose we have a baseline transactions matrix defined as $\mathbf{Z}(0)=\begin{bmatrix}100 & 55 & 5\\ 50 & 75 & 45\\ 25 & 10 & 110\end{bmatrix}$. We are provided with estimates of intermediate inputs and outputs, $\mathbf{v}(1)=\begin{bmatrix}265\\ 225\\ 325\end{bmatrix}$ and $\mathbf{u}(1)=\begin{bmatrix}325\\ 235\\ 255\end{bmatrix}$, respectively.

To compute an estimate of the transactions table for the next year, $\tilde{\mathbf{Z}}^Z(1)$, if **Z**(0), **v**(1) and **u**(1) are known, we use the RAS technique to biproportionately scale **Z**(0) iteratively to convergence of $\mathbf{u}(1)=\tilde{\mathbf{Z}}^Z(1)\mathbf{i}$ and $\mathbf{v}(1)=\mathbf{i}'\tilde{\mathbf{Z}}^Z(1)$ within .0001 for each element of **u**(1) and **v**(1) to yield: $\tilde{\mathbf{Z}}^Z(1)=\begin{bmatrix}167.5 & 104.5 & 53\\ 61.2 & 104.1 & 69.7\\ 36.3 & 16.5 & 202.2\end{bmatrix}$.

Alternatively, suppose we know the vector of total outputs, $\mathbf{x}(0)=\begin{bmatrix}750\\ 500\\ 1{,}000\end{bmatrix}$, corresponding to $\mathbf{Z}(0)$, and we also have an estimate of total outputs for next year, $\mathbf{x}(1)=\begin{bmatrix}1{,}000\\ 750\\ 1{,}500\end{bmatrix}$. Compute

$\mathbf{A}(0) = \mathbf{Z}(0)\hat{\mathbf{x}}(0) = \begin{bmatrix} .133 & .11 & .025 \\ .067 & .15 & .045 \\ .033 & .02 & .11 \end{bmatrix}$ and use it [rather than $\mathbf{Z}(0)$] along with $\mathbf{v}(1)$ and $\mathbf{u}(1)$ to generate an estimate of the technical coefficients matrix for next year using the RAS technique, we find $\tilde{\mathbf{A}}^A(1) = \begin{bmatrix} .168 & .139 & .035 \\ .061 & .139 & .047 \\ .036 & .022 & .135 \end{bmatrix}$. If we also compute the matrix of technical coefficients matrix from the $\tilde{\mathbf{Z}}^Z(1)$ and $\hat{\mathbf{x}}(1)^{-1}$, we find that $\tilde{\mathbf{A}}^Z(1) = \tilde{\mathbf{Z}}^Z(1)\hat{\mathbf{x}}(1)^{-1} = \begin{bmatrix} .168 & .139 & .035 \\ .061 & .139 & .047 \\ .036 & .022 & .135 \end{bmatrix}$, which is identical to $\tilde{\mathbf{A}}^Z(1)$. The explanation for why this is true generally is discussed in Section 9.4.3 of the text.

Computational Notes

We define, for two regions, matrices of interindustry transactions, `Z0` and `Z1`, and vectors of total outputs, `x0` and `x1`. We compute the intermediate outputs and inputs for each region, `u1`, `v1`, `u2`, and `v2`, as well as the matrices of technical coefficients `A0` and `A1`.

```
      Z0←3 3ρ100 55 25 50 75 45 25 10 110
      Z1←3 3ρ200 75 50 35 125 75 30 25 200
      x0←750 500 1000 ⋄ x1←1000 750 1500
      u1←+/Z1 ⋄ v1←+⌿Z1 ⋄ u0←+/Z0 ⋄ v0←+⌿Z0
      A0←Z0 AMAT x0 ⋄ A1←Z1 AMAT x1
      Z0
100 55  25
 50 75  45
 25 10 110
      u0
180 170 145
      v0
175 140 180
      x0
750 500 1000
      Z1
200  75  50
 35 125  75
 30  25 200
      u1
325 235 255
      v1
265 225 325
      x1
1000 750 1500
```

```
 A0
0.133     0.110     0.025
0.067     0.150     0.045
0.033     0.020     0.110
 A1
0.200     0.100     0.033
0.035     0.167     0.050
0.030     0.033     0.133
```

We first use RAS to produce an estimate, which we call AR, of A3 using A1 as the base matrix.

```
 AR←A0 RAS 3 3ρu1,v1,x1

 AR
0.168     0.139     0.035
0.061     0.139     0.046
0.036     0.022     0.135
```

Next, we adapt the RAS algorithm to modify transactions rather than technical coefficients in the APL function RAST.

```
[  0]  Z←Z0 RAST UV;tol;n;k;XD;test;u1;v1;r;s
[  1] ⍝Basic function for RAS biproportional scaling
[  2] ⍝of Z rather than A
[  3] ⍝INPUT: Z0 and rows of UV are u1 v1
[  4]  nn←2ρn←1↑ρZ0 ⋄ tol←0.001 ⋄ lim←500000
[  5]  test←'((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol)'
[  6]  Z←Z0 ⋄ u1←UV[1;] ⋄ v1←UV[2;]
[  7] ⍝BEGIN ITERATION
[  8]  u←+/Z ⋄ v←+⌿Z ⋄ k←0
[  9]  →CON×⍳1=⍎test
[ 10] ⍝----ROW ADJUSTMENT
[ 11] LOOP:Z←Z×⍉nnρr←u1÷u
[ 12]  u←+/Z ⋄ v←+⌿Z ⋄ k←k+1
[ 13]  →CON×⍳1=⍎test
[ 14] ⍝----COL ADJUSTMENT
[ 15]  Z←Z×nnρs←v1÷v
[ 16]  u←+/Z ⋄ v←+⌿Z ⋄ k←k+1
[ 17]  →CON×⍳1=⍎test
[ 18]  →LOOP×⍳lim>k
[ 19]  →0,0ρ⎕←'**** STOPPED: ',(⍕k),' ITERATIONS ****'
[ 20] CON:→0
```

We apply RAST to produce an estimate of Z3, which we call ZR, using Z1 as a base matrix and generate the corresponding technical coefficients matrix AZR.

```
     ZR←Z0 RAST 2 3ρu1,v1
     AZR←ZR AMAT x1
ZR
     167.5     104.5      53.0
      61.2     104.1      69.7
      36.3      16.5     202.2
AZR
     0.168     0.139     0.035
```

```
0.061     0.139     0.046
0.036     0.022     0.135
```

Finally, note another new APL function in `RAST`, used in lines 9, 13, 17, and 19, which are all conditional branching statement. The new monadic function is called *index generator* denoted with the character `⍳` (iota). The function takes as its argument a positive integer and returns as its explicit result a vector with from the index origin (which by default is zero but can be changed to 1) incremented by 1 until the number specified by the right argument. For example,

```
      ⍳3
1 2 3
      ⍳9
1 2 3 4 5 6 7 8 9
```

The one exception is if the right argument is 0, the function returns an empty vector. The function's use here takes advantage of how operations with an empty vector work, the most important feature of which is an empty vector times any value results in another empty vector. This means, for example, if an expression is a scalar value multiplied by the result of the index generator function (`⍳`) with the argument of a logical condition expression is true or 1, the result is the scalar value. That is, if the logical condition is false or 0, the result is the empty vector. So, in an APL function a branch condition `→LOOP×⍳lim>k` would branch to the line denoted by the label `LOOP` if the current value of the variable `lim` is larger than the current value of the variable `k`, but would just proceed to the next line of the function otherwise. The method of branching is actually not used frequently in modern APL implementations since it is dependent upon the value of the index origin, but the index generator has many other uses.

Problem 9.7: Incorporation of Partial Information into RAS Estimation

This exercise problem explores the prospects using of partial information about a target technical coefficients matrix to improve an RAS-estimated technical coefficients table compared with estimation absent such information.

Problem 9.7 Overview

For the economy in Problem 9.6, suppose we acquire a survey-based table of technical coefficients for next year of $\mathbf{A}(1) = \begin{bmatrix} .2 & .1 & .033 \\ .035 & .167 & .05 \\ .03 & .033 & .133 \end{bmatrix}$, which we consider to be the "real" target technical coefficients matrix since it is based on more comprehensive information.

At the beginning of the survey, however, suppose we know only $a(1)_{32} = .033$ of the nine survey-based coefficients and we use that value along with $\mathbf{A}(0)$, $\mathbf{v}(1)$ and $\mathbf{u}(1)$ to generate an intermediate estimate of the entire matrix of coefficients, $\tilde{\underset{\sim}{\mathbf{A}}}(1)$. To so this we first define the matrix of known coefficients for the target table as $\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & .033 & 0 \end{bmatrix}$ and the reference table,

$\bar{\mathbf{A}}(0) = \begin{bmatrix} .133 & .11 & .025 \\ .067 & .15 & .047 \\ .033 & 0 & .110 \end{bmatrix}$, where $a(0)_{32}$ (the location of the known coefficient) is set to 0. We must also revise $\mathbf{u}(1)$ and $\mathbf{v}(1)$ to reflect removal of the interindustry transaction associated with the know information, which we can compute as

$\bar{\mathbf{u}}(1) = \mathbf{u}(1) - \mathbf{Kx}(1) = \begin{bmatrix} 325 \\ 235 \\ 255 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 24.75 \end{bmatrix} = \begin{bmatrix} 325 \\ 235 \\ 230.25 \end{bmatrix}$ and

$\bar{\mathbf{v}}(1) = \bar{\mathbf{v}}(1) - \mathbf{iK\hat{x}}(1) = \begin{bmatrix} 265 \\ 225 \\ 325 \end{bmatrix} - \begin{bmatrix} 0 \\ 24.75 \\ 0 \end{bmatrix} = \begin{bmatrix} 265 \\ 200.25 \\ 325 \end{bmatrix}$.

The “intermediate estimate,” $\underset{\sim}{\tilde{\mathbf{A}}}(1)$, for this case, is then found by adding $\mathbf{K}$ to the result of applying the RAS procedure using $\bar{\mathbf{A}}(1), \bar{\mathbf{u}}(1), \bar{\mathbf{v}}(1)$, and $\mathbf{x}(1)$, to yield

$\underset{\sim}{\tilde{\mathbf{A}}}(1) = \begin{bmatrix} .169 & .134 & .037 \\ .062 & .133 & .049 \\ .034 & .033 & .131 \end{bmatrix}$, including the known value for $a(1)_{32}$. The MAPE for the RAS estimate, $\tilde{\mathbf{A}}(1)$, which excludes the additional information about $a(1)_{32}$, compared with the known $\mathbf{A}(1)$ is 24.05. The MAPE for the modified RAS estimate, $\tilde{\mathbf{A}}(1)$ (including the known coefficient), is 19.5, which we record for this case as $\underset{\sim}{\tilde{\mathbf{A}}}(1)^{(case1)} = 19.5$. The MAPE value is lower so the estimate with the additional information is better.

For a second case, we assume instead that we know only $a(1)_{33} = 0.133$, i.e., instead of $a(1)_{32} = 0.033$. If we apply the same procedure to determine $\tilde{\mathbf{A}}(1)$, we find that the MAPE for the modified RAS estimate, $\underset{\sim}{\tilde{\mathbf{A}}}(1)$, including the alternative known coefficient, is $\underset{\sim}{\tilde{\mathbf{A}}}(1)^{(case2)} = 24.18$. Recall that the MAPE of the estimate without additional information is 24.05, which is lower than that of the modified estimate in this case, so the estimate without additional information is better in this case. In general, introduction of more accurate exogenous information in applying RAS improves the resulting estimates, but it is not always the case as discussed in Section 9.4.6 of the text.

Computational Notes

We retrieve from Problem 9.6 the two technical coefficients matrices, `A0` and `A1`, along with the associated values of `u1`, `u2`, and `x1`. We create an RAS estimate `AR` of `A1` by using `A0` as the left argument of `RAS` and the values of `u1`, `v1`, and `x1` as the rows of the right argument. Calculate the mean absolute percentage error comparing `A1` with `AR`.

```
      AR←A0 RAS 3 3ρu1,v1,x1
      mape1←(÷9)×+/+/100×(|AR-A1)÷A1
A0
    0.1333    0.1100    0.0250
    0.0667    0.1500    0.0450
    0.0333    0.0200    0.1100
A1
    0.2000    0.1000    0.0333
    0.0350    0.1667    0.0500
    0.0300    0.0333    0.1333
u1
       325       235       255
v1
       265       225       325
x1
      1000       750      1500
AR
    0.1675    0.1393    0.0354
    0.0612    0.1388    0.0465
    0.0363    0.0220    0.1348

mape1
24.048867
```

We temporarily create `A0B` as a copy of `A0` but replace the location of the known coefficient by 0 and create **K** as a matrix of zeroes except for the location of the known information (we label the two cases of **K** as `K32` and `K33`). We also net out the know information from `u1` and `v1` which we label `u1b` and `v1b`. For the two cases compute RAS estimates of `A1`, labeling them `ARB32` and `ARB33`. Then we can calculate the mean absolute percentage error comparing `A1` with `ARB32` and `ARB33`, reporting the comparative results in `R`.

```
        R←2 4ρ0
        AR←A0 RAS 3 3ρu1,v1,x1
        mape1←(÷9)×+/+/100×(|AR-A1)÷A1

        A0B←A0 ⋄ A0B[3;2]←0
        K←3 3ρ0 ⋄ K[3;2]←0.033 ⋄ K32←K
        ARB32←ARB←K+A0B RAS 3 3ρ(u1b←u1-K+.×x1),(v1b←v1-+⌿K+.×DIAG x1),x1
        mape2←(÷9)×+/+/100×(|ARB-A1)÷A1
        R[1;]←3,2,mape1,mape2

        A0B←A0 ⋄ A0B[3;3]←0
        K←3 3ρ0 ⋄ K[3;3]←0.133 ⋄ K33←K
        ARB33←ARB←K+A0B RAS 3 3ρ(u1b←u1-K+.×x1),(v1b←v1-+⌿K+.×DIAG x1),x1
        mape2←(÷9)×+/+/100×(|ARB-A1)÷A1
        R[2;]←3,3,mape1,mape2

K32
    0.0000    0.0000    0.0000
```

```
    0.0000    0.0000    0.0000
    0.0000    0.0330    0.0000
ARB32
    0.1692    0.1337    0.0370
    0.0619    0.1333    0.0488
    0.0340    0.0330    0.1309
K33
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.1330


ARB33
    0.1664    0.1390    0.0362
    0.0605    0.1378    0.0474
    0.0381    0.0232    0.1330
R
3 2 24.048867 19.500293
3 3 24.048867 24.182855
```

Problem 9.8: Degenerate Cases in Application of RAS

This problem illustrates two degenerate cases that occur in applying RAS.

Problem 9.8

Consider the transactions matrix $\mathbf{Z}(0) = \begin{bmatrix} 100 & 55 & 25 \\ 0 & 75 & 25 \\ 25 & 10 & 110 \end{bmatrix}$ and projected vectors of intermediate inputs and outputs, $\mathbf{v}(1) = \begin{bmatrix} 125 \\ 140 \\ 160 \end{bmatrix}$ and $\mathbf{u}(1) = \begin{bmatrix} 180 \\ 100 \\ 145 \end{bmatrix}$, respectively. In this case $\mathbf{u}(1)$ and $\mathbf{v}(1)$ are identical to $\mathbf{u}(0)$ and $\mathbf{v}(0)$, respectively, so an RAS procedure attempting to produce $\tilde{\mathbf{Z}}(1)$ will converge immediately and is, of course, unnecessary.

If we project $v_1(1) = 100$ instead of 125, By reducing $v_1(1)$ to substantially below the existing value, without any other changes, then $\mathbf{i}'\mathbf{u}(1) \neq \mathbf{i}'\mathbf{v}(1)$. Successive RAS adjustments in this case fail to converge since both row and column constraints in the RAS procedure cannot be satisfied simultaneously.

Computational Notes

We first define the base `Z0` and the target `u1` and `v1`.

```
      Z0←3 3⍴100 55 25 0 75 25 25 10 110
      v1←125 140 160 ⋄ u1←180 100 145

      Z0
100 55  25
  0 75  25
 25 10 110
```

```
      u1
180 100 145
      v1
125 140 160
```

We attempt to create an RAS estimate of a new matrix of transactions based on `Z0`, `u1`, and `v1`

```
      Z0 RAST 2 3ρu1,v1

**** CONVERGENCE: 0 ITERATIONS ****
```

For the second case we modify the first element of the vector `v1` to a value of 100, and label it `v11` and attempt to create an RAS estimate of a new matrix of transactions based on `Z0`, `u1`, and `v11`.

```
      v11
100 140 160
      Z0 RAST 2 3ρu1,v11
**** STOPPED: 500000 ITERATIONS ****
```

Problem 9.9: Measuring Accuracy of RAS-Estimated Total Requirements Matrices

This exercise explores the degree to which the accuracy of RAS estimates of technical coefficients relates to that of the total requirements matrices.

Problem 9.9 Overview

We use the U.S. input-output tables for 1997 and 2005 (from Problem 9.1, expressed in current dollars rather than constant dollars).

The matrices $\mathbf{A}(1997)$, $\mathbf{A}(2005)$ and $\tilde{\mathbf{A}}(2005)$ [produced by using RAS with $\mathbf{A}(1997)$], $\mathbf{u}(2005)$, $\mathbf{v}(2005)$ and $\mathbf{x}(2005)$, were all computed in Problems 7.1 and 7.5. The MAPE for $\tilde{\mathbf{A}}(2005)$ compared with $\mathbf{A}(2005)$ is 49.03. The MAPE for $\tilde{\mathbf{L}}(2005) = [\mathbf{I} - \tilde{\mathbf{A}}(2005)]^{-1}$ compared with $\mathbf{L}(2005)$ is 12.33, where the matrices $\mathbf{L}(2005)$ and $\tilde{\mathbf{L}}(2005)$ are computed as:

L(2005)	1	2	3	4	5	6	7
1	1.3139	0.0102	0.0247	0.0789	0.0076	0.0103	0.0122
2	0.0462	1.1863	0.0515	0.1331	0.0584	0.0152	0.0296
3	0.0109	0.0034	1.0054	0.0075	0.0074	0.0116	0.0257
4	0.4324	0.1907	0.4421	1.5707	0.1332	0.1404	0.2098
5	0.1773	0.0865	0.1737	0.1969	1.1072	0.0714	0.0950
6	0.2861	0.2701	0.3053	0.3508	0.3136	1.4409	0.3600
7	0.0330	0.0231	0.0300	0.0486	0.0342	0.0329	1.0390

$\tilde{L}$(2005)	1	2	3	4	5	6	7
1	1.3426	0.0081	0.0218	0.0746	0.0084	0.0114	0.0126
2	0.0407	1.1812	0.0534	0.1223	0.0585	0.0188	0.0266
3	0.0124	0.0044	1.0073	0.0094	0.0095	0.0164	0.0078
4	0.3670	0.1485	0.3931	1.5503	0.1297	0.1533	0.2197
5	0.1607	0.0795	0.1686	0.1853	1.1006	0.0807	0.0876
6	0.3324	0.3030	0.3381	0.3682	0.3148	1.4147	0.3999
7	0.0254	0.0261	0.0278	0.0665	0.0398	0.0339	1.0187

Computational Notes

We retrieve from Problem 9.1 the values for `A1`, `A3`, `L3`, `u3`, `v3`, and `x3`.

```
u3
  265510.0   515254.5   152294.5 2911056.4 1329435.7 4780440.1  447314.2
v3
  189279.4   162793.1   653783.1 2998475.5 1218705.1 4153589.6 1024679.6
x3
  312753.9   396562.7 1302388.3 4485529.1 3355943.710477640.1 2526324.9
A1
    0.2611     0.0002     0.0017     0.0397     0.0012     0.0021     0.0000
    0.0017     0.1146     0.0070     0.0304     0.0226     0.0010     0.0000
    0.0035     0.0002     0.0011     0.0018     0.0044     0.0068     0.0000
    0.1745     0.1166     0.2679     0.3600     0.0737     0.0647     0.0020
    0.0745     0.0682     0.1108     0.0985     0.0836     0.0374     0.0007
    0.1156     0.2641     0.1617     0.1292     0.2272     0.2363     0.0039
    0.0004     0.0049     0.0008     0.0091     0.0085     0.0043     0.0000
A3
    0.2258     0.0000     0.0015     0.0384     0.0001     0.0017     0.0007
    0.0027     0.1432     0.0075     0.0675     0.0367     0.0004     0.0070
    0.0051     0.0002     0.0010     0.0018     0.0037     0.0071     0.0215
    0.1955     0.0877     0.2591     0.3222     0.0547     0.0566     0.1010
    0.0819     0.0422     0.1011     0.0994     0.0704     0.0334     0.0487
    0.0843     0.1276     0.1225     0.1172     0.1760     0.2783     0.2026
    0.0099     0.0095     0.0093     0.0219     0.0215     0.0188     0.0240
L3
    1.3139     0.0102     0.0247     0.0789     0.0076     0.0103     0.0122
    0.0462     1.1863     0.0515     0.1331     0.0584     0.0152     0.0296
    0.0109     0.0034     1.0054     0.0075     0.0074     0.0116     0.0257
    0.4324     0.1907     0.4421     1.5707     0.1332     0.1404     0.2098
    0.1773     0.0865     0.1737     0.1969     1.1072     0.0714     0.0950
    0.2861     0.2701     0.3053     0.3508     0.3136     1.4409     0.3600
    0.0330     0.0231     0.0300     0.0486     0.0342     0.0329     1.0390
```

We compute an RAS estimate of `A3`, labeled `AR`, using the RAS function with `A1` as the base matrix of technical coefficients (left argument) and a matrix with `u3`, `v3`, and `x3` as its rows as the right argument and compute the associated Leontief inverse `LR`.

```
AR
     0.245      0.000      0.002      0.036      0.001      0.002      0.001
     0.004      0.142      0.014      0.062      0.037      0.002      0.005
     0.005      0.000      0.002      0.003      0.005      0.011      0.002
```

```
      0.159      0.062      0.227      0.315      0.052      0.064      0.113
      0.074      0.039      0.102      0.093      0.064      0.040      0.042
      0.117      0.156      0.153      0.126      0.178      0.259      0.242
      0.001      0.011      0.003      0.034      0.026      0.019      0.001
LAR
      1.343      0.008      0.022      0.075      0.008      0.011      0.013
      0.041      1.181      0.053      0.122      0.058      0.019      0.027
      0.012      0.004      1.007      0.009      0.009      0.016      0.008
      0.367      0.149      0.393      1.550      0.130      0.153      0.220
      0.161      0.080      0.169      0.185      1.101      0.081      0.088
      0.332      0.303      0.338      0.368      0.315      1.415      0.400
      0.025      0.026      0.028      0.066      0.040      0.034      1.019
```

Finally, we compute the mean absolute percentage error of A3 compared with AR, labeled mape1 and the same measure of L3 compared with LAR, labeled mape2.

```
      mape1←(÷49)×+/+/100×(|A3-AR)÷A3
      mape2←(÷49)×+/+/100×(|L3-LAR)÷L3

      mape1, mape2
49.027628 12.327383
```

Chapter 10, Nonsurvey and Partial-Survey Methods: Extensions

Chapter 10 surveys a range of partial survey and nonsurvey estimation approaches for creating input–output tables at the regional level. Variants of the commonly used class of estimating procedures using location quotients are reviewed; these presume a regional estimate of input–output data can be derived using some information about a target region. Cross-hauling is discussed and approaches to address it are presented.

The RAS technique developed in Chapter 9 is applied using a base national table or a table for another region and some available data for the target region. Techniques for partial survey estimation of commodity flows between regions are also presented along with discussions of several real-world multinational applications, including the China–Japan Transnational Interregional Model and Leontief's World Model. The exercise problems for this chapter explore application of nonsurvey techniques for generating regional input-output models.

Problem 10.1: RAS Estimation of IO Tables for Regions with Similar Economies

The exercise explores the use of the RAS technique to develop and use input-output tables for target economies with similar basic structural characteristics.

Problem 10.1 Overview

Consider three different nations. The first, the economy of the Land of Lilliput, is described by the following input-output table:

	Interindustry Transactions		Total Outputs
	A	*B*	
A	1	6	20
B	4	2	15

The Land of Brobdingnag is described by another input-output table:

	Interindustry Transactions		Total Outputs
	A	*B*	
A	7	4	35
B	1	5	15

And finally, the economy of the distant land of the Houyhnhnms is described by yet another input-output table:

	Interindustry Transactions		Total Outputs
	A	*B*	
A	20	30.67	100
B	2.86	38.3	115

First, we compute the vectors of value-added, intermediate inputs, final-demand, and intermediate outputs for each economy, shown in the following table:

	Lilliput (L)	Brobdingnag (B)	Houyhnhnm (H)
Value Added	[15 7]	[27 6]	[77.14 46.03]
Intermediate Inputs ($\mathbf{v} = \mathbf{i}'\mathbf{Z}$)	[5 8]	[8 9]	[22.86 68.97]
Final Demands	$\begin{bmatrix} 13 \\ 9 \end{bmatrix}$	$\begin{bmatrix} 24 \\ 9 \end{bmatrix}$	$\begin{bmatrix} 49.33 \\ 73.84 \end{bmatrix}$
Intermediate Outputs ($\mathbf{u} = \mathbf{Zi}$)	$\begin{bmatrix} 7 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 11 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 50.67 \\ 41.16 \end{bmatrix}$

A Lilliputian economist is interested in examining the structure of the Brobdingnagian economy. Likewise, a Brobdingnagian economist is interested in examining the structure of the Lilliputian economy. However, each economist only has available to him the value-added, final-demand, and total-output vectors for the foreign economy. Each economist knows the RAS modification procedure and uses it with the technical coefficients matrix of her own economy serving as the base **A** matrix. To determine which of the two economists calculates a better estimate of the foreign economy's technical coefficients matrix in terms of mean absolute deviation (all elements of **A**), first we compute the true technical coefficients matrices for each economy: $\mathbf{A}^L = \begin{bmatrix} .050 & .400 \\ .200 & .133 \end{bmatrix}$ and $\mathbf{A}^B = \begin{bmatrix} .200 & .267 \\ .029 & .333 \end{bmatrix}$. We denote the L estimate of the $\mathbf{A}^B$ matrix as ${}^L\mathbf{A}^B = \begin{bmatrix} .088 & .529 \\ .141 & .071 \end{bmatrix}$; we use the metric of mean absolute deviation (MAD) to measure the relative accuracy of between ${}^L\mathbf{A}^B$ as an estimate of $\mathbf{A}^B$, which is 0.187. The B estimate of $\mathbf{A}^L$ is ${}^B\mathbf{A}^L = \begin{bmatrix} .207 & .190 \\ .043 & .343 \end{bmatrix}$, with a MAD comparing of ${}^B\mathbf{A}^L$ as an estimate of $\mathbf{A}^L$ found as 0.183. Therefore, the Brobdingnagian economist does slightly better.

Suppose now that an economist in the distant land of the Houyhnhnms learned of the two other economies from a world traveler. She becomes interested in the economic structures of these foreign lands but is only able to obtain the final-demand, value-added, and total-output vectors for each economy from the world traveler. The economist uses RAS with her own country's **A** matrix as a base to estimate the interindustry structure of the two distant lands. The two Houyhnhnm estimates are ${}^H\mathbf{A}^L = \begin{bmatrix} .207 & .190 \\ .043 & .343 \end{bmatrix}$ and ${}^H\mathbf{A}^B = \begin{bmatrix} .200 & .267 \\ .029 & .333 \end{bmatrix}$, respectively (note that $\mathbf{A}^H = \mathbf{A}^B = {}^H\mathbf{A}^B = \begin{bmatrix} .200 & .267 \\ .029 & .333 \end{bmatrix}$). The error, as measured by MAD, is 0.183 in the first case and, of course, zero in the second case since $\mathbf{A}^H = \mathbf{A}^B$, i.e., the Houyhnhnm and Brobdingnagian economies are identical.

Suppose now that the Land of Lilliput plans to build a new power plant which will require the following value of output (in millions of dollars) from each of the economy's industries (directly, so it can be thought of as a final demand presented to the Lilliputian economy) of $\mathbf{f} = [100 \quad 150]'$. To measure the accuracy of the Houyhnhnms' estimate of the total industrial activity (output) in the Lilliputian economy required to construct this power plant, measured as an average mean absolute deviation, we first compute the true impact as $\Delta\mathbf{x}^L = \mathbf{L}^L\Delta\mathbf{f} = \begin{bmatrix} 197.3 \\ 218.6 \end{bmatrix}$ for $\Delta\mathbf{f} = \begin{bmatrix} 100 \\ 150 \end{bmatrix}$ and $\mathbf{L}^L = (\mathbf{I} - \mathbf{A}^L)^{-1} = \begin{bmatrix} 1.166 & .538 \\ .269 & 1.278 \end{bmatrix}$. Using the same final demand vector with $(\mathbf{I} - {}^H\mathbf{A}^L)^{-1} = \begin{bmatrix} 1.281 & .371 \\ .083 & 1.546 \end{bmatrix}$ yields $\Delta[{}^H\mathbf{x}^L] = \begin{bmatrix} 183.8 \\ 240.3 \end{bmatrix}$. The mean absolute deviation between these two vectors is 17.6.

Computational Notes

We create the transactions matrix for the three Lilliputian, Houyhnhnm, and Brobdingnagian economies, `Z1`, `Z2`, and `Z3`, respectively, as well as the associated vectors of final demands, `f1`, `f2`, and `f3`; value-added, `w1`, `w2`, and `w3`; and total outputs, `x1`, `x2`, and `x3`. Then compute the corresponding matrices of technical coefficients, `A1`, `A2`, and `A3`.

```
      Z1←2 2⍴1 6 4 2 ⋄ Z2←2 2⍴7 4 1 5 ⋄ Z3←2 2⍴20 30.67 2.86 38.3
      x1←20 15 ⋄ x2←35 15 ⋄ x3←100 115
      f1←x1-u1←+/Z1 ⋄ f2←x2-u2←+/Z2 ⋄ f3←x3-u3←+/Z3
      w1←x1-v1←+⌿Z1 ⋄ w2←x2-v2←+⌿Z2 ⋄ w3←x3-v3←+⌿Z3
      A1←Z1 AMAT x1 ⋄ A2←Z2 AMAT x2 ⋄ A3←Z3 AMAT x3

A1,A2,A3
     0.050     0.400     0.200     0.267     0.200     0.267
     0.200     0.133     0.029     0.333     0.029     0.333
```

Now compute the RAS estimate of `A2`, named `ARR2`, using the function `RAS` with `A1` as the left argument and as the right argument a matrix with `u2`, `v2`, and `x2` as the rows of the matrix. Also compute the RAS estimate of `A1`, named `ARR1`, using the function `RAS` with A2 as the left argument and as the right argument a matrix with `u1`, `v1`, and `x1` as the rows of the matrix. In each case, compute the mean absolute deviation of `ARR2` compared with `A2` (`MAD2`) and the mean absolution deviation of `ARR1` compared with `A1` (`MAD1`).

```
      ARR2←A1 fras 3 2⍴u2,v2,x2 ⋄ MAD1←(÷n×n)×+/+/|ARR2-A2
      ARR1←A2 fras 3 2⍴u1,v1,x1 ⋄ MAD2←(÷n×n)×+/+/|ARR1-A1

      ARR2
     0.088     0.529
     0.141     0.071
      MAD1
0.18349075

      ARR1
     0.207     0.190
     0.043     0.343
```

```
      MAD1
0.1873887
```

Next, compute the RAS estimates of `A1`, named `ARS1`, using the function `RAS` with `A3` as the left argument and as the right argument a matrix with `u1`, `v1`, and `x1` as the rows of the matrix, and the RAS estimate of `A2`, named `ARS2`, using the function `RAS` with `A3` as the left argument and as the right argument a matrix with `u2`, `v2`, and x2 as the rows of the matrix. Also, compute the mean absolute deviation of `A1` compared with `ARS1` and `A2` compared with `ARS2`.

```
      ARS1←A3 fras 3 2ρu1,v1,x1 ⋄ MADH1←(÷n×n)×+/+/|ARS1-A1
      ARS2←A3 fras 3 2ρu2,v2,x2 ⋄ MADH2←(÷n×n)×+/+/|ARS2-A2
      ARS1
      0.207     0.190
      0.043     0.343
      MADH1
0.18343186
      ARS2
      0.200     0.267
      0.029     0.333
      MADH2
0.000048197817
```

Finally, define the new vector of final demands `Δf` and the Leontief inverses of `A1` and of `ARS1`, labeled `L1` and `LRS1`, respectively and compute the vector of total outputs, respectively, for each, `xreal` and `xest`, for the new vector of final demands. Then compute the mean absolute deviation of `xreal` compared with `xest`, labelled `MADPP`.

```
      Δf
100 150
      L1,                 LRS1
      1.166     0.538     1.281     0.371
      0.269     1.278     0.083     1.546
      xest
      183.8     240.3
      xreal
      197.3     218.6
      MADPP
17.555423
```

Problem 10.2: Expanded Case of Regional Estimation

This exercise expands the economies given in Problem 10.1 to three economic sectors.

Problem 10.2 Overview

The Land of Lilliput is described by the following input-output table:

	Interindustry Transactions			Total Outputs
	A	*B*	*C*	
A	1	6	6	20
B	4	2	1	15
C	4	1	1	12

The economy of the neighboring land of Brobdingnag is described by another input-output table:

	Interindustry Transactions			Total Outputs
	A	*B*	*C*	
A	7	4	8	35
B	1	5	1	15
C	6	2	7	30

The economy of the distant land of Houyhnhnms is described by yet another input-output table:

	Interindustry Transactions			Total Outputs
	A	*B*	*C*	
A	5.5	33	33	110
B	22	11	5.5	82.5
C	22	5.5	5.5	66

First, we find $\mathbf{A}^L = \begin{bmatrix} .050 & .400 & .500 \\ .200 & .133 & .083 \\ .200 & .067 & .083 \end{bmatrix}$ and $\mathbf{A}^B = \begin{bmatrix} .200 & .267 & .267 \\ .029 & .333 & .033 \\ .171 & .133 & .233 \end{bmatrix}$. The RAS estimates are $^B\mathbf{A}^L = \begin{bmatrix} .264 & .199 & .395 \\ .052 & .343 & .068 \\ .134 & .059 & .204 \end{bmatrix}$ and $^L\mathbf{A}^B = \begin{bmatrix} .033 & .460 & .365 \\ .106 & .123 & .049 \\ .261 & .151 & .120 \end{bmatrix}$. The table of value added, intermediate inputs, final demands, and intermediate outputs of the economies are given in the following table:

	Lilliput (L)	Brobdingnag (B)	Houyhnhnm (H)
Value Added ($\mathbf{v}'$)	[11 6 4]	[21 4 14]	[60.5 33.0 22.0]
Intermediate Inputs ($\mathbf{i}'\mathbf{Z}$)	[9 9 8]	[14 11 16]	[49.5 49.5 44.0]
Final Demands (**f**)	$\begin{bmatrix} 7 \\ 8 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 16 \\ 8 \\ 15 \end{bmatrix}$	$\begin{bmatrix} 38.5 \\ 44.0 \\ 33.0 \end{bmatrix}$
Intermediate Outputs (**u**)	$\begin{bmatrix} 13 \\ 7 \\ 6 \end{bmatrix}$	$\begin{bmatrix} 19 \\ 7 \\ 15 \end{bmatrix}$	$\begin{bmatrix} 71.5 \\ 38.5 \\ 33.0 \end{bmatrix}$

The mean absolute deviation (MAD) for the *L* estimate of *B* is 0.109, while the MAD for the *B* estimate of *L* is 0.121. The Houyhnhnm estimates of *L* and *B*, respectively are $^H\mathbf{A}^L = \begin{bmatrix} .050 & .400 & .500 \\ .200 & .133 & .083 \\ .200 & .067 & .083 \end{bmatrix}$ and $^H\mathbf{A}^B = \begin{bmatrix} .033 & .460 & .365 \\ .106 & .123 & .049 \\ .261 & .151 & .120 \end{bmatrix}$. Note that in this case that $\mathbf{A}^H = \mathbf{A}^L$, i.e., it is the Houyhnhnm and Lilliputian economies that are identical, so the error of the Houyhnhnm estimate of the Lilliputian economy, $^H\mathbf{A}^L$, compared with the true Lilliputian economy, $\mathbf{A}^L$, as measured by the MAD, is 0.0. The MAD for the Houyhnhnm estimate of the Brobdingnagian economy, ${}^H\mathbf{A}^B$, compared with the true Brobdingnagian economy, $\mathbf{A}^B$, is 0.109.

Computational Notes

We can use exactly the same APL expressions developed in Problem 10.1 for this problem.

```
Z1←3 3⍴1 6 6 4 2 1 4 1 1
Z2←3 3⍴7 4 8 1 5 1 6 2 7
Z3←5.5×Z1
x1←20 15 12 ⋄ x2←35 15 30 ⋄ x3←5.5×x1
```

Now we use the same APL expressions developed in Problem 10.1 with the following results:

```
A1
      0.050     0.400     0.500
      0.200     0.133     0.083
      0.200     0.067     0.083
A2
      0.200     0.267     0.267
      0.029     0.333     0.033
      0.171     0.133     0.233
A3
      0.050     0.400     0.500
      0.200     0.133     0.083
      0.200     0.067     0.083
A2R1
      0.033     0.460     0.365
      0.106     0.122     0.049
      0.261     0.151     0.120
MAD1
0.10915303
A1R2
      0.264     0.199     0.395
      0.052     0.343     0.068
      0.134     0.059     0.204
MAD2
0.12092175
ARS1
      0.050     0.400     0.500
      0.200     0.133     0.083
      0.200     0.067     0.083
MADH1
0
ARS2
      0.033     0.460     0.365
      0.106     0.122     0.049
      0.261     0.151     0.120
MADH2
0.10915303
```

Problem 10.3: Assessing Costs of Regional Estimation

This exercise illustrates the considerations of analysis costs in estimation and impact analysis.

Problem 10.3 Overview

Consider the following input-output transactions and total outputs table for Region 1:

	A	*B*	Total Output
A	1	2	10
B	3	4	10

We are interested in determining the impact of a particular final demand in another region (Region 2). Suppose we have the following data concerning Region 2.

	Value Added	Final Demand	Total Outputs
A	10	11	15
B	13	12	20

The cost of computing an RAS estimate of Region 2's input-output table using Region 1's **A** matrix as a base table is given by nc_1, where n is the number of RAS iterations, where for purposes here one iteration is defined by one row and one column adjustment, that is, $\mathbf{A}^k = \hat{\mathbf{r}}\mathbf{A}^{k-1}\hat{\mathbf{s}}$ (a row adjustment alone as the last iteration would also be counted as an iteration).

We ultimately wish to compute the impact of a new final demand in Region 2. This impact (the total outputs required to support the new final demand) can be computed exactly or by using the round-by-round approximation of the inverse. We know that: (1) The cost of computing the inverse exactly on a computer is c_1 and the cost of using this inverse in impact analysis is c_2 (let us assume that $c_2 = 10c_1$, i.e., the cost of computing the inverse is ten times the cost of using it in impact analysis). (2) The cost of a round-by-round approximation of impact analysis is mc_1, where m is the order of the round-by-round approximation, that is, $\mathbf{f} + \mathbf{Af} + \mathbf{A}^2\mathbf{f} + \cdots + \mathbf{A}^m\mathbf{f}$.

If we assume that a fourth-order round-by-round approximation is sufficiently accurate ($m = 4$), to determine whether the first or second method of impact analysis would minimize cost, we observe that the cost of using the first method, i.e., computing the exact inverse, is $c_1 + c_2$; with $c_2 = 10c_1$, and the total cost is $11c_1$. With $m = 4$, the cost of using the second method, i.e., round-by-round approximation in impact analysis, $4c_1$, so it is the least cost method in this case.

To determine the total cost of performing impact analysis, including the cost of the RAS approximation (tolerance of 0.01) and of the impact analysis scheme, we first note that, since the RAS procedure converges to within a tolerance of 0.01 in 2 iterations, the cost of the RAS estimate of region 2's coefficients matrix is $5c_1$. Then utilizing the result in a round-by-round application, with $m = 4$, gives a total cost of $6c_1$.

Finally, if we presume the budget for the entire impact-analysis calculation is $7c_1$, the level of tolerance that is affordable, among the options of 0.01, 0.001, 0.0001, 0.00001, or 0.000001, is found by in the following table of cost calculations:

RAS Tolerance	Number of Iterations	RAS Cost	Impact Analysis Cost	Total Cost
.01	3	$3c_1$	$4c_1$	$7c_1$
.001	4	$4c_1$	$4c_1$	$8c_1$
.0001	5	$5c_1$	$4c_1$	$9c_1$
.00001	6	$6c_1$	$4c_1$	$10c_1$
.000001	7	$7c_1$	$4c_1$	$11c_1$

Therefore, the maximum affordable tolerance is .01.

Computational Notes

We define, for the base region, the matrix of transactions `Z0` and vector of total outputs `x0`. For the target region we define the vectors of total outputs `x1`, value-added inputs `w1`, and final demands `f1`. From this we can compute values necessary to use the function `RAS`, which, along with `x1`, including the matrix of technical coefficients `A0` and the vectors of intermediate outputs `u1`, and intermediate inputs `v1`.

```
      Z0←2 2⍴1 2 3 4 ⋄ x0←10 10
      x1←15 20 ⋄ w1←10 13 ⋄ f1←11 12
      u1←x1-f1 ⋄ v1←x1-w1

Z0
     1     2
     3     4
x0
    10    10
A0
 0.100 0.200
 0.300 0.400
x1
    15    20
w1
    10    13
f1
    11    12
u1
     4     8
v1
     5     7
```

For solving this problem, it is helpful to modify the function `RAS` to show the number of iterations needed to reach convergence, which we name `RASI`. Note also we also remove the variable `tol` from the list of local variables so that it can be set externally as a global variable.

```
[ 0]  A←A0 RASI UVX;lim;n;nn;k;XD;u1;v1;x;r;s
[ 1] ⍝Basic function for RAS biproportional scaling
[ 2] ⍝that records the number of iterations
[ 3] ⍝to converge to tol (set globally)
[ 4] ⍝INPUT: A0 and rows of UVX are u1 v1 x
[ 5]  nn←2⍴n←1↑⍴A0 ⋄ lim←100
[ 6]  A←A0 ⋄ u1←UVX[1;] ⋄ v1←UVX[2;] ⋄ x←UVX[3;] ⋄ XD←(nn⍴x)×nn⍴1,n⍴0
[ 7]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←0
[ 8]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[ 9] ⍝----ROW ADJUSTMENT
[10] LOOP:A←A×⍉nn⍴r←u1÷u
[11]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←k+1
[12]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[13] ⍝----COL ADJUSTMENT
[14]  A←A×nn⍴s←v1÷v
[15]  u←A+.×x ⋄ v←+⌿A+.×XD ⋄ k←k+1
[16]  →(((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol))/CON
[17]  →(lim>k)/LOOP
[18]  →0,0⍴⎕←'**** STOPPED: ',(⍕k),' ITERATIONS ****'
[19] CON:→0,0⍴⎕←'**** CONVERGENCE IN ',(⍕k),' ITERATIONS ****'
```

Using RASI, with A0 as the left argument and, as the right argument, a matrix with `u1`, `v1`, and `x1` as its rows, we can evaluate the iterations necessary to achieve the candidate levels of convergence tolerance.

```
      ⎕←'tol= ',⍕tol←0.01
      A←A0 RASI 3 2⍴u1,v1,x1
      ⎕←'tol= ',⍕tol←0.001
      A←A0 RASI 3 2⍴u1,v1,x1
      ⎕←'tol= ',⍕tol←0.0001
      A←A0 RASI 3 2⍴u1,v1,x1
      ⎕←'tol= ',⍕tol←0.00001
      A←A0 RASI 3 2⍴u1,v1,x1
      ⎕←'tol= ',⍕tol←0.000001
      A←A0 RASI 3 2⍴u1,v1,x1
```

We could, of course, more efficiently consolidate this sequence of calculations into an APL function `RASI2`.

```
[0]  A0 RASI2 UV;k;tol;A
[1]  k←1
[2] L1:⎕←'tol= ',⍕tol←0.1×10*-k
[3]  A←A0 RASI UV
[4]  →(5≥k←k+1)/L1

     A0 RASI2 3 2⍴u1,v1,x1
tol= 0.01
**** CONVERGENCE IN 3 ITERATIONS ****
tol= 0.001
**** CONVERGENCE IN 4 ITERATIONS ****
tol= 0.0001
**** CONVERGENCE IN 5 ITERATIONS ****
tol= 0.00001
```

```
**** CONVERGENCE IN 6 ITERATIONS ****
tol= 0.000001
**** CONVERGENCE IN 7 ITERATIONS ****
```

Problem 10.4: Location Quotients

This exercise explores the behavior of the adjustment term that converts location-quotient Flagg Location quotient approach (*FLQ)* to an "augmented" *FLQ*, designated *AFLQ*, which adjusts for a measure of regional size.

Problem 10.4 Overview

First, recall that the *FLQ* is defined as an adjustment to the cross-industry quotient, *CIQ*, defined by $FLQ_{ij}^r = (\lambda)CIQ_{ij}^r$ where $\lambda = \left\{\log_2[1+(x^r/x^n)]\right\}^\delta$, $0 \le \delta < 1$, and the modified technical coefficients are defined by $a_{ij}^{rr} = \begin{Bmatrix} (FLQ_{ij}^r)a_{ij}^n & \text{if } FLQ_{ij}^r < 1 \\ a_{ij}^n & \text{if } FLQ_{ij}^r \ge 1 \end{Bmatrix}$. The *AFLQ* is defined by $AFLQ_{ij}^r = \begin{Bmatrix} \left[\log_2(1+LQ_j^r)\right]FLQ_{ij}^r & \text{if } LQ_j^r > 1 \\ FLQ_{ij}^r & \text{if } LQ_j^r \le 1 \end{Bmatrix}$ and the modified technical coefficients by $a_{ij}^{rr} = \begin{Bmatrix} (AFLQ_{ij}^r)a_{ij}^n & \text{if } LQ_j^r > 1 \\ (FLQ_{ij}^r)a_{ij}^n & \text{if } LQ_j^r \le 1 \end{Bmatrix}$. The adjustment term for *AFLQ,* $\lambda = \log_2(1+LQ_j^r) = \{\log_2[1+(x^r/x^n)]\}^\delta$, varies with the degree of specialization in a region, i.e., when $LQ_j^r > 1$, then $\left[\log_2(1+LQ_j^r) > 1\right]$, as discussed in Section 10.2.5.

The following table shows $\lambda = \log_2(1+LQ_j^r) = \{\log_2[1+(x^r/x^n)]\}^\delta$ for values of x^r/x^n = .01, .1, .25, .5, .75 and 1 cross tabulated with values of $\delta =$ 0, .1, .3, .5 and 1.

x^r/x^n	0.01	0.1	0.25	0.5	0.75	1.0
$\log_2[1+(x^r/x^n)]$	0.0144	0.1375	0.3219	0.5850	0.8074	1
$\{\log_2[1+(x^r/x^n)]\}^0$	1	1	1	1	1	1
$\{\log_2[1+(x^r/x^n)]\}^{0.1}$	0.6542	0.8200	0.8928	0.9478	0.9788	1
$\{\log_2[1+(x^r/x^n)]\}^{0.3}$	0.2800	0.5514	0.7118	0.8514	0.9378	1
$\{\log_2[1+(x^r/x^n)]\}^{0.5}$	0.1198	0.3708	0.5647	0.7648	0.8985	1
$\{\log_2[1+(x^r/x^n)]\}^1$	0.0144	0.1375	0.32	0.5850	0.8074	1

Computational Notes

First define the values of x^r/x^n to be evaluated, which in the APL workspace we assign to the variable `xx`, and the values of δ, which in the APL workspace we assign to the variable `q`.

To solve this problem, we introduce two new primitive functions. The first is the dyadic function *logarithm*, indicated by the character ⍟, which takes as its left argument the logarithm base and as its right argument the array for which the logarithm is to be calculated. For example,

```
      10⍟1000 10000
3 4
      2⍟100
6.6438562
```

As an aside, the monadic form of the primitive function logarithm gives the natural or Naperian logarithm of the right argument, as in

```
      ⍟1
0
      ⍟10
2.3025851
```

The second new primitive function to introduce is *exponential*, indicated by the character ⋆, which provides as its explicit result the left argument taken to the exponential power indicated by the right argument, as in

```
      10⋆1
10
      10⋆⍳3
10 100 1000
      2⋆.5
1.4142136
```

For this problem we use these new primitive functions to apply the formula $\lambda = \log_2(1+LQ_j^r) = \{\log_2[1+(x^r/x^n)]\}^{\delta}$ for values of $x^r/x^n = .01$, .1, .25, .5, .75 and 1 cross tabulated with values of $\delta =$ 0, .1, .3, .5 and 1, using the variables `xx` and `q` and assign the results to the variable `TAB`.

```
     TAB←(6 6⍴2⍟1+xx)⋆⍉6 6⍴q

     xx
0.01 0.1 0.25 0.5 0.75 1
     q
1 0 0.1 0.3 0.5 1

     TAB
0.014355293 0.13750352 0.32192809 0.5849625  0.80735492 1
1           1          1          1          1          1
0.65418594  0.8200331  0.89284462 0.94779148 0.97882815 1
0.27996492  0.55143477 0.71175031 0.85140934 0.9378197  1
0.11981358  0.37081468 0.56738708 0.76482841 0.89852931 1
0.014355293 0.13750352 0.32192809 0.5849625  0.80735492 1
```

Problem 10.5: Simple Location Quotients for Estimating Regional Technical Coefficients

This exercise illustrates the use of simple location quotients (SLQ) to estimate the matrix of regional technical coefficients.

Problem 10.5 Overview

First, we define the matrix of technical coefficients for a national economy, $\mathbf{A}^N$, and the vector of total outputs, $\mathbf{x}^N$, as

$$\mathbf{A}^N = \begin{bmatrix} .1830 & .0668 & .0087 \\ .1377 & .3070 & .0707 \\ .2084 & .2409 & .2999 \end{bmatrix} \quad \mathbf{x}^N = \begin{bmatrix} 518,288.6 \\ 4,953,700.6 \\ 14,260,843.0 \end{bmatrix}$$

as well as the corresponding values for a target region, $\mathbf{A}^R$ and $\mathbf{x}^R$, as

$$\mathbf{A}^R = \begin{bmatrix} .1092 & .0324 & .0036 \\ .0899 & .0849 & .0412 \\ .1603 & .1170 & .2349 \end{bmatrix} \quad \mathbf{x}^R = \begin{bmatrix} 8,262.7 \\ 95,450.8 \\ 170,690.3 \end{bmatrix}.$$

We calculate the simple location quotients by $LQ_i^r = \left(\dfrac{x_i^r / x^r}{x_i^n / x^n}\right)$, but set equal to 1 when the calculation of LQ_i^r exceeds 1. In this case, the matrix of simple location quotients is $\mathbf{SLQ} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ .8607 & .8607 & .8607 \end{bmatrix}$. The corresponding estimate of the matrix of regional technical coefficients, found by element-by-element multiplication of $\mathbf{A}^N$ by **SLQ,** is

$$\mathbf{A}^{(SLQ)} = \begin{bmatrix} .1830 & .0668 & .0087 \\ .1377 & .3070 & .0707 \\ .1794 & .2074 & .2581 \end{bmatrix}.$$

Computational Notes

First retrieve the national and regional matrices of transactions as `ZN` and `XR`, respectively, along with the corresponding vectors of total outputs, `xn` and `xr`. The we compute the corresponding matrices of technical coefficients, `AN` and `AR`.

```
ZN←94865.585 331072.05 124609.41 71381.324 1520546.4
ZN←3 3⍴ZN,1007903.7 108033.19 1193494.3 4276881.1
xn←518288.6 4953700.6 14260843
ZR←902.1 3093.8 619.4 743.2 8107.2
ZR←3 3⍴ZR,7027.3 1324.2 11167.8 40092.2
xr←8262.7 95450.8 170690.3
AN←ZN AMAT xn ⋄ AR←ZR AMAT xr
```

```
AN
    0.1830    0.0668    0.0087
    0.1377    0.3070    0.0707
    0.2084    0.2409    0.2999
xn
    518289   4953701  14260843
AR
    0.1092    0.0324    0.0036
    0.0899    0.0849    0.0412
    0.1603    0.1170    0.2349
xr
      8263     95451    170690
```

We compute the simple location quotients `SLQ` and the estimate of the regional technical coefficients using SLQ as `ASLQ`.

```
     SLQ←⍉3 3⍴(xr÷+/xr)÷(xn÷+/xn)
     SLQ←(SLQ≥1)+SLQ×SLQ<1
     ASLQ←(SLQ×A×SLQ<1)+A×SLQ≥1

SLQ
    1.0000    1.0000    1.0000
    1.0000    1.0000    1.0000
    0.8607    0.8607    0.8607

ASLQ
    0.1830    0.0668    0.0087
    0.1377    0.3070    0.0707
    0.1794    0.2074    0.2581
```

Problem 10.6: Cross Industry Quotients

This exercise illustrates the calculation of Cross-Industry Quotients (CIQ) using the national and regional data.

Problem 10.6 Overview

Consider the data specified in Problem 10.5. We calculate the cross-industry quotients by $CIQ_{ij}^{r} = \left(\dfrac{x_i^r / x_i^n}{x_j^r / x_j^n} \right)$, but again set equal to 1 when the calculation of CIQ_{ij}^{r} exceeds 1, or, in matrix terms, the matrix of cross industry quotients in this case is $\mathbf{CIQ} = \begin{bmatrix} 1 & .8274 & 1 \\ 1 & 1 & 1 \\ .7508 & .6212 & 1 \end{bmatrix}$.

The corresponding estimate of the matrix of regional technical coefficients using SLQ, found by element-by-element multiplication of $\mathbf{A}^N$ by $\mathbf{CIQ}$ is $\mathbf{A}^{(CIQ)} = \begin{bmatrix} .1830 & .0553 & .0087 \\ .1377 & .3070 & .0707 \\ .1565 & .1497 & .2999 \end{bmatrix}$.

Computational Notes

We use the same data introduced in Problem 10.5 for `AN`, `AR`, `xn`, and `xr`, but to calculate the cross-industry quotients we introduce a new APL primitive function known as *outer product*. This dyadic function takes as its arguments arbitrary numeric vectors and applies a specified dyadic primitive function element-by-element for all elements of the left and right arguments. For example, to generate a multiplication table, we can write

```
      1 2 3∘.×1 2 3
1 2 3
2 4 6
3 6 9
```

Note that the two characters preceding the multiplication sign are `∘.`which denote the outer product. The multiplication character can be replaced by many if not most primitive dyadic functions, as in

```
      1 2 3∘.+1 2 3
2 3 4
3 4 5
4 5 6
      1 2 3∘.=1 2 3
1 0 0
0 1 0
0 0 1
```

We use the outer product function to easily calculate the matrix of cross industry quotients `CIQ` and apply it to create the associated estimate of the matrix of regional technical coefficients using `CIQ`, which we name `ACIQ`.

```
     xrn←(xr÷xn)∘.÷(xr÷xn)
     CIQ←(xrn≥1)+xrn×xrn<1
     ACIQ←AN×CIQ

CIQ
    1.0000    0.8274    1.0000
    1.0000    1.0000    1.0000
    0.7508    0.6212    1.0000

ACIQ
    0.1830    0.0553    0.0087
    0.1377    0.3070    0.0707
    0.1565    0.1497    0.2999
```

Problem 10.7: Regional Estimation using RAS

The exercise uses the RAS technique to generate a regional estimate using the national and regional data.

Problem 10.7 Overview

Consider the economies specified in Problem 10.5 (and used in Problem 10.6). The intermediate outputs vector for the regional economy is given by

$\mathbf{u}^{(R)} = \mathbf{A}^{(R)}\mathbf{x}^{(R)} = \begin{bmatrix} 4{,}615.3 & 15{,}877.7 & 52{,}584.2 \end{bmatrix}'$ and the vector of intermediate inputs is given $\mathbf{v}^{(R)} = (\mathbf{i}'\mathbf{A}^{(R)}\hat{\mathbf{x}}^{(R)})' = \begin{bmatrix} 2{,}969.5 & 22{,}368.8 & 47{,}738.9 \end{bmatrix}'$.

Applying the RAS technique using $\mathbf{A}^N$, $\mathbf{u}^{(R)}$, $\mathbf{v}^{(R)}$, and $\mathbf{x}^{(R)}$, resulting estimate of the matrix of regional technical coefficients is $\mathbf{A}^{(RAS)} = \begin{bmatrix} .1241 & .0270 & .0059 \\ .0712 & .0945 & .0367 \\ .1640 & .1129 & .2370 \end{bmatrix}$.

Computational Notes

We again use the data introduced in Problem 10.5, this time for `AN`, `ZR`, `AR`, and `xr`. Using `ZR` we compute the vectors of regional intermediate outputs and inputs, `ur` and `vr`, respectively. Then we compute the RAS estimate of `AR`, which we call `ARAS`, by using the function `RAS` with `AN` as the left argument and, as the right argument, a matrix with `ur`, `vr`, and `xr` as its rows.

```
     ur←+/ZR ⋄ vr←+⌿ZR
     ARAS←AN RAS 3 3⍴ur,vr,xr

AN
     0.183     0.067     0.009
     0.138     0.307     0.071
     0.208     0.241     0.300
AR
     0.109     0.032     0.004
     0.090     0.085     0.041
     0.160     0.117     0.235
ZR
       902      3094       619
       743      8107      7027
      1324     11168     40092
xr
      8263     95451    170690

ur
    4615.3   15877.7   52584.2
vr
    2969.5   22368.8   47738.9

ARAS
     0.124     0.027     0.006
     0.071     0.095     0.037
     0.164     0.113     0.237
```

Problem 10.8: Comparing Nonsurvey Estimation Techniques

This exercise compares the estimates of regional technical coefficients from a matrix of national technical coefficients generated by simple location quotients (SLQ), cross industry quotients (CIQ), and RAS.

Problem 10.8 Overview

Consider the estimate generate in Problems 10.5, 10.6 and 10.7. In terms of mean absolute deviation from the actual regional technical coefficients, the mean absolute deviation (MAD) calculations for the three methods are: $MAD^{(SLQ)} = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left|a_{ij}^{(SLQ)} - a_{ij}^{(R)}\right| = .0606$;

$MAD^{(CIQ)} = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left|a_{ij}^{(CIQ)} - a_{ij}^{(R)}\right| = .0558$; and $MAD^{(RAS)} = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left|a_{ij}^{(RAS)} - a_{ij}^{(R)}\right| = .0073$.

The RAS technique produces the most accurate estimate in these examples since it shows the lowest value for *MAD* from the actual regional table.

Computational Notes

Once again using the data introduced in Problem 10.5 and used in Problems 10.6 and 10.7, retrieving `AN`, `xn`, `AR`, `xr`, `ur`, and `vr`. To facilitate calculations for this problem we create a simple dyadic user-defined function `MAD` to calculate the median absolute deviation comparing arrays named as the arguments.

```
[  0]  R←A1 MAD A2
[  1] ⍝Mean absolute deviation
[  2]  R←(+/+/|A1-A2)÷×/⍴A1
```

Using `MAD`, we can compare the median absolute deviation for `AR` compared with the nonsurvey estimates `ASLQ`, `ACIQ`, and `ARAS`, saving them as variables `MADSLQ`, `MADCIQ`, and `MADRAS`, respectively.

```
      SLQ←⍉3 3⍴(xr÷+/xr)÷(xn÷+/xn)
      SLQ←(SLQ≥1)+SLQ×SLQ<1 ⋄ ASLQ←A×SLQ
      A←AN ⋄ xrn←(xr÷xn)∘.÷(xr÷xn)
      CIQ←(xrn≥1)+xrn×xrn<1 ⋄ ACIQ←A×CIQ
      ARAS←AN RAS 3 3⍴ur,vr,xr
      MADSLQ←AR MAD ASLQ ⋄ MADCIQ←AR MAD ACIQ ⋄ MADRAS←AR MAD ARAS

AN
     0.183     0.067     0.009
     0.138     0.307     0.071
     0.208     0.241     0.300
xn
  518288.6 4953700.614260843.0
AR
     0.109     0.032     0.004
     0.090     0.085     0.041
     0.160     0.117     0.235
xr
    8262.7   95450.8  170690.3
ur
    4615.3   15877.7   52584.2
vr
    2969.5   22368.8   47738.9
SLQ,ASLQ
     1.000     1.000     1.000     0.183     0.067     0.009
```

```
     1.000      1.000      1.000      0.138      0.307      0.071
     0.861      0.861      0.861      0.179      0.207      0.258
CIQ,ACIQ
     1.000      0.827      1.000      0.183      0.055      0.009
     1.000      1.000      1.000      0.138      0.307      0.071
     0.751      0.621      1.000      0.156      0.150      0.300
ARAS
     0.124      0.027      0.006
     0.071      0.095      0.037
     0.164      0.113      0.237

MADSLQ,MADCIQ,MADRAS
    0.0606     0.0558     0.0073
```

Problem 10.9: More Detailed Comparisons of Nonsurvey Estimation Techniques

This exercise compares the performance of estimates of a variety of nonsurvey estimation techniques in estimating the technical coefficients and associate Leontief inverse coefficients for a known region from a table of national coefficients.

Problem 10.9 Overview

We use the three-sector, three-region Chinese MRIO data for 2000 specified in Problem 3.6 to estimate regions 2 (South China) and 3 (Rest of China) from the national data.

If we adopt the same error metrics used in Table 10.2 and using *LQ*, *CIQ*, *FLQ*, *AFLQ*, and *RPC* techniques to estimate $\mathbf{A}^2$ (for region 2) and $\mathbf{A}^3$ (for region 3) from $\mathbf{A}^n$ (the national table), the results are the following.

Results for Region 2 (South China) using 2000 Chinese IRIO data.

	Intraregional Input Coefficients	Leontief Inverse
Survey	$\begin{bmatrix} 0.1279 & 0.1086 & 0.0340 \\ 0.1348 & 0.4299 & 0.2191 \\ 0.0394 & 0.0814 & 0.1255 \end{bmatrix}$	$\begin{bmatrix} 1.1889 & 0.2418 & 0.1069 \\ 0.3130 & 1.8828 & 0.4839 \\ 0.0827 & 0.1861 & 1.1933 \end{bmatrix}$
Using $\mathbf{A}^n$		
LQ	$\begin{bmatrix} 0.1252 & 0.1301 & 0.0336 \\ 0.1517 & 0.4605 & 0.2411 \\ 0.0411 & 0.0867 & 0.1235 \end{bmatrix}$	$\begin{bmatrix} 1.2033 & 0.3113 & 0.1317 \\ 0.3804 & 2.0378 & 0.5751 \\ 0.0940 & 0.2161 & 1.2039 \end{bmatrix}$
CIQ	$\begin{bmatrix} 0.1252 & 0.1263 & 0.0351 \\ 0.1517 & 0.4605 & 0.2411 \\ 0.0429 & 0.0842 & 0.1235 \end{bmatrix}$	$\begin{bmatrix} 1.2019 & 0.3018 & 0.1311 \\ 0.3806 & 2.0324 & 0.5743 \\ 0.0953 & 0.2099 & 1.2024 \end{bmatrix}$
FLQ	$\begin{bmatrix} 0.1076 & 0.1085 & 0.0301 \\ 0.1406 & 0.4076 & 0.2228 \\ 0.0369 & 0.0723 & 0.1061 \end{bmatrix}$	$\begin{bmatrix} 1.1598 & 0.2241 & 0.0950 \\ 0.3025 & 1.7994 & 0.4587 \\ 0.0724 & 0.1548 & 1.1598 \end{bmatrix}$

FLQA	$\begin{bmatrix} 0.1076 & 0.1109 & 0.0301 \\ 0.1406 & 0.4163 & 0.2228 \\ 0.0369 & 0.0739 & 0.1061 \end{bmatrix}$	$\begin{bmatrix} 1.1613 & 0.2328 & 0.0972 \\ 0.3077 & 1.8307 & 0.4667 \\ 0.0734 & 0.1609 & 1.1613 \end{bmatrix}$
RPC	$\begin{bmatrix} 0.1155 & 0.1199 & 0.0310 \\ 0.1350 & 0.4097 & 0.2145 \\ 0.0396 & 0.0837 & 0.1192 \end{bmatrix}$	$\begin{bmatrix} 1.1738 & 0.2531 & 0.1029 \\ 0.2978 & 1.8187 & 0.4533 \\ 0.0811 & 0.1841 & 1.1830 \end{bmatrix}$
Using Round's $\mathbf{A}^r = \mathbf{A}^n \hat{\rho}^r$		
LQ	$\begin{bmatrix} 0.1263 & 0.1324 & 0.0346 \\ 0.1530 & 0.4687 & 0.2483 \\ 0.0414 & 0.0882 & 0.1272 \end{bmatrix}$	$\begin{bmatrix} 1.2080 & 0.3242 & 0.1401 \\ 0.3933 & 2.0810 & 0.6077 \\ 0.0971 & 0.2257 & 1.2138 \end{bmatrix}$
CIQ	$\begin{bmatrix} 0.1263 & 0.1285 & 0.0361 \\ 0.1530 & 0.4687 & 0.2483 \\ 0.0433 & 0.0856 & 0.1272 \end{bmatrix}$	$\begin{bmatrix} 1.2066 & 0.3143 & 0.1394 \\ 0.3935 & 2.0751 & 0.6068 \\ 0.0984 & 0.2192 & 1.2122 \end{bmatrix}$
FLQ	$\begin{bmatrix} 0.1086 & 0.1105 & 0.0310 \\ 0.1418 & 0.4148 & 0.2295 \\ 0.0373 & 0.0736 & 0.1093 \end{bmatrix}$	$\begin{bmatrix} 1.1629 & 0.2321 & 0.1003 \\ 0.3110 & 1.8281 & 0.4819 \\ 0.0744 & 0.1608 & 1.1668 \end{bmatrix}$
FLQA	$\begin{bmatrix} 0.1086 & 0.1128 & 0.0310 \\ 0.1418 & 0.4237 & 0.2295 \\ 0.0373 & 0.0752 & 0.1093 \end{bmatrix}$	$\begin{bmatrix} 1.1645 & 0.2413 & 0.1028 \\ 0.3166 & 1.8611 & 0.4906 \\ 0.0754 & 0.1672 & 1.1685 \end{bmatrix}$
RPC	$\begin{bmatrix} 0.1165 & 0.1221 & 0.0319 \\ 0.1361 & 0.4169 & 0.2209 \\ 0.0400 & 0.0851 & 0.1228 \end{bmatrix}$	$\begin{bmatrix} 1.1772 & 0.2623 & 0.1089 \\ 0.3064 & 1.8488 & 0.4768 \\ 0.0834 & 0.1914 & 1.1912 \end{bmatrix}$

	Total Intraregional Intermediate Inputs	Percentage Differences[a]	Average Percentage Difference[b]
Survey	**0.3022 0.6199 0.3786**		
Using $\mathbf{A}^n$			
LQ	0.3180 0.6773 0.3982	5.24 9.25 5.17	6.55
CIQ	0.3198 0.6710 0.3997	5.84 8.23 5.56	6.55
FLQ	0.2852 0.5884 0.3591	-5.63 -5.08 -5.15	-5.29
FLQA	0.2852 0.6010 0.3591	-5.62 -3.05 -5.15	-4.61
RPC	0.2901 0.6133 0.3646	-4.00 -1.08 -3.69	-2.92
Using Round's $\mathbf{A}^r = \mathbf{A}^n \hat{\rho}^r$			
LQ	0.3208 0.6892 0.4102	6.15 11.18 8.34	8.56
CIQ	0.3226 0.6829 0.4117	6.76 10.15 8.74	8.55
FLQ	0.2876 0.5989 0.3699	-4.81 -3.40 -2.30	-3.50
FLQA	0.2876 0.6117 0.3699	-4.81 -1.34 -2.30	-2.82

RPC	0.2926 0.6241 0.3756	-3.17 0.67 -0.79	1.54[c]

	Intraregional Output Multipliers	**Percentage Differences**[d]	**Average Percentage Difference**
Survey	**1.5846 2.3108 1.7841**		
Using $\mathbf{A}^n$			
LQ	1.6778 2.5651 1.9108	5.88 11.01 7.10	8.00
CIQ	1.6779 2.5441 1.9078	5.89 10.10 6.94	7.64
FLQ	1.5347 2.1784 1.7135	-3.15 -5.73 -3.96	-4.28
FLQA	1.5425 2.2245 1.7252	-2.66 -3.73 -3.30	-3.23
RPC	1.5527 2.2559 1.7392	-2.01 -2.37 -2.51	-2.30
Using Round's $\mathbf{A}^r = \mathbf{A}^n\hat{\rho}^r$			
LQ	1.6584 2.6309 1.9617	7.18 13.85 9.96	10.33
CIQ	1.6985 2.6087 1.9584	7.18 12.89 9.77	9.95
FLQ	1.5482 2.2210 1.7419	-2.30 -3.88 -1.96	-2.71
FLQA	1.5565 2.2696 1.7619	-1.78 -1.78 -1.24	-1.60
RPC	1.5670 2.3025 1.7769	-1.11 -0.36 -0.40	-0.62

[a] This is $\{[(\mathbf{i}'\tilde{\mathbf{A}} - \mathbf{i}'\mathbf{A}) \oslash \mathbf{i}'\mathbf{A}] \times 100\}$, where "$\oslash$" indicates element-by-element division.

[b] This is a simple, unweighted average. Various kinds of weightings (e.g., using some measure of the size of each sector) are frequently used.

[c] This is the average of the absolute values of the differences, so that the negatives and positives do not cancel out.

[d] Calculated as $\{[(\mathbf{i}'\tilde{\mathbf{L}} - \mathbf{i}'\mathbf{L}) \oslash \mathbf{i}'\mathbf{L}] \times 100\}$.

Results for Region 3 (Rest of China) using 2003 Chinese IRIO data.

	Intraregional Input Coefficients	**Leontief Inverse**
Survey	$\begin{bmatrix} 0.1356 & 0.1494 & 0.0329 \\ 0.1050 & 0.3176 & 0.1945 \\ 0.0364 & 0.1016 & 0.1122 \end{bmatrix}$	$\begin{bmatrix} 1.1950 & 0.2773 & 0.1050 \\ 0.2046 & 1.5624 & 0.3498 \\ 0.0725 & 0.1902 & 1.1707 \end{bmatrix}$
Using $\mathbf{A}^n$		
LQ	$\begin{bmatrix} 0.1311 & 0.1362 & 0.0352 \\ 0.1293 & 0.3925 & 0.2055 \\ 0.0429 & 0.0905 & 0.1290 \end{bmatrix}$	$\begin{bmatrix} 1.1992 & 0.2861 & 0.1159 \\ 0.2853 & 1.7742 & 0.4301 \\ 0.0887 & 0.1984 & 1.1984 \end{bmatrix}$
CIQ	$\begin{bmatrix} 0.1311 & 0.1362 & 0.0352 \\ 0.1015 & 0.3925 & 0.1789 \\ 0.0387 & 0.0905 & 0.1290 \end{bmatrix}$	$\begin{bmatrix} 1.1886 & 0.2822 & 0.1059 \\ 0.2210 & 1.7506 & 0.3684 \\ 0.0757 & 0.1944 & 1.1910 \end{bmatrix}$
FLQ	$\begin{bmatrix} 0.1057 & 0.1288 & 0.0247 \\ 0.0643 & 0.2485 & 0.1133 \\ 0.0245 & 0.0772 & 0.0938 \end{bmatrix}$	$\begin{bmatrix} 1.1341 & 0.2001 & 0.0559 \\ 0.1030 & 1.3661 & 0.1735 \\ 0.0394 & 0.1218 & 1.1198 \end{bmatrix}$

FLQA	$\begin{bmatrix} 0.1252 & 0.1288 & 0.0272 \\ 0.0762 & 0.2485 & 0.1250 \\ 0.0290 & 0.0772 & 0.1035 \end{bmatrix}$	$\begin{bmatrix} 1.1632 & 0.2059 & 0.0640 \\ 0.1260 & 1.3723 & 0.1951 \\ 0.0485 & 0.1248 & 1.1343 \end{bmatrix}$
RPC	$\begin{bmatrix} 0.1223 & 0.1270 & 0.0328 \\ 0.1263 & 0.3835 & 0.2008 \\ 0.0397 & 0.0837 & 0.1193 \end{bmatrix}$	$\begin{bmatrix} 1.1810 & 0.2572 & 0.1026 \\ 0.2676 & 1.7322 & 0.4048 \\ 0.0786 & 0.1762 & 1.1786 \end{bmatrix}$
Using Round's $\mathbf{A}^r = \mathbf{A}^n \hat{\rho}^r$		
LQ	$\begin{bmatrix} 0.1251 & 0.1295 & 0.0335 \\ 0.1234 & 0.3732 & 0.1957 \\ 0.0409 & 0.0860 & 0.1228 \end{bmatrix}$	$\begin{bmatrix} 1.1844 & 0.2587 & 0.1029 \\ 0.2583 & 1.7022 & 0.3895 \\ 0.0806 & 0.1790 & 1.1830 \end{bmatrix}$
CIQ	$\begin{bmatrix} 0.1251 & 0.1295 & 0.0335 \\ 0.0969 & 0.3732 & 0.1703 \\ 0.0369 & 0.0860 & 0.1228 \end{bmatrix}$	$\begin{bmatrix} 1.1754 & 0.2557 & 0.0945 \\ 0.2005 & 1.6827 & 0.3344 \\ 0.0691 & 0.1758 & 1.1768 \end{bmatrix}$
FLQ	$\begin{bmatrix} 0.1009 & 0.1224 & 0.0235 \\ 0.0614 & 0.2363 & 0.1078 \\ 0.0234 & 0.0734 & 0.0893 \end{bmatrix}$	$\begin{bmatrix} 1.1262 & 0.1855 & 0.0510 \\ 0.0956 & 1.3402 & 0.1612 \\ 0.0366 & 0.1128 & 1.1123 \end{bmatrix}$
FLQA	$\begin{bmatrix} 0.1195 & 0.1224 & 0.0259 \\ 0.0727 & 0.2363 & 0.1190 \\ 0.0277 & 0.0734 & 0.0985 \end{bmatrix}$	$\begin{bmatrix} 1.1533 & 0.1905 & 0.0583 \\ 0.1168 & 1.3455 & 0.1809 \\ 0.0449 & 0.1154 & 1.1258 \end{bmatrix}$
RPC	$\begin{bmatrix} 0.1167 & 0.1207 & 0.0312 \\ 0.1206 & 0.3646 & 0.1912 \\ 0.0379 & 0.0796 & 0.1136 \end{bmatrix}$	$\begin{bmatrix} 1.1679 & 0.2334 & 0.0915 \\ 0.2432 & 1.6661 & 0.3679 \\ 0.0717 & 0.1596 & 1.1651 \end{bmatrix}$

	Total Intraregional Intermediate Inputs			**Percentage Differences**[a]			**Average Percentage Difference**[b]
Survey	**0.2771**	**0.5687**	**0.3396**				
Using $\mathbf{A}^n$							
LQ	0.3033	0.6192	0.3696	9.47	8.88	8.85	9.07
CIQ	0.2713	0.6192	0.3430	-2.07	8.88	1.02	3.99[c]
FLQ	0.1945	0.4545	0.2317	-29.81	-20.08	-31.76	-27.22
FLQA	0.2304	0.4545	0.2557	-16.84	-20.08	-24.70	-20.54
RPC	0.2883	0.5942	0.3529	4.05	4.49	3.92	4.15
Using Round's $\mathbf{A}^r = \mathbf{A}^n \hat{\rho}^r$							
LQ	0.2895	0.5887	0.3519	4.47	3.52	3.64	3.88
CIQ	0.2590	0.5887	0.3266	-6.54	3.52	-3.81	4.62
FLQ	0.1856	0.4321	0.2206	-33.02	-24.02	-35.03	-30.69

FLQA	0.2199 0.4321 0.2434	-20.63 -24.02 -28.31	-24.32
RPC	0.2751 0.5649 0.3360	-0.70 -0.66 -1.05	-0.81

	Intraregional Output Multipliers	**Percentage Differences**[d]	**Average Percentage Difference**
Survey	**1.4721 2.0299 1.6256**		
Using $\mathbf{A}^n$			
LQ	1.5732 2.2587 1.7444	6.87 11.27 7.31	8.48
CIQ	1.4853 2.2272 1.6654	0.90 9.72 2.45	4.36
FLQ	1.2765 1.6880 1.3492	-13.29 -16.84 -17.00	-15.71
FLQA	1.3377 1.7031 1.3934	-9.13 -16.10 -14.28	-13.17
RPC	1.5272 2.1656 1.6860	3.75 6.68 3.72	4.72
Using Round's $\mathbf{A}^r = \mathbf{A}^n \hat{\rho}^r$			
LQ	1.5233 2.1400 1.6754	3.48 5.42 3.07	3.99
CIQ	1.4450 2.1142 1.6057	-1.84 4.15 -1.22	2.41[c]
FLQ	1.2584 1.6384 1.3245	-14.51 -19.29 -18.52	-17.44
FLQA	1.3150 1.6514 1.3651	-10.67 -18.65 -16.02	-15.11
RPC	1.4828 2.0591 1.6244	0.73 1.44 -0.07	0.75[c]

[a] This is $\{[(\mathbf{i}'\tilde{\mathbf{A}} - \mathbf{i}'\mathbf{A}) \oslash \mathbf{i}'\mathbf{A}] \times 100\}$, where "$\oslash$" indicates element-by-element division.

[b] This is a simple, unweighted average. Various kinds of weightings (e.g., using some measure of the size of each sector) are frequently used.

[c] This is the average of the absolute values of the differences, so that the negatives and positives do not cancel out.

[d] Calculated as $\{[(\mathbf{i}'\tilde{\mathbf{L}} - \mathbf{i}'\mathbf{L}) \oslash \mathbf{i}'\mathbf{L}] \times 100\}$.

Computational Notes

We begin by presuming that the full 3-region, 3-sector IRIO table is saved in the APL workspace as **Z** along with the IRIO vector of total outputs **x**. We generate the matrix **S** to aggregate **Z** to the national table `ZN` and **x** to the vector of national total outputs `xn`, from which we can generate `AN`, the matrix of national technical coefficients, and the associated Leontief inverse `LN`.

```
     S←I,I,I←3 3ρ1,3ρ0
     xn←S+.×x ⋄ ZN←S+.×Z+.×⍉S ⋄ LN←LINV AN←ZN AMAT xn

xn
     56178    151923     47588
ZN
      7366     20687      1673
      8523     69966     11473
      2409     13749      6137
AN
    0.1311    0.1362    0.0352
    0.1517    0.4605    0.2411
    0.0429    0.0905    0.1290
LN
```

```
    1.2155    0.3304    0.1405
    0.3865    2.0490    0.5827
    0.1000    0.2291    1.2155
```

For a region r (for this problem either 2 or 3), we can generate the relevant indexes in **Z** and **x** to retrieve `ZR` and `xr` in order to subsequently compute `AR` and the associated Leontief invers `LR`. Also, for calculations including the "Round adjustment," we calculate the necessary adjustment factor `rho` and modify `AN` by it to produce `ANR` and the associated Leontief inverse `LANR`.

```
     rx←(3×r-1)+⍳3
     xr←x[rx] ⋄ ZR←Z[rx;rx] ⋄ LR←LINV AR←ZR AMAT xr
     LANR←LINV ANR←AN+.×DIAG rho←((+⌿Z[;rx])÷xr)÷((+⌿ZN)÷xn)
```

Applying these expressions for region 2,

```
rx
4 5 6
xr
     27866     81253     23667
ZR
      3564      8828       806
      3757     34931      5186
      1099      6613      2969
AR
    0.1279    0.1086    0.0340
    0.1348    0.4299    0.2191
    0.0394    0.0814    0.1255
LR
    1.1889    0.2418    0.1069
    0.3130    1.8828    0.4839
    0.0827    0.1861    1.1933
rho
1.0086662 1.0177012 1.0301156
LANR
    1.2206    0.3443    0.1496
    0.3999    2.0932    0.6162
    0.1033    0.2395    1.2261
```

Applying the expressions for region 3,

```
rx
7 8 9
xr
     11661     21107      8910
ZR
      1581      3154       293
      1225      6704      1733
       425      2145      1000
AR
    0.1356    0.1494    0.0329
    0.1050    0.3176    0.1945
    0.0364    0.1016    0.1122
LR
```

```
    1.1950    0.2773    0.1050
    0.2046    1.5624    0.3498
    0.0725    0.1902    1.1707
rho
0.95436354 0.95073725 0.95214849
LANR
    1.1975    0.2946    0.1228
    0.3451    1.9380    0.5203
    0.0897    0.2038    1.1967
```

To solve this problem efficiently, for convenience, we can define APL functions to generate regional estimates using the *LQ*, *CIQ*, *FLQ*, *AFLQ*, and *RPC* methods:

```
[  0]  slq←xr fslq xn
[  1] ⍝compute vector of simple location quotients
[  2]  slq←(xr÷+/xr)÷(xn÷+/xn)

[  0]  CIQ←xr fciqx xn;slq;I;n
[  1] ⍝compute matrix of cross industry quotients CIQ
[  2] ⍝and matrix of CIQ with slq on diagonal
[  3]  CIQX←(xr÷xn)∘.÷(xr÷xn) ⋄ slq←xr fslq xn ⋄ I←(2⍴n)⍴1,(n←3)⍴0
[  4]  CIQ←(DIAG slq)+CIQX×~I

[  0]  FLQ←xr fflq xn;delta;lamda
[  1] ⍝compute matrix of Flagg quotients
[  2]  delta←0.3 ⋄ lamda←(2⍟1+(+/xr)÷+/xn)*delta
[  3]  FLQ←lamda×xr fciqx xn

[  0]  AFLQ←xr faflq xn;FLQ;slq;slqx;slqxx
[  1] ⍝compute matrix of adjusted Flagg quotients (AFLQ)
[  2]  FLQ←xr fflq xn ⋄ slq←xr fslq xn
[  3]  AFLQ←FLQ+.×DIAG slqxx←(slqx≤1)+(slqx>1)×slqx←((2⍟1+slq))

[  0]  RPC←ZR frpc Z;nn
[  1] ⍝compute vector of regional purchase coefficients (RPC)
[  2]  nn←⍴ZR
[  3]  RPC←(⍉nn⍴+/ZR)÷⍉nn⍴+⌿nn⍴+/Z[;rx]
```

We use these functions to compute the nonsurvey estimates using the *LQ*, *CIQ*, *FLQ*, *AFLQ*, and *RPC* techniques to estimate the regional technical coefficients for each as `ALQ`, `ACQ`, `AF`, `AFA`, and `ARPC`, respectively, and their associated Leontief inverses as `LALQ`, `LACQ`, `LAF`, `LAFA`, and `LARPC`. We collect these calculations in the niladic (no arguments) function `NSALL` that assumes `AN` and `xr` are defined as global variables.

```
[  0]  NSALL;ARR1;ARR2;n;nn
[  1] ⍝compute A and L for all nonsurvey methods
[  2] ⍝Global vars: AN, xn, ZR, xr
[  3]  nn←2⍴n←1↑⍴AN
[  4] ⍝SLQ
[  5]  slqx←(slq≥1)+(slq<1)×slq←xr fslq xn
[  6]  LQ←⍉nn⍴slq
[  7]  LALQ←LINV ALQ←(DIAG slqx)+.×AN
```

```
[  8] ⍝CIQ
[  9]  ARR1←(AN×(CQ≥1))+AN×CQ×((CQ←xr fciqx xn)<1)
[ 10]  ARR2←(AN×nn⍴slq≥1)+AN×nn⍴slq×slq<1
[ 11]  LACQ←LINV ACQ←(ARR2×I)+ARR1×~I←(2⍴n)⍴1,(n←⍴xn)⍴0
[ 12] ⍝FLQ
[ 13]  LAF←LINV AF←(AN×(F≥1))+AN×F×((F←xr fflq xn)<1)
[ 14] ⍝FLQA
[ 15]  FA←(FF×nn⍴slq≤1)+((FF←xr faflq xn)×nn⍴slq>1)
[ 16]  LAFA←LINV AFA←(FF×AN×nn⍴slq≤1)+(AN×(FF←xr faflq xn)×nn⍴slq>1)
[ 17] ⍝RPC
[ 18]  LARPC←LINV ARPC←AN×RPC←ZR frpc Z
```

Applying NSALL for region 2.

```
ASLQ,LALQ
    0.1830    0.0668    0.0087    1.2033    0.3113    0.1317
    0.1377    0.3070    0.0707    0.3804    2.0378    0.5751
    0.1794    0.2074    0.2581    0.0940    0.2161    1.2039
ACQ,LACQ
    0.1252    0.1263    0.0351    1.2019    0.3018    0.1311
    0.1517    0.4605    0.2411    0.3806    2.0324    0.5743
    0.0429    0.0842    0.1235    0.0953    0.2099    1.2024
AF,LAF
    0.1076    0.1085    0.0301    1.1598    0.2241    0.0950
    0.1406    0.4076    0.2228    0.3025    1.7994    0.4587
    0.0369    0.0723    0.1061    0.0724    0.1548    1.1598
AFA,LAFA
    0.1076    0.1109    0.0301    1.1613    0.2328    0.0972
    0.1406    0.4163    0.2228    0.3077    1.8307    0.4667
    0.0369    0.0739    0.1061    0.0734    0.1609    1.1613
ARPC,LARPC
    0.1155    0.1199    0.0310    1.1738    0.2531    0.1029
    0.1350    0.4097    0.2145    0.2978    1.8187    0.4533
```

Applying NSALL for region 2 with the Round adjustment.

```
ASLQ,LALQ
    0.1830    0.0668    0.0087    1.2080    0.3242    0.1401
    0.1377    0.3070    0.0707    0.3933    2.0810    0.6077
    0.1794    0.2074    0.2581    0.0971    0.2257    1.2138
ACQ,LACQ
    0.1263    0.1285    0.0361    1.2066    0.3143    0.1394
    0.1530    0.4687    0.2483    0.3935    2.0751    0.6068
    0.0433    0.0856    0.1272    0.0984    0.2192    1.2122
AF,LAF
    0.1086    0.1105    0.0310    1.1629    0.2321    0.1003
    0.1418    0.4148    0.2295    0.3110    1.8281    0.4819
    0.0373    0.0736    0.1093    0.0744    0.1608    1.1668
AFA,LAFA
    0.1086    0.1128    0.0310    1.1645    0.2413    0.1028
    0.1418    0.4237    0.2295    0.3166    1.8611    0.4906
    0.0373    0.0752    0.1093    0.0754    0.1672    1.1685
```

```
ARPC,LARPC
    0.1165     0.1221     0.0319     1.1772     0.2623     0.1089
    0.1361     0.4169     0.2209     0.3064     1.8488     0.4768
    0.0400     0.0851     0.1228     0.0834     0.1914     1.1912
```

Problem 10.10: RAS Estimate of Washington State from US National Data

This exercise problem applies the RAS technique to generate a matrix of technical coefficients for the state of Washington using the U.S. matrix of technical coefficients as a starting point.

Problem 10.10 Overview

The following are the 1997 matrix of technical coefficients and vector of total outputs for the State of Washington as well as the 2003 matrix of technical coefficients for the United States, where the sectors are defined as (1) agriculture, (2) mining, (3) construction, (4) manufacturing, (5) trade, transportation and utilities, (6) services, and (7) other:

$$\mathbf{A}^W = \begin{bmatrix} .1154 & .0012 & .0082 & .0353 & .0019 & .0033 & .0016 \\ .0008 & .0160 & .0057 & .0014 & .0022 & .0002 & .0001 \\ .0072 & .0084 & .0066 & .0043 & .0074 & .0196 & .0133 \\ .0868 & .0287 & .0958 & .0766 & .0289 & .0244 & .0205 \\ .0625 & .0278 & .0540 & .0525 & .0616 & .0317 & .0480 \\ .0964 & .1207 & .0704 & .0596 & .1637 & .1991 & .2224 \\ .0020 & .0031 & .0056 & .0019 & .0045 & .0051 & .0066 \end{bmatrix} \quad \mathbf{x}^W = \begin{bmatrix} 7{,}681.0 \\ 581.7 \\ 17{,}967.1 \\ 77{,}483.7 \\ 56{,}967.2 \\ 109{,}557.6 \\ 4{,}165.5 \end{bmatrix}$$

$$\mathbf{A}^{US} = \begin{bmatrix} .2225 & .0000 & .0012 & .0375 & .0001 & .0020 & .0010 \\ .0021 & .1360 & .0072 & .0453 & .0311 & .0003 & .0053 \\ .0034 & .0002 & .0012 & .0021 & .0035 & .0071 & .0214 \\ .1724 & .0945 & .2488 & .3204 & .0468 & .0572 & .1004 \\ .0853 & .0527 & .0912 & .0950 & .0643 & .0314 & .0526 \\ .0902 & .1676 & .1339 & .1261 & .1655 & .2725 & .1882 \\ .0101 & .0140 & .0103 & .0214 & .0206 & .0200 & .0247 \end{bmatrix}$$

To examine application of the RAS technique to estimate the Washington State table using the U.S. matrix of technical coefficients as a starting point, we first compute $\mathbf{Z}^W = \mathbf{A}^W \hat{\mathbf{x}}^W$ and then the vectors of total intermediate inputs and outputs for the real Washington State table:

$$\mathbf{u}(1) = \mathbf{Z}^W \mathbf{i} = \begin{bmatrix} 4{,}245.9 & 369.4 & 3{,}140.1 & 12{,}737.6 & 12{,}718 & 38{,}753.8 & 1{,}112.4 \end{bmatrix}'$$

$$\mathbf{v}(1) = (\mathbf{i}'\mathbf{Z}^W)' = \begin{bmatrix} 2{,}849.7 & 119.8 & 4{,}423 & 17{,}945.8 & 15{,}384.7 & 31{,}052.5 & 1{,}301.7 \end{bmatrix}'$$

Applying RAS using $\mathbf{A}^{US}$, $\mathbf{u}(1)$, $\mathbf{v}(1)$, and $\mathbf{x}^W$, the estimated matrix of technical coefficients for

Washington State is $^{US}\mathbf{A}^{W} = \begin{bmatrix} .2078 & .0000 & .0013 & .0299 & .0002 & .0027 & .0013 \\ .0001 & .0099 & .0005 & .0022 & .0031 & .0000 & .0005 \\ .0061 & .0004 & .0025 & .0032 & .0109 & .0177 & .0571 \\ .0526 & .0362 & .0883 & .0836 & .0246 & .0243 & .0456 \\ .0534 & .0415 & .0664 & .0508 & .0694 & .0273 & .0490 \\ .0493 & .1151 & .0852 & .0589 & .1561 & .2070 & .1531 \\ .0016 & .0028 & .0019 & .0029 & .0057 & .0044 & .0059 \end{bmatrix}$.

The mean absolute deviation between the estimated and actual Washington State matrices of technical coefficients is $MAD = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left|^{US}a_{ij}^{W} - a_{ij}^{W}\right| = 0.0098$.

Computational Notes

Retrieve the US national matrix of interindustry transactions `ZN` and vector of total outputs as `xn`. Also retrieve Washington state matrix of regional interindustry transactions as `ZR` and vector of total outputs as `xr`. We can then compute the corresponding matrices of technical coefficients, `AN` and `AR`, and the vectors of regional intermediate outputs and inputs, `ur` and `vr`, respectively. Finally, we use the function `RAS` to produce a nonsurvey estimate of `AR` using with `AN` specified as the left argument and as the right argument a matrix with `ur`, `vr`, and `xr` as the rows. We save explicit result as `ARAS` and compute the mean absolute deviations of `ARAS` with `AR` and save the result as `MADRAS`.

```
      AN←ZN AMAT xn ⋄ AR←ZR AMAT xr
      ur←+/ZR ⋄ vr←+⌿ZR
      MADRAS←AR MAD ARAS←AN RAS 3 3⍴ur,vr,xr

AN
      0.222     0.000     0.001     0.037     0.000     0.002     0.001
      0.002     0.136     0.007     0.045     0.031     0.000     0.005
      0.003     0.000     0.001     0.002     0.004     0.007     0.021
      0.172     0.094     0.249     0.320     0.047     0.057     0.100
      0.085     0.053     0.091     0.095     0.064     0.031     0.053
      0.090     0.168     0.134     0.126     0.166     0.272     0.188
      0.010     0.014     0.010     0.021     0.021     0.020     0.025
AR
      0.115     0.001     0.008     0.035     0.002     0.003     0.002
      0.001     0.016     0.006     0.001     0.002     0.000     0.000
      0.007     0.008     0.007     0.004     0.007     0.020     0.013
      0.087     0.029     0.096     0.077     0.029     0.024     0.021
      0.063     0.028     0.054     0.053     0.062     0.032     0.048
      0.096     0.121     0.070     0.060     0.164     0.199     0.222
      0.002     0.003     0.006     0.002     0.004     0.005     0.007
xr
    7681.0     581.7   17967.1   77483.7   56967.2  109557.6    4165.5
ur
    4245.9     369.4    3140.1   12737.6   12718.0   38753.8    1112.4
```

```
vr
    2849.7      119.8      4423.0    17945.8    15384.7    31052.5     1301.7
ARAS
     0.208      0.000      0.001      0.030      0.000      0.003      0.001
     0.000      0.010      0.000      0.002      0.003      0.000      0.000
     0.006      0.000      0.003      0.003      0.011      0.018      0.057
     0.053      0.036      0.088      0.084      0.025      0.024      0.046
     0.053      0.041      0.066      0.051      0.069      0.027      0.049
     0.049      0.115      0.085      0.059      0.156      0.207      0.153
     0.002      0.003      0.002      0.003      0.006      0.004      0.006

MADRAS
0.009841638
```

Problem 10.11: Refining Washington State Estimation with Additional Information

This exercise extends the estimation considered in exercise Problem 10.10, but in the case of selected additional available information

Problem 10.11 Overview

Presuming that while we do not know all the technical coefficients for the Washington State economy, $\mathbf{A}^W$, we do know several, namely a_{11}^W, a_{62}^W and a_{65}^W. To use the RAS technique incorporating that we know these coefficients, we begin with defining a matrix of the exogenously specified technical coefficients: $\mathbf{K} = \begin{bmatrix} .1154 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .1207 & 0 & 0 & .1637 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$. The corresponding modified vectors of total intermediate inputs and outputs for the real Washington State table are found as: $\mathbf{u}(1) = \mathbf{x}^W - \mathbf{Z}^W\mathbf{i} - \mathbf{K}\mathbf{x}^W =$

$[3{,}359.7 \quad 369.4 \quad 3{,}140.1 \quad 12{,}737.6 \quad 12{,}718 \quad 29{,}359.6 \quad 1{,}112.4]'$ and $\mathbf{v}(1) = \mathbf{x}^W - \mathbf{i}'\mathbf{Z}^W - \mathbf{i}'\mathbf{K}\hat{\mathbf{x}}^W =$

$[1{,}963.5 \quad 49.6 \quad 4{,}423 \quad 17{,}945.8 \quad 6{,}060.7 \quad 31{,}052.5 \quad 1{,}301.7]'$.

Applying RAS using $\mathbf{A}^{(US)}$, $\mathbf{u}(1)$, $\mathbf{v}(1)$, and $\mathbf{x}^W$, and, this time, $\mathbf{K}$, the new estimated matrix of technical coefficients for Washington State is

$$ {}^{US}\mathbf{A}^{W} = \begin{bmatrix} .1154 & .0001 & .0017 & .0377 & .0002 & .0035 & .0017 \\ .0002 & .0096 & .0005 & .0023 & .0030 & .0000 & .0005 \\ .0096 & .0004 & .0025 & .0031 & .0100 & .0180 & .0576 \\ .0830 & .0337 & .0890 & .0809 & .0229 & .0248 & .0463 \\ .0850 & .0389 & .0675 & .0496 & .0649 & .0281 & .0502 \\ .0752 & .1207 & .0830 & .0551 & .1637 & .2044 & .1502 \\ .0026 & .0026 & .0020 & .0029 & .0053 & .0046 & .0060 \end{bmatrix} $$

. The mean absolute deviation between the estimated and actual Washington State matrices of technical coefficients is $MAD = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left| {}^{US}a_{ij}^{W} - a_{ij}^{W} \right| = 0.0066$. In this case the constrained RAS procedure incorporating exogenous information improves the estimate considerably over the unconstrained case in Problem 10.10.

Computational Notes

Using the data from Problem 10.10, we retrieve `AN`, `AR`, `xr`, `ur`, and `vr` and repeat the estimation of `AR` using the function `RAS`, which we save for later comparison as `ARAS1` along with the mean absolution deviation comparing `ARAS1` with `AR`.

```
      MADRAS1←AR MAD ARAS1←AN RAS 3 7ρur,vr,xr

AN
    0.2225     0.0000     0.0012     0.0375     0.0001     0.0020     0.0010
    0.0021     0.1360     0.0072     0.0453     0.0311     0.0003     0.0053
    0.0034     0.0002     0.0012     0.0021     0.0035     0.0071     0.0214
    0.1724     0.0945     0.2488     0.3204     0.0468     0.0572     0.1004
    0.0853     0.0527     0.0912     0.0950     0.0643     0.0314     0.0526
    0.0902     0.1676     0.1339     0.1261     0.1655     0.2725     0.1882
    0.0101     0.0140     0.0103     0.0214     0.0206     0.0200     0.0247
AR
    0.1154     0.0012     0.0082     0.0353     0.0019     0.0033     0.0016
    0.0008     0.0160     0.0057     0.0014     0.0022     0.0002     0.0001
    0.0072     0.0084     0.0066     0.0043     0.0074     0.0196     0.0133
    0.0868     0.0287     0.0958     0.0766     0.0289     0.0244     0.0205
    0.0625     0.0278     0.0540     0.0525     0.0616     0.0317     0.0480
    0.0964     0.1207     0.0704     0.0596     0.1637     0.1991     0.2224
    0.0020     0.0031     0.0056     0.0019     0.0045     0.0051     0.0066
xr
    7681.0      581.7    17967.1    77483.7    56967.2   109557.6     4165.5
ur
    4245.9      369.4     3140.1    12737.6    12718.0    38753.8     1112.4
vr
    2849.7      119.8     4423.0    17945.8    15384.7    31052.5     1301.7
ARAS1
    0.2078     0.0000     0.0013     0.0299     0.0002     0.0027     0.0013
    0.0001     0.0099     0.0005     0.0022     0.0031     0.0000     0.0005
    0.0061     0.0004     0.0025     0.0032     0.0109     0.0177     0.0571
    0.0526     0.0362     0.0883     0.0836     0.0246     0.0243     0.0456
```

```
    0.0534     0.0415     0.0664     0.0508     0.0694     0.0273     0.0490
    0.0493     0.1151     0.0852     0.0589     0.1561     0.2070     0.1531
    0.0016     0.0028     0.0019     0.0029     0.0057     0.0044     0.0059

MADRAS1
0.009841638
```

We also specify the matrix of the locations of additional exogenous information **K** and replace the corresponding elements of `AN` with the specified exogenous information, calling the result `AA`, and net out the impact of those coefficients on `ur` and `vr` (denoted as `ub` and `vb`). Then use the `RAS` function with `AA` as the left argument and as the right argument a matrix with `ub`, `vb`, and `xb` as its rows. We save the result as `ARAS2` and compute mean absolution deviation comparing `ARAS2` with `AR`.

```
      KK←7 7⍴0 ⋄ KK[1;1]←KK[6;2]←KK[6;5]←1 ⋄ K←KK×AR ⋄ AB←(KK≠1)×AN
      ub←ur-+/K+.×DIAG xr ⋄ vb←vr-+⌿K+.×DIAG xr
      MADRAS2←AR MAD ARAS2←K+AB RAS 3 7⍴ub,vb,xr

K
    0.1154     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
    0.0000     0.1207     0.0000     0.0000     0.1637     0.0000     0.0000
    0.0000     0.0000     0.0000     0.0000     0.0000     0.0000     0.0000
ub
    3359.7      369.4     3140.1    12737.6    12718.0    29359.6     1112.4
vb
    1963.5       49.6     4423.0    17945.8     6060.7    31052.5     1301.7
ARAS2
    0.1154     0.0001     0.0017     0.0377     0.0002     0.0035     0.0017
    0.0002     0.0096     0.0005     0.0023     0.0030     0.0000     0.0005
    0.0096     0.0004     0.0025     0.0031     0.0100     0.0180     0.0576
    0.0830     0.0337     0.0890     0.0809     0.0229     0.0248     0.0463
    0.0850     0.0389     0.0675     0.0496     0.0649     0.0281     0.0502
    0.0752     0.1207     0.0830     0.0551     0.1637     0.2044     0.1502
    0.0026     0.0026     0.0020     0.0029     0.0053     0.0046     0.0060
MADRAS2
0.0066226259
```

Problem 10.12: Additional Cases of Exogenous Information

This exercise further explores use of constrained RAS estimation developed in Problem 10.11, this time assuming there is information from exogenous sources providing some alternative technical coefficients, namely a_{67}^{W}, a_{42}^{W} and a_{54}^{W} , to those provided in Problem 10.11.

Problem 10.12 Overview

The procedure is the same as the previous problem, i.e., the new exogenously specified technical coefficients are given by

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .0287 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .0525 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .2224 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

. The revised vectors of total intermediate inputs and outputs for the real Washington State table are:

$$\mathbf{u}(1) = \mathbf{x}^W - \mathbf{Z}^W \mathbf{i} - \mathbf{K}\mathbf{x}^W = [4{,}245.9 \quad 369.4 \quad 3{,}140.1 \quad 12{,}720.9 \quad 8{,}649.3 \quad 37{,}827.6 \quad 1{,}112.4]'$$

$$\mathbf{v}(1) = \mathbf{x}^W - \mathbf{i}'\mathbf{Z}^W - \mathbf{i}'\mathbf{K}\hat{\mathbf{x}}^W = [2{,}849.7 \quad 103.1 \quad 4{,}423 \quad 13{,}877.1 \quad 15{,}384.7 \quad 31{,}052.5 \quad 375.5]'$$

Applying RAS using $\mathbf{A}^{US}$, $\mathbf{u}(1)$, $\mathbf{v}(1)$, $\mathbf{x}^W$, and $\mathbf{K}$ yields the new estimated matrix of technical coefficients for Washington State:

$${}^{US}\mathbf{A}^W = \begin{bmatrix} .2088 & .0000 & .0013 & .0298 & .0002 & .0027 & .0007 \\ .0001 & .0104 & .0005 & .0022 & .0031 & .0000 & .0003 \\ .0063 & .0005 & .0026 & .0033 & .0113 & .0184 & .0329 \\ .0530 & .0287 & .0895 & .0834 & .0251 & .0247 & .0258 \\ .0527 & .0433 & .0659 & .0525 & .0692 & .0272 & .0271 \\ .0485 & .1200 & .0844 & .0575 & .1554 & .2059 & .2224 \\ .0016 & .0030 & .0019 & .0029 & .0058 & .0045 & .0033 \end{bmatrix}$$

. The mean absolute deviation between the estimated and actual Washington State matrices of technical coefficients is $MAD = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left|{}^{US}a_{ij}^W - a_{ij}^W\right| = 0.0077$ —not as good an estimate as that obtained in Problem 10.11, which resulted in $MAD = 0.0066$.

Finally, we can presume we can employ the exogenous information used in both Problems 10.10 and 10.11 in a combined case. For this combined case, the new exogenously specified technical coefficients are given by

$$\mathbf{K} = \begin{bmatrix} .1154 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .0287 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .0525 & 0 & 0 & 0 \\ 0 & .1207 & 0 & 0 & .1637 & 0 & .2224 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We once again compute the vectors of total intermediate inputs and outputs for the real Washington State table:

$$\mathbf{u}(1) = \mathbf{x}^W - \mathbf{Z}^W \mathbf{i} - \mathbf{K}\mathbf{x}^W = \begin{bmatrix} 3{,}359.7 & 369.4 & 3{,}140.1 & 12{,}720.9 & 8{,}649.3 & 28{,}433.4 & 1{,}112.4 \end{bmatrix}'$$

$$\mathbf{v}(1) = \mathbf{x}^W - \mathbf{i}'\mathbf{Z}^W - \mathbf{i}'\mathbf{K}\hat{\mathbf{x}}^W = \begin{bmatrix} 1{,}963.5 & 32.9 & 4{,}423 & 13{,}877.1 & 6{,}060.7 & 31{,}052.5 & 375.5 \end{bmatrix}'$$

Applying RAS using $\mathbf{A}^{US}$, $\mathbf{u}(1)$, $\mathbf{v}(1)$, $\mathbf{x}^W$, and $\mathbf{K}$, the new estimated matrix of technical coefficients for Washington State using both sets of exogenous information is

$${}^{US}\mathbf{A}^W = \begin{bmatrix} .1154 & .0001 & .0018 & .0376 & .0002 & .0036 & .0010 \\ .0002 & .0108 & .0005 & .0022 & .0031 & .0000 & .0003 \\ .0100 & .0005 & .0026 & .0031 & .0104 & .0187 & .0327 \\ .0847 & .0287 & .0908 & .0802 & .0234 & .0254 & .0259 \\ .0835 & .0423 & .0663 & .0525 & .0639 & .0277 & .0270 \\ .0746 & .1207 & .0822 & .0531 & .1637 & .2033 & .2224 \\ .0026 & .0030 & .0020 & .0028 & .0054 & .0047 & .0034 \end{bmatrix}$$

. The mean absolute deviation between the estimated and actual Washington State matrices of technical coefficients is $MAD = (\frac{1}{49})\sum_{i=1}^{7}\sum_{j=1}^{7}\left| {}^{US}a_{ij}^{W} - a_{ij}^{W} \right| = 0.0044$—perhaps not surprisingly better than either of the previous cases.

Computational Notes

We once again retrieve from Problem 10.10 the basic data `AN`, `AR`, `ur`, `vr`, and `xr`. We also retrieve the exogenous information **K** from Problem 10.11. We also specify a new case of exogenous information as `K2` and a third case combing **K** and `K2` as `K3`. We apply the expressions developed in Problem 10.11 to the case without exogenous information `ARAS0` and three exogenous information cases, `ARAS1`, `ARAS2`, and `ARAS3`, and the corresponding calculations of the mean absolute deviation of `AR` compared with `ARAS0`, `ARAS1`, `ARAS2`, and `ARAS3`, saved as `MADRAS0`, `MADRAS1`, `MADRAS2`, and `MADRAS3`, respectively.

```
⍝No addtl informaton
 MADRAS0←AR MAD ARAS0←AN RAS(3,n)⍴ur,vr,xr
⍝Case 1 of addtl information
 KK←7 7⍴0 ◇ KK[1;1]←KK[6;2]←KK[6;5]←1 ◇ K←KK×AR ◇ AB←(KK≠1)×AN
 ub←ur-+/K+.×DIAG xr ◇ vb←vr-+⌿K+.×DIAG xr
 MADRAS1←AR MAD ARAS1←K+AB RAS 3 7⍴ub,vb,xr
⍝Case 2 of addtl information
 KK2←7 7⍴0 ◇ KK2[6;7]←KK2[4;2]←KK2[5;4]←1 ◇ K2←KK2×AR ◇ AB←(KK2≠1)×AN
 ub2←ur-+/K2+.×DIAG xr ◇ vb2←vr-+⌿K2+.×DIAG xr
 MADRAS2←AR MAD ARAS2←K2+AB RAS 3 7⍴ub2,vb2,xr
⍝combined case
 KK3←7 7⍴0 ◇ KK3[1;1]←KK3[6;2]←KK3[6;5]←KK3[6;7]←KK3[4;2]←KK3[5;4]←1
 K3←KK3×AR ◇ AB←(KK3≠1)×AN
 ub3←ur-+/K3+.×DIAG xr ◇ vb3←vr-+⌿K3+.×DIAG xr
 MADRAS3←AR MAD ARAS3←K3+AB RAS 3 7⍴ub3,vb3,xr

MADRAS0
    0.0098
```

```
K
    0.1154    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.1207    0.0000    0.0000    0.1637    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
ub
    3359.7     369.4    3140.1   12737.6   12718.0   29359.6    1112.4
vr
    1963.5      49.6    4423.0   17945.8    6060.7   31052.5    1301.7
ARAS1
    0.1154    0.0001    0.0017    0.0377    0.0002    0.0035    0.0017
    0.0002    0.0096    0.0005    0.0023    0.0030    0.0000    0.0005
    0.0096    0.0004    0.0025    0.0031    0.0100    0.0180    0.0576
    0.0830    0.0337    0.0890    0.0809    0.0229    0.0248    0.0463
    0.0850    0.0389    0.0675    0.0496    0.0649    0.0281    0.0502
    0.0752    0.1207    0.0830    0.0551    0.1637    0.2044    0.1502
    0.0026    0.0026    0.0020    0.0029    0.0053    0.0046    0.0060
MADRAS1
0.0066226259

K2
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0287    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0525    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.2224
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
ub
    4245.9     369.4    3140.1   12720.9    8649.3   37827.6    1112.4
vb
    2849.7     103.1    4423.0   13877.1   15384.7   31052.5     375.5
ARAS2
      0.209     0.000     0.001     0.030     0.000     0.003     0.001
      0.000     0.010     0.000     0.002     0.003     0.000     0.000
      0.006     0.000     0.003     0.003     0.011     0.018     0.033
      0.053     0.029     0.090     0.083     0.025     0.025     0.026
      0.053     0.043     0.066     0.053     0.069     0.027     0.027
      0.049     0.120     0.084     0.057     0.155     0.206     0.222
      0.002     0.003     0.002     0.003     0.006     0.005     0.003
MADRAS2
0.0077221622

K3
    0.1154    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0287    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0525    0.0000    0.0000    0.0000
```

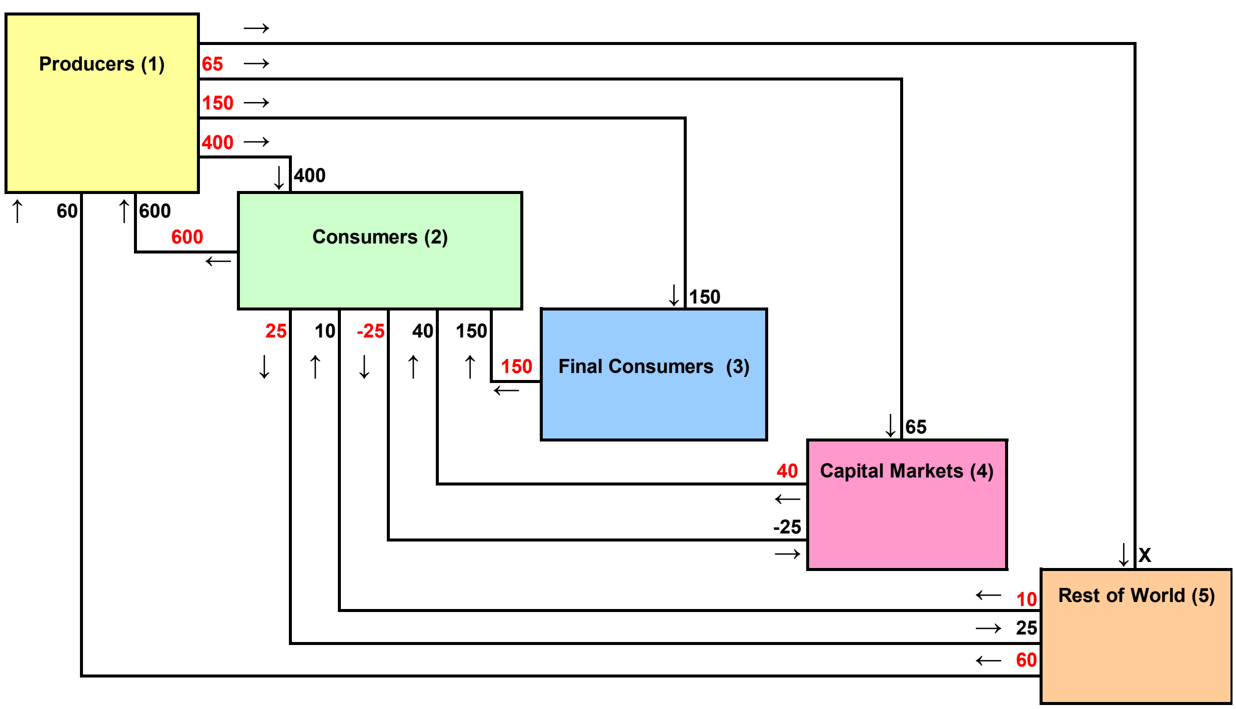
```
    0.0000    0.1207    0.0000    0.0000    0.1637    0.0000    0.2224
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
ub3
    3359.7     369.4    3140.1   12720.9    8649.3   28433.4    1112.4
vb3
    1963.5      32.9    4423.0   13877.1    6060.7   31052.5     375.5
ARAS2
     0.115     0.000     0.002     0.038     0.000     0.004     0.001
     0.000     0.011     0.001     0.002     0.003     0.000     0.000
     0.010     0.000     0.003     0.003     0.010     0.019     0.033
     0.085     0.029     0.091     0.080     0.023     0.025     0.026
     0.084     0.042     0.066     0.053     0.064     0.028     0.027
     0.075     0.121     0.082     0.053     0.164     0.203     0.222
     0.003     0.003     0.002     0.003     0.005     0.005     0.003
MADRAS3
0.0044082188
```

Chapter 11, Social Accounting Matrices

Chapter 11 expands the input–output framework to a broader class of economic analysis tools known as social accounting matrices (SAM) and other so-called "extended" input–output models to capture activities of income distribution in the economy in a more comprehensive and integrated way, including especially employment and social welfare features of an economy. The basic concepts of SAMs are explored and derived from the SNA introduced in Chapters 4 and 5, and the relationships between SAMs and input–output accounts are presented. The concept of SAM multipliers as well as the decomposition of SAM multipliers into components with specific economic interpretations are introduced and illustrated. Finally, techniques for balancing SAM accounts for internal accounting consistency are discussed and several illustrative applications of the use of SAMs are presented. The exercise problems for this chapter explore the construction of SAM accounts and models.

Problem 11.1 Quantifying the Circular Flow of Income and Expenditure

This exercise illustrates the relationships between a map of the circular flow of income and expenditure and a corresponding "macro-SAM." Consider a macro economy depicted in the figure below. Note the missing value, *X*, showing the exports from the Producers sector to the Rest of World sector. We can verify that this value is 45 from either the Producers balance equation, $X = (60 + 600) - (400 + 150 + 65) = 45$, or the Balance of Payment Account's Rest of World balance equation, $X = 25 - (10 + 60) = -45$.



We can express the chart as a basic macro-SAM, where a sector defined as consumption combines both intermediate and final consumption as a single sector by the following:

	Prod	Cons	Cap	ROW	
Producers		550	65	45	660
Consumers	600	150	-25	25	750
Capital Markets		40			40
Rest of World	60	10			70
	660	750	40	70	

If we express consumption as in the figure, i.e., with consumption separated into intermediate consumers (2) and final consumers (3), the SAM becomes:

	Prod	Cons	Fin Con	Cap	ROW	
Producers		400	150	65	45	660
Consumers	600			-25	25	600
Final Consumers		150				150
Capital Markets		40				40
Rest of World	60	10				70
	660	600	150	40	70	

Problem 11.2 Construction of a Fully Articulated SAM

This exercise illustrates construction of a "fully articulated" SAM, i.e., including the interindustry detail provided by input-output accounts. For the economy depicted in Problem 11.1, suppose the following input-output accounts are collected:

		Commodities		Industries		Final Demand	Totals	Grand Total
		Manuf.	Services	Manuf.	Services			
Commodities	Manuf.			94	96	110	300	660
	Services			94	117	148	360	
Industries	Manuf.	295	0				295	660
	Services	5	360				365	
Value Added				106	152	260		
Totals		300	360	295	365			
Grand Total		660		660				

To construct a "fully articulated" SAM, i.e., incorporating the interindustry detail provided by these input-output accounts, final demand must be allocated as part of consumer demand and commodity imports allocated to value added. There is no unique solution, but one such balanced fully articulated SAM is the following.

		Production		Consumption		Cap.	ROW	Total	
		Manuf.	Services	Manuf.	Services				
Production	Manuf.	0	0	158	96	28	18	300	660
	Services	0	0	94	203	37	25	360	
Consumption	Manuf.	284	0	0	0	-12	13	285	600
	Services	5	319	0	0	-13	4	315	
Capital		0	0	20	20	0	0	40	
ROW		10	41	3	5	0	0	60	
Total		300	360	276	324	40	60		
		660		600					

Problem 11.3 Expansion of SAM Accounts

This exercise problem expands SAM accounts to include sectors defined for consumer demand and exports, using SAM developed in Problem 11.2. Again, there is no unique solution, but the SAM must be balanced, i.e., row and column sums equal. One such SAM is the following:

		Commodities		Industries		Total Final Demand			Total	
		Manuf.	Services	Manuf.	Services	PCE	Cap.	Exports		
Commodities	Manuf.	0	0	94	96	64	28	18	300	660
	Services	0	0	94	117	86	37	25	360	
Industrieis	Manuf.	284	0	0	0	0	-12	13	285	600
	Services	5	319	0	0	0	-13	4	315	
Value Added	Consumer	0	0	73	77	0	0	0	150	250
	Capital	0	0	20	20	0	0	0	40	
	Imports	10	41	4	5	0	0	0	60	
Total		300	360	285	315	150	40	60		
		660		600		250				

Problem 11.4: Basic Construction of a Social Accounting Matrix (SAM)

This problem explores construction of a SAM matrix of total expenditure shares, $\overline{\mathbf{S}}$, and partitioning of $\overline{\mathbf{S}}$ to specify the SAM coefficient matrix, $\mathbf{S}$, and the "direct effect" multipliers using the table of SAM transactions developed in Problem 11.3.

Problem 11.4 Overview

First, we define the table of SAM transactions as $\overline{\mathbf{Z}}$ and the row or column totals of all transactions as $\overline{\mathbf{x}}$. Then we compute the matrix of total expenditure shares as

$$\overline{\mathbf{S}} = \overline{\mathbf{Z}}\hat{\mathbf{x}}^{-1} = \left[\begin{array}{cccc|ccc} 0 & 0 & .331 & .305 & .427 & .7 & .3 \\ 0 & 0 & .331 & .371 & .573 & .925 & .417 \\ .95 & 0 & 0 & 0 & 0 & -.3 & .217 \\ .017 & .884 & 0 & 0 & 0 & -.325 & .067 \\ \hline 0 & 0 & .257 & .244 & 0 & 0 & 0 \\ 0 & 0 & .07 & .063 & 0 & 0 & 0 \\ .033 & .116 & .011 & .016 & 0 & 0 & 0 \end{array}\right].$$

Notice that $\overline{\mathbf{S}}$ is partitioned into interindustry sectors (commodities and industries) and sectors exogenous to interindustry activity (final demand and value added). If we assume final demand and value-added sectors are considered exogenous transactions to this economy, the SAM coefficient matrix, $\mathbf{S}$, is the upper left partition of $\overline{\mathbf{S}}$, i.e., $\mathbf{S} = \begin{bmatrix} 0 & 0 & .331 & .305 \\ 0 & 0 & .331 & .371 \\ .95 & 0 & 0 & 0 \\ .017 & .884 & 0 & 0 \end{bmatrix}$.

The matrix of "direct effect" multipliers is then $\mathbf{M} = (\mathbf{I} - \mathbf{S})^{-1} = \begin{bmatrix} 1.812 & .726 & .840 & .822 \\ .865 & 1.835 & .894 & .945 \\ 1.721 & .690 & 1.798 & .781 \\ .794 & 1.634 & .804 & 1.849 \end{bmatrix}$.

Computational Notes

Define the sector transactions `ZZ` and normalize by the column sums **x** to create `SBAR`. Extract the upper left quadrant of `SBAR` as **S** and compute the Leontief inverse to generate the matrix of total multipliers `M1`.

```
ZZ←0 0 94 96 64 28 18 0 0 94 117 86 37 25 284
ZZ←XX,0 0 0 0 ¯12 13 5 319 0 0 0 ¯13 4 0 0 73 77
ZZ←7 7⍴ZZ,0 0 0 0 0 20 20 0 0 0 10 42 3 5 0 0 0

SBAR←ZZ AMAT x←+⌿ZZ
S←4 4↑SBAR
M1←LINV S
```

```
ZZ
         0         0        94        96        64        28        18
         0         0        94       117        86        37        25
       284         0         0         0         0       ¯12        13
         5       319         0         0         0       ¯13         4
         0         0        73        77         0         0         0
         0         0        20        20         0         0         0
        10        42         3         5         0         0         0
SBAR
     0.000     0.000     0.331     0.305     0.427     0.700     0.300
     0.000     0.000     0.331     0.371     0.573     0.925     0.417
     0.950     0.000     0.000     0.000     0.000    ¯0.300     0.217
```

```
        0.017     0.884     0.000     0.000     0.000    ¯0.325     0.067
        0.000     0.000     0.257     0.244     0.000     0.000     0.000
        0.000     0.000     0.070     0.063     0.000     0.000     0.000
        0.033     0.116     0.011     0.016     0.000     0.000     0.000
x
        299.0     361.0     284.0     315.0     150.0      40.0      60.0

S
        0.000     0.000     0.331     0.305
        0.000     0.000     0.331     0.371
        0.950     0.000     0.000     0.000
        0.017     0.884     0.000     0.000
M1
        1.812     0.726     0.840     0.822
        0.865     1.835     0.894     0.945
        1.721     0.690     1.798     0.781
        0.794     1.634     0.804     1.849
```

Problem 11.5: SAM Multipliers

The exercise illustrates the construction of direct, indirect, cross, and total SAM multipliers in the additive and multiplicative forms.

Problem 11.5 Overview

Using a highly aggregated SAM for the developing nation of Sri Lanka,[5] the basic SAM accounts are reflected in the following:

Sri Lanka SAM 1970	Value Added	Insti-tutions	Indirect Taxes	Surplus/ Deficit	Pro-duction	Rest of World	Total
Value Added	-	-	-	11,473	-	-	11,473
Institutions	11,360	2,052	1,368	-	-	3	14,783
Indirect Taxes	-	389	-	885	-	94	1,368
Production	-	11,312	-	4,660	-	2,113	18,085
Surplus/Deficit	-	(425)	-	-	-	425	-
Rest of World	113	1,455	-	1,067	-	-	2,635
Total	11,473	14,783	1,368	18,085	-	2,635	

We define the full table of SAM transactions as $\bar{\mathbf{Z}}$ and the row and/or column totals as $\bar{\mathbf{x}}$. If we consider the sectors, Surplus/Deficit and Rest of World, as external to the SAM, we first reorder the table so that the Surplus/Deficit and Rest of World sectors become sectors 5 and 6 in the table by $\tilde{\mathbf{Z}} = \mathbf{R}\bar{\mathbf{Z}}\mathbf{R}'$ and $\tilde{\mathbf{x}} = \mathbf{R}\bar{\mathbf{x}}$ where $\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$.

[5] Adapted from Pyatt and Round (1979), pp. 852-853.

With the reordered sectors, we calculate $\overline{\mathbf{S}} = \tilde{\mathbf{Z}}\hat{\hat{\mathbf{x}}}^{-1}$ and create partitions separating sectors 5 and 6 as the exogenous sectors. We can use the formulas developed in Section 11.10.4 to compute the direct, indirect, cross, and total multipliers, respectively, as the following:

$$\mathbf{N}_1 = \left[\begin{array}{cccc:cc} 5.7286 & 4.7756 & 4.7756 & 5.2105 & 0 & 0 \\ 7.2309 & 7.3209 & 7.3209 & 6.6610 & 0 & 0 \\ 0.5550 & 0.5605 & 1.5605 & 0.5772 & 0 & 0 \\ 7.4538 & 7.5279 & 7.5279 & 8.2133 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

$$\mathbf{N}_2 = \left[\begin{array}{cccc:cc} 0 & 0 & 0 & 0 & 26.9953 & 5.9390 \\ 0 & 0 & 0 & 0 & 34.7835 & 7.2176 \\ 0 & 0 & 0 & 0 & 3.2188 & 0.5400 \\ 0 & 0 & 0 & 0 & 42.5529 & 9.6888 \\ \hdashline -0.2506 & -0.2511 & -0.2511 & -0.2472 & 0 & 0 \\ 1.5382 & 1.5302 & 1.5302 & 1.6014 & 0 & 0 \end{array}\right]$$

$$\mathbf{N}_3 = \left[\begin{array}{cccc:cc} 1.5617 & 1.5182 & 1.5182 & 1.9068 & 0 & 0 \\ 1.4871 & 1.4301 & 1.4301 & 1.9388 & 0 & 0 \\ -0.0168 & -0.0224 & -0.0224 & 0.0271 & 0 & 0 \\ 2.8569 & 2.7889 & 2.7889 & 3.3954 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & -1 & -0.2075 \\ 0 & 0 & 0 & 0 & 6.2 & 1.3405 \end{array}\right]$$

$$\mathbf{N}_T = \left[\begin{array}{cccc:cc} 7.2903 & 6.2938 & 6.2938 & 7.1173 & 26.9953 & 5.9390 \\ 8.7180 & 8.7329 & 8.7329 & 8.5997 & 34.7835 & 7.2176 \\ 0.5382 & 0.5382 & 1.5382 & 0.6043 & 3.2188 & 0.5400 \\ 10.3106 & 10.3168 & 10.3168 & 11.6087 & 42.5529 & 9.6888 \\ \hdashline -0.2506 & -0.2511 & -0.2511 & -0.2472 & 0 & -0.2075 \\ 1.5382 & 1.5302 & 1.5302 & 1.6014 & 6.2 & 2.3405 \end{array}\right].$$

Computational Notes

We define the SAM transactions matrix as `ZS`, construct the reordering matrix `R` using the function `SCREATE` introduced earlier, and applying `R` to reorder the sectors of the SAM, saving the result as Z.

```
ZS←0 0 0 0 11473 0 11360 2052 1368
 ZS→ZS,0 0 3 0 389 0 0 885 94 0 ¯425
 ZS←ZS,0 0 0 425 0 11312 0 0 4660
 ZS←6 6⍴ZS,2113 113 1455 0 0 1067 0
 Z←R+.×ZS+.×⍉R←SCREATE' ',(⍕6 1⍴1 2 3 5 4 6),
```

The dyadic function SMULT (listed below) decomposes a SAM, such as **Z** into four partitions in order to calculate SAM multipliers. First, the left argument is a two-element vector nm, the first element of which is the number of sectors in the upper left partition, saved as the local variable n, and the second is number of sectors in the lower right partition, saved as the variable m. The value nn (the number of rows in endogenous partition of the overall SAM) is computed as the sum of n and m. **S** is then computed as the nn×nn upper left partition of **Z** normalized by its column sums. The upper left n×n partition of **S** is saved as **A**; the upper right m×n partition of **S** is saved as **C**; the lower left m×n partition of **S** is saved as **H** and the lower right m×m partition of **S** is saved as **O**. Also, **A** comprises the upper left partition of an nn×nn matrix **Q** with the rest of the elements of **Q** set to 0. **C** also comprises the upper right partition and **H** the lower left partition of an nn×nn matrix **R** with the rest of the matrix **R** set to 0. Then compute the matrix **T** which is subsequently used along with **Q** to compute the direct, indirect, cross, and total additive multipliers M1, M2, M3, and MT, respectively, and the direct, indirect, cross, and total multiplicative multipliers, N1, N2, N3, and NT, respectively, all as global variables in the APL workspace (the function does not produce an explicit result).

```
 nm SMULT Z;m;n;nn;S;Q;R;T;O;IQI;C;H;A;I
⍝compute decomposed multipliers for SAM=Z
⍝multiplicative multipliers are global results: M1, M2, M3, and MT
⍝additive multipliers are global results: N1, N2, N3, and NT (NT=MT)
⍝n=size of first partition; m second, nm total
 nn←(n←1↑nm)+m←¯1↑nm ⋄ S←(2⍴nn)↑Z AMAT+⌿Z ⋄ Q←R←(2⍴nn)⍴0 ⋄ I←(2⍴nn)⍴1,nn⍴0
 Q[⍳n;⍳n]←A←S[⍳n;⍳n] ⋄ R[⍳n;n+⍳m]←C←S[⍳n;n+⍳m]
 R[n+⍳m;⍳n]←H←S[n+⍳m;⍳n] ⋄ O←S[⍳m;⍳m]
 T←(IQI←INV Q)+.×R
 M3←INV T+.×T ⋄ M2←I+T ⋄ M1←INV Q ⋄ MT←M3+.×M2+.×M1
 N1←M1 ⋄ N2←(M2+.×M3+.×M1)-(M3+.×M1) ⋄ N3←(M3+.×M1)-M1 ⋄ NT←N1+N2+N3
```

For this problem we apply SMULT with the first four sectors of the economy defining the endogenous.

```
      4 2 SMULT Z

N1
    5.7286    4.7756    4.7756    5.2105    0.0000    0.0000
    7.2309    7.3029    7.3029    6.6610    0.0000    0.0000
    0.5550    0.5605    1.5605    0.5772    0.0000    0.0000
    7.4538    7.5279    7.5279    8.2133    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    1.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    1.0000
N2
    0.0000    0.0000    0.0000    0.0000   26.9953    5.9390
    0.0000    0.0000    0.0000    0.0000   34.7835    7.2176
    0.0000    0.0000    0.0000    0.0000    3.2188    0.5400
    0.0000    0.0000    0.0000    0.0000   42.5529    9.6888
```

```
   ¯0.2506   ¯0.2511   ¯0.2511   ¯0.2472    0.0000    0.0000
    1.5382    1.5302    1.5302    1.6014    0.0000    0.0000
N3
    1.5617    1.5182    1.5182    1.9068    0.0000    0.0000
    1.4871    1.4301    1.4301    1.9388    0.0000    0.0000
   ¯0.0168   ¯0.0224   ¯0.0224    0.0271    0.0000    0.0000
    2.8569    2.7889    2.7889    3.3954    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000   ¯1.0000   ¯0.2075
    0.0000    0.0000    0.0000    0.0000    6.2000    1.3405
NT
    7.2903    6.2938    6.2938    7.1173   26.9953    5.9390
    8.7180    8.7329    8.7329    8.5997   34.7835    7.2176
    0.5382    0.5382    1.5382    0.6043    3.2188    0.5400
   10.3106   10.3168   10.3168   11.6087   42.5529    9.6888
   ¯0.2506   ¯0.2511   ¯0.2511   ¯0.2472    0.0000   ¯0.2075
    1.5382    1.5302    1.5302    1.6014    6.2000    2.3405
```

Problem 11.6: Balancing a SAM

This problem illustrates use of the RAS technique to balance a SAM, i.e., to iteratively adjust the SAM transactions so that the row and column sums of the SAM are the same.

Problem 11.6 Overview

Consider the unbalanced SAM given in the table below. Suppose independent analysis indicates the total output of each sector; these are given in the additional column specified in the table.

	Prod.	Cons.	Capital	ROW	Totals	Estimated Totals
Producers	0	600	65	45	710	660
Consumers	700	0	-25	15	690	600
Capital	0	40	0	0	40	40
Rest of World	50	10	0	0	60	60
Totals	750	650	40	60	1,500	1,360

If we use the RAS technique to produce a balanced SAM with rows and columns both summing to the independent sector output estimates, the result is the following:

	Prod.	Cons.	Capital	ROW	Totals
Producers	-	560	40	60	660
Consumers	600	-	-	0	600
Capital	-	40	-	-	40
Rest of World	60	0	-	-	60
Totals	660	600	40	60	1,360

Computational Notes

We defined the unbalanced SAM as `Z2` and the presumed correct row and column totals as **u**. We apply the RAST function with the left argument as `Z2` and the right argument a two-row matrix with **u** comprises both rows. We store the result, the balanced SAM, as `ZNEW`.

```
Z2←4 4⍴0 600 65 45 700 0 ¯25 15 0 40 0 0 50 10 0 0
u←660 600 40 60
ZNEW←Z2 RAST 2 4⍴u
```

```
Z2
       0       600        65        45
     700         0       ¯25        15
       0        40         0         0
      50        10         0         0
u
     660       600        40        60
ZNEW
       0       560        40        60
     600         0         0         0
       0        40         0         0
      60         0         0         0
```

Problem 11.7: Additional Exogenous Information in Balancing SAMs

This problem explores the use of the RAS technique including additional exogenously specified information to balance a SAM using the unbalanced SAM given in Problem 11.6.

Problem 11.7 Overview

If, in addition to the estimated totals provided in the unbalanced table, we become aware that the elements $z_{23} = -25$, $z_{24} = 15$, and $z_{42} = 10$ in the balanced SAM are fixed, we can use the RAS procedure incorporating some fixed exogenous data for these elements (developed in chapter 10) to produce a balanced SAM:

	Prod.	Cons.	Capital	ROW	Totals
Producers	0	550	65	45	660
Consumers	610	0	-25	15	600
Capital	0	40	0	0	40
Rest of W	50	10	0	0	60
Totals	660	600	40	60	1,360

Computational Notes

We define the base SAM transactions matrix as `Z2`, the vector of target level of total outputs as **t**, and the matrix of known target transactions as `ZF`, which includes the target transactions and all other transactions are 0.

```
Z2←4 4⍴0 600 65 45 700 0 ¯25 15 0 40 0 0 50 10 0 0
t←660 600 40 60
ZF←4 4⍴0 0 0 0 0 0 ¯25 15 0 0 0 0 0 10 0 0
```

```
Z2
       0       600        65        45
     700         0       ¯25        15
       0        40         0         0
      50        10         0         0
t
     660       600        40        60
```

```
ZF
         0         0         0         0
         0         0       ¯25        15
         0         0         0         0
         0        10         0         0
```

To solve this problem, we modify the `RAST` function to include the steps for incorporating additional information of fixed coefficients in RAS biproportional scaling. The new function, `RASTRANSF`, like `RAST` developed earlier, takes as its right argument a two-row matrix with, in this case, the vector t repeated in the two rows. The left argument is a 2 x 4 x 4 cube, with `Z2` as the first "sheet" and `ZF` as the second. The function returns as its explicit result the biproportionately scaled transactions matrix that meets the condition that row and column totals are with a specified tolerance, `tol`, of the target row and column totals (if the RAS process successfully converges).

```
 Z←Z0F RASTRANSF UV;tol;n;k;XD;test;u1;v1;r;s;ZF;Z0
⍝Basic function for RAS biproportional scaling
⍝of Transactions with fixed elements
⍝GLOBAL INPUT: Z0F and rows of UV are u1 v1
⍝OUTPUT: Z
 Z0←Z0F[1;;] ⋄ ZF←Z0F[2;;]
 nn←2ρn←1↑ρZ0 ⋄ tol←0.1 ⋄ lim←500000
 test←'((⌈/|(v-v1))≤tol)∧((⌈/|(u-u1))≤tol)'
 Z←(Z0×ZF=0) ⋄ u1←UV[1;]-+/ZF ⋄ v1←UV[2;]-+⌿ZF
⍝BEGIN ITERATION
 u←+/Z ⋄ v←+⌿Z ⋄ k←0
 →CON×⍳1=⍎test
⍝----ROW ADJUSTMENT
LOOP:Z←Z×⍉nnρr←u1÷u
 u←+/Z ⋄ v←+⌿Z ⋄ k←k+1
 →CON×⍳1=⍎test
⍝----COL ADJUSTMENT
 Z←Z×nnρs←v1÷v
 u←+/Z ⋄ v←+⌿Z ⋄ k←k+1
 →CON×⍳1=⍎test
 →LOOP×⍳lim>k
 Z←Z+ZF
 →0,0ρ⎕←'**** STOPPED: ',(⍕k),' ITERATIONS ****'
CON:Z←Z+ZF
 '**** CONVERGENCE: ',(⍕k),' ITERATIONS ****'
```

We create the cube including `Z2` and `ZF` as `Z0F` and present it as the left argument of `RASTRANSF` and reshape t repeating it in two rows as the right argument and the balanced matrix (the explicit result) is saved as `ZNEW`.

```
Z0F←(2,ρZ2)ρ(,Z2),,ZF
ZNEW←Z0F RASTRANSF 2 4ρt
```

```
ZNEW
         0        550         65         45
       610          0        ¯25         15
         0         40          0          0
        50         10          0          0
```

Problem 11.8: SAM Multiplier for A U.S. "Macro-SAM"

The problem explores development of direct, indirect, cross, and total SAM multipliers in their multiplicative form using a "macro-SAM" for the U.S. economy for 1988.

Problem 11.8 Overview

The US SAM (as reported in Reinert and Roland-Holst, 1992, pp. 173-187) is the following:

US SAM 1988	Prod.	Comm.	Labor	Prop.	Enter-prises	House-holds	Govt.	Capital	ROW	Taxes	Errors	Total
Production		4831										4831
Commodities						3235	970	750	431			5386
Labor	2908											2908
Property	1556								117			1673
Enterprises				1589		95	93					1777
Households			2463		1045		556					4064
Government	377		445		138	587		96		18		1661
Capital					594	145			117		-10	846
Rest of World		537		84		2	42					665
Taxes		18										18
Errors & Omissions	-10											-10
Total	4831	5386	2908	1673	1777	4064	1661	846	665	18	-10	

If we consider the first five sectors as the endogenous sectors, the direct, indirect, cross, and total multipliers in their multiplicative form, respectively, are given by:

$$\mathbf{M}_1 = \left[\begin{array}{ccccc|c} 1 & .897 & 0 & 0 & 0 & \\ 0 & 1 & 0 & 0 & 0 & \\ .602 & .54 & 1 & 0 & 0 & \mathbf{0} \\ .322 & .289 & 0 & 1 & 0 & \\ .306 & .274 & 0 & .95 & 1 & \\ \hline & & \mathbf{0} & & & \mathbf{I} \end{array}\right],$$

$$\mathbf{M}_2 = \left[\begin{array}{c|c} \mathbf{I} & \begin{matrix} .714 & .524 & .795 & .581 & 0 & 0 \\ .796 & .584 & .887 & .648 & 0 & 0 \\ .43 & .315 & .479 & .35 & 0 & 0 \\ .23 & .169 & .256 & .363 & 0 & 0 \\ .24 & .216 & .243 & .345 & 0 & 0 \end{matrix} \\ \hline \begin{matrix} 0 & 0 & .847 & 0 & .588 & 0 \\ .078 & 0 & .153 & 0 & .078 & 0 \\ 0 & 0 & 0 & 0 & .334 & 0 \\ 0 & .10 & 0 & .05 & 0 & 0 \\ 0 & .003 & 0 & 0 & 0 & 0 \\ -.002 & 0 & 0 & 0 & 0 & 0 \end{matrix} & \mathbf{I} \end{array}\right],$$

$$\mathbf{M}_3 = \left[\begin{array}{c|c} \begin{matrix} 1.206 & .331 & 3.296 & .167 & 3.455 \\ .23 & 1.369 & 3.675 & .186 & 3.851 \\ .124 & .199 & 2.984 & .1 & 2.079 \\ .071 & .132 & 1.136 & 1.066 & 1.191 \\ .078 & .135 & 1.2 & .068 & 2.246 \end{matrix} & \mathbf{0} \\ \hline \mathbf{0} & \begin{matrix} 3.469 & 1.931 & 2.669 & 2.488 & 0 & 0 \\ .677 & 1.526 & .734 & .67 & 0 & 0 \\ .413 & .333 & 1.441 & .452 & 0 & 0 \\ .439 & .339 & .477 & 1.436 & 0 & 0 \\ .013 & .01 & .014 & .012 & 1 & 0 \\ -.007 & -.005 & -.008 & -.007 & 0 & 1 \end{matrix} \end{array}\right] \text{ and}$$

$$\mathbf{M} = \left[\begin{array}{ccccc|cccccc} 4.301 & 4.189 & 3.296 & 3.448 & 3.455 & 3.415 & 2.640 & 3.713 & 3.321 & 0 & 0 \\ 3.68 & 4.67 & 3.675 & 3.844 & 3.851 & 3.807 & 2.943 & 4.14 & 3.703 & 0 & 0 \\ 2.589 & 2.521 & 2.984 & 2.075 & 2.079 & 2.056 & 1.589 & 2.235 & 1.999 & 0 & 0 \\ 1.463 & 1.444 & 1.136 & 2.197 & 1.191 & 1.177 & .91 & 1.28 & 1.322 & 0 & 0 \\ 1.509 & 1.488 & 1.2 & 2.201 & 2.246 & 1.237 & .995 & 1.319 & 1.352 & 0 & 0 \\ \hline 3.080 & 3.011 & 3.233 & 3.052 & 3.082 & 3.469 & 1.931 & 2.669 & 2.488 & 0 & 0 \\ .849 & .828 & .807 & .758 & .762 & .677 & 1.526 & .734 & .67 & 0 & 0 \\ .504 & .497 & .401 & .736 & .751 & .413 & .333 & 1.441 & .452 & 0 & 0 \\ .44 & .538 & .423 & .494 & .444 & .439 & .339 & .477 & 1.436 & 0 & 0 \\ .012 & .016 & .012 & .013 & .013 & .013 & .01 & .014 & .012 & 1 & 0 \\ -.009 & -.009 & -.007 & -.007 & -.007 & -0.007 & -0.005 & -0.008 & -0.007 & 0 & 1 \end{array}\right]$$

Computational Notes

We define SAM transactions table `ZZ` and provide it as the right argument to the function `SMULT` develop earlier. The left argument is a two-element vector, the first of which defines the number of endogenous sectors and the second the number of exogenous sectors in the SAM. `SMULT` computes the direct, indirect, cross, and total additive multipliers are `M1`, `M2`, `M3`, and `MT` as global variables.

```
ZZ←0 4831 0 0 0 0 0 0 0 0 0 0 0 0 0 3235 970 750 431 0 0
ZZ←ZZ,2908 0 0 0 0 0 0 0 0 0 0 1556 0 0 0 0 0 0 0 117 0 0
```

```
      ZZ←ZZ,0 0 0 1589 0 95 93 0 0 0 0 0 0 2463 0 1045 0 556 0 0 0 0
      ZZ←ZZ,377 0 445 0 138 587 0 96 0 18 0 0 0 0 0 594 145 0 0 117
      ZZ←ZZ,0 ¯10 0 537 0 84 0 2 42 0 0 0 0 0 18 0 0 0 0 0 0 0 0 0
      ZZ←11 11ρZZ, ¯10 0 0 0 0 0 0 0 0 0 0
      5 6 SMULT ZZ
ZZ
     0   4831      0      0      0      0      0      0      0      0      0
     0      0      0      0      0   3235    970    750    431      0      0
  2908      0      0      0      0      0      0      0      0      0      0
  1556      0      0      0      0      0      0      0    117      0      0
     0      0      0   1589      0     95     93      0      0      0      0
     0      0   2463      0   1045      0    556      0      0      0      0
   377      0    445      0    138    587      0     96      0     18      0
     0      0      0      0    594    145      0      0    117      0    ¯10
     0    537      0     84      0      2     42      0      0      0      0
     0     18      0      0      0      0      0      0      0      0      0
   ¯10      0      0      0      0      0      0      0      0      0      0
 M1
 1.000  0.897  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.602  0.540  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.322  0.289  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.306  0.274  0.000  0.950  1.000  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000
M2
 1.000  0.000  0.000  0.000  0.000  0.714  0.524  0.795  0.581  0.000  0.000
 0.000  1.000  0.000  0.000  0.000  0.796  0.584  0.887  0.648  0.000  0.000
 0.000  0.000  1.000  0.000  0.000  0.430  0.315  0.479  0.350  0.000  0.000
 0.000  0.000  0.000  1.000  0.000  0.230  0.169  0.256  0.363  0.000  0.000
 0.000  0.000  0.000  0.000  1.000  0.242  0.216  0.243  0.345  0.000  0.000
 0.000  0.000  0.847  0.000  0.588  1.000  0.000  0.000  0.000  0.000  0.000
 0.078  0.000  0.153  0.000  0.078  0.000  1.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.334  0.000  0.000  1.000  0.000  0.000  0.000
 0.000  0.100  0.000  0.050  0.000  0.000  0.000  0.000  1.000  0.000  0.000
 0.000  0.003  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000
¯0.002  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000
M3
 1.206  0.331  3.296  0.167  3.455  0.000  0.000  0.000  0.000  0.000  0.000
 0.230  1.369  3.675  0.186  3.851  0.000  0.000  0.000  0.000  0.000  0.000
 0.124  0.199  2.984  0.100  2.079  0.000  0.000  0.000  0.000  0.000  0.000
 0.071  0.132  1.136  1.066  1.191  0.000  0.000  0.000  0.000  0.000  0.000
 0.078  0.135  1.200  0.068  2.246  0.000  0.000  0.000  0.000  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  3.469  1.931  2.669  2.488  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.677  1.526  0.734  0.670  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.413  0.333  1.441  0.452  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.439  0.339  0.477  1.436  0.000  0.000
 0.000  0.000  0.000  0.000  0.000  0.013  0.010  0.014  0.012  1.000  0.000
```

```
  0.000  0.000  0.000  0.000  0.000 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  1.000
MT
  4.301  4.189  3.296  3.448  3.455  3.415  2.640  3.713  3.321  0.000  0.000
  3.680  4.670  3.675  3.844  3.851  3.807  2.943  4.140  3.703  0.000  0.000
  2.589  2.521  2.984  2.075  2.079  2.056  1.589  2.235  1.999  0.000  0.000
  1.463  1.444  1.136  2.197  1.191  1.177  0.910  1.280  1.322  0.000  0.000
  1.509  1.488  1.200  2.201  2.246  1.237  0.995  1.319  1.352  0.000  0.000
  3.080  3.011  3.233  3.052  3.082  3.469  1.931  2.669  2.488  0.000  0.000
  0.849  0.828  0.807  0.758  0.762  0.677  1.526  0.734  0.670  0.000  0.000
  0.504  0.497  0.401  0.736  0.751  0.413  0.333  1.441  0.452  0.000  0.000
  0.440  0.538  0.423  0.494  0.444  0.439  0.339  0.477  1.436  0.000  0.000
  0.012  0.016  0.012  0.013  0.013  0.013  0.010  0.014  0.012  1.000  0.000
 ¯0.009 ¯0.009 ¯0.007 ¯0.007 ¯0.007 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  1.000
N1
  1.000  0.897  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.602  0.540  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.322  0.289  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.306  0.274  0.000  0.950  1.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000
N2
  0.000  0.000  0.000  0.000  0.000  3.415  2.640  3.713  3.321  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  3.807  2.943  4.140  3.703  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  2.056  1.589  2.235  1.999  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.177  0.910  1.280  1.322  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.237  0.995  1.319  1.352  0.000  0.000
  3.080  3.011  3.233  3.052  3.082  0.000  0.000  0.000  0.000  0.000  0.000
  0.849  0.828  0.807  0.758  0.762  0.000  0.000  0.000  0.000  0.000  0.000
  0.504  0.497  0.401  0.736  0.751  0.000  0.000  0.000  0.000  0.000  0.000
  0.440  0.538  0.423  0.494  0.444  0.000  0.000  0.000  0.000  0.000  0.000
  0.012  0.016  0.012  0.013  0.013  0.000  0.000  0.000  0.000  0.000  0.000
 ¯0.009 ¯0.009 ¯0.007 ¯0.007 ¯0.007  0.000  0.000  0.000  0.000  0.000  0.000
N3
  3.301  3.292  3.296  3.448  3.455  0.000  0.000  0.000  0.000  0.000  0.000
  3.680  3.670  3.675  3.844  3.851  0.000  0.000  0.000  0.000  0.000  0.000
  1.987  1.981  1.984  2.075  2.079  0.000  0.000  0.000  0.000  0.000  0.000
  1.141  1.155  1.136  1.197  1.191  0.000  0.000  0.000  0.000  0.000  0.000
  1.203  1.214  1.200  1.251  1.246  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  2.469  1.931  2.669  2.488  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.677  0.526  0.734  0.670  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.413  0.333  0.441  0.452  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.439  0.339  0.477  0.436  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.013  0.010  0.014  0.012  0.000  0.000
  0.000  0.000  0.000  0.000  0.000 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  0.000
NT
  4.301  4.189  3.296  3.448  3.455  3.415  2.640  3.713  3.321  0.000  0.000
  3.680  4.670  3.675  3.844  3.851  3.807  2.943  4.140  3.703  0.000  0.000
```

```
  2.589  2.521  2.984  2.075  2.079  2.056  1.589  2.235  1.999  0.000  0.000
  1.463  1.444  1.136  2.197  1.191  1.177  0.910  1.280  1.322  0.000  0.000
  1.509  1.488  1.200  2.201  2.246  1.237  0.995  1.319  1.352  0.000  0.000
  3.080  3.011  3.233  3.052  3.082  3.469  1.931  2.669  2.488  0.000  0.000
  0.849  0.828  0.807  0.758  0.762  0.677  1.526  0.734  0.670  0.000  0.000
  0.504  0.497  0.401  0.736  0.751  0.413  0.333  1.441  0.452  0.000  0.000
  0.440  0.538  0.423  0.494  0.444  0.439  0.339  0.477  1.436  0.000  0.000
  0.012  0.016  0.012  0.013  0.013  0.013  0.010  0.014  0.012  1.000  0.000
 ¯0.009 ¯0.009 ¯0.007 ¯0.007 ¯0.007 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  1.000
```

Problem 11.9: Equivalence of Additive and Multiplicative Direct Multipliers

This exercise expands the development of SAM multipliers to the multiplicative form using once again the macro-SAM specified in Problem 11.8.

Problem 11.9 Overview

If we compute the direct multipliers in their additive form, we discover that they are the same as those in the multiplicative form, i.e., $\mathbf{M}_1 = \mathbf{N}_1$, which turns out to be always the case as discussed in Section 11.10.5.

Computational Notes

The function `SMULT` as applied in Problem 11.8 also produces the direct, indirect, cross, and total multiplicative multipliers, `N1`, `N2`, `N3`, and `NT`, respectively, all as APL global variables.

```
N1
  1.000  0.897  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.602  0.540  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.322  0.289  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.306  0.274  0.000  0.950  1.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000
N2
  0.000  0.000  0.000  0.000  0.000  3.415  2.640  3.713  3.321  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  3.807  2.943  4.140  3.703  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  2.056  1.589  2.235  1.999  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.177  0.910  1.280  1.322  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  1.237  0.995  1.319  1.352  0.000  0.000
  3.080  3.011  3.233  3.052  3.082  0.000  0.000  0.000  0.000  0.000  0.000
  0.849  0.828  0.807  0.758  0.762  0.000  0.000  0.000  0.000  0.000  0.000
  0.504  0.497  0.401  0.736  0.751  0.000  0.000  0.000  0.000  0.000  0.000
  0.440  0.538  0.423  0.494  0.444  0.000  0.000  0.000  0.000  0.000  0.000
  0.012  0.016  0.012  0.013  0.013  0.000  0.000  0.000  0.000  0.000  0.000
 ¯0.009 ¯0.009 ¯0.007 ¯0.007 ¯0.007  0.000  0.000  0.000  0.000  0.000  0.000
N3
  3.301  3.292  3.296  3.448  3.455  0.000  0.000  0.000  0.000  0.000  0.000
  3.680  3.670  3.675  3.844  3.851  0.000  0.000  0.000  0.000  0.000  0.000
  1.987  1.981  1.984  2.075  2.079  0.000  0.000  0.000  0.000  0.000  0.000
```

```
  1.141  1.155  1.136  1.197  1.191  0.000  0.000  0.000  0.000  0.000  0.000
  1.203  1.214  1.200  1.251  1.246  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  2.469  1.931  2.669  2.488  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.677  0.526  0.734  0.670  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.413  0.333  0.441  0.452  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.439  0.339  0.477  0.436  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.013  0.010  0.014  0.012  0.000  0.000
  0.000  0.000  0.000  0.000  0.000 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  0.000
NT
  4.301  4.189  3.296  3.448  3.455  3.415  2.640  3.713  3.321  0.000  0.000
  3.680  4.670  3.675  3.844  3.851  3.807  2.943  4.140  3.703  0.000  0.000
  2.589  2.521  2.984  2.075  2.079  2.056  1.589  2.235  1.999  0.000  0.000
  1.463  1.444  1.136  2.197  1.191  1.177  0.910  1.280  1.322  0.000  0.000
  1.509  1.488  1.200  2.201  2.246  1.237  0.995  1.319  1.352  0.000  0.000
  3.080  3.011  3.233  3.052  3.082  3.469  1.931  2.669  2.488  0.000  0.000
  0.849  0.828  0.807  0.758  0.762  0.677  1.526  0.734  0.670  0.000  0.000
  0.504  0.497  0.401  0.736  0.751  0.413  0.333  1.441  0.452  0.000  0.000
  0.440  0.538  0.423  0.494  0.444  0.439  0.339  0.477  1.436  0.000  0.000
  0.012  0.016  0.012  0.013  0.013  0.013  0.010  0.014  0.012  1.000  0.000
 ¯0.009 ¯0.009 ¯0.007 ¯0.007 ¯0.007 ¯0.007 ¯0.005 ¯0.008 ¯0.007  0.000  1.000
```

Problem 11.10: Expanded Interindustry Detail of US SAM

This problem explores development of multipliers for a SAM expanded with interindustry detail using the SAM for the U.S. (1988) introduced in Problem 11.8, which is expanded with the interindustry detail shown in Table P11.10.

Table P11.10 SAM with Expanded Interindustry Detail for United States, 1988[6]

US SAM 1988 ($ billions)	1 Agric.	2 Mining	3 Const.	4 Nondur. Manuf	5 Durable Manuf.	6 Transp. & Util	7 Trade	8 Finance	9 Services	10 Labor	11 Propty	12 Enter-prises	13 House-holds	14 Govt.	15 Capital	16 Rest of World	17 Tariffs	18 Errors	Total
1 Agriculture	42	0	2	98	8	0	3	8	7	0	0	0	18	7	1	22	0	0	214
2 Mining	0	10	2	82	8	35	0	0	0	0	0	0	1	0	2	8	0	0	148
3 Construction	2	12	1	7	9	21	6	36	18	0	0	0	0	134	358	0	0	0	602
4 Nondurable Manuf.	30	1	35	370	83	37	24	14	149	0	0	0	453	38	4	93	0	0	1332
5 Durable Manuf.	4	3	175	55	480	19	7	4	81	0	0	0	236	97	296	187	0	0	1643
6 Transport & Utilities	5	1	17	66	65	78	46	31	84	0	0	0	310	34	13	26	0	0	774
7 Trade	8	1	72	57	73	11	14	7	50	0	0	0	529	11	56	43	0	0	932
8 Finance	10	3	10	18	25	14	52	20	79	0	0	0	771	16	22	25	0	0	1065
9 Services	5	1	53	68	74	31	124	93	214	0	0	0	917	632	0	27	0	0	2240
10 Labor	33	18	197	218	430	212	385	217	1198	0	0	0	0	0	0	0	0	0	2908
11 Property	60	56	32	142	69	207	147	511	332	0	0	0	0	0	0	117	0	0	1673
12 Enterprise	0	0	0	0	0	0	0	0	0	0	1589	0	96	92	0	0	0	0	1778
13 Households	0	0	0	0	0	0	0	0	0	2463	0	1046	0	556	0	0	0	0	4064
14 Government	8	12	7	28	18	35	127	113	30	445	0	138	587	0	96	0	16	0	1659
15 Capital	0	0	0	0	0	0	0	0	0	0	0	594	145	0	0	117	0	-10	846
16 Rest of World	8	31	0	115	295	75	0	12	2	0	83	0	2	42	0	0	0	0	665
17 Tariffs	0	0	0	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	16
18 Errors & Omissions	0	0	-1	-1	-1	-1	-1	-2	-2	0	0	0	0	0	0	0	0	0	-10
Total	214	148	602	1332	1643	774	932	1065	2240	2908	1673	1778	4064	1659	846	665	16	-10	

[6] As reported in Reinert and Roland-Holst (1992).

Problem 11.10 Overview

If we consider the first nine sectors as the endogenous sectors, the resulting total multipliers are the following:

$$\mathbf{i'M} = [3.245\ \ 3.053\ \ 3.380\ \ 3.647\ \ 3.581\ \ 2.949\ \ 2.769\ \ 2.588\ \ 2.645\ \ 1.000\ \ 1.000\ \ 1.000\ \ 3.302\ \ 2.691\ \ 4.000\ \ 3.160\ \ 1.000\ \ 1.000]$$

Computational Notes

We retrieve the US 1988 SAM from Appendix and define it as `ZZ` in the APL workspace. We then provide `ZZ` as the right argument to the function `SMULT` and, as the left argument, specify that all sectors the first 9 sectors are endogenous and the remaining 9 are endogenous, i.e., provide a two-element vector for which both elements are 9. `SMULT` delivers the matrix of total multipliers `MT` as a global variable and we compute the vector of total output multipliers `mt`.

```
      9 9 SMULT ZZ
      mt←+⌿MT
```

```
ZZ
     42      0      2     98      8      0      3      8      7      0      0      0     18      7      1     22      0      0
      0     10      2     82      8     35      0      0      0      0      0      0      1      0      2      8      0      0
      2     12      1      7      9     21      6     36     18      0      0      0      0    134    358      0      0      0
     30      1     35    370     83     37     24     14    149      0      0      0    453     38      4     93      0      0
      4      3    175     55    480     19      7      4     81      0      0      0    236     97    296    187      0      0
      5      1     17     66     65     78     46     31     84      0      0      0    310     34     13     26      0      0
      8      1     72     57     73     11     14      7     50      0      0      0    529     11     56     43      0      0
     10      3     10     18     25     14     52     20     79      0      0      0    771     16     22     25      0      0
      5      1     53     68     74     31    124     93    214      0      0      0    917    632      0     27      0      0
     33     18    197    218    430    212    385    217   1198      0      0      0      0      0      0      0      0      0
     60     56     32    142     69    207    147    511    332      0      0      0      0      0      0    117      0      0
      0      0      0      0      0      0      0      0      0      0   1589      0     96     92      0      0      0      0
      0      0      0      0      0      0      0      0      0   2463      0   1046      0    556      0      0      0      0
      8     12      7     28     18     35    127    113     30    445      0    138    587      0     96      0     16      0
      0      0      0      0      0      0      0      0      0      0      0    594    145      0      0    117      0    ¯10
      8     31      0    115    295     75      0     12      2      0     83      0      2     42      0      0      0      0
      0      0      0      8      8      0      0      0      0      0      0      0      0      0      0      0      0      0
      0      0     ¯1     ¯1     ¯1     ¯1     ¯1     ¯2     ¯2      0      0      0      0      0      0      0      0      0
MT
  1.278  0.025  0.031  0.150  0.044  0.023  0.017  0.019  0.020  0.000  0.000  0.000  0.037  0.022  0.033  0.079  0.000  0.000
  0.026  1.087  0.024  0.109  0.032  0.068  0.011  0.008  0.013  0.000  0.000  0.000  0.025  0.013  0.026  0.042  0.000  0.000
  0.029  0.103  1.019  0.033  0.024  0.047  0.028  0.049  0.019  0.000  0.000  0.000  0.026  0.093  0.443  0.020  0.000  0.000
  0.303  0.112  0.180  1.504  0.213  0.144  0.088  0.063  0.134  0.000  0.000  0.000  0.246  0.118  0.168  0.301  0.000  0.000
  0.125  0.209  0.499  0.212  1.583  0.150  0.070  0.066  0.097  0.000  0.000  0.000  0.171  0.179  0.773  0.498  0.000  0.000
  0.075  0.051  0.091  0.124  0.113  1.147  0.084  0.055  0.068  0.000  0.000  0.000  0.145  0.067  0.103  0.107  0.000  0.000
  0.083  0.058  0.170  0.106  0.113  0.051  1.037  0.027  0.044  0.000  0.000  0.000  0.173  0.048  0.182  0.122  0.000  0.000
  0.083  0.050  0.053  0.057  0.058  0.043  0.076  1.034  0.052  0.000  0.000  0.000  0.231  0.040  0.076  0.075  0.000  0.000
  0.123  0.111  0.200  0.172  0.159  0.123  0.244  0.172  1.152  0.000  0.000  0.000  0.362  0.474  0.164  0.148  0.000  0.000
  0.428  0.349  0.714  0.530  0.642  0.500  0.642  0.376  0.721  1.000  0.000  0.000  0.454  0.402  0.589  0.377  0.000  0.000
  0.497  0.505  0.218  0.357  0.214  0.410  0.281  0.561  0.250  0.000  1.000  0.000  0.285  0.151  0.210  0.198  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000  0.000  0.000
  0.083  0.111  0.060  0.077  0.055  0.077  0.161  0.122  0.036  0.000  0.000  0.000  0.070  1.028  0.060  0.050  0.000  0.000
  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  1.000  0.000  0.000  0.000
  0.110  0.280  0.121  0.209  0.323  0.166  0.032  0.037  0.041  0.000  0.000  0.000  0.076  0.053  0.171  1.139  0.000  0.000
  0.002  0.002  0.004  0.010  0.009  0.002  0.001  0.001  0.001  0.000  0.000  0.000  0.002  0.002  0.005  0.004  1.000  0.000
 ¯0.001 ¯0.001 ¯0.003 ¯0.002 ¯0.002 ¯0.002 ¯0.002 ¯0.002 ¯0.001  0.000  0.000  0.000 ¯0.001 ¯0.001 ¯0.002 ¯0.001  0.000  1.000
mt
  3.245  3.053  3.380  3.647  3.581  2.949  2.769  2.588  2.645  1.000  1.000  1.000  3.302  2.691  4.000  3.160  1.000  1.000
```

Chapter 12, Energy Input-Output Analysis

Chapter 12 explores the extension of the input–output framework to more detailed analysis of energy consumption associated with industrial production, including some of the complications that can arise when measuring input–output transactions in physical units of production rather than in monetary terms of the value of production.

The chapter reviews early efforts to develop energy input–output analysis and compares them with contemporary approaches and examines the strengths and limitations of the alternatives commonly used today. Special methodological considerations such as adjusting for energy conversion efficiencies are developed along with several illustrative applications, including estimation of the energy costs of goods and services, impacts of new energy technologies, and energy taxes.

Energy input-output analysis is increasingly being applied to global scale issues, such as the energy embodied in international trade of goods and services. Finally, the role of structural change of an input–output economy associated with changing patterns of energy use is illustrated, building on the more general approaches developed in Chapter 8.

The exercise problems for this chapter explore the use of input-output analysis to analyze the special case of energy production and use.

Problem 12.1: Basic Formulations of Energy Input-Output Analysis

This exercise problem develops two formulations of the energy input-output model from basic economic input-output accounts and supplemental information for tracking the flow of energy throughout an economy measured in physical units.

Problem 12. Overview

Consider the following three-sector input-output economy; two sectors are energy sectors (oil is the primary energy sector and refined petroleum is the secondary energy sector):

Interindustry Transactions ($\$10^6$)	Oil	Refined Petroleum	Manuf.	Final Demand	Total Output
Crude Oil	0	20	0	0	20
Refined Petroleum	2	2	2	24	30
Manufacturing	0	0	0	20	20

The energy sector transactions are also measured in quadrillions of Btus in the following table:

Energy Sector Transactions (10^{15} Btus)	Oil	Refined Petroleum	Manuf.	Final Demand	Total Output
Crude Oil	0	20	0	0	20
Refined Petroleum	1	1	1	17	20

To formulate an energy input-output model from these data, we first define the customary Leontief economic transactions matrix, vector of final demands, and vector of total outputs, respectively, all measured in millions of dollars as: $\mathbf{Z}=\begin{bmatrix}0 & 20 & 0\\ 2 & 2 & 2\\ 0 & 0 & 0\end{bmatrix}$, $\mathbf{f}=\begin{bmatrix}0\\ 24\\ 20\end{bmatrix}$ $\mathbf{x}=\begin{bmatrix}20\\ 30\\ 20\end{bmatrix}$. We can now compute the economic matrix of technical coefficients as $\mathbf{A}=\mathbf{Z}\hat{\mathbf{x}}^{-1}=\begin{bmatrix}0 & .667 & 0\\ .10 & .067 & .1\\ 0 & 0 & 0\end{bmatrix}$ and the corresponding matrix of total requirements as $\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}==\begin{bmatrix}1.077 & .769 & .077\\ .115 & 1.154 & .115\\ 0 & 0 & 1\end{bmatrix}$.

The matrix of energy transactions in physical units (quadrillions of Btus) is $\mathbf{E}=\begin{bmatrix}0 & 20 & 0\\ 1 & 1 & 1\end{bmatrix}$; the vector of energy consumption in final demand, $\mathbf{q}=\begin{bmatrix}0\\ 17\end{bmatrix}$, and total energy consumption, $\mathbf{g}=\begin{bmatrix}20\\ 20\end{bmatrix}$, are also measured in quadrillions of Btus (often referred to as Quads). The matrix of implied energy prices, defined as the element-by-element division of **E** by the corresponding elements in the energy rows of **Z** where transactions are nonzero and zero otherwise, is $\mathbf{P}=\begin{bmatrix}0 & 1 & 0\\ 2 & 2 & 2\end{bmatrix}$.

The traditional energy input-output formulation specifies the direct energy requirements as $\boldsymbol{\varepsilon}=\mathbf{D}(\mathbf{I}-\mathbf{A})^{-1}+\tilde{\mathbf{Q}}$ where $\mathbf{D}=\mathbf{E}\hat{\mathbf{x}}^{-1}=\begin{bmatrix}0 & .667 & 0\\ .05 & .033 & .05\end{bmatrix}$ and the elements of $\tilde{\mathbf{Q}}$ are defined as q_k / f_k for energy sectors and zero otherwise. In this case, $\tilde{\mathbf{Q}}=\begin{bmatrix}0 & 0 & 0\\ 0 & .708 & 0\end{bmatrix}$, so we find $\boldsymbol{\varepsilon}=\mathbf{D}(\mathbf{I}-\mathbf{A})^{-1}+\tilde{\mathbf{Q}}=\begin{bmatrix}.077 & .769 & .077\\ .058 & .785 & .058\end{bmatrix}$. Note that this suggests a million dollars' worth of final demand for manufacturing in this economy would require production of 0.785 Quads of refined petroleum but only 0.769 Quads of crude oil, which is not sensible since the structure of this economy is that refined petroleum is a secondary energy sector receiving all its energy input from the primary energy sector, Crude Oil, so the primary and secondary energy consumption (aside from any energy conversion efficiencies) should be the same, often referred to as an energy conservation condition.

To formulate these data instead as a hybrid units energy input output model, we first define the matrix of transactions in hybrid units, the vector of final demands, and the vector of total outputs, respectively, as $\mathbf{Z}^* = \begin{bmatrix} 0 & 20 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}$, $\mathbf{f}^* = \begin{bmatrix} 0 \\ 17 \\ 20 \end{bmatrix}$, and $\mathbf{x}^* = \begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}$ where energy rows are measured in Quads and nonenergy rows are measured in millions of dollars. We can now compute $\mathbf{A}^* = \begin{bmatrix} 0 & 1 & 0 \\ .05 & .05 & .05 \\ 0 & 0 & 0 \end{bmatrix}$ and $\mathbf{L}^* = (\mathbf{I} - \mathbf{A}^*)^{=1} == \begin{bmatrix} 1.056 & 1.111 & 0.056 \\ 0.056 & 1.111 & 0.056 \\ 0 & 0 & 1 \end{bmatrix}$ which is easy to see conforms to the energy conservation condition.

Computational Notes

We define the matrix of interindustry transactions **Z**, the vectors of total outputs x and of total final demands **f**, the matrix of energy transactions **E**, the vector of energy deliveries to final demand **q**, and the vector of total energy outputs **g**.

```
Z←3 3⍴0 20 0 2 2 2 0 0 0 ⋄ x←20 30 20 ⋄ f←0 24 20
E←2 3⍴0 20 0 1 1 1 ⋄ q←0 17 ⋄ g←20 20
```

For the first two sectors of this economy which are defined as energy sectors, we compute the matrix of implied energy prices **Q**, the matrix of direct energy requirements **D**, and the matrix of final demand energy coefficients **QT**. We also compute the matrix of technical coefficients **A** and the associated Leontief inverse **L** so that we can compute the matrix of total energy requirements **e**, using the traditional form of energy input-output analysis.

```
⍝-- implied energy prices
 Q←(Z[⍳2;]≠0)×E÷Z[⍳2;]
⍝-- direct energy requirements
 D←E+.×DIAG ÷x
 QT←2 3↑DIAG (q≠0)×q÷2↑f
⍝--  total energy requirements
 e←QT+D+.×L←LINV A←Z AMAT x
```

```
Q
     0.000     1.000     0.000
     0.500     0.500     0.500
D
     0.000     0.667     0.000
     0.050     0.033     0.050
QT
     0.000     0.000     0.000
     0.000     0.708     0.000
L
     1.077     0.769     0.077
     0.115     1.154     0.115
     0.000     0.000     1.000
e
     0.077     0.769     0.077
     0.058     0.785     0.058
```

To create the hybrid-units transactions matrix `ZS` we replace the energy rows of **Z** with **E**. For the vectors expressed in hybrid units of total outputs `xs` and of final demands `fs` we replace the energy sector elements of **x** and **f**, respectively, with **g** and **q**. Finally, we can compute hybrid-units direct requirements matrix `AS` and the associated hybrid-units total requirements matrix `LS`, the first two rows of which (the energy sectors) comprise the hybrid units total energy requirements matrix.

```
      ZS←E,[1]Z[3;] ⋄ xs←g,x[3] ⋄ fs←q,f[3]
      LS←LINV AS←ZS AMAT xs

      ZS
0 20 0
1  1 1
0  0 0
      fs
0 17 20
      xs
20 20 20
      AS
0.000     1.000     0.000
0.050     0.050     0.050
0.000     0.000     0.000
      LS
1.056     1.111     0.056
0.056     1.111     0.056
0.000     0.000     1.000
```

Problem 12.2: Energy Input-Output Analysis in Policy Analysis

The problem illustrates the typical use of the traditional energy input-output model in public policy analysis.

Problem 12.2 Overview

Consider the following input-output transactions table in value terms (millions of dollars) for two industries—*A* and *B*:

	A	*B*	Total Output
A	2	4	100
B	6	8	100

Suppose we have a direct energy requirements matrix for this economy that is given by:

$$\mathbf{D} = \begin{bmatrix} .2 & .3 \\ .1 & .4 \end{bmatrix} \begin{array}{l} 10^{15} \text{ Btus of oil per million dollars of output} \\ 10^{15} \text{ Btus of coal per million dollars of output} \end{array}$$

If for simplicity we ignore energy consumption by final demand, we compute the total energy requirements matrix, $\boldsymbol{\varepsilon} = \mathbf{DL} = \begin{bmatrix} .225 & .336 \\ .129 & .440 \end{bmatrix}$ where the matrices of technical requirements and total requirements are, respectively, $\mathbf{A} = \begin{bmatrix} .02 & .04 \\ .06 & .08 \end{bmatrix}$ and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.023 & .044 \\ .067 & 1.09 \end{bmatrix}$.

Suppose further that the final demands for industries *A* and *B* are projected to be \$200 million and \$100 million respectively for the next year. The net increase in energy (both oil and gas) required to support this new final demand (again, neglecting energy consumed directly by final demand) is found by $\Delta\mathbf{g} = \mathbf{DL}\Delta\mathbf{f} = \begin{bmatrix} 28.5 \\ 19.8 \end{bmatrix}$. We can determine how much of the total energy produced to supply this net increase in final demand is *direct* energy consumption $\Delta\mathbf{g}^{direct} = \mathbf{D}\Delta\mathbf{f} = \begin{bmatrix} 25.4 \\ 16.2 \end{bmatrix}$ where $\Delta\mathbf{f} = \mathbf{f}^{new} - \mathbf{f} = \begin{bmatrix} 200 \\ 100 \end{bmatrix} - \begin{bmatrix} 94 \\ 86 \end{bmatrix} = \begin{bmatrix} 106 \\ 14 \end{bmatrix}$. The amount of *indirect* energy consumption can be found as $\Delta\mathbf{g}^{indirect} = \Delta\mathbf{g} - \Delta\mathbf{g}^{direct} = \begin{bmatrix} 3.1 \\ 3.6 \end{bmatrix}$.

Finally, suppose an energy conservation measure in industry *B* causes the direct energy requirement of that industry for coal to be reduced from 0.4 to 0.3 (10^{15} Btus of coal per dollar of output of industry *B*). The resulting changes in direct and total energy requirements matrices are $\mathbf{D} = \begin{bmatrix} .2 & .3 \\ .1 & .3 \end{bmatrix}$ and $\mathbf{DL} = \begin{bmatrix} .225 & .336 \\ .122 & .331 \end{bmatrix}$, respectively. Hence the new change in total energy to support final demand, $\Delta\mathbf{f}$, is $\Delta\mathbf{g} = \mathbf{DL}\Delta\mathbf{f} = \begin{bmatrix} 28.5 \\ 17.6 \end{bmatrix}$. The direct portion is, once again, $\Delta\mathbf{g}^{direct} = \mathbf{D}\Delta\mathbf{f} = \begin{bmatrix} 25.4 \\ 14.8 \end{bmatrix}$ so the indirect portion is $\Delta\mathbf{g}^{indirect} = \Delta\mathbf{g} - \Delta\mathbf{g}^{direct} = \begin{bmatrix} 3.1 \\ 2.8 \end{bmatrix}$.

Hence, the differences in total energy consumption before and after the energy conservation measure are given by $\begin{bmatrix} 28.5 \\ 19.8 \end{bmatrix} - \begin{bmatrix} 28.5 \\ 17.6 \end{bmatrix} = \begin{bmatrix} 0 \\ 2.2 \end{bmatrix}$; the differences in direct energy consumption are given by $\begin{bmatrix} 25.4 \\ 16.2 \end{bmatrix} - \begin{bmatrix} 25.4 \\ 14.8 \end{bmatrix} = \begin{bmatrix} 0 \\ 1.4 \end{bmatrix}$; and the differences in indirect energy consumption are given by $\begin{bmatrix} 3.1 \\ 3.6 \end{bmatrix} - \begin{bmatrix} 3.1 \\ 2.8 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.8 \end{bmatrix}$.

Computational Notes

We define the matrix of interindustry transactions **Z**, the vectors of total outputs **x** and of final demands **f**, as well as the matrix of direct energy coefficients **D**, and a vector of new final demands presented to the economy `fn`.

```
Z←2 2⍴2 4 6 8 ⋄ x←100 100 ⋄ f←94 86
D←2 2⍴0.2 0.3 0.1 0.4 ⋄ fn←200 100
```

We can then compute the matrix of technical requirements **A** and the Leontief inverse **L** in order to compute the matrix of total energy requirements **e** using the traditional energy input-output formulation. We can then generate the vector of changes in final demand `Δf` and the associated interindustry energy consumption attributed to those changes in terms of the vector of direct energy consumption DX and the vector of indirect energy consumption IX, which if summed is the total interindustry energy consumption TX.

```
      e←D+.×L←LINV A←Z AMAT x
      IX←(TX←e+.×Δf)-DX←D+.×Δf←fn-f

      Z
15 20
      f
94 86
      x
100 100

      D
0.200     0.300
0.100     0.400
      A
0.020     0.040
0.060     0.080
      L
1.023     0.044
0.067     1.090
      e
0.225     0.336
0.129     0.440
      Δf
106 14
      DX
25.400    16.200
      TX
28.514    19.840
      IX
3.114     3.640
```

We specify the modified matrix of direct requirements as D2 and compute the revised total energy requirements matrix **e2**. For the same vector of changes in final demand **Δf**, we compute the vector of total interindustry consumption TX2, which is the sum of the vector of the direct energy consumption DX2 and the vector of indirect energy consumptions IX2. Finally, we compute the change in energy consumption associated with the change for **D** to D2 as ΔTX, which is the sum the direct and indirect components ΔDX and ΔIX.

```
      D2←2 2ρ0.2 0.3 0.1 0.3
      e2←D2+.×L
      IX2←(TX2←e2+.×Δf)-DX2←D2+.×Δf
      ΔDX←DX-DX2 ⋄ ΔIX←IX-IX2 ⋄ ΔTX←TX-TX2

      D2
0.2 0.3
0.1 0.3
      DX2
25.400    14.800
```

```
     TX2
28.514    17.607
     IX2
3.114     2.807
     ΔDX
0.000     1.400
     ΔIX
0.000     0.833
     ΔTX
0.000     2.233
```

Problem 12.3: Total Energy Impacts of Changes in Nonenergy Final Demand

This problem uses the energy input-output formulation to illustrate computation of the total energy impacts of a change in nonenergy final demand.

Problem 12.3 Overview

Consider the following input-output table ($\$10^6$):

	Transactions			Total Output
	Autos	Oil	Electricity	
Autos	2	6	1	10
Oil	0	0	20	20
Electricity	3	2	1	30

Assume that there is a matrix of implied inverse energy prices for this economy given by the following (inverse because the measure is millions of dollars per billion Btu rather than vice versa):

	Autos	Oil	Electricity	Final Demand
Oil	0	0	0.4082	0
Electricity	0.3333	0.2857	0.5	1.2912

We define the basic economic data of the matrix transactions and vectors of total outputs and final demands, respectively, as $\mathbf{Z} = \begin{bmatrix} 2 & 6 & 1 \\ 0 & 0 & 20 \\ 3 & 2 & 1 \end{bmatrix}$, $\mathbf{x} = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix}$, and

$$\mathbf{f} = \mathbf{x} - \mathbf{Zi} = \begin{bmatrix} 10 \\ 20 \\ 30 \end{bmatrix} - \begin{bmatrix} 9 \\ 20 \\ 6 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 24 \end{bmatrix}.$$

We can compute the energy transactions physical units (billions of Btus) by, first, defining the matrix of implied inverse energy prices from the table as $\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0.408 \\ 0.333 & 0.286 & 0.5 \end{bmatrix}$. If we multiply, element by element, $\mathbf{Q}$ and the energy rows of $\mathbf{Z}$, the energy transactions matrix measured in physical units is $\mathbf{E}=\begin{bmatrix}0 & 0 & 49\\ 9 & 7 & 2\end{bmatrix}$.

Using the energy sector elements of the computed economic final demands, $\begin{bmatrix}0\\ 24\end{bmatrix}$, multiplied, element by element, by the inverse energy prices for final demand from the table, $\begin{bmatrix}0\\ 1.2912\end{bmatrix}$, yields the vector of energy consumption in final demand measured in physical units, $\mathbf{q}=\begin{bmatrix}0\\ 31\end{bmatrix}$, from which we can now compute the total energy consumption as $\mathbf{g}=\mathbf{Ei}+\mathbf{q}=\begin{bmatrix}49\\ 49\end{bmatrix}$. We can express the energy flows as the energy rows in a hybrid units transactions matrix and corresponding vectors of final demands and total outputs:
$\mathbf{Z}^*=\begin{bmatrix}2 & 6 & 1\\ 0 & 0 & 49\\ 9 & 7 & 2\end{bmatrix}$, $\mathbf{f}^*=\begin{bmatrix}1\\ 0\\ 31\end{bmatrix}$ and $\mathbf{x}^*=\begin{bmatrix}10\\ 49\\ 49\end{bmatrix}$ and $\mathbf{A}^*=\mathbf{Z}^*(\hat{\mathbf{x}}^*)^{-1}=\begin{bmatrix}.2 & .122 & .02\\ 0 & 0 & 1\\ .9 & .143 & .041\end{bmatrix}$. The direct energy requirements matrix is then defined as the energy rows of $\mathbf{A}^*$. For this economy $\mathbf{G}=\begin{bmatrix}0 & 49 & 0\\ 0 & 0 & 49\end{bmatrix}$ so the direct energy coefficients can be computed as $\mathbf{G}(\hat{\mathbf{x}}^*)^{-1}\mathbf{A}^*=\begin{bmatrix}0 & 0 & 1\\ .9 & .143 & .041\end{bmatrix}$.

If a final demand vector of \$2 million worth of autos and 18 quadrillion Btus of electricity is presented to this economy, the total amount of energy (of each type) required to support this final demand is found by first retrieving the energy rows of $\mathbf{L}^*$,
$\mathbf{L}^*=(\mathbf{I}-\mathbf{A}^*)^{-1}=\begin{bmatrix}1.556 & 0.229 & 0.272\\ 1.716 & 1.428 & 1.526\\ 1.716 & 0.428 & 1.526\end{bmatrix}$,which defines the total energy requirements matrix,

$\boldsymbol{\alpha}=\mathbf{G}(\hat{\mathbf{x}}^*)^{-1}\mathbf{L}^*=\begin{bmatrix}1.716 & 1.428 & 1.526\\ 1.716 & 0.428 & 1.526\end{bmatrix}$. Then, for $\Delta\mathbf{f}^*=\begin{bmatrix}2\\ 0\\ 18\end{bmatrix}$ we compute the total energy consumption as $\Delta\mathbf{g}=\boldsymbol{\alpha}\Delta\mathbf{f}^*=\begin{bmatrix}30.887\\ 30.887\end{bmatrix}$.

If we alternatively use the traditional energy input-output formulation, using the energy prices defined above for final demand we can first compute $\Delta\mathbf{f} = \begin{bmatrix} 2 \\ 0 \\ 23.2 \end{bmatrix}$ and then

$$\Delta\mathbf{g} = \boldsymbol{\varepsilon}\Delta\mathbf{f} = \begin{bmatrix} 48.2 \\ 450.9 \end{bmatrix} \text{ for } \mathbf{A} = \begin{bmatrix} .2 & .3 & .033 \\ 0 & 0 & .667 \\ .3 & .1 & .033 \end{bmatrix},\ \mathbf{L} = \begin{bmatrix} 1.385 & .451 & .359 \\ .308 & 1.174 & .821 \\ .462 & .262 & 1.231 \end{bmatrix} \text{ and}$$

$\boldsymbol{\varepsilon} = \mathbf{DL} + \tilde{\mathbf{Q}} = \begin{bmatrix} .754 & .427 & 2.010 \\ 1.385 & .835 & 19.280 \end{bmatrix}$ where $\tilde{\mathbf{Q}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 18.587 \end{bmatrix}$ is the matrix of *implied inverse energy prices* for final demand. The elements of $\tilde{\mathbf{Q}} = [\tilde{q}_k]$ are defined by

$$\tilde{q}_k = \begin{cases} 1/p_{kf}, & \text{when energy sector } k \text{ and industry sector } j \text{ are the same sector} \\ 0, & \text{otherwise} \end{cases}$$

Computational Notes

Define the economic interindustry transactions table **Z** and the vectors of total outputs **x** and of total final demands **f**. Also define the implied energy price matrices for interindustry transactions **P** and for final demands **Q**.

```
      Z←3 3⍴2 6 1 0 0 20 3 2 1 ⋄ x←10 20 30 ⋄ f←1 0 24
      P←2 3⍴0 0 0.4082 0.3333 0.2857 0.5 ⋄ Q←2 3⍴0 0 0 0 0 1.2912
```

```
Z
         2          6          1
         0          0         20
         3          2          1
f
         1          0         24
x
        10         20         30
P
    0.0000     0.0000     0.4082
    0.3333     0.2857     0.5000


Q
    0.0000     0.0000     0.0000
    0.0000     0.0000     1.2912
```

Compute the energy transactions matrix **E** and, using **Q**, compute the energy deliveries to final demand **q**. Then, compute the vector of total energy outputs **g** and the hybrid-units matrix of transactions `ZS` and the vectors expressed in hybrid units of total final demand `fs` and of total outputs `xs`. Finally, compute the hybrid-units matrix of direct requirements `AS`.

```
      E←(P≠0)×Z[2 3;]÷P+(P=0)
      q←Q+.×f
      g←q++/E
```

```
      ZS←Z[1;],[1]E ◇ fs←f[1],q ◇ xs←x[1],g
      AS←ZS AMAT xs

E
        0.0       0.0      49.0
        9.0       7.0       2.0
q
        0.0      31.0
g
       49.0      49.0
ZS
        2.0       6.0       1.0
        0.0       0.0      49.0
        9.0       7.0       2.0
fs
        1.0       0.0      31.0
xs
       10.0      49.0      49.0


AS
      0.200     0.122     0.020
      0.000     0.000     1.000
      0.900     0.143     0.041
```

From the elements of **g**, construct the matrix to **G** to enable computing the matrix `GS` for extracting the energy rows from hybrid-units matrices `AS`, defining the hybrid-units matrix of direct energy requirements `DS`, and extracting the energy rows the hybrid units matrix of total requirements `LS`, defining the matrix of total energy requirements `ALPHA`.

```
      G←2 3ρ0,g[1],0 0 0 g[2]
      GS←G+.×DIAG ÷xs
      DS←GS+.×AS
      ALPHA←GS+.×LS←LINV AS

G
          0        49         0
          0         0        49
GS
          0         1         0
          0         0         1
DS
      0.000     0.000     1.000
      0.900     0.143     0.041
LS
      1.557     0.230     0.272
      1.717     1.428     1.526
      1.716     0.428     1.526
ALPHA
      1.717     1.428     1.526
      1.716     0.428     1.526
```

Define the new vector of final demands in hybrid units `Δf` and use `ALPHA` to compute the vector of total energy consumption `Δgs` necessary to deliver the new final demands.

```
      Δgs←ALPHA+.×Δfs←2 0 18

Δfs
2 0 18
Δgs
30.898167 30.898167
```

For the traditional energy input-output formulation we must express the new vector of final demands in economic units rather than hybrid units, which we call `Δf`. From **Z** and **x** we can compute the matrix of technical coefficients in economic rather than hybrid-units **A** and the Leontief inverse **L**. From **E** and **x** we can compute the matrix of direct energy coefficients **D** and the energy coefficients for deliveries to final demand `Gf` from which we finally compute the total energy coefficients `EPS` and use `EPS` to compute the total energy consumption supporting `Δf`, which we label `Δg`.

```
      Δf←Δfs×1 1 1.2912
      L←LINV A←Z AMAT x
      D←E÷2 3ρx
      Gf←2 3ρ0 ⋄ Gf[2;3]←f[3]÷Q[2;3]
      EPS←Gf+D+.×L
      Δg←EPS+.×Δf

Δf
       2.0       0.0       23.2
A
     0.200     0.300     0.033
     0.000     0.000     0.667
     0.300     0.100     0.033
L
     1.385     0.451     0.359
     0.308     1.174     0.821
     0.462     0.262     1.231
D
     0.000     0.000     1.633
     0.900     0.350     0.067
Gf
     0.000     0.000     0.000
     0.000     0.000    18.587
EPS
     0.754     0.427     2.010
     1.385     0.835    19.280
Δg
      48.2     450.9
```

Problem 12.4: Energy Conservation Conditions

This problem explores the conditions for energy conservation in an input-output model. The energy conservation conditions in an input-output model can be expressed as $\boldsymbol{\alpha}\hat{\mathbf{x}} = \boldsymbol{\alpha}\mathbf{Z} + \mathbf{G}$ where

$\boldsymbol{\alpha}$ is the matrix of total energy coefficients, **Z** is the matrix of interindustry transactions, **x** is the vector of total outputs, and **G** is the matrix of primary energy outputs.

We can show that the hybrid-units formulation of the energy input-output model—that is, where **x** is replaced by $\mathbf{x}^*$ and **Z** is replaced by $\mathbf{Z}^*$—satisfies these conditions in general: $\boldsymbol{\alpha}\hat{\mathbf{x}}^* = \boldsymbol{\alpha}\mathbf{Z}^* + \mathbf{G}$ and $\mathbf{Z}^* = \mathbf{A}^*\hat{\mathbf{x}}^*$, so $\boldsymbol{\alpha}\hat{\mathbf{x}}^* = \boldsymbol{\alpha}\mathbf{A}^*\hat{\mathbf{x}}^* + \mathbf{G}$. Rearranging, this becomes $\boldsymbol{\alpha}(\mathbf{I}-\mathbf{A}^*)\hat{\mathbf{x}}^* = \mathbf{G}$ or $\boldsymbol{\alpha} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I}-\mathbf{A}^*)^{-1}$, which is the definition of the total energy requirements matrix in the hybrid-units energy input-output formulation.

Given the following two tables of total energy coefficients, we adopt the convention that crude oil is a primary energy sector while refined petroleum and electricity are both secondary energy sectors.

Case 1	Crude Oil	Refined Petroleum	Electricity	Autos
Crude Oil	0	.6	.5	.3
Refined Petroleum	0	.4	.5	.2
Electricity	0	.2	0	.1

Case 2	Crude Oil	Refined Petroleum	Electricity	Autos
Crude Oil	0	.6	.5	.3
Refined Petroleum	0	.4	.2	.1
Electricity	0	.2	0	.1

Case 1 satisfies the energy conservation conditions, since $\boldsymbol{\alpha}_{ref.pet} + \boldsymbol{\alpha}_{elec.} = \boldsymbol{\alpha}_{crude} = \begin{bmatrix} .6 & .5 & .3 \end{bmatrix}$ i.e., the sum of all secondary energy consumed for energy type in the economy equals the total primary energy consumed by each energy sector. Case 2 fails to satisfy the energy conservation conditions, since $\boldsymbol{\alpha}_{ref.pet} + \boldsymbol{\alpha}_{elec.} = \begin{bmatrix} .6 & .2 & .2 \end{bmatrix} \neq \boldsymbol{\alpha}_{crude} = [.6 \quad .5 \quad .3]$

Problem 12.5: Comparison of Total Energy Requirements

This problem compares the total energy requirements matrices for the traditional and contemporary energy input-output formulations.

Problem 12.5 Overview

Consider an input-output economy defined (in \$10^6 units) by $\mathbf{Z} = \begin{bmatrix} 0 & 10 & 0 \\ 5 & 5 & 5 \\ 0 & 0 & 0 \end{bmatrix}$, $\mathbf{f} = \begin{bmatrix} 0 \\ 25 \\ 20 \end{bmatrix}$, and $\mathbf{x} = \begin{bmatrix} 10 \\ 40 \\ 20 \end{bmatrix}$. The first two of the three industries are energy industries with patterns of output allocation expressed in energy terms (10^{15} Btus) for interindustry transactions, $\mathbf{E} = \begin{bmatrix} 0 & 40 & 0 \\ 5 & 5 & 15 \end{bmatrix}$, and for final demand, $\mathbf{g} = \begin{bmatrix} 0 \\ 15 \end{bmatrix}$. First, we compute the direct and total requirements as

$$\mathbf{A} = \begin{bmatrix} 0 & .25 & 0 \\ .5 & .13 & .25 \\ 0 & 0 & 0 \end{bmatrix} \text{ and } \mathbf{L} = \begin{bmatrix} 1.17 & .33 & .08 \\ .67 & 1.33 & .33 \\ 0 & 0 & 1 \end{bmatrix}.$$

With the traditional energy input-output analysis formulation we have

$$\boldsymbol{\varepsilon} = \mathbf{DL} + \tilde{\mathbf{Q}} = \begin{bmatrix} .67 & 1.33 & .33 \\ .67 & .94 & .83 \end{bmatrix} \text{ where } \mathbf{D} = \mathbf{E}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} 0 & 1 & 0 \\ .5 & .13 & .75 \end{bmatrix} \text{ and } \tilde{\mathbf{Q}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.6 & 0 \end{bmatrix}.$$

With the hybrid-units formulation we have $\mathbf{Z}^* = \begin{bmatrix} 0 & 40 & 0 \\ 5 & 5 & 15 \\ 0 & 0 & 0 \end{bmatrix}$, $\mathbf{x}^* = \begin{bmatrix} 40 \\ 40 \\ 20 \end{bmatrix}$,

$$\mathbf{G} = \begin{bmatrix} 40 & 0 & 0 \\ 0 & 40 & 0 \end{bmatrix} \text{ and } \mathbf{A}^* = \begin{bmatrix} 0 & 1 & 0 \\ .13 & .13 & .75 \\ 0 & 0 & 0 \end{bmatrix}, \text{ so } \boldsymbol{\alpha} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^*)^{-1} = \begin{bmatrix} 1.167 & 1.333 & 1.0 \\ .167 & 1.333 & 1.0 \end{bmatrix}.$$

Computational Notes

Define the matrix of interindustry transactions **Z** and the vector of total outputs **x**. Also, define the matrix of energy transactions **E** and the vector of energy deliveries to final demand **q**. We compute the vector of final demands **f** and the vector of total energy consumption **g**.

```
     Z←3 3⍴0 10 0 5 5 5 0 0 0 ⋄ x←10 40 20
     E←2 3⍴0 40 0 5 5 15 ⋄ q←0 15
     f←x-+/Z ⋄ g←q++/E

Z
         0         10          0
         5          5          5
         0          0          0
f
         0         25         20
x
        10         40         20
E
         0         40          0
         5          5         15
q
         0         15
g
        40         40
```

For the traditional method, we compute the matrix of technical coefficients **A**, the Leontief inverse **L**, the direct energy requirements **D**, the matrix of coefficients for energy deliveries to final demand `QT`, and finally the total energy coefficients, `EPS`.

```
     L←INV A←Z AMAT x ⋄ D←E+.×DIAG ÷x ⋄ QT←(DIAG 0 0.6),2 1ρ0
     EPS←QT+D+.×L

A
     0.000     0.250     0.000
     0.500     0.125     0.250
     0.000     0.000     0.000
L
     1.167     0.333     0.083
     0.667     1.333     0.333
     0.000     0.000     1.000
D
     0.000     1.000     0.000
     0.500     0.125     0.750
QT
     0.000     0.000     0.000
     0.000     0.600     0.000
EPS
     0.667     1.333     0.333
     0.667     0.933     0.833
```

For the hybrid-units formulation, we assemble the hybrid units transactions table `ZS` with the energy rows from **E** and the other rows from **Z**. Similarly, the hybrid-units vector of total final demands fs is assembled from the **q** for the energy sectors and **f** for the other sectors. With `ZS` and `fs` we can compute the hybrid-units vector of total outputs `xs` and the we can then we can compute the hybrid-units matrices of direct requirements `AS` and the hybrid units matrix of total requirements `LS`. From the elements of **g**, we construct the matrix **G** to enable computing the matrix `GS` for extracting the energy rows from `LS`, defining the matrix of total energy requirements `ALPHA`.

```
     ZS←E,[1]Z[3;] ⋄ fs←q,f[3] ⋄ xs←fs++/ZS
     LS←LINV AS←ZS AMAT xs
     G←(DIAG g),2 1ρ0 ⋄ GS←G+.×(DIAG ÷xs)
     ALPHA←GS+.×LS

ZS
         0        40         0
         5         5        15
         0         0         0
fs
         0        15        20
xs
        40        40        20
AS
     0.000     1.000     0.000
     0.125     0.125     0.750
     0.000     0.000     0.000
LS
     1.167     1.333     1.000
     0.167     1.333     1.000
     0.000     0.000     1.000
```

```
G
    40.000      0.000      0.000
     0.000     40.000      0.000
GS
     1.000      0.000      0.000
     0.000      1.000      0.000
ALPHA
     1.167      1.333      1.000
     0.167      1.333      1.000
```

Problem 12.6: Hybrid-Units Energy Input-Output Model and Impact Analysis

This problem illustrates use of the hybrid-units energy input-output model in impact analysis.

Problem 12.6 Overview

Consider the following hybrid units transactions matrix and vector of total outputs, i.e., the first three rows of the energy sectors (oil, coal, and electricity) are measured in millions of Btu and the last row, manufacturing, is measured in millions of dollars: $\mathbf{Z}^* = \begin{bmatrix} 0 & 0 & 40 & 0 \\ 0 & 0 & 60 & 0 \\ 2 & 3 & 12 & 48 \\ 15 & 20 & 30 & 40 \end{bmatrix}$ and $\mathbf{x}^* = \begin{bmatrix} 40 \\ 60 \\ 100 \\ 200 \end{bmatrix}$. Using $\mathbf{A}^* = \mathbf{Z}^*(\hat{\mathbf{x}}^*)^{-1}$ we can compute $\mathbf{A}^* = \begin{bmatrix} 0 & 0 & .4 & 0 \\ 0 & 0 & .6 & 0 \\ .05 & .05 & .12 & .24 \\ .375 & .333 & .3 & .24 \end{bmatrix}$ and

$$\mathbf{L}^* = (\mathbf{I} - \mathbf{A}^*)^{-1} = \begin{bmatrix} 1.1024 & .0945 & .6299 & .1890 \\ .1535 & 1.1417 & .9449 & .2835 \\ .2559 & .2362 & 1.5748 & .4724 \\ .6767 & .6086 & 1.2795 & 1.6339 \end{bmatrix}.$$

If we project a final demand for manufactured goods will increase by \$200 billion, the change in final demand can be written as $(\Delta\mathbf{f}^*)' = \begin{bmatrix} 0 & 0 & 0 & 200 \end{bmatrix}$ so the corresponding change in total energy consumption can be expressed as $\Delta\mathbf{g} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}\mathbf{L}^*\Delta\mathbf{f}^* = \begin{bmatrix} 37.7953 \\ 56.6929 \\ 0 \end{bmatrix}$ where $\mathbf{G}(\hat{\mathbf{x}}^*)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$. The total primary energy intensity is $\mathbf{i}'\Delta\mathbf{g} = 94.4882$.

Computational Notes

We everything assumed in hybrid units, we define the interindustry transactions matrix **Z** and vector of total outputs **x**, and the vector of expected changes in final demand `Δf`. We compute the matrix of technical coefficients **A** and the Leontief inverse **L**.

```
      Z←4 4ρ0 0 40 0 0 0 60 0 2 3 12 48 15 20 30 40
      x←40 60 100 200 ⋄ Δf←0 0 0 200
      L←INV A←Z AMAT x

Z
         0         0        40         0
         0         0        60         0
         2         3        12        48
        15        20        30        40


x
        40        60       100       200
A
    0.0000    0.0000    0.4000    0.0000
    0.0000    0.0000    0.6000    0.0000
    0.0500    0.0500    0.1200    0.2400
    0.3750    0.3333    0.3000    0.2000
L
    1.1024    0.0945    0.6299    0.1890
    0.1535    1.1417    0.9449    0.2835
    0.2559    0.2362    1.5748    0.4724
    0.6767    0.6086    1.2795    1.6339
```

We construct the matrix GS to extract only the primary energy sectors from hybrid-units matrices and compute the total primary energy coefficients ALPHA and subsequently the value for primary energy intensity pei.

```
      GS←3 4ρ1 0 0 0 0 1 0 0 0 0 0 0
      ALPHA←GS+.×L
      pei←+/Δg←ALPHA+.×Δf

GXI
         1         0         0         0
         0         1         0         0
         0         0         0         0
ALPHA
    1.1024    0.0945    0.6299    0.1890
    0.1535    1.1417    0.9449    0.2835
    0.0000    0.0000    0.0000    0.0000
Δf
         0         0         0       200
Δg
   37.7953   56.6929    0.0000

pei
   94.4882
```

Problem 12.7: Impacts of New Energy Technologies

This problem illustrates the use of energy input-output analysis to evaluate the relative impact of alternative energy technologies on total energy consumption.

Problem 12.7 Overview

For the economy specified in Problem 12.6, two alternative technologies are proposed for generating electric power, which involve alternative new specifications for the matrix of technical coefficients depicting different "recipes" for electric power production in the economy, $\mathbf{A}^{*(I)}$ and $\mathbf{A}^{*(II)}$. For the original electric power generation column of the technical coefficients matrix is given by $\mathbf{A}^*$, suppose the two alternative changed columns of the technical coefficients matrix corresponding to the alternative technologies are given by $\mathbf{A}_{\bullet 3}^{*(I)} = \begin{bmatrix} .2 \\ .7 \\ .1 \\ .4 \end{bmatrix}$ and $\mathbf{A}_{\bullet 3}^{*(II)} = \begin{bmatrix} .5 \\ .4 \\ .12 \\ .4 \end{bmatrix}$ as well as a vector of new final demands of $\Delta\mathbf{f}^* = \begin{bmatrix} 0 \\ 0 \\ 20 \\ 30 \end{bmatrix}$ is presented to the economy.

To determine which economy [matrix incorporating the specifications $\mathbf{A}^*$, $\mathbf{A}^{*(I)}$ or $\mathbf{A}^{*(II)}$] reflects the most energy intensive manufacturing, i.e., which one of the two new technologies consumes the least primary energy per unit of final demand of manufacturing and how much less primary energy does that technology consume than the other to support final demand $\Delta\mathbf{f}^*$, first using $\mathbf{A}^* = \mathbf{Z}^*(\hat{\mathbf{x}}^*)^{-1}$ for the alternative power generation technologies, *I* and *II*, we can specify the corresponding matrices of technical coefficients and total requirements as

$$\mathbf{A}^{*(I)} = \begin{bmatrix} 0 & 0 & .2 & 0 \\ 0 & 0 & .7 & 0 \\ .05 & .05 & .1 & .24 \\ .375 & .333 & .4 & .20 \end{bmatrix}, \ (\mathbf{I} - \mathbf{A}^{*(I)})^{-1} = \begin{bmatrix} 1.0506 & .0467 & .3113 & .0934 \\ .1770 & 1.1634 & 1.0895 & .3268 \\ .2529 & .2335 & 1.5564 & .4669 \\ .6927 & .6234 & 1.3781 & 1.6634 \end{bmatrix},$$

$$\mathbf{A}^{*(II)} = \begin{bmatrix} 0 & 0 & .5 & 0 \\ 0 & 0 & .4 & 0 \\ .05 & .05 & .12 & .24 \\ .375 & .333 & .4 & .2 \end{bmatrix} \text{ and } (\mathbf{I} - \mathbf{A}^{*(II)})^{-1} = \begin{bmatrix} 1.1313 & .1212 & .8081 & .2424 \\ .1051 & 1.0970 & .6465 & .1939 \\ .2626 & .2424 & 1.6162 & .4848 \\ .7054 & .6351 & 1.4562 & 1.6869 \end{bmatrix}.$$

If we designate the technical coefficients of original economy by $\mathbf{A}^{*(0)}$, the total energy consumption associate with the new final demand, $\Delta\mathbf{f}^*$, is $\Delta\mathbf{g}^{(0)} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^{*(0)})^{-1}\Delta\mathbf{f}$ and for the technical coefficients modified by the two alternative technologies, $\Delta\mathbf{g}^{(I)} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^{*(I)})^{-1}\Delta\mathbf{f}$ and $\Delta\mathbf{g}^{(II)} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^{*(II)})^{-1}\Delta\mathbf{f}$, respectively where

$$\mathbf{G}(\hat{\mathbf{x}}^*)^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{ and } \Delta\mathbf{f}^* = \begin{bmatrix} 0 \\ 0 \\ 20 \\ 30 \end{bmatrix}, \text{ we can write}$$

$$\Delta\mathbf{g} = \left[\Delta\mathbf{g}^{(0)} \;\vdots\; \Delta\mathbf{g}^{(I)} \;\vdots\; \Delta\mathbf{g}^{(II)} \right] = \left[\begin{array}{c:c:c} 18.2677 & 9.0272 & 23.4343 \\ 27.4016 & 31.5953 & 18.7475 \\ 0 & 0 & 0 \end{array}\right]$$ which provides the total energy of each fuel type to support $\Delta\mathbf{f}^*$.

The total primary energy intensity is given by $\mathbf{i}'(\Delta\mathbf{g}) = \begin{bmatrix} 45.6693 & 40.6226 & 42.1818 \end{bmatrix}$, so employment of technology *I* consumes 1.5592 less primary energy than employment of technology *II*. Both new technologies *I* and *II* are more efficient than the base technology.

Computational Notes

We define the matrix of interindustry transactions **Z** and the vector of total outputs **x**. We also define the new final demand `Δf`. Now we compute the matrix of technical coefficients for the original economy `A0` and the associated Leontief inverse `L0`. We also construct the two modified matrices of technical coefficients reflected the new technologies `A1` and `A2` along with their respective Leontief inverses, `L1` and `L2`.

```
      Z←4 4ρ0 0 40 0 0 0 60 0 2 3 12 48 15 20 30 40
      x←40 60 100 200 ⋄ Δf←0 0 20 30
      L0←LINV A2←A1←A0←Z AMAT x
      A1[;3]←0.2 0.7 0.1 0.4 ⋄ A2[;3]←0.5 0.4 0.12 0.4
      L1←LINV A1 ⋄ L2←LINV A2
```

```
A0
    0.0000    0.0000    0.4000    0.0000
    0.0000    0.0000    0.6000    0.0000
    0.0500    0.0500    0.1200    0.2400
    0.3750    0.3333    0.3000    0.2000
L0
    1.1024    0.0945    0.6299    0.1890
    0.1535    1.1417    0.9449    0.2835
    0.2559    0.2362    1.5748    0.4724
    0.6767    0.6086    1.2795    1.6339
A1
    0.0000    0.0000    0.2000    0.0000
    0.0000    0.0000    0.7000    0.0000
    0.0500    0.0500    0.1000    0.2400
    0.3750    0.3333    0.4000    0.2000
L1
    1.0506    0.0467    0.3113    0.0934
    0.1770    1.1634    1.0895    0.3268
    0.2529    0.2335    1.5564    0.4669
    0.6927    0.6234    1.3781    1.6634
```

```
A2
     0.0000    0.0000    0.5000    0.0000
     0.0000    0.0000    0.4000    0.0000
     0.0500    0.0500    0.1200    0.2400
     0.3750    0.3333    0.4000    0.2000
L2
     1.1313    0.1212    0.8081    0.2424
     0.1051    1.0970    0.6465    0.1939
     0.2626    0.2424    1.6162    0.4848
     0.7054    0.6351    1.4562    1.6869

A0
     0.0000    0.0000    0.4000    0.0000
     0.0000    0.0000    0.6000    0.0000
     0.0500    0.0500    0.1200    0.2400
     0.3750    0.3333    0.3000    0.2000
L0
     1.1024    0.0945    0.6299    0.1890
     0.1535    1.1417    0.9449    0.2835
     0.2559    0.2362    1.5748    0.4724
     0.6767    0.6086    1.2795    1.6339

A1
     0.0000    0.0000    0.2000    0.0000
     0.0000    0.0000    0.7000    0.0000
     0.0500    0.0500    0.1000    0.2400
     0.3750    0.3333    0.4000    0.2000
L1
     1.0506    0.0467    0.3113    0.0934
     0.1770    1.1634    1.0895    0.3268
     0.2529    0.2335    1.5564    0.4669
     0.6927    0.6234    1.3781    1.6634
A2
     0.0000    0.0000    0.5000    0.0000
     0.0000    0.0000    0.4000    0.0000
     0.0500    0.0500    0.1200    0.2400
     0.3750    0.3333    0.4000    0.2000
L2
     1.1313    0.1212    0.8081    0.2424
     0.1051    1.0970    0.6465    0.1939
     0.2626    0.2424    1.6162    0.4848
     0.7054    0.6351    1.4562    1.6869x
```

We now compute the vectors total outputs needed to support the new final demand for the base case and the two cases of alternative technologies as columns of a matrix `Δx`. Using the matrix to extract only the primary energy rows `GXI`, compute the matrix of total energy consumption `Δg` corresponding to `Δf` and compute the column sums to generate the thee-element vector of primary energy intensities for the three cases `pei`.

```
      Δx1←L1+.×Δf ⋄ Δx2←L2+.×Δf ⋄ Δx0←L+.×Δf
      Δx←⍉3 4⍴Δx0,Δx1,Δx2
```

```
      GXI←3 4⍴1 0 0 0 0 1 0 0 0 0 0 0
      pei←+⌿Δg←GXI+.×Δx

GXI
         1         0         0         0
         0         1         0         0
         0         0         0         0


Δf
         0         0        20        30
Δx0
      18.3       9.0      23.4
      27.4      31.6      18.7
      45.7      45.1      46.9
      74.6      77.5      79.7
Δg
   18.2677    9.0272   23.4343
   27.4016   31.5953   18.7475
    0.0000    0.0000    0.0000
pei
   45.6693   40.6226   42.1818
```

Problem 12.8: Technical Change and Total Energy Consumption

This problem explores calculation of the total energy consumption in an economy associated with an energy saving manufacturing process technology.

Problem 12.8 Overview

Using the original energy-economy defined in Problem 12.6, for the direct requirements matrix, $\mathbf{A}^*$, suppose an energy conserving manufacturing process is developed that can be depicted as a new column of the matrix of technical coefficients for manufacturing, given by $\mathbf{A}_{\bullet 4}^{*(new)} = \begin{bmatrix} 0 \\ 0 \\ .12 \\ .20 \end{bmatrix}$.

The technical coefficient matrix incorporating the new manufacturing technology is

$$\mathbf{A}^{*(new)} = \begin{bmatrix} 0 & 0 & .2 & 0 \\ 0 & 0 & .7 & 0 \\ .05 & .05 & .1 & .12 \\ .375 & .333 & .4 & .2 \end{bmatrix} \text{ and } \mathbf{L}^{*(new)} = \begin{bmatrix} 1.0580 & .0546 & .5461 & .0819 \\ .0870 & 1.0819 & .8191 & .1229 \\ .1451 & .1365 & 1.3652 & .2048 \\ .5866 & .5276 & 1.1092 & 1.4164 \end{bmatrix}.$$

So, for $\Delta\mathbf{g}^{(0)} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^{*(0)})^{-1}\Delta\mathbf{f}$ and $\Delta\mathbf{g}^{(new)} = \mathbf{G}(\hat{\mathbf{x}}^*)^{-1}(\mathbf{I} - \mathbf{A}^{*(new)})^{-1}\Delta\mathbf{f}$ we can write

$$\mathbf{i}'(\Delta\mathbf{g}) = \mathbf{i}'\left[\Delta\mathbf{g}^{(0} \;\vdots\; \Delta\mathbf{g}^{(new)}\right] = \left[\begin{array}{c:c} 18.2677 & 13.3788 \\ 27.4016 & 20.0683 \\ 0.0000 & 0.0000 \end{array}\right]\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 45.6693 & 33.4471 \end{bmatrix}.$$

Hence the primary energy saved by adopting the new technology is $45.6693 - 33.4471 = 12.2222$.

Computational Notes

Retrieve the data for the economy defined in Problem 12.6, including **Z**, **x**, and `Δf`. Now we compute the matrix of technical coefficients for the original economy `A0` and the associated Leontief inverse `L0`, as well the modified matrix of technical coefficients reflecting the new technologies `A1` and the associated Leontief inverse `L1`.

```
A0
    0.0000    0.0000    0.4000    0.0000
    0.0000    0.0000    0.6000    0.0000
    0.0500    0.0500    0.1200    0.2400
    0.3750    0.3333    0.3000    0.2000
L0
    1.1024    0.0945    0.6299    0.1890
    0.1535    1.1417    0.9449    0.2835
    0.2559    0.2362    1.5748    0.4724
    0.6767    0.6086    1.2795    1.6339
A1
    0.0000    0.0000    0.4000    0.0000
    0.0000    0.0000    0.6000    0.0000
    0.0500    0.0500    0.1200    0.1200
    0.3750    0.3333    0.3000    0.2000
L1
    1.0506    0.0467    0.3113    0.0934
    0.1770    1.1634    1.0895    0.3268
    0.2529    0.2335    1.5564    0.4669
    0.6927    0.6234    1.3781    1.6634
```

We now compute the vectors total outputs needed to support the new final demand for the base case and the alternative technology case as columns of a matrix `Δx`. Using the matrix to extract only the primary energy rows `GXI`, compute the matrix of total energy consumption `Δg` corresponding to `Δf` and compute the column sums to generate the two-element vector of primary energy intensities for the two cases `pei`, and compute the difference in primary energy consumption `npei`.

```
 Δx
      18.3      13.4
      27.4      20.1
      45.7      33.4
      74.6      64.7
GXI
         1         0         0         0
         0         1         0         0
         0         0         0         0
Δg
   18.2677   13.3788
   27.4016   20.0683
    0.0000    0.0000
pei
   45.6693   33.4471
npei
   12.2222
```

Problem 12.9: Analyzing Impacts of and Oil Supply Reduction

This problem explores the use of an energy input-output model in analyzing the implications of an oil supply reduction.

Problem 12.9 Overview

Again using the original energy economy introduced in Problem 12.6 but with the added information that the energy prices to final demand are given by $\mathbf{p}_f = [p_{kf}] = \begin{bmatrix} 2 \\ 1 \\ 3 \end{bmatrix}$, from the original matrix of technical coefficients, $\mathbf{A}^{*(0)}$, we can compute

$(\mathbf{I} - \mathbf{A}^{*(0)}) = \begin{bmatrix} 1 & 0 & -.4 & 0 \\ 0 & 1 & -.6 & 0 \\ -.05 & -.05 & .88 & -.24 \\ -.375 & -.333 & -.3 & .8 \end{bmatrix}$. The GDP for the original economy can be found by

$GDP = \mathbf{i}'\tilde{\mathbf{Q}}\mathbf{f}^* = \mathbf{i}'\tilde{\mathbf{Q}}(\mathbf{I} - \mathbf{A}^{*(0)})\mathbf{x}^* = 105$ where $\tilde{\mathbf{Q}} = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 3 & 0 \end{bmatrix}$ and

$(\mathbf{x}^*)' = \begin{bmatrix} 40 & 60 & 100 & 200 \end{bmatrix}$.

For a 10 percent reduction in availability of oil supply, the vector of total outputs becomes $(\mathbf{x}^*)' = \begin{bmatrix} 36 & 60 & 100 & 200 \end{bmatrix}$. Hence, we can compute the GDP as the sum of the corresponding final demand (measured in dollars) which we determine once again by $GDP = \mathbf{i}'\tilde{\mathbf{Q}}\mathbf{f}^* = \mathbf{i}'\tilde{\mathbf{Q}}(\mathbf{I} - \mathbf{A}^{*(0)})\mathbf{x}^* = 97.6$. The reduction in GDP due to the oil shortage is $105 - 97.6 = 7.4$. When the new technologies are incorporated into the technical coefficients matrix it becomes $\mathbf{A}^{*(new)} = \begin{bmatrix} 0 & 0 & .2 & 0 \\ 0 & 0 & .7 & 0 \\ .05 & .05 & .1 & .12 \\ .375 & .333 & .4 & .2 \end{bmatrix}$ and

$(\mathbf{I} - \mathbf{A}^{*(new)}) = \begin{bmatrix} 1 & 0 & -.2 & 0 \\ 0 & 1 & -.7 & 0 \\ -.05 & -.05 & .9 & -.12 \\ -.375 & -.333 & -.4 & .8 \end{bmatrix}$ and, as before, we compute GDP by

$GDP = \mathbf{i}'\tilde{\mathbf{Q}}\mathbf{f}^* = \mathbf{i}'\tilde{\mathbf{Q}}(\mathbf{I} - \mathbf{A}^{*(new)})\mathbf{x}^* = 205.6$. This turns out to be not a reduction at all but an increase in *GDP* of 100.6.

Computational Notes

Retrieve the data for the economy defined in Problem 12.6, including **Z**, **x**, and `Δf`. Now we compute the matrix of technical coefficients for the original economy `A0` as well as the matrix **I** minus A0 (with a suitable identity matrix **I**), which we call IA0. We do the same for the modified matrix of technical coefficients reflecting the new technologies A2 to produce IA2.

```
Z←4 4⍴0 0 40 0 0 0 60 0 2 3 12 48 15 20 30 40
x←40 60 100 200 ⋄ I←4 4⍴1,4⍴0
A0←A2←Z AMAT x
A2[;4]←0 0 0.12 0.2
A2[;3]←0.2 0.7 0.1 0.4
IA2←I-A2
```

```
A0
    0.0000    0.0000    0.4000    0.0000
    0.0000    0.0000    0.6000    0.0000
    0.0500    0.0500    0.1200    0.2400
    0.3750    0.3333    0.3000    0.2000
IA0
    1.0000    0.0000   ¯0.4000    0.0000
    0.0000    1.0000   ¯0.6000    0.0000
   ¯0.0500   ¯0.0500    0.8800   ¯0.2400
   ¯0.3750   ¯0.3333   ¯0.3000    0.8000
A2
    0.0000    0.0000    0.2000    0.0000
    0.0000    0.0000    0.7000    0.0000
    0.0500    0.0500    0.1000    0.1200
    0.3750    0.3333    0.4000    0.2000
IA2
    1.0000    0.0000   ¯0.2000    0.0000
    0.0000    1.0000   ¯0.7000    0.0000
   ¯0.0500   ¯0.0500    0.9000   ¯0.1200
   ¯0.3750   ¯0.3333   ¯0.4000    0.8000
```

We use a matrix of energy prices QT, IA0, and **x** to compute the GDP for the original economy, which we call GDP0. We compute a new vector of total outputs reflection a 10% reduction in oil availability x1, and use it with QT to create the changed GDP for the original economy GDP1 and for the economy incorporating new technologies GDP2. Compute a vector of the differences in GDP with GDP1 and GDP2 compared with GDP0, which comprise a vector we call ∆GDP.

```
GDP0←+/QT+.×(IA0←I-A0)+.×x
x1←x-x×0.1 0 0 0
GDP1←+/QT+.×(IA1←I-A0)+.×x1
GDP2←+/QT+.×(IA2←I-A2)+.×x1
∆GDP←(GDP1,GDP2)-GDP0
```

```
QT
     2.000     0.000     0.000     0.000
     0.000     1.000     0.000     0.000
     0.000     0.000     3.000     0.000
GDP0
     105.0
x1
      36.0      60.0     100.0     200.0
```

```
GDP1
      97.6
GDP2
     205.6
∆GDP
     ¯7.4     100.6
```

Problem 12.10: Structural Change and Energy Input-Output Analysis

This problem uses energy input-output analysis to examine structural change using US input-output tables for two years.

Problem 12.10 Overview

Below are 9-sector 1963 and 1980 input-output tables for the United States expressed in hybrid units (quadrillions of Btus for energy sectors and millions of dollars for non-energy sectors). The first five sectors are energy sectors: (1) coal, (2) oil, (3) refined petroleum products, (4) electricity, and (5) natural gas. The remaining four sectors are non-energy sectors: (6) natural resources, (7) manufacturing, (8) transportation, and (9) services.

1980	1	2	3	4	5	6	7	8	9	Total Output
1	0.0012	0.0000	0.0007	1.5464	0.0000	0.0000	0.0002	0.0000	0.0000	18,597
2	0.0001	0.0319	0.8960	0.0001	0.8707	0.0000	0.0001	0.0000	0.0000	36,842
3	0.0063	0.0024	0.0612	0.3344	0.0008	0.0005	0.0002	0.0023	0.0002	31,215
4	0.0026	0.0021	0.0035	0.0822	0.0020	0.0000	0.0001	0.0000	0.0001	7,827
5	0.0006	0.0461	0.0301	0.4856	0.0720	0.0001	0.0003	0.0000	0.0001	19,244
6	0.2092	1.4027	0.5040	7.8254	0.4350	0.0896	0.0628	0.0355	0.0289	6,194,571
7	2.6323	0.8480	2.4090	3.5155	0.1804	0.2672	0.3780	0.0493	0.0626	18,081,173
8	0.1773	0.0806	2.1831	4.8195	0.0794	0.0199	0.0251	0.1289	0.0141	2,240,904
9	1.8576	2.6159	2.7945	8.5173	1.2302	0.1831	0.1238	0.1224	0.2027	23,803,723
1963	1	2	3	4	5	6	7	8	9	Total Output
1	0.0019	0.0000	0.0008	1.7415	0.0010	0.0000	0.0004	0.0001	0.0000	12,476
2	0.0000	0.0423	0.7996	0.0007	0.9308	0.0000	0.0003	0.0000	0.0000	30,384
3	0.0015	0.0011	0.0600	0.1973	0.0031	0.0004	0.0003	0.0021	0.0002	19,878
4	0.0015	0.0007	0.0018	0.0963	0.0002	0.0000	0.0001	0.0000	0.0000	3,128
5	0.0001	0.0035	0.0330	0.7046	0.0919	0.0000	0.0003	0.0001	0.0001	13,194
6	0.0456	0.4582	0.5926	7.9623	0.6565	0.1111	0.0835	0.0415	0.0426	4,865,092
7	0.8684	0.4081	1.1700	1.0933	0.0937	0.2340	0.4035	0.0498	0.0496	11,333,710
8	0.1105	0.0655	1.1964	4.5632	0.3965	0.0231	0.0256	0.0863	0.0121	1,131,226
9	0.4794	2.2388	1.9461	8.0643	1.1016	0.1121	0.0881	0.1203	0.1721	10,588,385

To determine the amounts of the change in total energy use of each energy type between 1963 and 1980 and the components of that change that are attributable to change in production functions, to change in final demand, and to the interaction between the changes in production functions and final demand between the two years, we begin by calculating the total requirements matrices $\mathbf{L}^{*(80)} = (\mathbf{I} - \mathbf{A}^{*(80)})^{-1}$ and $\mathbf{L}^{*(63)} = (\mathbf{I} - \mathbf{A}^{*(63)})^{-1}$. From the available data we must calculate final demands as $\mathbf{f}^{*(80)} = \mathbf{x}^{*(80)} - \mathbf{A}^{*}\mathbf{x}^{*(80)}$ and $\mathbf{f}^{*(63)} = \mathbf{x}^{*(63)} - \mathbf{A}^{*}\mathbf{x}^{*(63)}$. The total requirements matrices and vectors of final demand are the following:

1980	1	2	3	4	5	6	7	8	9	Final Demand
1	1.0081	0.0059	0.016	1.718	0.0099	0.0003	0.0007	0.0002	0.0002	3,258
2	0.0164	1.0923	1.0933	1.0115	1.0301	0.0014	0.0015	0.0032	0.0006	-10,684
3	0.0115	0.0076	1.0851	0.4513	0.0106	0.001	0.0007	0.003	0.0004	10,461
4	0.0038	0.0035	0.0089	1.1062	0.006	0.0002	0.0003	0.0001	0.0001	3,155
5	0.0058	0.0578	0.098	0.6569	1.1341	0.0005	0.0008	0.0004	0.0003	4,066
6	0.7803	2.1173	3.4665	15.249	2.6707	1.154	0.1361	0.0722	0.0559	3,596,887
7	5.0854	2.9969	8.6363	27.246	3.696	0.5453	1.7127	0.1654	0.1614	7,804,130
8	0.479	0.3575	3.3772	9.3886	0.5332	0.0517	0.0613	1.1667	0.0288	925,557
9	3.5349	4.7573	10.33	30.909	6.5227	0.369	0.3197	0.2444	1.3022	15,022,410
1963	1	2	3	4	5	6	7	8	9	Final Demand
1	1.0058	0.0026	0.0094	1.9521	0.0049	0.0004	0.0011	0.0002	0.0002	2,199
2	0.0056	1.0532	0.9444	1.0968	1.0861	0.0012	0.0021	0.0026	0.0007	-2,359
3	0.0032	0.0033	1.0732	0.2727	0.0094	0.0008	0.0008	0.0026	0.0004	8,630
4	0.0019	0.0011	0.0037	1.1145	0.0016	0.0001	0.0002	0.0001	0.0001	1,037
5	0.0025	0.0061	0.0483	0.8888	1.1087	0.0004	0.0009	0.0004	0.0003	3,540
6	0.2843	0.8465	2.0483	14.209	1.88	1.1866	0.1853	0.0793	0.0749	2,820,771
7	1.6793	1.3446	4.2429	14.296	2.1657	0.4915	1.7788	0.1478	0.1362	4,989,750
8	0.2025	0.1894	1.7661	7.6651	0.7549	0.0485	0.0604	1.1074	0.0233	456,425
9	0.8733	3.1616	6.1679	21.334	5.0473	0.2267	0.2346	0.202	1.2401	6,933,979

If we denote the energy rows of $\mathbf{L}^{*(80)}$ as $\boldsymbol{\alpha}^{80}$, the vector of total energy output as $\mathbf{g}^{80}$, and final demand as $\mathbf{f}^{80}$ (now in all cases dropping the $*$ for simplicity) with the analogous designations for 1963, we can compute the changes in energy consumption as

$$\mathbf{g}^{80} - \mathbf{g}^{63} = \boldsymbol{\alpha}^{63}(\mathbf{f}^{80} - \mathbf{f}^{63}) + (\boldsymbol{\alpha}^{80} - \boldsymbol{\alpha}^{63})\mathbf{f}^{63} + (\boldsymbol{\alpha}^{80} - \boldsymbol{\alpha}^{63})(\mathbf{f}^{80} - \mathbf{f}^{63}) = \begin{bmatrix} 6{,}121.4 \\ 6{,}457.3 \\ 11{,}337.2 \\ 4{,}698.8 \\ 6{,}049.6 \end{bmatrix}$$

where the effect caused by changing final demand is $\boldsymbol{\alpha}^{63}(\mathbf{f}^{80} - \mathbf{f}^{63}) = \begin{bmatrix} 10{,}467.2 \\ 9{,}433.3 \\ 9{,}967.9 \\ 3{,}738.7 \\ 8{,}021.1 \end{bmatrix}$;

the effect caused by changes in production functions is $(\boldsymbol{\alpha}^{80} - \boldsymbol{\alpha}^{63})\mathbf{f}^{63} = \begin{bmatrix} -2460.4 \\ -1257.5 \\ 756.8 \\ 613.3 \\ -502.4 \end{bmatrix}$;

and the effect of interaction of final demand and production function changes is

$$(\boldsymbol{\alpha}^{80} - \boldsymbol{\alpha}^{63})(\mathbf{f}^{80} - \mathbf{f}^{63}) = \begin{bmatrix} -1885.4 \\ -1718.5 \\ 612.5 \\ 346.8 \\ -1469.1 \end{bmatrix}.$$

Computational Notes

We presume that the matrices of technical coefficients for 1963 and 1980, `A1` and `A2`, respectively, and the corresponding vectors of total commodity outputs, `q1` and `q2`, are in the APL workspace. We compute the matrices of total requirements, `L1` and `L2`, as well as the vectors of commodity final demands, `e1` and `e2`. We define the matrix `QTI` to extract the energy rows from `L1` and `L2`, and compute the matrices of total energy coefficients, `ei1` and `ei2`.

```
      L1←LINV A1 ⋄ L2←LINV A2 ⋄ e1←q1-A1+.×q1 ⋄ e2←q2-A2+.×q2
      QTI←(5 5ρ1,5ρ0),5 4ρ0
      ei1←QTI+.×L1
      ei2←QTI+.×L2
```

```
L1
    1.0058    0.0026    0.0094    1.9521    0.0049    0.0004    0.0011    0.0002    0.0002
    0.0056    1.0532    0.9444    1.0968    1.0861    0.0012    0.0021    0.0026    0.0007
    0.0032    0.0033    1.0732    0.2727    0.0094    0.0008    0.0008    0.0026    0.0004
    0.0019    0.0011    0.0037    1.1145    0.0016    0.0001    0.0002    0.0001    0.0001
    0.0025    0.0061    0.0483    0.8888    1.1087    0.0004    0.0009    0.0004    0.0003
    0.2843    0.8465    2.0483   14.2093    1.8800    1.1866    0.1853    0.0793    0.0749
    1.6793    1.3446    4.2429   14.2964    2.1657    0.4915    1.7788    0.1478    0.1362
    0.2025    0.1894    1.7661    7.6651    0.7549    0.0485    0.0604    1.1074    0.0233
    0.8733    3.1616    6.1679   21.3339    5.0473    0.2267    0.2346    0.2020    1.2401
L2
    1.0081    0.0059    0.0160    1.7180    0.0099    0.0003    0.0007    0.0002    0.0002
    0.0164    1.0923    1.0933    1.0115    1.0301    0.0014    0.0015    0.0032    0.0006
    0.0115    0.0076    1.0851    0.4513    0.0106    0.0010    0.0007    0.0030    0.0004
    0.0038    0.0035    0.0089    1.1062    0.0060    0.0002    0.0003    0.0001    0.0001
    0.0058    0.0578    0.0980    0.6569    1.1341    0.0005    0.0008    0.0004    0.0003
    0.7803    2.1173    3.4665   15.2493    2.6707    1.1540    0.1361    0.0722    0.0559
    5.0854    2.9969    8.6363   27.2464    3.6960    0.5453    1.7127    0.1654    0.1614
```

```
    0.4790    0.3575    3.3772    9.3886    0.5332    0.0517    0.0613    1.1667    0.0288
    3.5349    4.7573   10.3297   30.9091    6.5227    0.3690    0.3197    0.2444    1.3022
e1
    2199.4   ¯2359.0    8630.4    1037.0    3540.0 2820771.1 4989750.4  456424.6 6933978.6
e2
    3257.6  ¯10683.6   10461.1    3154.7    4066.4 3596886.9 7804129.8  925557.215022409.9

QTI
  1   0   0   0   0   0   0   0   0
  0   1   0   0   0   0   0   0   0
  0   0   1   0   0   0   0   0   0
  0   0   0   1   0   0   0   0   0
  0   0   0   0   1   0   0   0   0
ei1
    1.0058    0.0026    0.0094    1.9521    0.0049    0.0004    0.0011    0.0002    0.0002
    0.0056    1.0532    0.9444    1.0968    1.0861    0.0012    0.0021    0.0026    0.0007
    0.0032    0.0033    1.0732    0.2727    0.0094    0.0008    0.0008    0.0026    0.0004
    0.0019    0.0011    0.0037    1.1145    0.0016    0.0001    0.0002    0.0001    0.0001
    0.0025    0.0061    0.0483    0.8888    1.1087    0.0004    0.0009    0.0004    0.0003
ei2
    1.0081    0.0059    0.0160    1.7180    0.0099    0.0003    0.0007    0.0002    0.0002
    0.0164    1.0923    1.0933    1.0115    1.0301    0.0014    0.0015    0.0032    0.0006
    0.0115    0.0076    1.0851    0.4513    0.0106    0.0010    0.0007    0.0030    0.0004
    0.0038    0.0035    0.0089    1.1062    0.0060    0.0002    0.0003    0.0001    0.0001
    0.0058    0.0578    0.0980    0.6569    1.1341    0.0005    0.0008    0.0004    0.0003
```

Now we use the total energy coefficients matrices and commodity final demand vectors to compute the structural decomposition components for the final demand effect `t1`, the production effect `t2`, and the interaction effect `t3`, summing the three for the total `t`.

```
      t1←ei1+.×e2-e1
      t2←(ei2-ei1)+.×e1
      t3←(ei2-ei1)+.×e2-e1
      t←t1+t2+t3

t1
   10467.2    9433.3    9967.9    3738.7    8021.1
t2
   ¯2460.4   ¯1257.5     756.8     613.3    ¯502.4
t3
   ¯1885.4   ¯1718.5     612.5     346.8   ¯1469.1
t
    6121.4    6457.3   11337.2    4698.8    6049.6
```

Chapter 13, Environmental Input–Output Analysis

Chapter 13 reviews the extension of the input–output framework to incorporate activities of environmental pollution generation and elimination associated with economic activities as well as the linkages of input–output to models of ecosystems. The chapter begins with the augmented Leontief model for incorporating pollution generation and elimination, from which many subsequent approaches have been developed.

The chapter then describes the now widespread application of input-output analysis to environmental life cycle assessment and establishing a "pollution footprint" for industrial activity. Environmental input-output is also now widely used to evaluate global environmental issues. The special case of a analyzing the relationship between global climate change and industrial activity with a carbon footprint is then explored along with using input-output to attribute pollution generation to the demands driving consumption compared with the more traditional attribution of pollution generation to the sectors of industrial production necessary to meet that demand.

The exercise problems for this chapter explore the features of environmentally extended input-output models and their applications.

Problem 13.1: Generalized Input-Output Model

This problem explores the basic features of a generalized input-output model configuration applied to assessing energy, pollution, and employment associated with industrial activity.

Problem 13.1 Overview

Assume that we have the following direct coefficient matrices for energy, air pollution, and employment ($\mathbf{D}^e$, $\mathbf{D}^v$ and $\mathbf{D}^l$, respectively) for two industries, 1 and 2: $\mathbf{D}^e = \begin{bmatrix} 0.1 & 0.2 \\ 0.2 & 0.3 \end{bmatrix}$,

$\mathbf{D}^v = \begin{bmatrix} 0.2 & 0.5 \\ 0.2 & 0.3 \end{bmatrix}$ and $\mathbf{D}^l = \begin{bmatrix} 0.2 & 0.5 \end{bmatrix}$. Notice that industry 2 is both a high-polluting and high-employment industry.

Suppose that the local government has an opportunity to spend a total of \$10 million on a regional development project. Two projects are candidates: (1) Project 1 would spend appropriated dollars in the ratio of 60 percent to industry 1 and 40 percent to industry 2; the minimum size of this project is \$4 million; (2) Project 2 would spend appropriated dollars in the ratio of 30 percent to industry 1 and 70 percent to industry 2; the minimum size of this project is \$2 million. The government can adopt either project or a combination of the two projects (as long as the minimum size of each project is at least maintained and that the total budget is not overrun). In other words, we might describe the options available to the government as:

$$\begin{bmatrix} \beta_a \\ \beta_b \end{bmatrix} = \alpha_1 \begin{bmatrix} 0.6 \\ 0.4 \end{bmatrix} + \alpha_2 \begin{bmatrix} 0.3 \\ 0.7 \end{bmatrix}$$

where α_1 and α_2 are budgets allocated to projects 1 and 2, respectively. β_a and β_b are the total final demands presented to the regional economy by the combination of projects for industries *A* and *B*, respectively.

Suppose that four alternative compositions of these projects are being considered (1) $\begin{cases} \alpha_1 = 4 \\ \alpha_2 = 2 \end{cases}$, (2) $\begin{cases} \alpha_1 = 5 \\ \alpha_2 = 5 \end{cases}$, (3) $\begin{cases} \alpha_1 = 10 \\ \alpha_2 = 0 \end{cases}$ and (4) $\begin{cases} \alpha_1 = 0 \\ \alpha_2 = 10 \end{cases}$. The following table of constraints describes the local regulation on energy consumption and environmental pollution in the region:

	Maximum Allowable Changes Collectively by All Industries
Oil Consumption (10^{15} Btus)	3.0
Coal Consumption (10^{15} Btus)	no limit
SO_2 Emissions (tons)	14.5
NO_x Emissions (tons)	10

Finally, suppose that the regional economy is currently described by the following input-output transactions table (in millions of dollars):

	A	*B*	Total Output
A	1	3	10
B	5	1	10

If we are interested in determining which of the proposed combinations of projects (1), (2), (3) or (4) permit the region to operate within the above constraints on energy consumption and air pollution emission and within the established budget constraint, we begin by retrieving the matrix of economic transactions, $\mathbf{Z} = \begin{bmatrix} 1 & 3 \\ 5 & 1 \end{bmatrix}$, and the vector of total outputs, $\mathbf{x} = \begin{bmatrix} 10 \\ 10 \end{bmatrix}$, from the table to calculate the economic direct and total requirements matrices: $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .1 & .3 \\ .5 & .1 \end{bmatrix}$ and

$$\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.364 & .455 \\ .758 & 1.364 \end{bmatrix}.$$

Then we define the direct impact matrix as a concatenation of the three individual impact matrices, $\mathbf{D}^e$, $\mathbf{D}^v$ and $\mathbf{D}^l$, as: $\mathbf{D} = \begin{bmatrix} \mathbf{D}^e \\ \mathbf{D}^v \\ \mathbf{D}^l \end{bmatrix} = \left[\begin{array}{cc} .1 & .2 \\ .2 & .3 \\ \hdashline .2 & .5 \\ .2 & .3 \\ \hdashline .2 & .5 \end{array}\right]$ from which we can compute the total impact coefficients matrix, $\mathbf{DL} = \begin{bmatrix} .288 & .318 \\ .5 & .5 \\ \hdashline .652 & .773 \\ .500 & .500 \\ \hdashline .652 & .773 \end{bmatrix}$. We can represent the four candidate projects by the following matrix, the columns of which are the final demand change vectors:

$$\Delta\mathbf{F} = \overbrace{\begin{bmatrix} 3 & 4.5 & 6 & 3 \\ 3 & 5.5 & 4 & 7 \end{bmatrix}}^{Projects}.$$

Total allocated budgets can be represented by the column sums of $\Delta\mathbf{F}$, found as $\mathbf{i}'[\Delta\mathbf{F}] = \begin{bmatrix} 6 & 10 & 10 & 10 \end{bmatrix}$; that is, all four candidate projects satisfy the budget constraint of \$10 million. We can compute matrix of total impacts as

$$\Delta\mathbf{X} = \mathbf{DL}\Delta\mathbf{F} = \begin{bmatrix} \mathbf{x}^1 & \mathbf{x}^2 & \mathbf{x}^3 & \mathbf{x}^4 \end{bmatrix} = \begin{bmatrix} 1.8 & 3.0 & 3.0 & 3.1 \\ 3.0 & 5.0 & 5.0 & 5.0 \\ 4.3 & 7.2 & 7.0 & 7.4 \\ 3.0 & 5.0 & 5.0 & 5.0 \\ 4.3 & 7.2 & 7.0 & 7.4 \end{bmatrix}$$ where the columns are the vectors of total impacts for each scenario 1, 2, 3, and 4, respectively. Note that Project 4, using 3.1×10^{15} Btus of oil, exceeds the established consumption limit of 3.0×10^{15} Btus.

If our goal is to maximize employment, Project 2 should be chosen since it produces the highest level of employment among the three feasible projects, i.e., from among the first three scenarios that comply with established energy or environmental constraints (from the bottom row of $\Delta\mathbf{X}$).

Computational Notes

For the regional economy specified, we define the interindustry transactions matrix **Z**, the vector of total outputs **x**, the matrix of direct impact coefficients **D**, and the matrix of project costs **C**. We compute the matrix of technical coefficients **A**, the Leontief inverse **L**, and the matrix of total impact coefficients **T**.

```
        D←5 2ρ0.1×1 2 2 3 2 5 2 3 2 5
        Z←2 2ρ1 3 5 1 ⋄ x←10 10 ⋄ C←2 2ρ0.6 0.3 0.4 0.7
        L←LINV A←Z AMAT x
        T←D+.×L

Z
         1          3
         5          1
x
        10         10
A
     0.100      0.300
```

```
      0.500      0.100
L
      1.364      0.455
      0.758      1.364
D
      0.100      0.200
      0.200      0.300
      0.200      0.500
      0.200      0.300
      0.200      0.500
C
      0.600      0.300
      0.400      0.700
T
      0.288      0.318
      0.500      0.500
      0.652      0.773
      0.500      0.500
      0.652      0.773
```

We construct a matrix of the four candidate project combinations with the costs for each of the and compute a matrix `ΔF` the columns of which are the final demand change vectors associated with the four scenarios of combinations of candidate projects, and the column sums of which are the total costs of the scenarios `TC`. Finally using **T** and `ΔF` we can compute total impacts of the scenarios `ΔX`, the columns of which are total impacts of each scenario.

```
        ALPHA←8 2⍴ALPHA1,ALPHA1←4 2⍴4 2 5 5 10 0 0 10
        PC←ALPHA×CC←8 2⍴C
        TC←+⌿ΔF←⍉4 2⍴+/PC
        ΔX←T+.×ΔF
```

```
ΔF
      3.000      4.500      6.000      3.000
      3.000      5.500      4.000      7.000
TC
      6.000     10.000     10.000     10.000
ΔX
      1.818      3.045      3.000      3.091
      3.000      5.000      5.000      5.000
      4.273      7.182      7.000      7.364
      3.000      5.000      5.000      5.000
      4.273      7.182      7.000      7.364
```

Problem 13.2: Generalized Impact Assessment

This problem illustrates construction of a generalized impact assessment model from available data.

Problem 13.2 Overview

Consider a regional economy that has two primary industries, *A* and *B*. In producing these two products it was observed that in the previous year air pollution emissions associated with this industrial activity included 3 pounds of SO_2 and 1 pound of NO_x emitted per dollars' worth of output of industry A, and 5 pounds of SO_2 and 2 pounds of NO_x emitted per dollars' worth of output of industry B.

It was also observed that industries A and B consumed 1×10^6 tons and 6×10^6 tons of coal, respectively, during that year. Industry A also consumed 2×10^6 barrels of oil. Total employment in the region was 100,000 (40 percent of which were employed by industry A and the rest by industry B) and the regional planning agency constructed the following input-output table of interindustry activity and total output in the region (in $\$10^6$):

	A	B	Total Output
A	2	6	20
B	6	12	30

If the projected vector of final demands for the next year is $\mathbf{f}^{new}=\begin{bmatrix}15\\25\end{bmatrix}$, we can estimate for the next year the total consumption of each energy type (coal and oil), the total pollution emission (of each type), and the level of total employment by first by retrieving the matrix of economic transactions, $\mathbf{Z}=\begin{bmatrix}2 & 6\\6 & 12\end{bmatrix}$, and the vector of total outputs, $\mathbf{x}=\begin{bmatrix}20\\30\end{bmatrix}$, from the table to calculate the economic direct and total requirements matrices: $\mathbf{A}=\mathbf{Z}\hat{\mathbf{x}}^{-1}=\begin{bmatrix}.1 & .2\\.3 & .4\end{bmatrix}$ and $\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}=\begin{bmatrix}1.250 & .417\\.625 & 1.875\end{bmatrix}$. From the available data we assemble the direct impact coefficient matrices for energy, emissions, and employment impact and concatenate them to yield an overall matrix of direct impact coefficients, $\mathbf{D}==\left[\begin{array}{cc}1 & 0\\.05 & .2\\\hdashline 3 & 5\\1 & 2\\\hdashline .002 & .002\end{array}\right]$.

Now we can compute the total impacts as $\mathbf{x}^*=\mathbf{DLf}^{new}=\begin{bmatrix}\mathbf{x}^{e*}\\\mathbf{x}^{v*}\\\mathbf{x}^{l*}\end{bmatrix}=\left[\begin{array}{c}29.167\\12.708\\\hdashline 368.75\\141.667\\\hdashline .171\end{array}\right]$. That is, $\mathbf{x}^{e*}=\begin{bmatrix}29.167\\12.708\end{bmatrix}$ shows 29,167,000 tons of coal and 12,708,000 barrels of oil will be consumed in production next year; $\mathbf{x}^{v*}=\begin{bmatrix}368.75\\141.667\end{bmatrix}$ shows that 368,750,000 pounds of SO_2 and 141,667,000 pounds of NO_x will be emitted in the course of that industrial production; and $\mathbf{x}^{l*}=[0.171]$

shows that 171,000 workers will be employed. Total economic output is found as
$\mathbf{x}^{new} = \mathbf{L}\mathbf{f}^{new} = \begin{bmatrix} 29.167 \\ 56.25 \end{bmatrix}$; that is, $x_1^{new} = \$29{,}167{,}000$ and $x_2^{new} = \$56{,}250{,}000$.

Computational Notes

We define the matrix of direct impact coefficients **D**, the matrix of industry transactions **Z**, the vector of total outputs **x**, and the projected new final demand `f2`. The compute the matrix of technical coefficients **A**, the Leontief inverse **L**, and the matrix of total impact coefficients **T**. For the new vector of final demands `f2`, we compute the vector of total outputs `x2` and the vector of total impacts `xs2`.

```
      D←5 2⍴1 0 0.05 0.2 3 5 1 2 0.002 0.002
      Z←2 2⍴2 6 6 12 ⋄ x←20 30 ⋄ f2←15 25⍝
      T←D+.×L←LINV A←Z AMAT x ⍝
      x2←L+.×f2 ⋄ xs2←T+.×f2
```

```
D
     1.000     0.000
     0.050     0.200
     3.000     5.000
     1.000     2.000
     0.002     0.002
Z
         2         6
         6        12
x
        20        30
A
     0.100     0.200
     0.300     0.400


L
     1.250     0.417
     0.625     1.875
T
     1.250     0.417
     0.188     0.396
     6.875    10.625
     2.500     4.167
     0.004     0.005

f2
      15.0      25.0
x2
      29.2      56.3
xs2
      29.2      12.7     368.8     141.7       0.2
```

Problem 13.3: Generalized Input-Output Analysis for Regional Planning

This problem explores typical regional planning consideration in application of a generalized input-output impact model.

Problem 13.3 Overview

Suppose a regional planning agency initiates a regional development planning effort. Four projects are being considered that would represent government purchases of regionally produced products of the output of three industries, A, B, and C, which would appear as final demands presented to the regional economy, as depicted in the following table.

Regional Industry	Project Expenditure (millions of dollars)			
	Project 1	Project 2	Project 3	Project 4
A	2	4	2	2
B	2	0	0	2
C	2	2	4	3

Additional information is available, including the matrix of technical coefficients, $\mathbf{A} = \begin{bmatrix} 0.04 & 0.23 & 0.38 \\ 0.33 & 0.52 & 0.47 \\ 0 & 0 & 0.1 \end{bmatrix}$, and relationships between the following quantities and total output given by the following:

	Industry		
	A	B	C
Pollution emission (grams/$ output)	4.2	7	9.1
Energy Consumption (bbls oil/$ output)	7.6	2.6	0.5
Employment (workers/ $ output)	0.73	0.33	0.63

To determine which of the four projects contributes most to gross regional output, we begin by computing the total economic requirements matrix, $\mathbf{L} = \begin{bmatrix} 1.247 & .598 & .839 \\ .857 & 2.494 & 1.665 \\ 0 & 0 & 1.111 \end{bmatrix}$, and from the table we can assemble the direct impact matrix as $\mathbf{D} = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & .5 \\ .73 & .33 & .63 \end{bmatrix}$. The table of prospective project expenditures retrieved directly from the table is $\Delta\mathbf{F} = \begin{bmatrix} 2 & 4 & 2 & 2 \\ 2 & 0 & 0 & 2 \\ 2 & 2 & 4 & 3 \end{bmatrix}$ from which we can compute $\Delta\mathbf{X}^* = \mathbf{DL}\Delta\mathbf{F} = \begin{bmatrix} 112.986 & 95.527 & 123.618 & 138.27 \\ 67.98 & 69.341 & 68.44 & 79.236 \\ 8.628 & 8.496 & 9.832 & 10.49 \end{bmatrix}$, the corresponding total impacts where each column shows the total impacts of the corresponding column in $\mathbf{\Delta F}$ for each project. Since the sum of final demands equals the contribution to gross regional product (GRP), we can also note that Project 4 contributes most to GRP, i.e., that project shows the largest column sum of $\mathbf{\Delta F}$, $\mathbf{i}'[\Delta\mathbf{F}] = \begin{bmatrix} 6 & 6 & 6 & 7 \end{bmatrix}$. Project 4 also consumes the most energy (79.236 × 10^6 bbls of oil) and contributes the most to regional employment (10.490 × 10^6 workers).

Computational Notes

We define a matrix `ΔY` that includes the list of project expenditures for each project as its columns, the matrix of technical coefficients **A**, and the matrix of direct impact coefficients **D**. We then compute the Leontief inverse **L**, and the matrix of total impact coefficients **T**. We compute the vector of contributions to gross regional product for the three projects `GRP` which is simply a vector comprising the column sums of `ΔY`. Finally we compute the matrix `XS`, the columns of which are total impacts of the each of the three projects.

```
      ΔY←3 4⍴2 4 2 2 2 0 0 2 2 2 4 3
      A←3 3⍴0.04 0.23 0.38 0.33 0.52 0.47 0 0 0.1
      D←3 3⍴4.2 7 9.1 7.6 2.6 0.5 0.73 0.33 0.63
      T←D+.×L←LINV A
      GRP←+⌿ΔY
      XS←T+.×ΔY
```

```
A
     0.040     0.230     0.380
     0.330     0.520     0.470
     0.000     0.000     0.100
L
     1.247     0.598     0.839
     0.857     2.494     1.665
     0.000     0.000     1.111
D
     4.200     7.000     9.100
     7.600     2.600     0.500
     0.730     0.330     0.630
ΔY
         2         4         2         2
         2         0         0         2
         2         2         4         3
GRP
         6         6         6         7
T
    11.239    19.969    25.285
    11.707    11.026    11.257
     1.193     1.259     1.861
XS
     113.0      95.5     123.6     138.3
      68.0      69.3      68.4      79.2
       8.6       8.5       9.8      10.5
```

Problem 13.4: Illustrating Environmental-Employment Tradeoffs

This problem explores the potential tradeoffs between environmental and employment considerations using input-output analysis.

Problem 13.4 Overview

Consider an input-output economy defined by interindustry transactions and total outputs,
$\mathbf{Z} = \begin{bmatrix} 140 & 350 \\ 800 & 50 \end{bmatrix}$ and $\mathbf{x} = \begin{bmatrix} 1,000 \\ 1,000 \end{bmatrix}$.

Suppose this is an economy in deep economic trouble. The federal government has at its disposal policy tools that can be implemented to stimulate demand for goods from one sector or the other. Also suppose that the plants in sector 1 discharge 0.3 lbs. of airborne particulate substances for every dollar of output (0.3 lbs/$ output), while sector 2 pollutes at 0.5 lbs/$ output. Finally, let labor input coefficients be 0.005 and 0.07 for sectors 1 and 2, respectively.

To assess whether or not a conflict of interest would arise between unions and environmentalists in determining the sector toward which the government should direct its policy effort, first, from **Z** and **x** we compute $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .14 & .35 \\ .8 & .05 \end{bmatrix}$ and $\mathbf{L} = \begin{bmatrix} 1.769 & .652 \\ 1.49 & 1.601 \end{bmatrix}$. Since from the data provided, $\mathbf{D} = \begin{bmatrix} .3 & .5 \\ .005 & .07 \end{bmatrix}$ we can compute $\mathbf{D}(\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 1.276 & .996 \\ .113 & .115 \end{bmatrix}$.
Therefore, for $\mathbf{f}^{new} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{x}^* = \begin{bmatrix} 1.276 \\ \hdashline .113 \end{bmatrix}$, meaning that for each new dollar's worth of final demand for the output of sector 1, there will be 1.276 pounds of pollutant emitted and 0.113 new workers. Similarly, with $\mathbf{f}^{new} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, we find $\mathbf{x}^* = \begin{bmatrix} .996 \\ \hdashline .115 \end{bmatrix}$, meaning that for each new dollar's worth of final demand for the output of sector 2, there will be 0.996 pounds of pollutant emitted and 0.115 new workers. Thus, there would not be a conflict between unions and environmentalists in this case; each dollar's worth of new demand for sector 2 generates less pollution and also generates more employment (notice that this is true despite the fact that sector 2's direct-pollution coefficient per dollar of output is larger than sector 1's direct-pollution coefficient).

Computational Notes

We define the transactions matrix **Z**, the vector of total outputs **x**, the matrix of direct impact coefficients **D**, and the vectors of increases in final demand growth for sectors 1 or 2, respectively as `f1` and `f2`. Now we compute the matrix of technical coefficients **A**, the Leontief inverse **L**, and the matrix of total impact coefficients **T** from which we can compute the vectors of total impacts, `xs1` and `xs2`, resulting from the changes in final demand `f1` and `f2`, respectively.

```
      Z←2 2⍴140 350 800 50 ⋄ x←1000 1000
      D←2 2⍴0.3 0.5 0.005 0.07 ⋄ f1←1 0 ⋄ f2←0 1
      T←D+.×L←LINV A←Z AMAT x
      xs1←T+.×f1 ⋄ xs2←T+.×f2

Z
      140       350
      800        50
x
     1000      1000
A
    0.140     0.350
    0.800     0.050
L
    1.769     0.652
    1.490     1.601
D
    0.300     0.500
    0.005     0.070
T
    1.276     0.996
    0.113     0.115
f1
        1         0
xs1
    1.276     0.113
f2
        0         1
xs2
    0.996     0.115
```

Problem 13.5: The Augmented Leontief Input-Output Model

This problem explores the basic features of the pollution-activity augmented Leontief input-output formulation.

Problem 13.5 Overview

Consider the following table of interindustry transactions and total industry outputs (the same transactions as in Problem 13.4 but with different total outputs):

		Purchasing Sector		Total Output
		1	2	
Selling Sector	1	140	350	2,000
	2	800	50	1,850

An amount of pollution generated by sector 1 is 10 units and by sector 2 is 25 units. Pollution abatement reduced pollution by 5 units in sector 1 and 12 units in sector 2. Total pollution permitted by local regulation is 12 units. if final demands for both sectors increase by 100, we can use the pollution-activity-augmented Leontief formulation to determine is the level of output for each industry by first augmenting the basic economic transactions matrix,

$\mathbf{Z} = \begin{bmatrix} 140 & 350 \\ 800 & 50 \end{bmatrix}$ with the pollution abatement and elimination data to yield

$$\overline{\mathbf{Z}} = \begin{bmatrix} 140 & 350 & 5 \\ 800 & 50 & 12 \\ 10 & 25 & 0 \end{bmatrix}.$$

Total pollution output is found by adding pollution generation in the interindustry matrix to the pollution tolerated (reflected as a negative value in final demand), i.e., we define $x_p = 10 + 25 - 12 = 23$, which we can augment to the total industry outputs vector to yield $\overline{\mathbf{x}} = \begin{bmatrix} 2,000 & 1,850 & 23 \end{bmatrix}'$. The vector of final demands is $\overline{\mathbf{f}} = \overline{\mathbf{x}} - \overline{\mathbf{Z}}\mathbf{i} = \begin{bmatrix} 1,505 & 988 & -12 \end{bmatrix}'$ and the matrix of technical coefficients is $\overline{\mathbf{A}} = \overline{\mathbf{Z}}\hat{\overline{\mathbf{x}}}^{-1} = \begin{bmatrix} .07 & .189 & .217 \\ .4 & .027 & .522 \\ .005 & .014 & 0 \end{bmatrix}$, from which we can compute $\overline{\mathbf{L}} = (\mathbf{I} - \overline{\mathbf{A}})^{-1} = \begin{bmatrix} 1.178 & .234 & .378 \\ .491 & 1.133 & .698 \\ .013 & .016 & 1.011 \end{bmatrix}$.

For an increase in final demand of both sectors by 100, $\Delta\overline{\mathbf{f}} = \begin{bmatrix} 100 & 100 & 0 \end{bmatrix}'$, the changes in total outputs and pollution are found as $\Delta\overline{\mathbf{x}} = \begin{bmatrix} 141.2 & 162.4 & 2.9 \end{bmatrix}'$. Hence, the new levels of outputs and pollution are $\overline{\mathbf{x}}^{new} = \begin{bmatrix} 2,141.2 & 2,012.4 & 25.9 \end{bmatrix}'$.

Computational Notes

We define the transactions matrix **Z**, the vector of total outputs **x**, and the vector of specified new final demands `Δf`. We the compute the vector of current final demands **f**, the matrix of technical coefficients **A**, and the Leontief inverse **L**, from which we can compute the vector of the changes in total outputs `Δx` and the vector of new total outputs `x2`.

```
      Z←3 3⍴140 350 5 800 50 12 10 25 0
      x←2000 1850 23 ⋄ Δf←100 100 0
      f←x-+/Z
      L←INV A←Z AMAT x
      Δx←L+.×Δf ⋄ x2←x+Δx

Z
      140       350         5
      800        50        12
       10        25         0
f
     1505       988       ¯12
x
     2000      1850        23
A
    0.070     0.189     0.217
```

```
     0.400     0.027     0.522
     0.005     0.014     0.000
L
     1.178     0.234     0.378
     0.491     1.133     0.698
     0.013     0.016     1.011


Δf
       100       100         0
Δx
     141.2     162.4       2.9
x2
    2141.2    2012.4      25.9
```

Problem 13.6: Regional and National Impacts of Public Works Initiatives

This problem compares regional and national pollution, energy consumption, and employment impacts of a public works initiative.

Problem 13.6 Overview

In Problems 10.5 and 10.6 national and regional input-output tables are defined with three sectors (natural resources, manufacturing, and services) with the following matrices of technical coefficients and vectors of total outputs, respectively,

$$\mathbf{A}^N = \begin{bmatrix} .1830 & .0668 & .0087 \\ .1377 & .3070 & .0707 \\ .1603 & .2409 & .2999 \end{bmatrix}, \ \mathbf{x}^N = \begin{bmatrix} 518,288.6 \\ 4,953,700.6 \\ 14,260,843.0 \end{bmatrix}, \ \mathbf{A}^R = \begin{bmatrix} .1092 & .0324 & .0036 \\ .0899 & .0849 & .0412 \\ .1603 & .1170 & .2349 \end{bmatrix} \text{ and}$$

$$\mathbf{x}^R = \begin{bmatrix} 8,262.7 \\ 95,450.8 \\ 170,690.3 \end{bmatrix}$$. We define the energy use, pollution, and employment coefficients that apply to both the regional and national economies in the following table:

Environmental, Energy, and Employment Impact Coefficients

	Industry		
	Nat. Res.	Manuf.	Services
Pollution emission (grams/$ output)	4.2	7	9.1
Energy Consumption (bbls oil/$ output)	7.6	2.6	0.5
Employment (workers/ $ output)	7.3	3.3	6.3

Suppose a major new public works initiative by the federal government is characterized by the following vector of increases in federal spending: $\Delta\mathbf{f}' = \begin{bmatrix} 250 & 3,000 & 7,000 \end{bmatrix}$, of which 20 percent will be spent in the region (assume the 20 percent applies linearly to all expenditures). We can determine the percentage changes in total impacts on pollution, energy use, employment, and total industrial output of each industry sector for the region compared with those of the nation as a whole by first defining, from the table, the common direct impact coefficients as

$$\mathbf{D} = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & .5 \\ 7.3 & 3.3 & 6.3 \end{bmatrix}.$$

The baseline environmental, energy, and employment impacts for the nation and the region, respectively, are found by $\mathbf{x}^{*N} = \mathbf{D}\mathbf{x}^N = [166{,}626{,}387.6 \quad 23{,}949{,}036.4 \quad 109{,}974{,}029.7\,]'$ and $\mathbf{x}^{*R} = \mathbf{D}\mathbf{x}^R = [2{,}256{,}140.7 \quad 396{,}313.8 \quad 1{,}450{,}654.2\,]'$. Then the total impacts, including the economic impacts of the new public works project for the nation and the region, respectively, are

$$\Delta\overline{\mathbf{x}}^N = \begin{bmatrix} \mathbf{D}^{*N}(\mathbf{I}-\mathbf{A}^N)^{-1} \\ \hdashline (\mathbf{I}-\mathbf{A}^N)^{-1} \end{bmatrix} \Delta\mathbf{f}^N = \begin{bmatrix} 10.7832 & 16.2768 & 14.7759 \\ 10.4543 & 5.2368 & 1.3729 \\ 12.5176 & 9.4836 & 10.1120 \\ \hdashline 1.2516 & 0.1306 & 0.0287 \\ 0.2881 & 1.5256 & 0.1576 \\ 0.3857 & 0.5549 & 1.4892 \end{bmatrix} \begin{bmatrix} 250 \\ 3{,}000 \\ 7{,}000 \end{bmatrix} = \begin{bmatrix} 154{,}957.3 \\ 27{,}934.5 \\ 102{,}364.1 \\ \hdashline 906.1 \\ 5{,}752.2 \\ 12{,}185.3 \end{bmatrix}$$

$$\Delta\overline{\mathbf{x}}^R = \begin{bmatrix} \mathbf{D}^{*R}(\mathbf{I}-\mathbf{A}^R)^{-1} \\ \hdashline (\mathbf{I}-\mathbf{A}^R)^{-1} \end{bmatrix} \Delta\mathbf{f}^R = \begin{bmatrix} 7.9150 & 9.5207 & 12.4438 \\ 9.0188 & 3.2720 & 0.8721 \\ 10.2453 & 5.0627 & 8.5550 \\ \hdashline 1.1281 & 0.0409 & 0.0075 \\ 0.1223 & 1.1048 & 0.0601 \\ 0.2550 & 0.1775 & 1.3178 \end{bmatrix} \begin{bmatrix} 50 \\ 600 \\ 1{,}400 \end{bmatrix} = \begin{bmatrix} 23{,}529.5 \\ 3{,}635.2 \\ 15{,}527.0 \\ \hdashline 91.5 \\ 753.1 \\ 1{,}964.2 \end{bmatrix}$$

where $\Delta\mathbf{f}^R = .2\Delta\mathbf{f}^N$. Hence, the comparative percentage changes from $\mathbf{x}^{*(N)}$ and $\mathbf{x}^{*(R)}$ are:

	Nation	Region
Nat. Res.	0.09	1.04
Manuf.	0.12	0.92
Services	0.09	1.07
Pollution	0.17	1.11
Energy	0.12	0.79
Employ.	0.09	1.15

Computational Notes

We define the matrices of technical coefficients for the national and regional economies, `AN` and `AR`, respectively along with the corresponding vectors of total outputs, `xn` and `xr`. In addition, we define the matrix of direct impact coefficients **D** and the vector of planned new federal expenditures `Δf`. We can compute the vector of total impacts of current industrial activity as `xsn` for the national economy and `xsr` for the regional economy.

```
        AN←3 3ρ0.183 0.0668 0.0087 0.1377 0.307 0.0707 0.1603 0.2409 0.2999
        xn←518288.6 4953700.6 14260843
        AR←3 3ρ0.1092 0.0324 0.0036 0.0899 0.0849 0.0412 0.1603 0.117 0.2349
        xr←8262.7 95450.8 170690.3
        D←3 3ρ4.2 7 9.1 7.6 2.6 0.5 7.3 3.3 6.3
        Δf←250 3000 7000 ⋄
        xsn←(D+.×xn),xn ⋄ xsr←(D+.×xr),xr
```

```
AN
    0.1830    0.0668    0.0087
    0.1377    0.3070    0.0707
    0.1603    0.2409    0.2999
xn
  518288.6 4953700.614260843.0
AR
    0.1092    0.0324    0.0036
    0.0899    0.0849    0.0412
    0.1603    0.1170    0.2349
xr
    8262.7   95450.8  170690.3
D
     4.200     7.000     9.100
     7.600     2.600     0.500
     7.300     3.300     6.300
Δf
     250.0    3000.0    7000.0
xsn
 166626387.6  23949036.4 109974029.7    518288.6   4953700.6  14260843.0
xsr
   2256140.7    396313.8   1450654.2      8262.7     95450.8    170690.
```

We compute the vector of planned new federal expenditures for the regional economy `Δfr`, the matrices of technical requirements for the nation and the region, `AN` and `AR`, respectively and the corresponding Leontief inverses, `LN` and `LR`, along with the corresponding matrices of total impact coefficients, `TN` and `TR`. We then compute the vectors of total national impacts `Δxsn` and total regional impacts `Δxsr`, respectively, and the percentage changes from current levels in the matrix `PC`.

```
LN
    1.2516    0.1306    0.0287
    0.2881    1.5256    0.1576
    0.3857    0.5549    1.4892
LR
    1.1281    0.0409    0.0075
    0.1223    1.1048    0.0601
    0.2550    0.1775    1.3178
TN
   10.7832   16.2768   14.7759
   10.4543    5.2368    1.3729
```

```
   12.5176    9.4836   10.1120
    1.2516    0.1306    0.0287
    0.2881    1.5256    0.1576
    0.3857    0.5549    1.4892
TR
    7.9150    9.5207   12.4438
    9.0188    3.2720    0.8721
   10.2453    5.0627    8.5550
    1.1281    0.0409    0.0075
    0.1223    1.1048    0.0601
    0.2550    0.1775    1.3178
Δxsn
  154957.3   27934.5  102364.1      906.1     5752.2    12185.3
Δfr
      50.0     600.0    1400.0
Δxsr
   23529.5    3635.2   15527.0       91.5      753.1     1964.2

PC
      0.09      1.04
      0.12      0.92
      0.09      1.07
      0.17      1.11
      0.12      0.79
      0.09      1.15
```

Problem 13.7: Analyzing Implications of an Energy Shortage

This problem explores the implications of an energy shortage on economic performance using the regional economy specified in Problem 13.6 prior to the projected final demand for that problem.

Problem 13.7 Overview

Recall for the economy specified in Problem 13.6, the matrix of technical coefficients and vector of total outputs, respectively, were $\mathbf{A}^R = \begin{bmatrix} .1092 & .0324 & .0036 \\ .0899 & .0849 & .0412 \\ .1603 & .1170 & .2349 \end{bmatrix}$ and $\mathbf{x}^R = \begin{bmatrix} 8,262.7 \\ 95,450.8 \\ 170,690.3 \end{bmatrix}$. The matrix of direct impact coefficients was specified as $\mathbf{D} = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & .5 \\ 7.3 & 3.3 & 6.3 \end{bmatrix}$.

The levels of pollution, energy consumption, and employment accompanying the baseline levels of total industry output are found by

$$\mathbf{x}^{*R} = \mathbf{D}\mathbf{x}^R = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & .5 \\ 7.3 & 3.3 & 6.3 \end{bmatrix} \begin{bmatrix} 8,262.7 \\ 95,450.8 \\ 170,690.3 \end{bmatrix} = \begin{bmatrix} 2,256,140.7 \\ 396,313.8 \\ 1,450,654.2 \end{bmatrix}.$$

We can show a ten percent reduction in energy availability defined by

$$\mathbf{x}^{*new} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & .9 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2{,}256{,}140.7 \\ 396{,}313.8 \\ 1{,}450{,}654.2 \end{bmatrix} = \begin{bmatrix} 2{,}256{,}140.7 \\ 356{,}682.4 \\ 1{,}450{,}654.2 \end{bmatrix}.$$

The corresponding limits on total industry output can be found conveniently as

$$\mathbf{x}^{new} = \mathbf{D}^{-1}\mathbf{x}^{*new} = \begin{bmatrix} -0.0766 & 0.0732 & 0.1049 \\ 0.2301 & 0.2079 & -0.3488 \\ -0.0317 & -0.1937 & 0.2199 \end{bmatrix} \begin{bmatrix} 2{,}256{,}140.7 \\ 356{,}682.4 \\ 1{,}450{,}654.2 \end{bmatrix} = \begin{bmatrix} 5{,}362.0 \\ 87{,}210.5 \\ 178{,}367.8 \end{bmatrix}$$, which in turn means

$$\mathbf{f}^{new} = (\mathbf{I}-\mathbf{A})\mathbf{x}^{new} = \begin{bmatrix} 0.8908 & -0.0324 & -0.0036 \\ -0.0899 & 0.9151 & -0.0412 \\ -0.1603 & -0.1170 & 0.7651 \end{bmatrix} \begin{bmatrix} 5{,}362.0 \\ 87{,}210.5 \\ 178{,}367.8 \end{bmatrix} = \begin{bmatrix} 1{,}308.7 \\ 71{,}975.5 \\ 125{,}406.0 \end{bmatrix}.$$

Finally, the change in GDP is $\mathbf{i}'\mathbf{f}^{new} - \mathbf{i}'\mathbf{f} = -2{,}637.7$ or a 1.31 percent reduction in GDP, where the original final demand is $\mathbf{f} = \mathbf{x} - \mathbf{A}\mathbf{x} = \begin{bmatrix} 3{,}653.3 \\ 79{,}571.8 \\ 118{,}102.9 \end{bmatrix}$.

Computational Notes

We define the matrix of technical coefficients **A**, the vector of total outputs x, and the matrix of direct impact coefficients **D**. We then compute the matrix inverse of **D**, which we call `DI`, the matrix found by subtracting `A` from a suitably size identity matrix, which we call `IA`. Finally, we can compute the vector of current final demands **f**, and the vector of current total impacts **x**, and the vector of direct impacts `xs`.

```
A←3 3ρ0.1092 0.0324 0.0036 0.0899 0.0849 0.0412 0.1603 0.117 0.2349
x←8262.7 95450.8 170690.3 ⋄ D←3 3ρ4.2 7 9.1 7.6 2.6 0.5 7.3 3.3 6.3
DI←⌹D ⋄ I←3 3ρ1,3ρ0 ⋄ IA←I-A ⋄ f←x-A+.×x ⋄ xs←D+.×x
```

```
A
    0.1092     0.0324     0.0036
    0.0899     0.0849     0.0412
    0.1603     0.1170     0.2349
D
    4.2000     7.0000     9.1000
    7.6000     2.6000     0.5000
    7.3000     3.3000     6.3000
DI
   ¯0.0766     0.0732     0.1049
    0.2301     0.2079    ¯0.3488
   ¯0.0317    ¯0.1937     0.2199
IA
    0.8908    ¯0.0324    ¯0.0036
   ¯0.0899     0.9151    ¯0.0412
```

```
   ¯0.1603   ¯0.1170    0.7651
f
    3653.3   79571.8  118102.9
x
    8262.7   95450.8  170690.3
xs
 2256140.7  396313.8 1450654.2
```

We compute the vector of total impacts reflecting at shortfall in energy availability `xs2` and the corresponding vector of total outputs `x2` and using `IA` we can then compute the corresponding vector of final demands `f2`. Knowing `f2` we can compute the GNP change `ΔGNP`, also expressed in percentage terms `PCGNP`.

```
       x2←DI+.×xs2←1 0.9 1×xs
       f2←IA+.×x2
       ΔGNP←(+/f2)-+/f
       PCGNP←100×((+/f2)-+/f)÷+/f

xs2
 2256140.7  356682.4 1450654.2
x2
    5362.0   87210.5  178367.8
f2
    1308.7   71975.5  125406.0
ΔGNP
  ¯2637.70
PCGNP
     ¯1.31
```

Problem 13.8: Input-Output and Linear Programming

This problem explores input-output analysis configured as a linear programming (LP) problem using alternative objective functions.

Problem 13.8 Overview

Consider a traditional input-output economy is specified by the technical requirements matrix and vector of final demands, respectively, as $\mathbf{A} = \begin{bmatrix} .3 & .1 \\ .2 & .5 \end{bmatrix}$ and $\mathbf{f} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$, for which the vector of total outputs is found by $\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{f} = \begin{bmatrix} 7.575 \\ 10.03 \end{bmatrix}$. The vector of value-added coefficients for this economy are found as $\mathbf{v} = \mathbf{x} - \mathbf{i}'\mathbf{A} = \begin{bmatrix} .5 \\ .4 \end{bmatrix}$.

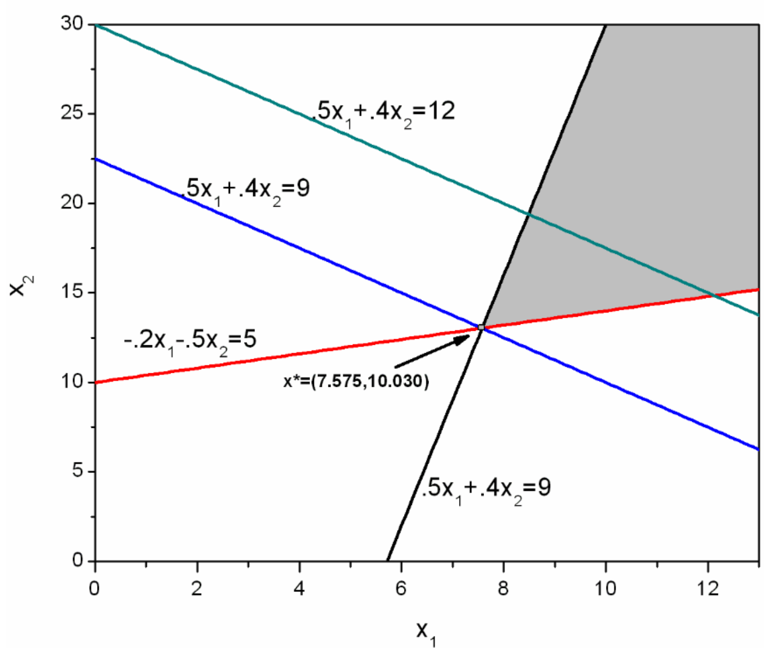
To specify this model as an equivalent LP formulation, we can write

$$\begin{gathered} Min\ .5x_1 + .4x_2 \\ .7x_1 - .1x_2 \geq 4 \\ -.2x_1 + .5x_2 \geq 5 \\ x_1, x_2 \geq 0 \end{gathered}$$

which we can interpret as minimizing the gross domestic product (the sum of all value added) subject to total industry production that at least satisfies all industry final demands. In matrix terms this is expressed as

$$\begin{gathered} Min\ \mathbf{v}'\mathbf{x} \\ (\mathbf{I} - \mathbf{A})\mathbf{x} \geq \mathbf{f} \\ \mathbf{x} \geq 0 \end{gathered}$$

The graphical solution is



Note that it turns out that the solution to this LP problem has a "dual" formulation (discussed in Chapter 13) of maximizing the value of total final demand (or maximizing gross domestic product) subject to the technical coefficients and supply availability of value-added factors, which may seem more intuitive. These dual LP problems have the same result such that maximized value of final demand equals the minimized cost of value-added factors and that value is the gross domestic product, or the familiar equality of national product to national income.

Now, suppose that for this economy pollution is generated at a rate of 2.5 units per dollar of output of industry 1 and 2 units per dollar of output of industry 2 for this economy. If we replace the objective function for the LP problem with minimizing pollution emissions instead of maximizing GDP, we can express this as

$$Min\ 2.5x_1 + 2x_2$$
$$.7x_1 - .1x_2 \geq 4$$
$$-.2x_1 + .5x_2 \geq 5$$
$$x_1, x_2 \geq 0$$

Note that the solution to this LP problem is the same as with the original objective function, which should not be surprising, in this particular case, since the minimum production levels defined in the constraint equations are already determined uniquely by the input-output relationships $(\mathbf{I} - \mathbf{A}) \geq \mathbf{f}$.

Computational Notes

Methods for solving LP problems computationally are beyond the scope of this text, but for convenience we use the monadic APL function `LINPROG` (listed in the appendix for this volume) for this purpose in this Workbook. The function argument is a matrix **D** with partitions defined by the following:

```
⍝------C,u,0
⍝------A,S,b
⍝-----C is row vector of objective function coefficients (nm←⍴C)
⍝-----u is scalar u=1 for maximization u=¯1 for minimization
⍝-----A is matrix of constraint equation coefficients (nn×nm)
⍝-----S is col vector of signs: ¯1 for ≤; 0 for =; 1 for ≥' for
⍝-----   constraint equations;(nn←⍴S=⍴b) the no. of const eqns
⍝-----b is col vector of rhs terms of constraint equations (nn)
```

The explicit result of applying `LINPROG` is a vector including (1) a scalar code indicating solution status (see the documentation of the function), (2) the final value of the objective function, and (3) the final values of the decision variables.

For this problem, we define the matrix in the format needed to use LINPROG specifying the first LP problem as `D1` and for the second problem `D2`, with the corresponding results `r1` and `r2`, respectively.

```
      r1←1↓LINPROG D1←3 4⍴0.5 0.4 ¯1 0 0.7 ¯0.1 1 4 ¯0.2 0.5 1 5
      r2←1↓LINPROG D2←3 4⍴2.5 2 ¯1 0 0.7 ¯0.1 1 4 ¯0.2 0.5 1 5

D1
 0.5  0.4 ¯1 0
 0.7 ¯0.1  1 4
¯0.2  0.5  1 5
r1
9.000005 7.5757576 13.030303
D2
 2.5  2   ¯1 0
 0.7 ¯0.1  1 4
¯0.2  0.5  1 5
r2
45.000005 7.5757576 13.030303
```

Problem 13.9: Goal Programming

This problem expands the LP formulation for the economy specified in Problem 13.8 to a goal programming (GP) formulation which can accommodate multiple objective functions.

Problem 13.9 Overview

Suppose that, in addition to the environmental criteria specified in Problem 13.8, we also know that employment is generated at a rate of 6 and 3 units per dollars' worth of output for industries 1 and 2, respectively, and that there is high priority employment target of 7.5 units for industry 2.

To formulate this situation as finding the vector of total outputs that meets the employment target for industry 2 as the highest priority, then meets final demand requirements for both industries as the next highest priority, and minimizes total pollution generation to the extent possible as the next priority, and, if possible, limiting pollution to a total level of 10 units between the two industries, we specify the following GP problem:

$$Min\ \ P_1(d_3^-) + P_2(d_1^- + d_2^-) + P_3(d_4^+)$$

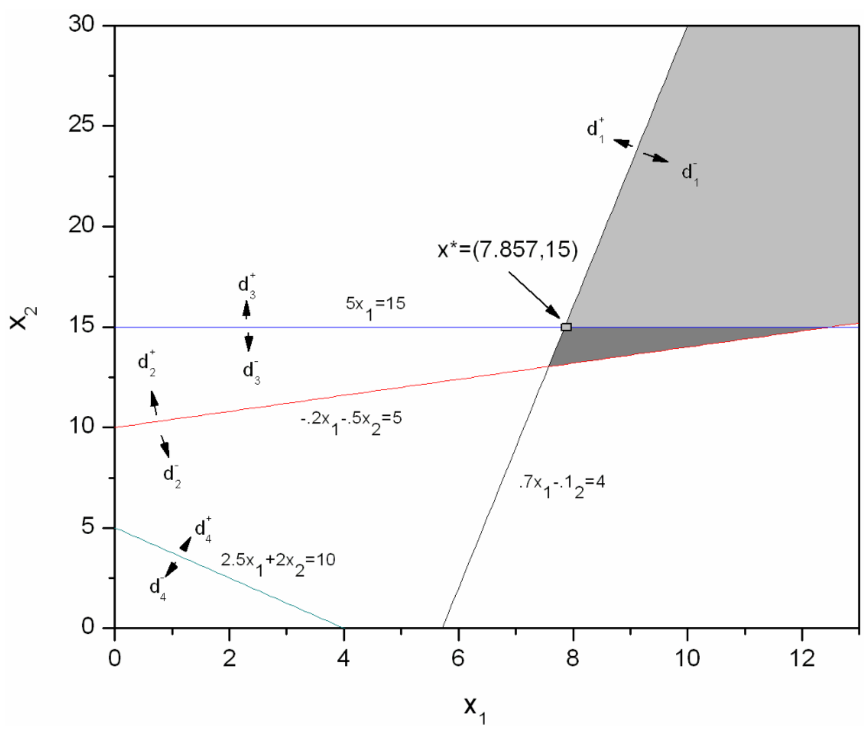
$$.7x_1 - .1x_2 + d_1^+ + d_1^- = 4$$

$$-.2x_1 + .5x_2 + d_2^+ + d_2^- = 5$$

$$.5x_2 + d_3^+ + d_3^- = 7.5$$

$$2.5x_1 + 2x_2 + d_4^+ + d_4^- = 10$$

The graphical solution to this GP problem is shown below with the preferred vector of total outputs computed as $\mathbf{x}^* = (7.857, 15)$.



Computational Notes

Computational methods for solving goal programming problems are beyond the scope of this textbook, but one useful feature of linear goal programming problems is that they can be formulated as a linear programming problem with different order magnitude weights on variables in pre-emptive priority levels, in which we can use the function `LINPROG` developed earlier.

For this problem, we define the goal programming problem in the format for `LINPROG` by specifying the weight for the first priority goal as 1,000, for the second priority goal as 100, and the third priority goal as 10 in the objective function specified in the first row of the matrix `D` (see the earlier description of LINPROG for additional details), which for this GP problem we define as `DGP`. Except for the first row, the first two columns of `DGP` are the constraint equation coefficients, the next four columns are initial coefficients for the positive deviational variables (all with the value -1), the next four columns are the initial coefficients for the negative deviational variables (all with a value of 1), the next to last column specifying equality constraints (value of 0), and the last column includes the right-hand-side values of the constraint equations. `DGP` is then used as the right argument in applying `LINPROG` and the second and third elements of the explicit result are the values of the x_1 and x_2. which are extracted in the vector `xs`.

```
      I←4 4ρ1,4ρ0
      D←(⍉2 4ρ0.7 ¯0.2 0 2.5 ¯0.1 0.5 0.5 2)
      D←D,(-I),I,⍉2 4ρ(4ρ0),4 5 7.5 10
      DGP←(1 12ρ0 0 0 0 0 10 100 100 1000 0 ¯1 0),[1]D
      xs←2↑2↓r←LINPROG DGP

      DGP
 0    0    0  0  0 10 100 100 1000 0 ¯1  0
 0.7 ¯0.1 ¯1  0  0  0   1   0    0 0  0  4
¯0.2  0.5  0 ¯1  0  0   0   1    0 0  0  5
 0    0.5  0  0 ¯1  0   0   0    1 0  0  7.5
 2.5  2    0  0  0 ¯1   0   0    0 1  0 10
      xs
7.8571429 15
```

We can streamline the formulation of the GP problem with the dyadic APL Function `GOALPROG` which takes a left argument that specifies the goal programming objective function and a right argument that specifies the constraint matrix.

```
 r←P GOALPROG C;np;nv;nd;I;k;OBJ;W;i;j;obj;R
⍝GP with pre-emptive priority levels
⍝P--row 1 specifies pre-emptive level (weight)
⍝---row 2 neg dev var (minus#) or pos dev var(Pos#)
⍝C--const and obj fn equations, last col is RHS
⍝P is 2×np ; C is nv×(nd+1)
 np←¯1↑ρP ⋄ nv←1↑ρC ⋄ nd←¯1+¯1↑ρC ⋄ W←⊖10*⍳5
 I←(2ρnv)ρ1,nvρ0 ⋄ OBJ←(np,(2×nv))ρ0 ⋄ k←1
L1:i←P[1;k] ⋄ j←(|P[2;k])+nv×P[2;k]<0
 OBJ[i;j]←W[P[1;k]]
 →(np≥k←k+1)/L1
```

```
[ 11]  R←C[;⍳nd],(-I),I,((nv,1)⍴0),C[;nd+1]
[ 12]  obj←(nd⍴0),(+⌿OBJ),¯1 0
[ 13]  R←((1,⍴obj)⍴obj),[1]R
[ 14]  r←LINPROG R
```

The left argument **P** is a two-row matrix. An element in the first row specifies the preemptive priority level for a deviational variable specified in the same column in the second row. A negative number in the second row indicates a negative deviational variable, e.g., ¯3 would indicate d_3^- , and a positive number indicates a positive deviational variable, e.g., 2 would indicate d_2^+. Hence, a goal programming objective function specified as $Min\ \ P_1(d_3^-)+P_2(d_1^-+d_2^-)+P_3(d_4^+)$ would be specified in the left argument of GOALPROG as

```
      P←2 4⍴1 2 2 3 ¯3 ¯1 ¯2 4
      P
 1  2  2 3
¯3 ¯1 ¯2 4
```

The right argument to `GOALPROG` is the matrix of constraint equations **C** with all but the last column specifying the constraint equation coefficients and the last column specifying the right-hand-side values of the constraint equations. Hence, a set of goal programming constraint equations specified as

$$.7x_1-.1x_2+d_1^++d_1^-=4$$

$$-.2x_1+.5x_2+d_2^++d_2^-=5$$

$$.5x_2+d_3^++d_3^-=7.5$$

$$2.5x_1+2x_2+d_4^++d_4^-=10$$

would be specified in the right argument of `GOALPROG` as

```
      C←4 3⍴.7 ¯.1 4 ¯.2 .5 5 0 .5 7.5 2.5 2 10
      C
 0.7 ¯0.1  4
¯0.2  0.5  5
 0    0.5  7.5
 2.5  2   10
```

The function `GOALPROG` converts these left and right arguments into a tableau that is then solved with the function `LINPROG` defined earlier.

```
 0     0    0  0  0 1000 10000 10000 100000 0 ¯1  0
 0.7 ¯0.1 ¯1  0  0    0     1     0      0 0  0  4
¯0.2  0.5  0 ¯1  0    0     0     1      0 0  0  5
 0    0.5  0  0 ¯1    0     0     0      1 0  0  7.5
 2.5  2    0  0  0   ¯1     0     0      0 1  0 10

      P GOALPROG C
0 39642.85714 7.857142857 15 0 0.9285714286 0 39.64285714 0 0 0 0
```

Recall the format of the result: the first element, which is reported as 0, indicates an optimal LP solution was reached; the second element is the final value of the LP objective function (not important for the GP solution); and the next two values are the values of the decision variables for the GP solution, so the solution is more precisely specified as

```
      2↑2↓ P GOALPROG C
7.857142857 15
```

Problem 13.10: Estimating Changes in Carbon Dioxide Emissions

This problem illustrates the estimation of the change in U.S. carbon dioxide emissions between two reference years.

Problem 13.10 Overview

We use highly aggregated (seven industry sectors) versions of the 1997 and 2007 U.S. input-output tables (industry-by-industry and assume industry-based technology, after redefinitions) provided in text's Appendix SD1.

First, we retrieve the supply and use matrices and, recalling that the supply matrix is the transpose of the use matrix, we can specify:

$$\mathbf{V}^{1997} = \begin{bmatrix} 258{,}234 & 0 & 0 & 45 & 0 & 796 & 0 \\ 0 & 162{,}842 & 0 & 9{,}119 & 0 & 0 & 0 \\ 0 & 0 & 762{,}267 & 0 & 0 & 0 & 0 \\ 0 & 1{,}007 & 0 & 3{,}752{,}428 & 0 & 28{,}347 & 3{,}789 \\ 126 & 381 & 0 & 0 & 2{,}262{,}980 & 259 & 742 \\ 0 & 240 & 0 & 0 & 11 & 6{,}577{,}434 & 1{,}866 \\ 1{,}002 & 0 & 0 & 0 & 76{,}744 & 165{,}709 & 1{,}326{,}951 \end{bmatrix}$$

$$\mathbf{V}^{2007} = \begin{bmatrix} 347{,}665 & 0 & 0 & 12 & 0 & 1{,}559 & 0 \\ 0 & 437{,}065 & 0 & 24{,}850 & 0 & 0 & 0 \\ 0 & 0 & 1{,}436{,}071 & 0 & 0 & 0 & 0 \\ 0 & 828 & 0 & 5{,}176{,}967 & 0 & 33{,}134 & 5{,}050 \\ 439 & 15 & 0 & 0 & 3{,}784{,}910 & 2{,}194 & 1{,}382 \\ 0 & 149 & 0 & 0 & 10 & 12{,}218{,}068 & 1{,}377 \\ 1{,}413 & 0 & 0 & 0 & 120{,}309 & 296{,}914 & 2{,}348{,}118 \end{bmatrix}$$

$$\mathbf{U}^{1997} = \begin{bmatrix} 48{,}986 & 86 & 1{,}067 & 155{,}059 & 1{,}282 & 3{,}447 & 1{,}306 \\ 1{,}195 & 17{,}051 & 7{,}663 & 126{,}256 & 32{,}295 & 1{,}439 & 10{,}910 \\ 879 & 1{,}958 & 189 & 15{,}114 & 6{,}110 & 38{,}589 & 33{,}763 \\ 44{,}105 & 19{,}986 & 205{,}959 & 1{,}532{,}339 & 97{,}639 & 333{,}465 & 165{,}270 \\ 25{,}240 & 12{,}589 & 85{,}547 & 335{,}127 & 210{,}522 & 198{,}233 & 61{,}843 \\ 28{,}584 & 29{,}237 & 68{,}715 & 320{,}296 & 413{,}007 & 1{,}774{,}829 & 203{,}053 \\ 720 & 808 & 4{,}873 & 25{,}755 & 43{,}718 & 52{,}333 & 17{,}330 \end{bmatrix}$$

$$\mathbf{U}^{2007} = \begin{bmatrix} 64,432 & 127 & 1,841 & 206,823 & 1,524 & 5,602 & 3,832 \\ 1,995 & 41,923 & 13,422 & 428,689 & 83,277 & 4,020 & 24,343 \\ 1,764 & 3,806 & 188 & 16,102 & 12,568 & 129,186 & 58,171 \\ 62,374 & 40,138 & 343,216 & 1,954,459 & 210,331 & 542,427 & 327,316 \\ 37,563 & 18,993 & 150,821 & 489,276 & 393,027 & 363,846 & 112,297 \\ 34,714 & 54,852 & 118,577 & 386,380 & 745,794 & 3,643,107 & 433,804 \\ 823 & 1,401 & 1,611 & 38,698 & 59,939 & 90,726 & 26,504 \end{bmatrix}$$

We can compute the vectors of total industry outputs, $\mathbf{x} = \mathbf{V}\mathbf{i}$, total commodity outputs, $\mathbf{q} = \mathbf{i}'\mathbf{V}$ for both years as:

$$\mathbf{x}^{1997} = \mathbf{V}^{1997}\mathbf{i} = [259,362 \;\; 164,470 \;\; 762,267 \;\; 3,761,592 \;\; 2,339,735 \;\; 6,772,545 \;\; 1,333,348]'$$

$$\mathbf{q}^{1997} = \mathbf{i}'\mathbf{V}^{1997}[259,075 \;\; 171,961 \;\; 762,267 \;\; 3,785,571 \;\; 2,264,488 \;\; 6,579,551 \;\; 1,570,406]'$$

$$\mathbf{x}^{2007} = \mathbf{V}^{2007}\mathbf{i} = [349,517 \;\; 438,057 \;\; 1,436,071 \;\; 5,201,829 \;\; 3,905,229 \;\; 12,551,869 \;\; 2,355,927]'$$

$$\mathbf{q}^{2007} = \mathbf{i}'\mathbf{V}^{2007} = [349,236 \;\; 461,915 \;\; 1,436,071 \;\; 5,215,979 \;\; 3,788,940 \;\; 12,219,604 \;\; 2,766,754]'$$

We can the compute the matrix of industry commodity requirements, **B**, and the matrix of commodity output proportions, **D**, for both years and under the assumption of an industry-by-industry model assuming industry-based technology we can specify the direct requirements matrix as $\mathbf{A} = \mathbf{DB}$ for each of the two years as:

$$\mathbf{A}^{1997} = \begin{bmatrix} 0.188273 & 0.000544 & 0.001409 & 0.041103 & 0.000568 & 0.000540 & 0.000996 \\ 0.004773 & 0.098467 & 0.010171 & 0.032766 & 0.013171 & 0.000320 & 0.008047 \\ 0.003389 & 0.011905 & 0.000248 & 0.004018 & 0.002611 & 0.005698 & 0.025322 \\ 0.169072 & 0.121839 & 0.268290 & 0.404378 & 0.042252 & 0.049956 & 0.123601 \\ 0.097359 & 0.076731 & 0.112182 & 0.089133 & 0.089964 & 0.029265 & 0.046382 \\ 0.110184 & 0.177859 & 0.090139 & 0.085177 & 0.176504 & 0.261988 & 0.152266 \\ 0.009150 & 0.011224 & 0.011481 & 0.011109 & 0.023286 & 0.014123 & 0.016394 \end{bmatrix}$$

$$\mathbf{A}^{2007} = \begin{bmatrix} 0.183530 & 0.000305 & 0.001287 & 0.039591 & 0.000413 & 0.000481 & 0.001643 \\ 0.006251 & 0.090990 & 0.009982 & 0.079768 & 0.020434 & 0.000509 & 0.010439 \\ 0.005047 & 0.008688 & 0.000131 & 0.003095 & 0.003218 & 0.010292 & 0.024691 \\ 0.177407 & 0.091459 & 0.237452 & 0.373278 & 0.054040 & 0.043692 & 0.138432 \\ 0.107608 & 0.043339 & 0.104929 & 0.094028 & 0.100577 & 0.029013 & 0.047656 \\ 0.099311 & 0.125233 & 0.082564 & 0.074299 & 0.190964 & 0.290211 & 0.184119 \\ 0.008570 & 0.007135 & 0.006298 & 0.011266 & 0.020864 & 0.014109 & 0.015542 \end{bmatrix}$$

The corresponding matrices of total requirements, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$, for the two years are:

$$\mathbf{L}^{1997} = (\mathbf{I} - \mathbf{A}^{1997})^{-1} = \begin{bmatrix} 1.252977 & 0.015617 & 0.027750 & 0.089978 & 0.007161 & 0.007798 & 0.014963 \\ 0.024627 & 1.122265 & 0.032976 & 0.068182 & 0.021330 & 0.006615 & 0.020653 \\ 0.008970 & 0.017526 & 1.005313 & 0.011175 & 0.006220 & 0.009337 & 0.029176 \\ 0.407744 & 0.285896 & 0.506676 & 1.768336 & 0.120680 & 0.133877 & 0.264423 \\ 0.187822 & 0.138849 & 0.188470 & 0.200829 & 1.125215 & 0.061665 & 0.094020 \\ 0.291291 & 0.346223 & 0.243804 & 0.289671 & 0.296740 & 1.393885 & 0.275576 \\ 0.025275 & 0.024644 & 0.026054 & 0.030630 & 0.032644 & 0.023243 & 1.026551 \end{bmatrix}$$

$$\mathbf{L}^{2007} = (\mathbf{I} - \mathbf{A}^{2007})^{-1} = \begin{bmatrix} 1.244908 & 0.010357 & 0.022731 & 0.082327 & 0.007667 & 0.006880 & 0.015992 \\ 0.049530 & 1.120132 & 0.053254 & 0.154067 & 0.038688 & 0.013456 & 0.039350 \\ 0.012215 & 0.013642 & 1.005408 & 0.011260 & 0.008746 & 0.016252 & 0.030428 \\ 0.407543 & 0.199923 & 0.430289 & 1.689458 & 0.140185 & 0.121968 & 0.280757 \\ 0.205969 & 0.086585 & 0.175881 & 0.205806 & 1.142868 & 0.064162 & 0.101938 \\ 0.288758 & 0.249039 & 0.226796 & 0.279912 & 0.339136 & 1.450302 & 0.335832 \\ 0.024442 & 0.015987 & 0.018918 & 0.029612 & 0.031088 & 0.023802 & 1.026593 \end{bmatrix}$$

Finally, we specify the vector of units of carbon dioxide emissions generated per dollar of total output in 1997 as $\mathbf{d}^{1997} = [2 \quad 3 \quad 4 \quad 7 \quad 10 \quad 5 \quad 4]'$. If we presume that the availability of new technology reduces the emissions per dollar of output in the year 2007 for the manufacturing sector by 10 percent and the construction sector by 15 percent, then we can specify the new emissions coefficients for 2007 as $\mathbf{d}^{2007} = [2 \quad 3 \quad 3.4 \quad 6.3 \quad 10 \quad 5 \quad 4]'$. Hence the total pollution impacts are $\Delta p = \mathbf{i}'[\mathbf{T}^{2007}\mathbf{f}^{2007} - \mathbf{T}^{19997}\mathbf{f}^{1997]}] = 57{,}916{,}899$ where $\mathbf{T}^{2007} = [\mathbf{d}^{2007}]'\mathbf{L}^{2007}$ and $\mathbf{T}^{1997} = [\mathbf{d}^{1997}]'\mathbf{L}^{1997}$ or equivalently $[\mathbf{d}^{2007}]'\mathbf{x}^{2007} - [\mathbf{d}^{1997}]'\mathbf{x}^{1997}$. In this case the improvement in pollution coefficients (reduction pollution generation per dollar of total output) was offset by growth in output levels so the net result was an increase in pollution impacts.

Computational Notes

We define the vector of emissions coefficients for 1997 as `DC1`. We presume the Make and Use matrices for 1997 and 2007 are retrieved in the APL workspace as `V1` and `U1`, respectively, for 1997 and `V2` and `U2` for 2007. Applying the specified reductions in emissions due to new technology we compute the vector of emissions coefficients for 2007 `DC2`. From `V1` and `V2` we compute the total commodity outputs and industry outputs `q1`, `q2`, `x1`, and `x2`. We compute the matrices of industry commodity requirements, `B1` and `B2`, and the matrices of commodity output proportions, `D1` and `D2`. Finally, assuming industry-based technology we compute `A1` and `A2`, the corresponding Leontief inverses, `L1` and `L2`, and vectors of final demand `f1` and `f2`.

```
DC1←1 7ρ2 3 4 7 10 5 4 ⋄ DC2←DC1×1 7ρ1 1 0.85 0.9 1 1 1
q1←+/V1 ⋄ x1←+⌿V1 ⋄ q2←+/V2 ⋄ x2←+⌿V2
D1←V1 AMAT q1 ⋄ B1←U1 AMAT x1 ⋄ D2←V2 AMAT q2 ⋄ B2←U2 AMAT x2
L1←LINV A1←D1+.×B1 ⋄ L2←LINV A2←D2+.×B2
f1←x1-+/Z1←A1+.×DIAG x1 ⋄ f2←x2-Z2←+/A2+.×DIAG x2
```

DC1

0.9968	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.9470	0.0000	0.0024	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0059	0.0000	0.9912	0.0000	0.0043	0.0024
0.0005	0.0022	0.0000	0.0000	0.9993	0.0000	0.0005
0.0000	0.0014	0.0000	0.0000	0.0000	0.9997	0.0012
0.0039	0.0000	0.0000	0.0000	0.0339	0.0252	0.8450

DC2

0.9955	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000
0.0000	0.9462	0.0000	0.0048	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0018	0.0000	0.9925	0.0000	0.0027	0.0018
0.0013	0.0000	0.0000	0.0000	0.9989	0.0002	0.0005
0.0000	0.0003	0.0000	0.0000	0.0000	0.9999	0.0005
0.0040	0.0000	0.0000	0.0000	0.0318	0.0243	0.8487

V1

258234	0	0	45	0	796	0
0	162842	0	9119	0	0	0
0	0	762267	0	0	0	0
0	1007	0	3752428	0	28347	3789
126	381	0	0	2262980	259	742
0	240	0	0	11	6577434	1866
1002	0	0	0	76744	165709	1326951

q1

259075	171961	762267	3785571	2264488	6579551	1570406

x1

259362	164470	762267	3761592	2339735	6772545	1333348

V2

347665	0	0	12	0	1559	0
0	437065	0	24850	0	0	0
0	0	1436071	0	0	0	0
0	828	0	5176967	0	33134	5050
439	15	0	0	3784910	2194	1382
0	149	0	0	10	12218068	1377
1413	0	0	0	120309	296914	2348118

q2

349236	461915	1436071	5215979	3788940	12219604	2766754

x2

349517	438057	1436071	5201829	3905229	12551869	2355927

A1

0.1883	0.0005	0.0014	0.0411	0.0006	0.0005	0.0010
0.0048	0.0985	0.0102	0.0328	0.0132	0.0003	0.0080
0.0034	0.0119	0.0002	0.0040	0.0026	0.0057	0.0253
0.1691	0.1218	0.2683	0.4044	0.0423	0.0500	0.1236
0.0974	0.0767	0.1122	0.0891	0.0900	0.0293	0.0464
0.1102	0.1779	0.0901	0.0852	0.1765	0.2620	0.1523
0.0091	0.0112	0.0115	0.0111	0.0233	0.0141	0.0164

L1

1.2530	0.0156	0.0277	0.0900	0.0072	0.0078	0.0150
0.0246	1.1223	0.0330	0.0682	0.0213	0.0066	0.0207
0.0090	0.0175	1.0053	0.0112	0.0062	0.0093	0.0292

```
    0.4077     0.2859     0.5067     1.7683     0.1207     0.1339     0.2644
    0.1878     0.1388     0.1885     0.2008     1.1252     0.0617     0.0940
    0.2913     0.3462     0.2438     0.2897     0.2967     1.3939     0.2756
    0.0253     0.0246     0.0261     0.0306     0.0326     0.0232     1.0266
A2
    0.1835     0.0003     0.0013     0.0396     0.0004     0.0005     0.0016
    0.0063     0.0910     0.0100     0.0798     0.0204     0.0005     0.0104
    0.0050     0.0087     0.0001     0.0031     0.0032     0.0103     0.0247
    0.1774     0.0915     0.2375     0.3733     0.0540     0.0437     0.1384
    0.1076     0.0433     0.1049     0.0940     0.1006     0.0290     0.0477
    0.0993     0.1252     0.0826     0.0743     0.1910     0.2902     0.1841
    0.0086     0.0071     0.0063     0.0113     0.0209     0.0141     0.0155
L2
    1.2449     0.0104     0.0227     0.0823     0.0077     0.0069     0.0160
    0.0495     1.1201     0.0533     0.1541     0.0387     0.0135     0.0393
    0.0122     0.0136     1.0054     0.0113     0.0087     0.0163     0.0304
    0.4075     0.1999     0.4303     1.6895     0.1402     0.1220     0.2808
    0.2060     0.0866     0.1759     0.2058     1.1429     0.0642     0.1019
    0.2888     0.2490     0.2268     0.2799     0.3391     1.4503     0.3358
    0.0244     0.0160     0.0189     0.0296     0.0311     0.0238     1.0266
```

We compute the total emissions coefficients matrices `T1` and `T2`, compute the total emissions for each year as `E1` and `E2`, and compute the change in total emissions `ΔE`.

```
       T1←DC1+.×L1 ⋄ T2←DC2+.×L2
       E1←T1+.×f1 ⋄ E2←T2+.×f2
       ΔE←+/E2-E1

T1
    8.9057     8.6876    10.9303    16.3867    13.8144     8.6890     8.4838
E1
  92985813.0
T2
    8.8487     6.8620     9.3029    14.8848    14.2930     8.8661     8.8272
E2
 150902712.1
ΔE
57916899
```

Problem 13.11: Attribution of Pollution Production or Consumption

This problem illustrates the attribution of pollution emissions, in this case CO_2 emissions, to either consumption or production using an interregional input-output (IRIO) model.

Problem 13.11 Overview

Consider the 3 region 2 sector IRIO interindustry technical coefficients matrix defined by

$$\mathbf{A}=\left[\begin{array}{cc:cc:cc} .222 & .121 & .027 & .023 & .007 & .014 \\ .217 & .028 & .014 & .015 & .021 & .012 \\ \hdashline .02 & .025 & .126 & .088 & .019 & .019 \\ .002 & .025 & .06 & .141 & .002 & .019 \\ \hdashline .012 & .02 & .019 & .005 & .192 & .179 \\ .022 & .017 & .005 & .013 & .195 & .164 \end{array}\right]$$

. The corresponding Leontief inverse is then

$$\mathbf{L}=(\mathbf{I}-\mathbf{A})^{-1}=\left[\begin{array}{cc:cc:cc} 1.335 & .17 & .048 & .045 & .025 & .033 \\ .3 & .1069 & .029 & .031 & .039 & .03 \\ \hdashline .043 & .04 & 1.155 & .125 & .038 & .039 \\ .018 & .037 & .085 & 1.217 & .015 & .034 \\ \hdashline .04 & .038 & .033 & .018 & 1.308 & .282 \\ .051 & .036 & .019 & .025 & .307 & 1.265 \end{array}\right]$$

. We define a vector of CO_2 emission coefficients as $\mathbf{g}'=\left[\begin{array}{cc:cc:cc} .9 & .4 & .3 & 1.0 & .2 & .7 \end{array}\right]$.

For a new vector of final demands presented to this IRIO economy, defined by

$$\mathbf{f}^{new}=\left[\begin{array}{c} 1500 \\ 2000 \\ \hdashline 55 \\ 40 \\ \hdashline 5 \\ 3 \end{array}\right]$$

, we can calculate the vector of the total CO_2 emissions associated with the total economic production for each sector in each region attributed to where the pollution is generated as

$$\mathbf{e}^{D}=\hat{\mathbf{g}}\mathbf{L}\mathbf{f}^{new}=\left[\begin{array}{c} 2112.0 \\ 1036.7 \\ \hdashline 63.9 \\ 153.9 \\ \hdashline 29.1 \\ 109.1 \end{array}\right]$$

. To attribute the emissions to consumption rather than production, i.e., where the consumption occurs that generates the demand for the production that generates the emissions, we specify the impacts as

$$\mathbf{e}^{C}=\mathbf{g}\mathbf{L}\hat{\mathbf{f}}^{new}=\left[\begin{array}{c} 2091.7 \\ 1320.2 \\ \hdashline 27.64 \\ 51.33 \\ \hdashline 2.7 \\ 3.1 \end{array}\right].$$

Note that both vectors sum to the same level of total CO_2 emissions, i.e., $\mathbf{i}'\mathbf{e}^{D}=\mathbf{i}'\mathbf{e}^{C}=34,967$, but $\mathbf{e}^{D}$ attributes the pollution generated to the sectors where the emissions were generated during production while $\mathbf{e}^{C}$ attributes the emissions to final consumers, i.e., final demand sectors that generated the demand for that industrial production and associate emissions. Since this economy is dominated by region 1, as is evident by total GDP in regions 1, 2, and 3 of 3500, 95, and 8, respectively (the sum of final demands in each region), the total emissions when attributable to final demand are 9 percent higher for region 1 and 63 and 96 percent lower in regions 2 and 3, respectively, when emissions are attributed to final consumption rather than the source of production.

Computational Notes

We first define technical coefficients matrix **A**, the vector of final demands **f**, and the emissions coefficients **g**. We then compute the Leontief inverse in order to compute the vector of total emissions attributed to production `ed`, the vector of total emissions attributed to consumption `ec`, and the sum of total emissions `tec`.

```
A←0.222 0.121 0.027 0.023 0.007 0.014 0.217 0.028 0.014
A←A,0.015 0.021 0.012 0.02 0.025 0.126 0.088 0.019 0.019
A←A,0.002 0.025 0.06 0.141 0.002 0.019 0.012 0.02 0.019
A←6 6⍴A,0.005 0.192 0.179 0.022 0.017 0.005 0.013 0.195 0.164
g←0.9 0.4 0.3 1 0.2 0.7
f←1500 2000 55 40 5 3

L←LINV A
ed←(DIAG g)+.×L+.×f
ec←g+.×L+.×DIAG f
tec←+/ec
```

```
A
   0.222    0.121   0.027   0.023   0.007   0.014
   0.217    0.028   0.014   0.015   0.021   0.012
   0.020    0.025   0.126   0.088   0.019   0.019
   0.002    0.025   0.060   0.141   0.002   0.019
   0.012    0.020   0.019   0.005   0.192   0.179
   0.022    0.017   0.005   0.013   0.195   0.164
f
1500.000 2000.000  55.000  40.000   5.000   3.000
g
   0.900    0.400   0.300   1.000   0.200   0.700
L
   1.335    0.170   0.048   0.044   0.025   0.032
   0.300    1.069   0.030   0.030   0.038   0.030
   0.043    0.040   1.155   0.121   0.038   0.038
   0.016    0.035   0.082   1.174   0.014   0.032
   0.040    0.038   0.033   0.017   1.308   0.282
   0.051    0.036   0.018   0.025   0.307   1.264
ed
2112.229 1036.775  63.608 146.054  29.110 108.882
ec
2091.726 1320.187  27.642  51.327   2.697   3.079
tec
3496.658
```

Problem 13.12: Attribution of Global Emissions

This problem explores the same issues regarding attribution of CO_2 emissions attributed to consumption versus production as in Problem 13.11, but for a 3-region, 3-sector global IRIO model.

Problem 13.12 Overview

Consider the Global IRIO transactions tables aggregated to 3 regions (the US, China, and Rest of World) and 3-Sector industry sectors (Agriculture and Mining, Manufacturing, and Services & Utilities) for the years 2005 and 2015 given in the text's Appendix SD2.

First, we retrieve the matrices of IRIO transactions, **Z**, and total outputs, **x**, and specify for the two years:

$$\mathbf{Z}^{2005} = \left[\begin{array}{ccc|ccc|ccc} 355 & 247 & 155 & 2 & 1 & 0 & 18 & 11 & 7 \\ 114 & 1{,}173 & 1{,}199 & 1 & 17 & 3 & 19 & 195 & 80 \\ 263 & 810 & 4{,}985 & 1 & 5 & 4 & 21 & 87 & 159 \\ \hline 1 & 1 & 0 & 237 & 208 & 71 & 7 & 8 & 4 \\ 2 & 42 & 35 & 85 & 1{,}149 & 388 & 9 & 158 & 71 \\ 0 & 3 & 4 & 57 & 236 & 368 & 2 & 13 & 14 \\ \hline 42 & 197 & 32 & 10 & 59 & 4 & 1{,}800 & 1{,}352 & 702 \\ 18 & 264 & 190 & 10 & 213 & 42 & 573 & 6123 & 3{,}449 \\ 12 & 60 & 132 & 7 & 45 & 31 & 1{,}052 & 3{,}100 & 10{,}903 \end{array}\right] \quad \mathbf{x}^{2005} = \left[\begin{array}{c} 1{,}346 \\ 4{,}568 \\ 17{,}168 \\ \hline 846 \\ 2{,}675 \\ 1{,}867 \\ \hline 6{,}760 \\ 16{,}816 \\ 37{,}192 \end{array}\right]$$

$$\mathbf{Z}^{2015} = \left[\begin{array}{ccc|ccc|ccc} 503 & 269 & 224 & 10 & 5 & 2 & 35 & 20 & 13 \\ 108 & 1{,}180 & 1{,}228 & 3 & 57 & 14 & 27 & 250 & 122 \\ 374 & 974 & 7{,}023 & 6 & 22 & 19 & 39 & 133 & 304 \\ \hline 2 & 0 & 1 & 1{,}336 & 905 & 325 & 18 & 5 & 9 \\ 6 & 108 & 121 & 423 & 5{,}879 & 2{,}001 & 39 & 502 & 283 \\ 1 & 9 & 12 & 413 & 1{,}563 & 2{,}519 & 6 & 43 & 42 \\ \hline 39 & 148 & 33 & 52 & 254 & 16 & 3{,}415 & 1{,}968 & 1{,}096 \\ 19 & 280 & 225 & 25 & 533 & 119 & 868 & 7{,}857 & 4{,}714 \\ 15 & 63 & 196 & 18 & 111 & 69 & 1{,}635 & 3{,}981 & 15{,}278 \end{array}\right],$$

$$\mathbf{x}^{2015} = \left[\begin{array}{c} 1{,}838 \\ 5{,}284 \\ 23{,}699 \\ \hline 3{,}937 \\ 12{,}429 \\ 10{,}818 \\ \hline 11{,}465 \\ 22{,}489 \\ 52{,}297 \end{array}\right]$$

From $\mathbf{Z}^{2005}$, $\mathbf{x}^{2005}$, $\mathbf{Z}^{2015}$, and $\mathbf{x}^{2015}$, we can calculate the associated vectors of final demand:

$$\mathbf{f}^{2005} = \mathbf{x}^{2005} - \mathbf{Z}^{2005}\mathbf{i} = [551 \quad 1{,}768 \quad 10{,}835 \;|\; 310 \quad 736 \quad 1{,}171 \;|\; 2{,}563 \quad 5{,}934 \quad 21{,}851]$$

$$\mathbf{f}^{2015} = \mathbf{x}^{2015} - \mathbf{Z}^{2015}\mathbf{i} = [757 \quad 2{,}296 \quad 14{,}805 \;|\; 1{,}335 \quad 3{,}069 \quad 6{,}211 \;|\; 4{,}446 \quad 7{,}849 \quad 30{,}931].$$

Also, we can compute the matrices of technical coefficients, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1}$, and total requirements, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$ for each year as:

$$\mathbf{A}^{2005} = \left[\begin{array}{ccc|ccc|ccc} 0.2634 & 0.0541 & 0.0090 & 0.0025 & 0.0003 & 0.0002 & 0.0026 & 0.0006 & 0.0002 \\ 0.0849 & 0.2567 & 0.0698 & 0.0011 & 0.0062 & 0.0018 & 0.0027 & 0.0116 & 0.0021 \\ 0.1953 & 0.1772 & 0.2903 & 0.0014 & 0.0020 & 0.0019 & 0.0031 & 0.0051 & 0.0043 \\ \hline 0.0006 & 0.0001 & 0.0000 & 0.2796 & 0.0778 & 0.0379 & 0.0010 & 0.0005 & 0.0001 \\ 0.0018 & 0.0093 & 0.0020 & 0.1006 & 0.4294 & 0.2078 & 0.0013 & 0.0094 & 0.0019 \\ 0.0002 & 0.0007 & 0.0002 & 0.0668 & 0.0881 & 0.1973 & 0.0003 & 0.0008 & 0.0004 \\ \hline 0.0308 & 0.0432 & 0.0018 & 0.0116 & 0.0221 & 0.0022 & 0.2663 & 0.0804 & 0.0189 \\ 0.0135 & 0.0578 & 0.0111 & 0.0116 & 0.0796 & 0.0225 & 0.0847 & 0.3641 & 0.0927 \\ 0.0090 & 0.0131 & 0.0077 & 0.0078 & 0.0168 & 0.0168 & 0.1556 & 0.1844 & 0.2931 \end{array}\right]$$

$$\mathbf{A}^{2015} = \left[\begin{array}{ccc|ccc|ccc} 0.2736 & 0.0510 & 0.0094 & 0.0026 & 0.0004 & 0.0002 & 0.0030 & 0.0009 & 0.0002 \\ 0.0586 & 0.2232 & 0.0518 & 0.0008 & 0.0046 & 0.0013 & 0.0023 & 0.0111 & 0.0023 \\ 0.2034 & 0.1842 & 0.2964 & 0.0015 & 0.0017 & 0.0018 & 0.0034 & 0.0059 & 0.0058 \\ \hline 0.0013 & 0.0000 & 0.0001 & 0.3394 & 0.0728 & 0.0301 & 0.0016 & 0.0002 & 0.0002 \\ 0.0034 & 0.0203 & 0.0051 & 0.1074 & 0.4730 & 0.1849 & 0.0034 & 0.0223 & 0.0054 \\ 0.0004 & 0.0017 & 0.0005 & 0.1048 & 0.1257 & 0.2329 & 0.0005 & 0.0019 & 0.0008 \\ \hline 0.0210 & 0.0280 & 0.0014 & 0.0132 & 0.0204 & 0.0014 & 0.2978 & 0.0875 & 0.0209 \\ 0.0102 & 0.0529 & 0.0095 & 0.0063 & 0.0429 & 0.0110 & 0.0757 & 0.3494 & 0.0901 \\ 0.0082 & 0.0119 & 0.0083 & 0.0045 & 0.0089 & 0.0064 & 0.1426 & 0.1770 & 0.2921 \end{array}\right]$$

$$\mathbf{L}^{2005} = \left[\begin{array}{ccc|ccc|ccc} 1.3780 & 0.1078 & 0.0282 & 0.0059 & 0.0041 & 0.0022 & 0.0064 & 0.0050 & 0.0017 \\ 0.1996 & 1.3971 & 0.1407 & 0.0083 & 0.0239 & 0.0113 & 0.0123 & 0.0316 & 0.0097 \\ 0.4304 & 0.3810 & 1.4527 & 0.0090 & 0.0167 & 0.0103 & 0.0149 & 0.0253 & 0.0138 \\ \hline 0.0027 & 0.0042 & 0.0013 & 1.4298 & 0.2151 & 0.1233 & 0.0035 & 0.0054 & 0.0017 \\ 0.0120 & 0.0305 & 0.0093 & 0.3099 & 1.8781 & 0.5020 & 0.0101 & 0.0335 & 0.0102 \\ 0.0023 & 0.0053 & 0.0017 & 0.1532 & 0.2245 & 1.3114 & 0.0023 & 0.0061 & 0.0022 \\ \hline 0.0812 & 0.1098 & 0.0196 & 0.0441 & 0.0971 & 0.0383 & 1.4006 & 0.1995 & 0.0643 \\ 0.0767 & 0.1692 & 0.0478 & 0.0858 & 0.2842 & 0.1306 & 0.2440 & 1.6776 & 0.2283 \\ 0.0642 & 0.1005 & 0.0358 & 0.0591 & 0.1484 & 0.0873 & 0.3728 & 0.4834 & 1.4891 \end{array}\right]$$

$$\mathbf{L}^{2015} = \left[\begin{array}{ccc|ccc|ccc} 1.3923 & 0.0984 & 0.0261 & 0.0069 & 0.0041 & 0.0019 & 0.0075 & 0.0055 & 0.0020 \\ 0.1357 & 1.3229 & 0.0999 & 0.0070 & 0.0176 & 0.0076 & 0.0105 & 0.0282 & 0.0093 \\ 0.4393 & 0.3773 & 1.4556 & 0.0100 & 0.0151 & 0.0087 & 0.0169 & 0.0279 & 0.0175 \\ \hline 0.0060 & 0.0090 & 0.0030 & 1.5737 & 0.2478 & 0.1216 & 0.0070 & 0.0116 & 0.0041 \\ 0.0252 & 0.0702 & 0.0226 & 0.4243 & 2.0900 & 0.5222 & 0.0270 & 0.0863 & 0.0289 \\ 0.0065 & 0.0164 & 0.0055 & 0.2847 & 0.3770 & 1.4060 & 0.0073 & 0.0205 & 0.0075 \\ \hline 0.0571 & 0.0785 & 0.0140 & 0.0519 & 0.0912 & 0.0308 & 1.4639 & 0.2215 & 0.0727 \\ 0.0545 & 0.1398 & 0.0374 & 0.0610 & 0.1719 & 0.0699 & 0.2221 & 1.6349 & 0.2169 \\ 0.0491 & 0.0796 & 0.0315 & 0.0440 & 0.0933 & 0.0440 & 0.3513 & 0.4557 & 1.4824 \end{array}\right]$$

If we estimate CO_2 emission indices as $\mathbf{g} = [.2 \quad .3 \quad .1 \;\vdots\; .3 \quad .5 \quad .2 \;\vdots\; .1 \quad .2 \quad .1]$ per million US dollars and, for simplicity, we assume these indices do not change between 2005 and 2015, we have all we need to compute the vectors of generated emissions from producing sectors as

$$\mathbf{e}^{p2005} = \hat{\mathbf{g}}\mathbf{L}^{2005}\mathbf{f}^{2005} = [269 \quad 1{,}370 \quad 1{,}717 \mid \quad 254 \quad 1{,}337 \quad 373 \mid \quad 676 \quad 3{,}363 \quad 3{,}719]$$
$$\mathbf{e}^{p2015} = \hat{\mathbf{g}}\mathbf{L}^{2015}\mathbf{f}^{2015} = [368 \quad 1{,}585 \quad 2{,}370 \mid 1{,}181 \quad 6{,}214 \quad 2{,}164 \mid 1{,}147 \quad 4{,}498 \quad 5{,}230].$$

The corresponding vectors of generated emissions attributed to consuming sectors are

$$\mathbf{e}^{c2005} = \mathbf{g}\mathbf{L}^{2005}\hat{\mathbf{f}}^{2005} = [229 \quad 974 \quad 2{,}314 \mid \quad 200 \quad 838 \quad 695 \mid \quad 613 \quad 2{,}590 \quad 4{,}626]$$
$$\mathbf{e}^{c2015} = \mathbf{g}\mathbf{L}^{2015}\hat{\mathbf{f}}^{2015} = [303 \quad 1{,}238 \quad 3{,}050 \mid 1{,}025 \quad 3{,}852 \quad 3{,}750 \mid 1{,}108 \quad 3{,}593 \quad 6{,}836].$$

For convenience we compute the sums of emissions for all sectors in each region, defining the vectors of total regional emissions (from producing sectors) for the two years as

$$\mathbf{e}^{rp2005} = [3{,}356 \quad 1{,}965 \quad 7{,}758] \text{ and } \mathbf{e}^{rp2015} = [4{,}323 \quad 9{,}558 \quad 10{,}874].$$

The vectors of total regional emissions (attributed to consuming sectors) for the two years are

$$\mathbf{e}^{rc2005} = [3{,}518 \quad 1{,}733 \quad 7{,}828] \text{ and } \mathbf{e}^{rc2015} = [4{,}591 \quad 8{,}627 \quad 11{,}538].$$

The percentage shifts for each region for attributing emissions to consumption rather than production are, for 2005 a 5 and 1 percent increase in the US and ROW, respectively, and a 12 percent decrease in China. For 2015 there is a 6 percent increase in both the US and ROW and a 10 percent decrease in China.

Computational Notes

We presume the matrices of global IRIO transactions for 2005 and 2015 are present in the APL workspace as `Z1` and `Z2`, respectively, along with the corresponding vectors of total outputs `x1` and `x2`. We define the vector of global emissions coefficients **g** and compute the vectors of total final demands, `f1` and `f2`. We compute the matrices of total pollution flows `GP1` and `GP2` and compute the rows sums of each to yield total emissions attributed to production, `ep1` and `ep2`, as well as the column sums of each to yield total emissions attributed to consumption, `ec1` and `ec2`.

```
        f1←x1-+/Z1  ⋄ f2←x2-+/Z2
        g←0.2 0.3 0.1 0.3 0.5 0.2 0.1 0.2 0.1
        L1←INV A1←Z1 AMAT x1 ⋄ L2←INV A2←Z2 AMAT x2
        GP1←(DIAG g)+.×L1+.×DIAG f1 ⋄ GP2←(DIAG g)+.×L2+.×DIAG f2
        ep1←+/GP1 ⋄ ep2←+/GP2 ⋄ ec1←+⌿GP1 ⋄ ec2←+⌿GP2
```

```
f1
     551      1768     10835       310       736      1171      2563      5934     21851
f2
     757      2296     14805      1335      3069      6211      4446      7849     30931
A1
  0.2634    0.0541    0.0090    0.0025    0.0003    0.0002    0.0026    0.0006    0.0002
  0.0849    0.2567    0.0698    0.0011    0.0062    0.0018    0.0027    0.0116    0.0021
  0.1953    0.1772    0.2903    0.0014    0.0020    0.0019    0.0031    0.0051    0.0043
  0.0006    0.0001    0.0000    0.2796    0.0778    0.0379    0.0010    0.0005    0.0001
  0.0018    0.0093    0.0020    0.1006    0.4294    0.2078    0.0013    0.0094    0.0019
```

0.0002	0.0007	0.0002	0.0668	0.0881	0.1973	0.0003	0.0008	0.0004
0.0308	0.0432	0.0018	0.0116	0.0221	0.0022	0.2663	0.0804	0.0189
0.0135	0.0578	0.0111	0.0116	0.0796	0.0225	0.0847	0.3641	0.0927
0.0090	0.0131	0.0077	0.0078	0.0168	0.0168	0.1556	0.1844	0.2931

A2

0.2736	0.0510	0.0094	0.0026	0.0004	0.0002	0.0030	0.0009	0.0002
0.0586	0.2232	0.0518	0.0008	0.0046	0.0013	0.0023	0.0111	0.0023
0.2034	0.1842	0.2964	0.0015	0.0017	0.0018	0.0034	0.0059	0.0058
0.0013	0.0000	0.0001	0.3394	0.0728	0.0301	0.0016	0.0002	0.0002
0.0034	0.0203	0.0051	0.1074	0.4730	0.1849	0.0034	0.0223	0.0054
0.0004	0.0017	0.0005	0.1048	0.1257	0.2329	0.0005	0.0019	0.0008
0.0210	0.0280	0.0014	0.0132	0.0204	0.0014	0.2978	0.0875	0.0209
0.0102	0.0529	0.0095	0.0063	0.0429	0.0110	0.0757	0.3494	0.0901
0.0082	0.0119	0.0083	0.0045	0.0089	0.0064	0.1426	0.1770	0.2921

L1

1.3780	0.1078	0.0282	0.0059	0.0041	0.0022	0.0064	0.0050	0.0017
0.1996	1.3971	0.1407	0.0083	0.0239	0.0113	0.0123	0.0316	0.0097
0.4304	0.3810	1.4527	0.0090	0.0167	0.0103	0.0149	0.0253	0.0138
0.0027	0.0042	0.0013	1.4298	0.2151	0.1233	0.0035	0.0054	0.0017
0.0120	0.0305	0.0093	0.3099	1.8781	0.5020	0.0101	0.0335	0.0102
0.0023	0.0053	0.0017	0.1532	0.2245	1.3114	0.0023	0.0061	0.0022
0.0812	0.1098	0.0196	0.0441	0.0971	0.0383	1.4006	0.1995	0.0643
0.0767	0.1692	0.0478	0.0858	0.2842	0.1306	0.2440	1.6776	0.2283
0.0642	0.1005	0.0358	0.0591	0.1484	0.0873	0.3728	0.4834	1.4891

L2

1.3923	0.0984	0.0261	0.0069	0.0041	0.0019	0.0075	0.0055	0.0020
0.1357	1.3229	0.0999	0.0070	0.0176	0.0076	0.0105	0.0282	0.0093
0.4393	0.3773	1.4556	0.0100	0.0151	0.0087	0.0169	0.0279	0.0175
0.0060	0.0090	0.0030	1.5737	0.2478	0.1216	0.0070	0.0116	0.0041
0.0252	0.0702	0.0226	0.4243	2.0900	0.5222	0.0270	0.0863	0.0289
0.0065	0.0164	0.0055	0.2847	0.3770	1.4060	0.0073	0.0205	0.0075
0.0571	0.0785	0.0140	0.0519	0.0912	0.0308	1.4639	0.2215	0.0727
0.0545	0.1398	0.0374	0.0610	0.1719	0.0699	0.2221	1.6349	0.2169
0.0491	0.0796	0.0315	0.0440	0.0933	0.0440	0.3513	0.4557	1.4824

GP1

152	38	61	0	1	1	3	6	7
33	741	457	1	5	4	9	56	63
24	67	1574	0	1	1	4	15	30
0	2	4	133	47	43	3	10	11
3	27	51	48	691	294	13	100	112
0	2	4	9	33	307	1	7	10
4	19	21	1	7	4	359	118	141
8	60	103	5	42	31	125	1991	998
4	18	39	2	11	10	96	287	3254

GP2

211	45	77	2	3	2	7	9	12
31	911	444	3	16	14	14	67	86
33	87	2155	1	5	5	8	22	54
1	6	13	630	228	227	9	27	38
10	81	167	283	3207	1622	60	339	447
1	8	16	76	231	1746	6	32	46
4	18	21	7	28	19	651	174	225
8	64	111	16	105	87	198	2567	1342
4	18	47	6	29	27	156	358	4585

ep1

269	1370	1717	254	1337	373	676	3363	3719

ep2

368	1585	2370	1181	6214	2164	1147	4498	5230

ec1

229	974	2314	200	838	695	613	2590	4626

ec2

Finally, compute the vectors of total emissions by region attributed to production for the two years, respectively, `erp1` and `erp2`, along with vectors of total emissions by region attributed to consumption, `erc1` and `erc2`. Finally compute the percentage difference between the two years for production and consumption, respectively, as `perc1` and `perc2`.

```
      erp1←+/3 3ρep1 ⋄ erp2←+/3 3ρep2
      erc1←+/3 3ρec1 ⋄ erc2←+/3 3ρec2
      perc1←100×(erc1-erp1)÷erp1
      perc2←100×(erc2-erp2)÷erp2

erp1
     3356      1965      7758


erp2
     4323      9559     10874
erc1
     3518      1733      7828
erc2:
     4591      8627     11538

perc1
        5       ¯12         1
perc2
        6       ¯10         6s
```

Chapter 14, Mixed and Dynamic Models

Chapter 14 describes so called mixed input-output models that are driven by a mix of output and final demand specifications rather than driven either solely by specification by final demand or total output. This chapter also introduces dynamic input–output models that more explicitly capture the role of capital investment and utilization in the production process. The exercise problems for this chapter illustrate key features of several mixed and dynamic model configurations.

Dynamic Models

Problem 14.1: Basic Characteristics of the Dynamic Input-Output Model

This problem illustrates the basic structure of a dynamic input-output model.

Problem 14.1 Overview

Consider an input-output economy with technical coefficients defined as $\mathbf{A} = \begin{bmatrix} 0.3 & 0.1 \\ 0.2 & 0.5 \end{bmatrix}$ and capital coefficients defined as $\mathbf{B} = \begin{bmatrix} .01 & .003 \\ .005 & .020 \end{bmatrix}$. Current final demand is $\mathbf{f}^0 = \begin{bmatrix} 100 \\ 100 \end{bmatrix}$ and the projections for the next three years for final demand are given by $\mathbf{f}^1 = \begin{bmatrix} 125 \\ 160 \end{bmatrix}$, $\mathbf{f}^2 = \begin{bmatrix} 150 \\ 175 \end{bmatrix}$ and $\mathbf{f}^3 = \begin{bmatrix} 185 \\ 200 \end{bmatrix}$.

For $\mathbf{A}$ and $\mathbf{B}$ as defined we specify the dynamic model as $\mathbf{Bx}^{t+1} = (\mathbf{I} - \mathbf{A} + \mathbf{B})\mathbf{x}^t - \mathbf{f}^t$ or $\mathbf{x}^t = (\mathbf{I} - \mathbf{A} + \mathbf{B})^{-1}(\mathbf{Bx}^{t+1} + \mathbf{f}^t)$, which we can write as $\mathbf{x}^t = \mathbf{G}^{-1}(\mathbf{Bx}^{t+1} + \mathbf{f}^t)$ where $\mathbf{G} = (\mathbf{I} - \mathbf{A} + \mathbf{B})$. For this case we compute $\mathbf{G} = (\mathbf{I} - \mathbf{A} + \mathbf{B}) = \begin{bmatrix} .69 & -.103 \\ -.205 & .48 \end{bmatrix}$ and $\mathbf{G}^{-1} = \begin{bmatrix} 1.548 & .332 \\ .661 & 2.225 \end{bmatrix}$. The "dynamic multipliers" are defined as $\mathbf{R} = \mathbf{G}^{-1}\mathbf{B} = \begin{bmatrix} .017 & .011 \\ .018 & .046 \end{bmatrix}$, $\mathbf{R}^2\mathbf{G}^{-1} = \begin{bmatrix} .001 & .002 \\ .003 & .006 \end{bmatrix}$ and $\mathbf{R}^3\mathbf{G}^{-1} = \begin{bmatrix} .00006 & .00009 \\ .00018 & .00029 \end{bmatrix}$.

Then we can construct the difference equations in matrix terms as

$$\mathbf{D} = \begin{bmatrix} \mathbf{G} & -\mathbf{B} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{G} & -\mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{G} & -\mathbf{B} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{G}^{-1} & \mathbf{RG}^{-1} & \mathbf{R}^2\mathbf{G}^{-1} & \mathbf{R}^3\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{G}^{-1} & \mathbf{RG}^{-1} & \mathbf{R}^2\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}^{-1} & \mathbf{RG}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}^{-1} \end{bmatrix} \text{ so that } \begin{bmatrix} \mathbf{x}^0 \\ \cdots \\ \mathbf{x}^1 \\ \cdots \\ \mathbf{x}^2 \\ \cdots \\ \mathbf{x}^3 \end{bmatrix} = \mathbf{D} \begin{bmatrix} \mathbf{f}^0 \\ \cdots \\ \mathbf{f}^1 \\ \cdots \\ \mathbf{f}^2 \\ \cdots \\ \mathbf{f}^3 \end{bmatrix}$$

or, for the

base year and the three projected years, $\mathbf{x}^0 = \begin{bmatrix} 197.7 \\ 315 \end{bmatrix}$, $\mathbf{x}^1 = \begin{bmatrix} 257.7 \\ 468.3 \end{bmatrix}$, $\mathbf{x}^2 = \begin{bmatrix} 302.8 \\ 521.2 \end{bmatrix}$ and $\mathbf{x}^3 = \begin{bmatrix} 352.8 \\ 567.3 \end{bmatrix}$.

Computational Notes

Define the matrices of technical coefficients **A** and of capital coefficients **B** as well as the final demands for the current and future three years, `f0`, `f1`, `f2` and `f3`. For convenience define a three-sector identity matrix **I** and matrix of zeroes **O**. Compute the matrix forming the dynamic model **G** and its matrix inverse `GI`. Compute the dynamic multipliers **R**, `RGI`, and `R2GI` and construct matrix of difference equations `LD`. Assemble a vector `ff` which catenates the four final demand vectors as its columns. Finally compute the corresponding vector of total outputs `xx`.

```
A←2 2ρ0.3 0.1 0.2 0.5 ◇ B←2 2ρ0.01 0.003 0.005 0.02
f0←100 100 ◇ f1←125 160 ◇ f2←150 175 ◇ f3←185 200
I←2 2ρ1 0 0 ◇ O←2 2ρ0

GI←⌹G←I-A+B
R←GI+.×B ◇ R2GI←R+.×RGI←R+.×GI
LD←GI,RGI,R2GI,R3GI←R+.×R2GI
LD←LD,[1]O,GI,RGI,R2GI
LD←LD,[1]O,O,GI,RGI
LD←LD,[1]O,O,O,GI
ff←f0,f1,f2,f3
xx←LD+.×ff
```

```
A
   0.30000   0.10000
   0.20000   0.50000
B
   0.01000   0.00300
   0.00500   0.02000
G
   0.69000  ¯0.10300
  ¯0.20500   0.48000
GI
   1.54796   0.33217
   0.66111   2.22520
R
   0.01714   0.01129
   0.01774   0.04649
R2GI
   0.00124   0.00176
   0.00331   0.00563
LD
   1.54796   0.33217   0.03399   0.03081   0.00124   0.00176   0.00006   0.00009
   0.66111   2.22520   0.05819   0.10933   0.00331   0.00563   0.00018   0.00029
   0.00000   0.00000   1.54796   0.33217   0.03399   0.03081   0.00124   0.00176
   0.00000   0.00000   0.66111   2.22520   0.05819   0.10933   0.00331   0.00563
   0.00000   0.00000   0.00000   0.00000   1.54796   0.33217   0.03399   0.03081
   0.00000   0.00000   0.00000   0.00000   0.66111   2.22520   0.05819   0.10933
   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   1.54796   0.33217
```

```
b    0.00000    0.00000    0.00000    0.00000    0.00000    0.00000    0.66111    2.22520
ff
       100        100        125        160        150        175        185        200
xx
     197.7      315.0      257.7      468.3      302.8      521.2      352.8      567.3
```

Problem 14.2: Turnpike Growth in Dynamic Input-Output Models

This problem illustrates the basic concepts of turnpike growth in a dynamic input-output model.

Problems 14.2 Overview

Consider the following closed dynamic input-output model, $\mathbf{Ax}+\mathbf{B}(\mathbf{x}^t-\mathbf{x})=\mathbf{x}$ where

$\mathbf{x}^t$ and $\mathbf{x}$ are the vectors of future total outputs and current total outputs, respectively $\mathbf{A}=\begin{bmatrix}0.5 & 0.1\\ 0.1 & 0.5\end{bmatrix}$ and $\mathbf{B}=\begin{bmatrix}0 & 0.1\\ 0.1 & 0\end{bmatrix}$ are the matrices of technical and capital coefficients, respectively.

Assume that $\mathbf{x}^t=\lambda\mathbf{x}$, where λ is some scalar (called the turnpike growth rate), the dynamic input-output model is expressed as $\mathbf{Ax}+\mathbf{B}(\lambda\mathbf{x}\ -\mathbf{x})=\mathbf{x}$. Rearranging terms, this becomes $\mathbf{B}\lambda\mathbf{x}=(\mathbf{I}-\mathbf{A}+\mathbf{B})\mathbf{x}$ or $\mathbf{B}^{-1}(\mathbf{I}-\mathbf{A}+\mathbf{B})\mathbf{x}=\lambda\mathbf{x}$, which we write more succinctly as $\mathbf{Qx}^t=\lambda\mathbf{x}^t$ where $\mathbf{Q}=\mathbf{B}^{-1}(\mathbf{I}-\mathbf{A}+\mathbf{B})$.

In this case we compute $\mathbf{B}^{-1}=\begin{bmatrix}0 & 10\\ 10 & 0\end{bmatrix}$ and then $\mathbf{Q}=\begin{bmatrix}0 & 5\\ 5 & 0\end{bmatrix}$. To calculate the turnpike growth, we solve the characteristic equation $|\mathbf{Q}-\mathbf{I}|=0$, and we find $\lambda_{\max}=5$.

Computational Notes

We define the matrices of technical and capital coefficients, **A** and **B**, respectively, and for convenience we compute an appropriately sized identity matrix I and the matrix inverse of **B**, which we name `BI`, from which we construct the dynamic model coefficients **Q**.

```
A←2 2ρ0.5 0.1 0.1 0.5
B←2 2ρ0 0.1 0.1 0
I←2 2ρ1 0 0 1 ⋄ BI←⌹B
Q←BI+.×((I-A)+B)
```

To solve this problem, we introduce the monadic function `EIG`. Computing eigenvalues for all but the simplest matrices with the methods defining eigenvalues covered in Appendix A of the text is tedious and impractical. Computational methods for computing eigenvalues are beyond the scope of this text but we use one such method (the so-called power method for the finding dominant eigenvalue and eigenvector) in the function `EIG` (listed in the appendix to this volume) which takes as the right argument as square matrix and produces as the explicit result the dominant eigenvalue catenated with the corresponding eigenvector.

For this problem, we can use `EIG` to compute the turnpike growth rate `Lambda`.

```
      Lambda←1↑EIG Q
```

```
A
     0.500     0.100
     0.100     0.500
B
     0.000     0.100
     0.100     0.000
BI
     0.000    10.000
    10.000     0.000
Q
     0.000     5.000
     5.000     0.000
Lambda
5
```

Problem 14.3: Implications of Changes in Capital Coefficeints

This problem illustrates the implications of changes in capital coefficients on the turnpike growth formulation of a dynamic input-output model.

Problem 14.3 Overview

Consider the closed dynamic input-output model $\mathbf{Ax} + \mathbf{B}(\mathbf{x}^t - \mathbf{x}) = \mathbf{x}$, where $\mathbf{A} = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$. Under the assumption of turnpike growth, we calculate $\mathbf{Q} = \mathbf{B}^{-1}(\mathbf{I} - \mathbf{A} + \mathbf{B}) = \begin{bmatrix} 10 & -2 \\ -3 & 7 \end{bmatrix}$ and solving the characteristic question $|\mathbf{Q} - \mathbf{I}| = 0$ find that the turnpike growth rate is $\lambda_{\max} = 11.37$.

If both the capital coefficients for the first industry (the first column of **B**) are changed to 0.1, then $\mathbf{B} = \begin{bmatrix} .1 & 0 \\ .1 & .1 \end{bmatrix}$. Hence, we find $\mathbf{Q} = \begin{bmatrix} 10 & -2 \\ -12 & 9 \end{bmatrix}$ and $\lambda_{\max} = 14.42$, which is an increase associated with the change in capital coefficients and indicates an improvement in the apparent overall "health" of the economy.

Computational Notes

Once again, we define the matrices of technical and capital coefficients, **A** and **B**, respectively, and for convenience we compute an appropriately sized identity matrix **I** and the matrix inverse of **B**, which we name `BI`, from which we construct the dynamic model coefficients **Q**.

```
        A←2 2ρ0.1 0.2 0.3 0.4 ⋄ B←2 2ρ0.1 0 0 0.1
        I←2 2ρ1 0 0 1 ⋄ BI←⌹B
        Q←BI+.×((I-A)+B)
```

```
A
     0.100     0.200
     0.300     0.400
```

```
B
     0.100     0.000
     0.000     0.100
BI
    10.000     0.000
     0.000    10.000
Q
    10.000    ¯2.000
    ¯3.000     7.000
```

We compute the revised matrix of capital coefficients `B2` as specified (and its matrix inverse `B2I`) in order to compute the corresponding dynamic model coefficients `Q2`. Then we use the function `EIG` to compute the turnpike growth rates for `Q1` and `Q2`, which we denote as `Lambda1` and `Lambda2`.

```
       B2←B ⋄ B2[;1]←0.1 ⋄ B2I←⌹B2
       Q2←B2I+.×((I-A)+B2)
       Lambda1←1↑EIG Q ⋄ Lambda2←1↑EIG Q2
```

```
B2
     0.100     0.000
     0.100     0.100
B2I
    10.000     0.000
   ¯10.000    10.000
Q2
    10.000    ¯2.000
   ¯12.000     9.000
Lambda1
11.372281
Lambda2
14.424429
```

Problem 14.4: Dynamic Multipliers

This problem illustrates the basic concepts of dynamic multipliers in dynamic input-output models.

Problem 14.4 Overview

Consider an input-output economy with technical coefficients defined as $\mathbf{A} = \begin{bmatrix} 0.2 & 0.1 \\ 0.3 & 0.5 \end{bmatrix}$ and

capital coefficients defined as $\mathbf{B} = \begin{bmatrix} .02 & .002 \\ .003 & .01 \end{bmatrix}$.

As in earlier problems, for **A** and **B** we specify the dynamic model as $\mathbf{Bx}^{t+1} = (\mathbf{I} - \mathbf{A} + \mathbf{B})\mathbf{x}^t - \mathbf{f}^t$

or $\mathbf{x}^t = (\mathbf{I} - \mathbf{A} + \mathbf{B})^{-1}(\mathbf{Bx}^{t+1} + \mathbf{f}^t)$, which we can write as $\mathbf{x}^t = \mathbf{G}^{-1}(\mathbf{Bx}^{t+1} + \mathbf{f}^t)$ where

$\mathbf{G}=(\mathbf{I}-\mathbf{A}+\mathbf{B})$. For this case, we compute $\mathbf{G}=(\mathbf{I}-\mathbf{A}+\mathbf{B})=\begin{bmatrix} .78 & -.102 \\ -.303 & .49 \end{bmatrix}$ and

$\mathbf{G}^{-1}=\begin{bmatrix} 1.395 & .29 \\ .863 & 2.220 \end{bmatrix}$, so the "dynamic multipliers" are $\mathbf{R}=\mathbf{G}^{-1}\mathbf{B}=\begin{bmatrix} .029 & .006 \\ .024 & .024 \end{bmatrix}$,

$\mathbf{R}^2\mathbf{G}^{-1}=\begin{bmatrix} .002 & .001 \\ .002 & .002 \end{bmatrix}$ and $\mathbf{R}^3\mathbf{G}^{-1}=\begin{bmatrix} .00006 & .00004 \\ .00010 & .00007 \end{bmatrix}$ such that

$\Delta\mathbf{x}^{-3}=\mathbf{R}^3\mathbf{G}^{-1}\Delta\mathbf{f}^0$, $\Delta\mathbf{x}^{-2}=\mathbf{R}^2\mathbf{G}^{-1}\Delta\mathbf{f}^0$ and $\Delta\mathbf{x}^{-1}=\mathbf{R}\mathbf{G}^{-1}\Delta\mathbf{f}^0$ or, in expanded matrix terms,

$$\begin{bmatrix} \mathbf{x}^{-3} \\ \mathbf{x}^{-2} \\ \mathbf{x}^{-1} \\ \mathbf{x}^{0} \end{bmatrix}=\begin{bmatrix} \mathbf{G}^{-1} & \mathbf{R}\mathbf{G}^{-1} & \mathbf{R}^2\mathbf{G}^{-1} & \mathbf{R}^3\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{G}^{-1} & \mathbf{R}\mathbf{G}^{-1} & \mathbf{R}^2\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{G}^{-1} & \mathbf{R}\mathbf{G}^{-1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{G}^{-1} \end{bmatrix}\begin{bmatrix} \mathbf{f}^{-3} \\ \mathbf{f}^{-2} \\ \mathbf{f}^{-1} \\ \mathbf{f}^{0} \end{bmatrix}.$$

If we assume the current vector of final demands is $\mathbf{f}^0=\begin{bmatrix} 185 \\ 200 \end{bmatrix}$ and the vectors for final demand for the previous three years are given by $\mathbf{f}^{-1}=\begin{bmatrix} 150 \\ 175 \end{bmatrix}$, $\mathbf{f}^{-2}=\begin{bmatrix} 125 \\ 160 \end{bmatrix}$, and $\mathbf{f}^{-3}=\begin{bmatrix} 100 \\ 100 \end{bmatrix}$, we can specify $\Delta\mathbf{f}=\begin{bmatrix} 100 & 100 & \vdots & 125 & 160 & \vdots & 150 & 175 & \vdots & 185 & 200 \end{bmatrix}'$ and compute

$\Delta\mathbf{x}=\begin{bmatrix} 177.9 & 325.4 & \vdots & 231.7 & 482.5 & \vdots & 272.6 & 539.9 & \vdots & 316.1 & 603.6 \end{bmatrix}'$.

Computational Notes

Define the matrices of technical coefficients **A** and of capital coefficients **B** as well as the final demands for the current and three previous years, `f0`, `f1`, `f2` and `f3`. For convenience define a three-sector identity matrix **I** and matrix of zeroes **O**. Compute the matrix forming the dynamic model **G** and its matrix inverse `GI`. Compute the dynamic multipliers **R**, `RGI`, and `R2GI` and construct matrix of difference equations `LD`. Assemble a vector `ff` which catenates the four final demand vectors as its columns. Finally compute the corresponding vector of total outputs `xx`.

```
A←2 2⍴0.2 0.1 0.3 0.5 ⋄ B←2 2⍴0.02 0.002 0.003 0.01
f3←100 100 ⋄ f2←125 160 ⋄ f1←150 175 ⋄ f0←185 200
I←2 2⍴1 0 0 ⋄ O←2 2⍴0

GI←⌹G←I-A+B
R←GI+.×B ⋄ R2GI←R+.×RGI←R+.×GI
LD←GI,RGI,R2GI,R3GI←R+.×R2GI
LD←LD,[1]O,GI,RGI,R2GI
LD←LD,[1]O,O,GI,RGI
```

```
      LD←LD,[1]0,0,0,GI
      ff←f3,f2,f1,f0
      xx←LD+.×ff

A
   0.20000   0.10000
   0.30000   0.50000
B
   0.02000   0.00200
   0.00300   0.01000
G
   0.78000  ¯0.10200
  ¯0.30300   0.49000
GI
   1.39484   0.29036
   0.86253   2.22036
R
   0.02877   0.00569
   0.02391   0.02393
R2GI
   0.00160   0.00095
   0.00237   0.00194
R3GI
   0.00006   0.00004
   0.00010   0.00007
LD
   1.39484   0.29036   0.04504   0.02099   0.00160   0.00095   0.00006   0.00004
   0.86253   2.22036   0.05399   0.06007   0.00237   0.00194   0.00010   0.00007
   0.00000   0.00000   1.39484   0.29036   0.04504   0.02099   0.00160   0.00095
   0.00000   0.00000   0.86253   2.22036   0.05399   0.06007   0.00237   0.00194
   0.00000   0.00000   0.00000   0.00000   1.39484   0.29036   0.04504   0.02099
   0.00000   0.00000   0.00000   0.00000   0.86253   2.22036   0.05399   0.06007
   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   1.39484   0.29036
   0.00000   0.00000   0.00000   0.00000   0.00000   0.00000   0.86253   2.22036
ff
       100       100       125       160       150       175       185       200
xx
     177.9     325.4     231.7     482.5     272.6     539.9     316.1     603.6
```

Mixed Models

Problem 14.5: Basic Characteristics of a Mixed Input-Output Model

This exercise problem illustrates the basic characteristics of a mixed input-output model.

Problem 14.5 Overview

Consider an input-output economy specified by an interindustry transactions matrix,

$\mathbf{Z} = \begin{bmatrix} 14 & 76 & 46 \\ 54 & 22 & 5 \\ 68 & 71 & 94 \end{bmatrix}$ and vector of final demands, $\mathbf{f} = \begin{bmatrix} 100 \\ 200 \\ 175 \end{bmatrix}$ where the three industrial sectors are manufacturing, oil, and electricity.

Suppose the economic forecasts determine that total domestic output for oil and electricity will remain unchanged in the next year and final demand for manufactured goods will increase by 30 percent. That is, the projection is a mixture of total outputs and final demands rather than only final demands (or total outputs). In such a situation, we can construct a mixed input-output model by first determining the economy's total outputs as $\mathbf{x} = \mathbf{f} + \mathbf{Zi} = \begin{bmatrix} 236 \\ 281 \\ 408 \end{bmatrix}$ so that we can compute the matrix of technical coefficients, $\mathbf{A} = \mathbf{Z}(\hat{\mathbf{x}})^{-1} = \begin{bmatrix} .059 & .270 & .113 \\ .229 & .078 & .012 \\ .288 & .253 & .230 \end{bmatrix}$. For sector 1 (manufactured goods), the level of final demand is exogenously specified and for sectors 2 and 3 (oil and electricity), levels of total output are specified for each sector, so we partition $\mathbf{A}$ as $\mathbf{A} = \left[\begin{array}{c|c} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \hline \mathbf{A}_{21} & \mathbf{A}_{22} \end{array}\right] = \left[\begin{array}{c|cc} .059 & .27 & .113 \\ \hline .229 & .078 & .012 \\ .288 & .253 & .23 \end{array}\right]$, and with a vector of exogenously specified values $\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right]$ and the vector of endogenously determined values designated by $\left[\begin{array}{c} \mathbf{x} \\ \hline \mathbf{f} \end{array}\right]$ we write $\mathbf{M}\left[\begin{array}{c} \mathbf{x} \\ \hline \mathbf{f} \end{array}\right] = \mathbf{N}\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right]$ where $\mathbf{M} = \left[\begin{array}{c|c} (\mathbf{I} - \mathbf{A}_{11}) & \mathbf{0} \\ \hline -\mathbf{A}_{21} & -\mathbf{I} \end{array}\right] = \left[\begin{array}{c|cc} .941 & 0 & 0 \\ \hline -.229 & -1 & 0 \\ -.288 & 0 & -1 \end{array}\right]$ and $\mathbf{N} = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{A}_{12} \\ \hline \mathbf{0} & -(\mathbf{I} - \mathbf{A}_{22}) \end{array}\right] = \left[\begin{array}{c|cc} 1 & .270 & .113 \\ \hline 0 & -.922 & .012 \\ 0 & .253 & -.770 \end{array}\right]$.

It follows that $\mathbf{M}^{-1} = \left[\begin{array}{c|cc} 1.063 & 0 & 0 \\ \hline -.243 & -1 & 0 \\ -.306 & 0 & -1 \end{array}\right]$.

For the case where the economic forecasts determine that total domestic output for oil and electricity will remain unchanged in the next year and final demand for manufactured goods will increase by 30 percent, we specify $\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right] = \left[\begin{array}{c} 130 \\ \hline 281 \\ 408 \end{array}\right]$ and find $\left[\begin{array}{c} \mathbf{x} \\ \hline \mathbf{f} \end{array}\right] = \mathbf{M}^{-1}\mathbf{N}\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right] = \left[\begin{array}{c} 267.9 \\ \hline 192.7 \\ 165.8 \end{array}\right]$. That is, total output of manufactured goods will be 267.9, and final demands presented to the economy for oil and electricity are 192.7 and 165.8, respectively.

If instead the final demand for manufactured goods increased by 50 percent instead of 30 percent, we find the new projections of final demand for oil and electricity and the total output of manufacturing as $\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right] = \left[\begin{array}{c} 150 \\ \hline 281 \\ 408 \end{array}\right]$ we find $\left[\begin{array}{c} \mathbf{x} \\ \hline \mathbf{f} \end{array}\right] = \mathbf{M}^{-1}\mathbf{N}\left[\begin{array}{c} \mathbf{f} \\ \hline \overline{\mathbf{x}} \end{array}\right] = \left[\begin{array}{c} 289.2 \\ \hline 187.8 \\ 159.7 \end{array}\right]$.

Computational Notes

We first specify the matrix of interindustry transactions **Z** and the vector of total final demands **f**. Then compute the corresponding vector of total outputs **x** and matrix of technical coefficients **A**.

```
      Z←3 3ρ14 76 46 54 22 5 68 71 94
      f←100 200 175
      A←Z AMAT x←f++/Z
```

```
Z
        14        76        46
        54        22         5
        68        71        94
f
       100       200       175
A
    0.0593    0.2705    0.1127
    0.2288    0.0783    0.0123
    0.2881    0.2527    0.2304
```

To help facilitate solving this problem we introduce the dyadic function `MIXED` which configures a mixed input output model. The function takes as its left argument the technical coefficients matrix **A**. The right argument is a two-row vector `yx`, the first row of which specifies the known exogenous values for final demand and zero otherwise and the second row specifies the known values for total outputs and zero otherwise. The function returns as the explicit result a vector `xy` which include the computed values for the unknown values indicated by zeroes in `yx`. The function also computes global variables **M** (and its inverse `MI`) and **N** used in configuring the mixed model.

```
[  0]  xy←A MIXED yx;P;Q;R;S
[  1] ⍝mixed model yx[1;]=yexog,0 for endog; yx[2;]xexog,0 for endog
[  2]  nmx←+/nyx←yx≠0 ⋄ nm←1↑ρA ⋄ n←nmx[1] ⋄ m←nmx[2]
[  3]  Inm←(2ρnm)ρ1,nmρ0 ⋄ In←(2ρn)↑Inm ⋄ Im←(2ρm)↑Inm ⋄ Onm←(n,m)ρ0 ⋄ Omn←(m,n)ρ0
[  4] ⍝reorder A to AR
[  5]  qx←nyx×(2,nm)ρ⍳nm ⋄ q←((qx[1;]≠0)/qx[1;]),(qx[2;]≠0)/qx[2;]
[  6]  AR←q REORDER A
[  7] ⍝construct mixed exogenous-endogenous model (y exog first)
[  8]  IAR←Inm-A
[  9]  P←(n,n)↑IAR ⋄ R←IAR[n+⍳m;⍳n] ⋄ Q←-IAR[⍳n;n+⍳m] ⋄ S←-IAR[n+⍳m;n+⍳m]
[ 10]  M←(P,Onm),[1]R,-Im ⋄ N←(In,Q),[1]Omn,S
[ 11]  xy←(MI←⌹M)+.×N+.×(+⌿yx)[q]
```

As somewhat of an aside, note that with this problem the order of the sectors for the exogenously specified quantities is that the exogenous element of final demand is specified first with the exogenously specified elements to total outputs following. In general, this may not be case so it is helpful (although not necessary) generally to reorder the sectors so that it is the case, which in the function `MIXED` is accomplished with another function named `REORDER`. The function `REORDER` takes **A** as the right argument and the left argument is a vector specifying the order of indexes to accomplish this (computed in `MIXED`, but, of course, could be specified in other ways) and returns, as the explicit result, the matrix `AR`, which is comprised of elements of **A** reordered as specified.

For example, consider the matrix

```
      Q← 3 3ρι9
      Q
1 2 3
4 5 6
7 8 9
```

To exchange the 2nd and 3rd rows and columns, we would use `REORDER`:

```
      1 3 2 REORDER Q
1 3 2
7 9 8
4 6 5
```

For this problem, to apply `MIXED`, we configure the known and unknown values of projected final demands and total outputs in the matrix `YX1`, which is in the format expected by `MIXED` noted above. The vector of exogenously specified variables `yx1`, is computed as the column sums of `YX1`. Again, as an aside, it is import to remember that if the sectors are not ordered with the exogenously specified final demands listed first, `MIXED` will reorder the sectors so that they are (not necessary for this problem since sectors are already in this order). `MIXED` will return the computed values in the variable `xy1` and we preserve the working matrices **M**, **N**, `NI` as `M1`, `N1`, and `MI`, respectively.

```
      xy1←A MIXED YX1←2 3ρ(1.3×f[1]),0 0 0,x[2 3]
YX1
     130.0       0.0       0.0
       0.0     281.0     408.0
yx1
     130.0     281.0     408.0
xy1
     267.9     192.7     165.8
M1
     0.941     0.000     0.000
    ¯0.229    ¯1.000     0.000
    ¯0.288     0.000    ¯1.000
N1
     0.941     0.000     0.000
    ¯0.229    ¯1.000     0.000
    ¯0.288     0.000    ¯1.000
MI1
     1.063     0.000     0.000
    ¯0.243    ¯1.000     0.000
    ¯0.306     0.000    ¯1.000
```

For the modified case where the final demand for manufactured goods increased by 50 percent instead of 30 percent, we compute

```
      xy2←A MIXED YX2←2 3ρ(1.5×f[1]),0 0 0,x[2 3]

YX2
     150.0       0.0       0.0
```

```
      0.0      281.0      408.0
yx2
    150.0      281.0      408.0


xy2
    289.2      187.8      159.7
M2
    0.941      0.000      0.000
   ¯0.229     ¯1.000      0.000
   ¯0.288      0.000     ¯1.000
N2
    1.000      0.270      0.113
    0.000     ¯0.922      0.012
    0.000      0.253     ¯0.770
MI2
    1.063      0.000      0.000
   ¯0.243     ¯1.000      0.000
   ¯0.306      0.000     ¯1.000
```

Problem 14.6: Modeling New Economic Sector Additions

This problem explores modeling establishment of a new economic sector using input-output analysis.

Problem 14.6 Overview

Revisiting the economy of Problem 2.1, consider the prospect of adding a new sector, finance and insurance (sector 3), to this economy. First, we can recall from Problem 2.1 that, for this economy, the interindustry transactions matrix, $\mathbf{Z} = \begin{bmatrix} 500 & 350 \\ 320 & 360 \end{bmatrix}$, and the vector of total outputs, $\mathbf{x} = \begin{bmatrix} 1{,}000 \\ 800 \end{bmatrix}$, from which we can compute the matrix of technical coefficients, $\mathbf{A} = \mathbf{Z}\hat{\mathbf{x}}^{-1} = \begin{bmatrix} .500 & .438 \\ .320 & .450 \end{bmatrix}$, and the total requirements matrix, $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} 4.074 & 3.241 \\ 2.370 & 3.704 \end{bmatrix}$.

Initially we know that the total output of this new sector will be $x_3 = \$900$ during the current year (its first year of operation), and that its needs for agricultural and manufactured goods are captured by technical coefficients $a_{13} = 0.001$ and $a_{23} = 0.07$. In the absence of any further information, we can estimate to be the impact of this new sector on the economy by first constructing a final demand vector by multiplying each of the new technical coefficients by the associated known total output to yield $\Delta\mathbf{f} = \begin{bmatrix} a_{13}x_3 \\ a_{23}x_3 \end{bmatrix} = \begin{bmatrix} (.001)(900) \\ (.07)(900) \end{bmatrix} = \begin{bmatrix} 0.9 \\ 63.0 \end{bmatrix}$. The impact of the new sector is found by $\Delta\mathbf{x} = \mathbf{L}\Delta\mathbf{f} = \begin{bmatrix} 207.8 \\ 235.47 \end{bmatrix}$.

Suppose we learn subsequently that: (1) that the agriculture and manufacturing sectors bought \$20 and \$40 in finance and insurance services last year from foreign firms (i.e., that they imported these inputs), and (2) that sector 3 will use \$15 of its own product for each \$100 worth of its output. We now have enough information to endogenize the sector into the interindustry transactions matrix, including specifying $a_{33} = 15/100 = 0.15$, $a_{31} = 20/x_1 = 20/1000 = 0.02$, and $a_{32} = 40/x_2 = 40/800 = 0.05$ so the new, expanded technical coefficient matrix becomes $\overline{\mathbf{A}} = \begin{bmatrix} .500 & .438 & .001 \\ .320 & .450 & .070 \\ .020 & .050 & .150 \end{bmatrix}$ so the new total requirements matrix is $\overline{\mathbf{L}} = \begin{bmatrix} 4.130 & 3.310 & .277 \\ 2.433 & 3.782 & .314 \\ .240 & .300 & 1.201 \end{bmatrix}$.

Hence, the new, expanded total outputs vector is $\overline{\mathbf{x}} = \begin{bmatrix} 1,000 \\ 800 \\ 900 \end{bmatrix}$ so the new interindustry transactions matrix is found as $\overline{\mathbf{Z}} \equiv \mathbf{A}\hat{\overline{\mathbf{x}}} = \begin{bmatrix} 500 & 350 & .9 \\ 320 & 360 & 63 \\ 20 & 40 & 135 \end{bmatrix}$. The third column of $\mathbf{Z}$ describes the interindustry purchases of the three sectors' outputs by the new finance and insurances services sector, which we can also describe as a new (at least in the first year) final demand to the expanded regional economy, $\Delta\overline{\mathbf{f}} = \begin{bmatrix} 0.9 \\ 63.0 \\ 135.0 \end{bmatrix}$. We can now use the new expanded total requirements matrix, $\mathbf{L}$, to compute the total output in the economy to support introduction of the new economic sector, $\Delta\overline{\mathbf{x}} = \overline{\mathbf{L}}\Delta\overline{\mathbf{f}} = \begin{bmatrix} 249.7 \\ 282.9 \\ 181.3 \end{bmatrix}$.

Computational Notes

We first define the two-sector transactions matrix **Z** and the vector of total outputs **x**. We then compute the corresponding vector of total final demands **f** and the matrices of technical coefficients and the Leontief inverse, **A** and **L**, respectively.

```
      Z←2 2⍴500 350 320 360 ⋄ x←1000 800
      f←x-+/Z ⋄ L←LINV A←Z AMAT x

f
     150.0     120.0
A
    0.5000    0.4375
    0.3200    0.4500
L
    4.0741    3.2407
    2.3704    3.7037
```

We compute the impact of the new sector specified as `Δf` in terms of total outputs `Δx`.

```
      Δx←L+.×Δf←0.001 0.07×900

Δf
       0.9      63.0
Δx
     207.8     235.5
```

We specify the expanded matrix of technical coefficients as `A2` and compute the corresponding Leontief inverse `L2` and corresponding vectors of total outputs and total final demands as `x2` and `f2`, respectively, as well as the associated matrix of interindustry transactions `Z2`.

```
      A2←3 3⍴0 ⋄ A2[⍳2;⍳2]←A ⋄ A2[;3]←0.001 0.07 0.15 ⋄ A2[3;1 2]←20 40÷x
      L2←LINV A2 ⋄ x2←x,900 ⋄ f2←f,¯1↑A2+.×x2 ⋄ Z2←A2+.×DIAG x2

A2
    0.5000    0.4375    0.0010
    0.3200    0.4500    0.0700
    0.0200    0.0500    0.1500
L2
    4.1296    3.3101    0.2775
    2.4332    3.7823    0.3143
    0.2403    0.3004    1.2015
f2
     150.0     120.0     195.0
x2
    1000.0     800.0     900.0

Z2
     500.0     350.0       0.9
     320.0     360.0      63.0
      20.0      40.0     135.0
```

Finally, we specify the new third column of `Z2` reduced by the specified level of imports as a new final demand `Δf2` and compute the corresponding vector of total outputs `Δx2`.

```
      Δf2←Δf,f2[3]-(20+40)
      Δx2←L2+.×Δf2

Δf2
       0.9      63.0     135.0
Δx2
     249.7     282.9     181.3
```

Problem 14.7: Uses of Mixed Input-Output Models

This problem illustrates use of a mixed input-output model applied to planning with availability of variable data, e.g., some estimated final demands for products of some sectors and some projected total outputs the balance of sectors in the economy.

Problem 14.7 Overview

We revisit the Czaria economy from Problem 7.1, recalling that

$$\mathbf{A} = \begin{bmatrix} 0.168 & 0.155 & 0.213 & 0.212 \\ 0.194 & 0.193 & 0.168 & 0.115 \\ 0.105 & 0.025 & 0.126 & 0.124 \\ 0.178 & 0.101 & 0.219 & 0.186 \end{bmatrix}$$

. Next year's projected total outputs in millions of dollars for agriculture, mining, and civilian manufacturing in Czaria are 4,558, 5,665 and 5,079, respectively, and final demand of military manufactured products is projected to be \$2,050 million.

To compute the GDP and total gross production of the economy next year, we can fashion a mixed model by first reordering the industry sectors so that those with exogenously specified final demands are listed first (in this case only sector 3) and those with exogenously specified total outputs are listed second (in this case sectors 1, 2, and 4). With the reordered sectors, we can compute

$$\mathbf{M} = \left[\begin{array}{c|c} (\mathbf{I}-\mathbf{A}_{11}) & \mathbf{0} \\ \hline -\mathbf{A}_{21} & -\mathbf{I} \end{array}\right] = \left[\begin{array}{c|ccc} .832 & 0 & 0 & 0 \\ \hline -.194 & -1 & 0 & 0 \\ -.105 & 0 & -1 & 0 \\ -.178 & 0 & 0 & -1 \end{array}\right]$$

and

$$\mathbf{N} = \left[\begin{array}{c|c} \mathbf{I} & \mathbf{A}_{12} \\ \hline \mathbf{0} & -(\mathbf{I}-\mathbf{A}_{22}) \end{array}\right] = \left[\begin{array}{c|ccc} 1 & .155 & .213 & .212 \\ \hline 0 & -.807 & .168 & .115 \\ 0 & .025 & -.874 & .124 \\ 0 & .101 & .219 & -.814 \end{array}\right]$$

, which satisfies the condition $\mathbf{M}\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{f} \end{bmatrix} = \mathbf{N}\begin{bmatrix} \mathbf{f} \\ \hline \mathbf{x} \end{bmatrix}$ or $\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{f} \end{bmatrix} = \mathbf{M}^{-1}\mathbf{N}\begin{bmatrix} \mathbf{f} \\ \hline \mathbf{x} \end{bmatrix}$ where $\begin{bmatrix} \mathbf{f} \\ \hline \mathbf{x} \end{bmatrix}$ is the vector of exogenously specified values and $\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{f} \end{bmatrix}$ is the vector of endogenously determined values.

To compute the endogenously determined values we first compute

$$\mathbf{M}^{-1} = \left[\begin{array}{c|ccc} 1.202 & 0 & 0 & 0 \\ \hline -.233 & -1 & 0 & 0 \\ -.126 & 0 & -1 & 0 \\ -.214 & 0 & 0 & -1 \end{array}\right]$$

and then

$$\mathbf{M}^{-1}\mathbf{N} = \left[\begin{array}{c|ccc} 1.202 & .186 & .256 & .255 \\ \hline -.233 & .771 & -.218 & -.164 \\ -.126 & -.045 & .847 & -.151 \\ -.214 & -.134 & -.265 & .769 \end{array}\right]$$

so that

$$\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{f} \end{bmatrix} = \mathbf{M}^{-1}\mathbf{N}\begin{bmatrix} \mathbf{f} \\ \hline \mathbf{x} \end{bmatrix} = \left[\begin{array}{c} 6,058 \\ \hline 967 \\ 3,572 \\ 1,355 \end{array}\right] \text{ for } \begin{bmatrix} \mathbf{f} \\ \hline \mathbf{x} \end{bmatrix} = \left[\begin{array}{c} 2,050 \\ \hline 4,558 \\ 5,665 \\ 5,079 \end{array}\right]$$

. Total output of sector 3 will be 6055, and amounts of sector 1, 2, and 4 production that are available for final demand are 969, 3573, and 1347, respectively. GDP is the sum of all final demands (7,944) and total gross production is the sum of all total outputs (21,360).

Computational Notes

We revisit from Problem 7.1 the matrix of technical coefficients **A** and specify the projected values for final demands and total outputs `yx`, but in a matrix `YX`, which is in the format expected by the function `MIXED`.

```
       A←0.168 0.155 0.213 0.212 0.194 0.193 0.168 0.115
       A←4 4⍴A,0.105 0.025 0.126 0.124 0.178 0.101 0.219 0.186
       yx←+⌿YX←2 4⍴2050 0 0 0 0 4558 5665 5079

A
     0.168     0.155     0.213     0.212
     0.194     0.193     0.168     0.115
     0.105     0.025     0.126     0.124
     0.178     0.101     0.219     0.186
YX
      2050         0         0         0
         0      4558      5665      5079
yx
      2050      4558      5665      5079
```

The result of applying `MIXED` is the vector `xy`. To compute the gross domestic product `GDP`, we select and sum the final demand elements from `xy` and `yx` and, to compute the total gross production `GX`, we select and sum the total outputs elements.

```
      xy←A MIXED YX
      GDP←+/yx[1],xy[2 3 4]
      GX←+/yx[2 3 4],xy[1]

xy
    6057.6     967.3    3571.4    1355.1
GDP
    7943.8
GX
   21359.6
```

Problem 14.8: Mixed Model Application with the U.S. Input-Output Data

This problem illustrates use of a mixed input-output model applied to planning with availability of variable data, e.g., some estimated final demands for products of some sectors and some projected total outputs the balance of sectors in the economy.

Problem 14.8 Overview

To illustrate the mixed-modeling process, we use a highly aggregated industry by industry, industry technology-based input-output model for the 2005 U.S. economy specified as a technical coefficients matrix, **A**, and make matrix, **V**, given for 7 industries: (1) agriculture, (2) mining, (3) construction, (4) manufacturing, (5) trade, transportation, and utility services, (6) services, and (7) other industries. We first compute the baseline vector of total outputs as

$$\mathbf{x} = \mathbf{V}\mathbf{i} = \begin{bmatrix} 312{,}754 \\ 396{,}563 \\ 1{,}302{,}388 \\ 4{,}485{,}529 \\ 3{,}355{,}944 \\ 10{,}477{,}640 \\ 2{,}526{,}325 \end{bmatrix}$$ and vector of total final demands as $$\mathbf{f} = \mathbf{x} - \mathbf{A}\mathbf{x} = \begin{bmatrix} 47{,}244 \\ -118{,}692 \\ 1{,}150{,}094 \\ 1{,}574{,}473 \\ 2{,}026{,}508 \\ 5{,}697{,}200 \\ 2{,}079{,}011 \end{bmatrix}$$. Note that the negative final demand for mining indicates net importation of products such as petroleum.

A	1	2	3	4	5	6	7
1	0.2258	0.0000	0.0015	0.0384	0.0001	0.0017	0.0007
2	0.0027	0.1432	0.0075	0.0675	0.0367	0.0004	0.0070
3	0.0051	0.0002	0.0010	0.0018	0.0037	0.0071	0.0215
4	0.1955	0.0877	0.2591	0.3222	0.0547	0.0566	0.1010
5	0.0819	0.0422	0.1011	0.0994	0.0704	0.0334	0.0487
6	0.0843	0.1276	0.1225	0.1172	0.1760	0.2783	0.2026
7	0.0099	0.0095	0.0093	0.0219	0.0215	0.0188	0.0240

V	1	2	3	4	5	6	7
1	310,868	0	0	65	0	1,821	0
2	0	373,811	0	22,752	0	0	0
3	0	0	1,302,388	0	0	0	0
4	0	0	0	4,454,957	0	26,106	4,467
5	0	808	0	0	3,354,043	47	1,046
6	0	556	0	0	152	10,473,161	3,771
7	4,657	1,410	0	4,111	115,428	339,582	2,061,136

Suppose our economic forecast projects, for 2010, a 10 percent growth in final demand for agriculture, mining, and construction, a 5 percent growth in final demand for manufactured goods, and a 6 percent growth in total output for the trade, transportation, utilities, services and other industries. So, the vector of exogenously specified data is $$\begin{bmatrix} \overline{\mathbf{f}} \\ \hdashline \overline{\mathbf{x}} \end{bmatrix} = \begin{bmatrix} 51{,}968 \\ -130{,}561 \\ 1{,}265{,}103 \\ 1{,}653{,}196 \\ \hdashline 3{,}557{,}300 \\ 11{,}106{,}299 \\ 2{,}677{,}904 \end{bmatrix}$$. The sectors are already conveniently ordered such that the four sectors with exogenously specified final demands are listed first and the remaining three with exogenously specified total outputs follow, so we can compute

$$\mathbf{M}=\left[\begin{array}{c:c}(\mathbf{I}-\mathbf{A}_{11}) & \mathbf{0}\\ \hdashline -\mathbf{A}_{21} & -\mathbf{I}\end{array}\right]=\left[\begin{array}{cccc:ccc}0.7742 & 0.0000 & -0.0015 & -0.0384 & 0 & 0 & 0\\ -0.0027 & 0.8568 & -0.0075 & -0.0675 & 0 & 0 & 0\\ -0.0051 & -0.0002 & 0.9990 & -0.0018 & 0 & 0 & 0\\ -0.1955 & -0.0877 & -0.2591 & 0.6778 & 0 & 0 & 0\\ \hdashline -0.0819 & -0.0422 & -0.1011 & -0.0994 & -1 & 0 & 0\\ -0.0843 & -0.1276 & -0.1225 & -0.1172 & 0 & -1 & 0\\ -0.0099 & -0.0095 & -0.0093 & -0.0219 & 0 & 0 & -1\end{array}\right] \text{ and}$$

$$\mathbf{N}=\left[\begin{array}{c:c}\mathbf{I} & \mathbf{A}_{12}\\ \hdashline \mathbf{0} & -(\mathbf{I}-\mathbf{A}_{22})\end{array}\right]=\left[\begin{array}{cccc:ccc}1 & 0 & 0 & 0 & 0.0001 & 0.0017 & 0.0007\\ 0 & 1 & 0 & 0 & 0.0367 & 0.0004 & 0.0070\\ 0 & 0 & 1 & 0 & 0.0037 & 0.0071 & 0.0215\\ 0 & 0 & 0 & 1 & 0.0547 & 0.0566 & 0.1010\\ \hdashline 0 & 0 & 0 & 0 & -0.9296 & 0.0334 & 0.0487\\ 0 & 0 & 0 & 0 & 0.1760 & -0.7217 & 0.2026\\ 0 & 0 & 0 & 0 & 0.0215 & 0.0188 & -0.9760\end{array}\right]\text{, which satisfies the}$$

condition $\mathbf{M}\begin{bmatrix}\mathbf{x}\\ \hdashline \mathbf{f}\end{bmatrix}=\mathbf{N}\begin{bmatrix}\bar{\mathbf{f}}\\ \hdashline \bar{\mathbf{x}}\end{bmatrix}$ or $\begin{bmatrix}\mathbf{x}\\ \hdashline \mathbf{f}\end{bmatrix}=\mathbf{M}^{-1}\mathbf{N}\begin{bmatrix}\bar{\mathbf{f}}\\ \hdashline \bar{\mathbf{x}}\end{bmatrix}$ where $\begin{bmatrix}\bar{\mathbf{f}}\\ \hdashline \bar{\mathbf{x}}\end{bmatrix}$ and $\begin{bmatrix}\mathbf{x}\\ \hdashline \mathbf{f}\end{bmatrix}$ are the vectors of exogenously specified values and the vector of endogenously determined values, respectively. To compute the endogenously determined values we first compute

$$\mathbf{M}^{-1}=\left[\begin{array}{cccc:ccc}1.3109 & 0.0077 & 0.0215 & 0.0752 & 0 & 0 & 0\\ 0.0345 & 1.1794 & 0.0398 & 0.1195 & 0 & 0 & 0\\ 0.0074 & 0.0005 & 1.0018 & 0.0031 & 0 & 0 & 0\\ 0.3855 & 0.1551 & 0.3943 & 1.5138 & 0 & 0 & 0\\ \hdashline -0.1479 & -0.0659 & -0.1439 & -0.1619 & -1 & 0 & 0\\ -0.1610 & -0.1694 & -0.1759 & -0.1994 & 0 & -1 & 0\\ -0.0218 & -0.0147 & -0.0185 & -0.0351 & 0 & 0 & -1\end{array}\right]\text{ and then}$$

$$\mathbf{M}^{-1}\mathbf{N}=\left[\begin{array}{cccc:ccc}1.3109 & 0.0077 & 0.0215 & 0.0752 & 0.0046 & 0.0067 & 0.0090\\ 0.0345 & 1.1794 & 0.0398 & 0.1195 & 0.0500 & 0.0076 & 0.0212\\ 0.0074 & 0.0005 & 1.0018 & 0.0031 & 0.0039 & 0.0073 & 0.0218\\ 0.3855 & 0.1551 & 0.3943 & 1.5138 & 0.0900 & 0.0893 & 0.1628\\ \hdashline -0.1479 & -0.0659 & -0.1439 & -0.1619 & 0.9178 & -0.0439 & -0.0687\\ -0.1610 & -0.1694 & -0.1759 & -0.1994 & -0.1938 & 0.7088 & -0.2279\\ -0.0218 & -0.0147 & -0.0185 & -0.0351 & -0.0241 & -0.0210 & 0.9719\end{array}\right]\text{ so that}$$

$$\begin{bmatrix}\mathbf{x}\\ \cdots\\ \mathbf{f}\end{bmatrix} = \mathbf{M}^{-1}\mathbf{N}\begin{bmatrix}\bar{\mathbf{f}}\\ \cdots\\ \bar{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}333,767\\ 414,773\\ 1,426,583\\ 4,748,959\\ \hdashline 2,144,061\\ 6,034,580\\ 2,203,480\end{bmatrix} \text{ for } \begin{bmatrix}\bar{\mathbf{f}}\\ \cdots\\ \bar{\mathbf{x}}\end{bmatrix} = \begin{bmatrix}51,968\\ -130,561\\ 1,265,103\\ 1,653,196\\ \hdashline 3,557,300\\ 11,106,299\\ 2,677,904\end{bmatrix}\text{, specified above.}$$

Computational Notes

We presume the specified make matrix **V** and the matrix of technical coefficients **A** are defined in the APL workspace and we compute the vector of total outputs **x** and the vector of final demands **f**. We compute the forecasted levels of final demand leaving unknown levels with a value of zero in a vector `f2` and the forecasted levels of total outputs leaving unknown levels with a value of zero in a vector `x2`. From `f2` and `x2` we assemble the matrix `YX` (the format expected by the function `MIXED`, developed earlier) with the column sums defining the vector of exogenous inputs `yx` and compute the endogenous vector of total outputs and final demands in the vector `xy`.

```
      x←+/V ⋄ f←x-A+.×x
      f2←1.1 1.1 1.1 1.05 0 0 0×f
      x2←0 0 0 0 1.06 1.06 1.06×x
      yx←+⌿YX←2 7⍴f2,x2
      xy←A MIXED YX
```

```
x
     312754      396563     1302388     4485529     3355944    10477640     2526325
f
      47244     ¯118692     1150094     1574473     2026508     5697200     2079011


f2
    51968.2   ¯130561.0   1265103.2   1653196.3         0.0         0.0         0.0
x2
        0.0         0.0         0.0         0.0   3557300.3  11106298.5   2677904.4
YX
    51968.2   ¯130561.0   1265103.2   1653196.3         0.0         0.0         0.0
        0.0         0.0         0.0         0.0   3557300.3  11106298.5   2677904.4
yx
    51968.2   ¯130561.0   1265103.2   1653196.3   3557300.3  11106298.5   2677904.4
xy
   333767.0    414773.5   1426582.7   4748959.3   2144060.8   6034580.2   2203479.9
```

Chapter 15, Selected Larger Scale Illustrations

In this section we use the APL functions developed so far supplemented with several more to work with larger scale examples of selected problems explored in previous chapters.

15.1 National Carbon Footprints Using the Larger Scale Databases

In Problem 13.12 we used a global IRIO transactions tables aggregated to 3 regions (the US, China, and Rest of World) and 3-Sector industry sectors (Agriculture and Mining, Manufacturing, and Services & Utilities) for the years 2005 and 2015 given in Appendix SD2 of the text. The highly aggregated version used in that problem is actually a spatial and sectoral aggregation of the OECD Inter-Country Input-Output (ICIO) Tables for those two years (https://www.oecd.org/sti/ind/inter-country-input-output-tables.htm).[7] These tables are available in several different formats, but one convenient for downloading to work in APL is the so-called CSV format. CSV which is the acronym for Comma Separated Values which is widely-used data exchange format that is simply a file of values using a comma (or other designated character) to separate values. The file downloaded in this format can then be saved as a normal Excel spreadsheet.

The OECD data can be downloaded via Microsoft Excel and easily converted to a Dyalog APL workspace with a user-defined APL function shown below, `XLFROM`, that utilizes a utility function provided by Dyalog APL for interfacing with Excel named `LoadXL`, which is provided in one of the library workspaces provided with Dyalog APL, `loaddata.dws`.

```
[  0]  R←mn XLFROM Range
[  1] ⍝ Retrieve specified Range (assumed numeric array of shape mn)
[  2] ⍝ from open Excel spreadsheet (format: 'a1:c2'
[  3] ⍝ Convert to APL numeric array and provide as explict result R
[  4]  ⍎'R←LoadXL ⍬ ⍬ ''',Range,''''
[  5]  R←mn⍴⍎,⍕R
```

There is a new character in this function, `⍬` , which is simply a keyboard character for an empty vector. The function `LoadXL` provides for a variety of additional features as well, but for present purposes it is important that the format of the specified Excel range of cells be all *numeric* (without additional formatting, e.g., comma delimiters for thousands).

Depending upon the computing resources available, especially the system memory, it may be convenient to download this large array (2484 x 2484 elements for the transactions matrix) in blocks, as in the function, `READDATA`, specified below.

[7] The OCED IRIO tables are configured as 38 OECD member nations (but including Mexico split out as two regions [domestic production and global manufacturing] and 31 non-OECD regions. China is also split out as two regions [domestic production and global manufacturing] and there is a rest-of-world (ROW) region, each of which are provided for 36 industry sectors, so overall there are 69×36=2,484 interregional sectors.

```
[  0]  m READDATA n;IDX;mn;i;j;D;R;ul;lr
[  1] ⍝ Read m regions by n sectors from Excel (block by block)
[  2] ⍝ m is no. of regions; n number of industry sectors
[  3] ⍝ CreateIDX creates Excel col range label from col number
[  4] ⍝ Excel must be open to sheet with array to be read at 'A1'
[  5] ⍝ Result in Global T
[  6]  mn←m×n ⋄ i←j←1 ⋄ T←(2⍴mn)⍴0
[  7]  IDX←CreateIDX mn
[  8] ⍝ Read blocks by rows
[  9] L:ul←(ul≠' ')/ul←IDX[j;],⍕i
[ 10]  lr←(lr≠' ')/lr←IDX[j+n-1;],⍕i+n-1
[ 11]  ⍎'D←LoadXL ⍬ ⍬ ',''''',ul,':',lr,''''
[ 12]  R←(n,n)⍴⍎,(⍕D),' '
[ 13]  T[(i-1)+⍳n;(j-1)+⍳n]←R
[ 14]  →(mn≥j←j+n)/L
[ 15]  j←1
[ 16]  →(mn≥i←i+n)/L
```

The function `CreateIDX` converts column index numbers (1,2, …) to the corresponding Microsoft Excel column code format (A, B, … ,Z, AA, BB, …).

```
[  0]  R←CreateIDX N;x;xx;n;A;B;C
[  1] ⍝Excel column indices from number indices up to number N (≤18,278))
[  2]  n←⍴x←'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[  3]  xx←(n,n)⍴x
[  4]  A←(3,¯1↑⍴A)⍴(3×¯1↑⍴A)↑A←(1,n)⍴x
[  5]  B←(3,¯1↑⍴B)⍴(3×¯1↑⍴B)↑B←(2,n×n)⍴(,⍉xx),,xx
[  6]  C←(3,(n×n×n))⍴(,⍉((n×n),n)⍴x),(,(⍉(n×n),n)⍴,⍉xx),,(n×n×n)⍴x
[  7]  R←(N,3)↑⍉A,B,C
```

Utilizing `READDATA`, we now have the full global IRIO transactions table (2484 × 2484) as a global variable T in the active workspace. We can also use `XLFROM` to download the corresponding vector of total outputs `XC`. With sufficient computer memory available, we can now use the APL functions developed throughout this volume to compute the matrix of technical coefficients `AR` and the corresponding Leontief inverse `LR`.

```
LR←INV AR←T AMAT XC
```

Let us now aggregate the global IRIO model by region and industry sector to focus on carbon-emitting industry for three regions: the US, China, and the Rest-of-World. These are the same three regions defined in Problem 13.12 but we retain more sectoral detail (15 sectors) for this illustration. We can use the APL function, `SCREATE`, defined earlier to create the sectoral aggregation matrix `SI` from the aggregation code `SICODE`, and to create the spatial aggregation matrix `SR` from the aggregation code `SRCODE`:

```
      SRCODE
36
42 68 69
(⍳35),(36+⍳5),(42+⍳25)
      SICODE
1
```

```
2 3 4
5
6 7 8
9
10
11 12
13 14
15 16 17
18 19 20
21
22
23 24
25 26 27 28 29 30 31 32 33 34 35
36
```

In the following APL function, IRIOAGG, we call on SCREATE to create the relevant aggregation matrices SR and SI and utilize three additional functions, TO3D, AGGZ, and TO3D, to aggregate the 69 region, 36 sector table (2484 × 2484) to 3 regions, 15 sectors (45 × 45).

```
 Z←MNmn IRIOAGG T;I;O;sx;k;ZZZ;ZZT;ZZ;nm;n;m;k;M;N;NM;Mn;SR;SI;S
⍝ Aggregate (MN×MN) IRIO Transactions to mn×mn
⍝ --M=# of unagg regs; N=# and unagg industry secs;
⍝ --m and n=# of agg regs and secs
⍝ --T is unaggregated (MN×MN) IRIO transactions matrix
⍝ --SICODE=ind agg code (text mtx), SRCODE-= reg agg code (txt mtx)
⍝ --Aggregations matrices: SI, SR, S
⍝ --calls functions SCREATE, AGGZ
 M←MNmn[1] ⋄ N←MNmn[2] ⋄ m←MNmn[3] ⋄ n←MNmn[4]
 nm←n×m ⋄ NM←N×M ⋄ Mn←M×n
⍝ AGGREGATE BY INDUSTRY
⍝ reshape T to 3D array -- (MN×N×N) to use aggregation by sheet (AGGZ)
 ZZZ←(M,N)TO3D T
 ZZT←(SI←SCREATE SICODE)AGGZ ZZZ
⍝ RESHAPE to IRIO FORMAT
⍝ZZ is (MN×n×n)--reshape to (M×n) x (M×n)IRIO format
 ZZ←TO2D ZZT
⍝ AGGREGATE Sector-Aggregated ZZ BY REGION k=1,2...m
⍝--SRCODE is region agg code (text mtx)
 O←0×I←(n,n)⍴1,n⍴0 ⋄ S←(0,Mn)⍴0 ⋄ k←1
⍝Build Aggregation Matrix S
 SR←SCREATE SRCODE
L:sx←M⍴'O'
 ⍎'sx[',SRCODE[k;],']←''I'''
 sx←¯1↓,⍉(2,M)⍴sx,M⍴','
 ⍎'S←S,[1]',sx
 →(m≥k←k+1)/L
⍝Aggregate ZZ using aggregation matrix S
 Z←S+.×ZZ+.×⍉S
```

```
 M←mn TO3D R;n;m;mn;i;j;k;nm;r
⍝ Create 3d array of sheets from IRIO format (sequence blocks by rows)
⍝ r=no. of sheets; m=no of row and col blocks; n=no. of sectors
 m←mn[1] ⋄ n←mn[2] ⋄ nm←1↑⍴R ⋄ r←m×m ⋄ M←(r,2⍴n)⍴0 ⋄ i←j←1 ⋄ k←0
```

```
L1:M[k←k+1;;]←R[(i-1)+⍳n;(j-1)+⍳n]
 →(nm≥j←j+n)/L1
 j←1
 →(nm≥i←i+n)/L1
```

```
 ZZM←S AGGZ ZZQ;i;n
⍝AGGREGATE ROWS AND COLS OF SHEETS IN 3D ARRAY
 ZZM←((n←1↑⍴ZZQ),2⍴1↑⍴S)⍴0×i←1
L1:ZZM[i;;]←S+.×ZZQ[i;;]+.×⍉S
 →(n≥i←i+1)/L1
```

```
 R←TO2D M;n;m;r;mn;i;j;k
⍝ 3d array of sheets to IRIO format
⍝ r=no. of sheets; m=no of row and col blocks; n=no. of sectors
 r←1↑⍴M ⋄ m←r*0.5 ⋄ n←¯1↑⍴M ⋄ mn←m×n ⋄ R←(2⍴mn)⍴0 ⋄ i←j←1 ⋄ k←0
L1:R[(i-1)+⍳n;(j-1)+⍳n]←M[k←k+1;;]
 →(mn≥j←j+n)/L1
 j←1
 →(mn≥i←i+n)/L1
```

Finally, we also need a function similar to `IRIOAGG`, defined as `IRIOAGGX` below, to aggregate the vector of total outputs to the same regional and sectoral aggregation as the matix of interindustry transactions.

```
 x←MNmn IRIOAGGX X;I;O;sx;k;XX;XXX;nm;n;m;k;M;N;NM;Mn;S;SI
⍝ Aggregate (MN×MN) IRIO Total Outputs to mn×mn
⍝ --M and N=no. of unagg regs and secs; m and n=no. of agg regs and secs
⍝ --X is unaggregated (⍴=MN) IRIO total outputs vector
⍝ --Global SICODE, SRCODE, SI, SR, S; call functions SCREATE
 M←MNmn[1] ⋄ N←MNmn[2] ⋄ m←MNmn[3] ⋄ n←MNmn[4]
 nm←n×m ⋄ NM←N×M ⋄ Mn←M×n
⍝ AGGREGATE BY INDUSTRY
⍝ --SICODEis industry agg code (text mtx)
⍝ reshape X to N×M table, aggregate and then reshape back to vector
 XX←⍉(M,N)⍴X
 SI←SCREATE SICODE
 XXX←,⍉SI+.×XX
⍝ AGGREGATE Sector-Aggregated XXX BY REGION k=1,2...m
⍝--SRCODE is region agg code (text mtx)
 O←0×I←(n,n)⍴1,n⍴0 ⋄ S←(0,Mn)⍴0 ⋄ k←1
⍝Build Aggregation Matrix S
L:sx←M⍴'O'
 ⍎'sx[',SRCODE[k;],']←''I'''
 sx←¯1↓,⍉(2,M)⍴sx,M⍴','
 ⍎'S←S,[1]',sx
 →(m≥k←k+1)/L
⍝Aggregate ZZ using aggregation matrix S
 x←S+.×XXX
```

We presume that the matrices of interindustry transactions for 2005 and 2015 are defined as global variables `Z1` and `Z2` in the APL workspace along with the corresponding vectors of total outputs `x1` and `x2`.

Finally, we now define the carbon-dioxide intensity coefficients. As an illustrative proxy we use the fraction of total global emissions by industry aggregated for 2005 and 20015 (OECD, 2021):

```
      e2005
0.021 0.029 0.012 0.014 0.031 0.029 0.059 0.076 0.006 0.02 0.381 0.014 0.112 0.046 0.15

      e2015
0.02 0.031 0.01 0.01 0.028 0.032 0.06 0.097 0.005 0.02 0.385 0.015 0.11 0.04 0.138
```

As an illustration, if we apply the fractional shares to the global total industrial emissions for 2005 and 2015, we obtain the vectors of total global emissions by industry, `e1` and `e2`, which, in turn if divided by total global outputs by industry, `X1` and `X2`, we generate vectors of emissions per dollar of industry output, `g1` and `g2`.

```
      X1←+/⍉3 15⍴x1 ⋄ X2←+/⍉3 15⍴x2
      e1←27070×e2005 ⋄ e2←32277×e2015
      g1←e1÷X1 ⋄ g2←e2÷X2
```

We now have all we need to compute the values of total embodied emissions attributed to production, `ep1` and `ep2`, and attributed to consumption, `ec1` and `ec2`:

```
      f1←x1-+/Z1  ⋄ f2←x2-+/Z2
      L1←INV A1←Z1 AMAT x1 ⋄ L2←INV A2←Z2 AMAT x2
      GP1←(DIAG 45⍴g1)+.×L1+.×DIAG f1 ⋄ GP2←(DIAG 45⍴g2)+.×L2+.×DIAG f2
      ep1←+/GP1 ⋄ ep2←+/GP2 ⋄ ec1←+⌿GP1 ⋄ ec2←+⌿GP2
```

```
      7 0⍕3 15⍴ep1
  67    112     59     67    196    179    269    316     27    124   1848     78    699    434    459
  77     60     25     52     53     76    224    267     25     37    865     24    120     32      0
 425    612    241    260    590    530   1104   1474    111    381   7601    277   2212    779   3601
      7 0⍕3 15⍴ec1
  47     10    233     70    129    178     69     59    175    366    730    560    636   1669    459
  66      5     95     98      5     63     20     46    313    176    130    423    106    311      0
 441     72    844    401    374    467    245    361   1016   1213   3240   2028   2026   3492   3601
      7 0⍕3 15⍴ep2
1011   2406   1704    989   2828   9477   1881   8483    922   9285 100054  10257 151915 109471    415
3865   4350   2527   3616   3437  16991   6400  31135   3394  10106 162017  18357 100847  34113      0
8509  18786   7546   5242  11356  30929   8916  41116   4328  29635 602670  47237 486090 196849   3703
      7 0⍕3 15⍴ec2
 2355    376  14184   3518   3653  10994   1993   2360   5510  29424  49228  29870 101348 173985    415
 9984    523  16861  11428    999  12714   2076   7560  54848  35945  27815  99616  41340  70386      0
30966   5570  67140  28419  17779  34884   9370  17023  57024  82144 253913 151885 316241 417793   3703
```

Perhaps of more interest are these results aggregated by region, i.e., summing all the emissions in each region as `EP1` or `EC1` for 2005 and `EP1`or `EC2` for 2015 in each case for production or consumption, respectively. Note that, as expected, the sum of total emissions for all regions across for each year are the same for each year, found by summing the elements of `EP1` or `EC1` for 2005 and `EP2` or `EC2` for 2015.

```
      8 0⍕, (+/EP1),EP1←+/3 15⍴ep1
   27070    4934    1936   20200
      8 0⍕, (+/EC1),EC1←+/3 15⍴ec1
   27070    5390    1858   19822
```

```
      8 0⍕, (+/EP2),EP2←+/3 15⍴ep2
 2315162  411097  401154 1502912
      8 0⍕, (+/EC2),EC2←+/3 15⍴ec2
 2315162  429213  392095 1493854
```

15.2 The U.S. National Input-Output Model

The U.S. National Input-Output Tables are available from the Bureau of Economic Analysis of the U.S. Department of Commerce at a variety of sectoral aggregations up to 405 industry sectors. The tables are presented in the form of supply and use tables for benchmark years and are available as Microsoft Excel spreadsheets (https://www.bea.gov/data/economic-accounts/industry). We can download and open the 2012 405-sector Supply and Use table spreadsheets and use the APL function XLFROM defined earlier to define **S** and **U** in the APL workspace. With the Supply spreadsheet open, it is helpful to delete the row and column headings so that the (1,1) entry of the Supply table is located at A1 in the spreadsheet and remove the formatting from the cells to be downloaded.

```
      S←405 405 XLFROM 'A1:OO405'
```

With the Use spreadsheet open, again, it is helpful to delete the row and column headings so that the (1,1) entry of the Use table is located at A1 in the spreadsheet and remove the formatting from the cells to be downloaded.

```
      U←405 405 XLFROM 'A1:OO405'
```

We can now assemble any of the commodity-by-industry model configurations, such as an industry-by-industry based model under the assumption of industry-based technology defined by $\mathbf{A} = \mathbf{DB}$ and $\mathbf{L} = (\mathbf{I} - \mathbf{A})^{-1}$ where $\mathbf{D} = \mathbf{V}\hat{\mathbf{q}}^{-1}$ with $\mathbf{q} = (\mathbf{V}')\mathbf{i}$ and $\mathbf{B} = \mathbf{U}\hat{\mathbf{x}}^{-1}$ with $\mathbf{x} = \mathbf{Vi}$, recalling that $\mathbf{S} = \mathbf{V}'$ (the supply matrix **S** is the transpose of the make matrix **V**). In APL terms, this is

```
      V←⍉S ⋄ D←V+.×DIAG ÷q←+⌿V ⋄ B←U+.×DIAG ÷x←+/V
      L←INV A←D+.×B
```

15.3 RAS Estimation Regional Tables from National Data

In Problem 10.10 we explored RAS estimation of an input-output table for the state of Washington using the U.S. national input-output table and estimates of regional intermediate inputs and outputs for Washington at a level of 7 industry sectors. In the following we use the same technique, but we retain the survey-based 2012 Washington State 52-sector industry aggregation (https://ofm.wa.gov/washington-data-research/economy-and-labor-force/washington-input-output-model/2012-washington-input-output-model) and aggregate the 405-sector U.S. 2012 transactions table to that level of aggregation before testing RAS performance.

We start with a concordance table **C**, which in this case is a two-column table, the first column of for each row specifies the industry index of the Washington table corresponding to the US table industry index listed in the second column. Hence **C** is of shape 405 × 2 and we can use it in the APL function SCREATEC (listed below) to generate the aggregation matrix necessary for aggregating the national table to the regional table's level of industry aggregation.

```
 S←SCREATEC C;n;m;k
⍝--Create Aggregation mtx S
⍝--from concordance matrix C
⍝--C[;1]=agg idx; C[;2]=unagg idx
 n←1↑⍴C ⋄ m←⌈/C[;1]
 S←(m,n)⍴0 ⋄ k←1
L1:S[C[k;1];k]←1
 →(n≥k←k+1)/L1
```

We retrieve **C** and apply SCREATEC to generate the relevant aggregation matrix.

```
      S←SCREATEC C←405 2 XLFROM 'A7:b411'
```

We presume the Washington State tables for 2007 and 2012 (downloaded from the website noted above) are resident in the APL Workspace as global variables Z1 and Z2, respectively, along with their corresponding vectors of total outputs, x1 and x2.

Using the aggregation matrix **S**, we can create the aggregated national matrix of interindustry transactions ZN and aggregated vector of total outputs XN to compute the corresponding matrix of technical coefficients AN that we will use to estimate the regional table by RAS.

```
      ZN←S+.×ZZN+.×⍉S
      XN←S+.×XXN
      AN←ZN AMAT XN
```

Finally, we generate two RAS estimates. The first is updating the 2007 Washington State table to 2012 with RAS using the vectors of intermediate outputs, intermediate inputs, and total outputs from the survey-based 2012 table. The second is estimating the 2012 Washington State table from the 2012 US national table, using the vectors of intermediate outputs, intermediate inputs, and total outputs from the survey-based 2012 table. First, we compute the matrix of technical coefficients for the 2007 Washington State table A1 and define the interindustry outputs and inputs for the 2012 Washington State table as u2 and v2, respectively.

```
      A1←Z1 AMAT x1
      u2←+/Z2 ⋄ v2←+⌿Z2
```

We can now compute the RAS estimates of A2 from A1 and AN, which we define as AE1 and AE2, respectively, and compute the mean absolute deviation of each compared with A2.

```
      A2 MAD AE1←A1 RAS 3 52⍴u2, v2, x2
0.000004684813586
      A2 MAD AE2←AN RAS 3 52⍴u2, v2, x2
0.005583551083
```

As might easily be predicted, the updating of the 2007 regional table is a far better estimate of the 2012 state table than is modifying the national table.

15.4 Generalized Input-Output Analysis and Goal Programming

In Chapter 13, we illustrated a goal programming (GP) solution for the planning form of the generalized input-output formulation. Problem 13.9 illustrated a small-scale problem that could be solved graphically as well as analytically by formulating the problem as a linear programming problem using the APL function LINPROG. Now we revisit the slightly larger scale planning problem posed in Problem 13.3, but reconfigure it as a GP problem. Recall that the matrix of technical coefficients, $\mathbf{A} = \begin{bmatrix} 0.04 & 0.23 & 0.38 \\ 0.33 & 0.52 & 0.47 \\ 0 & 0 & 0.1 \end{bmatrix}$, and the direct impact coefficients matrix as, $\mathbf{D} = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & .5 \\ .73 & .33 & .63 \end{bmatrix}$ with the environmental, energy, and employment coefficients as the three rows respectively. We also assemble $\Delta\mathbf{F} = \begin{bmatrix} 2 & 4 & 2 & 2 \\ 2 & 0 & 0 & 2 \\ 2 & 2 & 4 & 3 \end{bmatrix}$, the table of prospective project expenditures, and compute the total economic requirements matrix, $\mathbf{L} = \begin{bmatrix} 1.247 & .598 & .839 \\ .857 & 2.494 & 1.665 \\ 0 & 0 & 1.111 \end{bmatrix}$.

In Problem 13.3 we determined that Project 4 among the four projects depicted in $\Delta\mathbf{F}$ contributed the most to GRP, which can be easily seen as the column sums of $\Delta\mathbf{F}$. Consequently (not surprisingly) Project 4 also consumed the most energy and contributed the most to employment. We configure the planning form as

$$\tilde{\mathbf{x}} = \mathbf{G}\mathbf{x} = \begin{bmatrix} \mathbf{D} \\ \hdashline (\mathbf{I}-\mathbf{A}) \end{bmatrix}\mathbf{x} = \begin{bmatrix} 4.2 & 7 & 9.1 \\ 7.6 & 2.6 & 0.5 \\ 0.73 & 0.33 & 0.63 \\ \hdashline 0.96 & -0.23 & -0.38 \\ -0.33 & 0.48 & -0.47 \\ 0 & 0 & 0.9 \end{bmatrix}\begin{bmatrix} 6.21 \\ 11.7 \\ 3.33 \end{bmatrix} = \begin{bmatrix} 138.27 \\ 79.24 \\ 10.49 \\ \hdashline 2.0 \\ 2.0 \\ 3.0 \end{bmatrix} = \begin{bmatrix} \mathbf{x}^* \\ \hdashline \mathbf{f} \end{bmatrix}.$$

Suppose we now consider imposing an environmental emissions constraint (specified by the first row of coefficients of **D**) for total emissions less than 120. This would limit project 3 and 4 since emissions for those projects pursued independently would be than 123.62 and138.27, respectively. If we want to maximize employment (specified by the third row of coefficients of **D**) but we have set the emissions constraint at 120, we can pose this as a GP problem as the following:

$$Min\ \ P_1 d_1^+ + P_2(d_4^- + d_5^- + d_6^-) + P_3 d_3^-$$

$$\begin{bmatrix} \mathbf{G} & -\mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{d}^+ \\ \mathbf{d}^- \end{bmatrix} = \tilde{\mathbf{x}} \text{ where } \mathbf{d}^+ = \begin{bmatrix} d_1^+ & d_2^+ & d_3^+ & d_4^+ & d_5^+ & d_6^+ \end{bmatrix}' \text{ and}$$

$\mathbf{d}^- = \begin{bmatrix} d_1^- & d_2^- & d_3^- & d_4^- & d_5^- & d_6^- \end{bmatrix}'$ are the vectors of positive and negative deviational variables, respectively, or

$$Min \;\; P_1 d_1^+ + P_2(d_4^- + d_5^- + d_6^-) + P_3 d_3^-$$

$$\left[\begin{array}{ccc|cccccc|cccccc} 4.2 & 7 & 9.1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 7.6 & 2.6 & 0.5 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0.73 & 0.33 & 0.63 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0.96 & -0.23 & -0.38 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -0.33 & 0.48 & -0.47 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \hline d_1^+ \\ d_2^+ \\ d_3^+ \\ d_4^+ \\ d_5^+ \\ d_6^+ \\ \hline d_1^- \\ d_2^- \\ d_3^- \\ d_4^- \\ d_5^- \\ d_6^- \end{bmatrix} = \begin{bmatrix} 120 \\ 79.24 \\ 10.49 \\ \hline 2.0 \\ 2.0 \\ 3.0 \end{bmatrix}.$$

The optimal solution is $\mathbf{x}^* = \begin{bmatrix} 5.6 & 10.49 & 2.53 \end{bmatrix}'$ with impacts of $\mathbf{Dx}^* = \begin{bmatrix} 120 & 71.1 & 9.14 \end{bmatrix}'$ and a diminished GRP of $\mathbf{i}'(\mathbf{I}-\mathbf{A})\mathbf{x}^* = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 2.28 \end{bmatrix} = 6.28$ compared with the corresponding values for Project 4 alone of is $\mathbf{x}^4 = \begin{bmatrix} 6.2 & 11.7 & 3.33 \end{bmatrix}'$ with impacts of $\mathbf{Dx}^4 = \begin{bmatrix} 138.27 & 79.24 & 10.49 \end{bmatrix}'$ and a diminished GRP of $\mathbf{i}'(\mathbf{I}-\mathbf{A})\mathbf{x}^4 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix} = 7.0$.

We can use the APL tools developed already to configure this goal programming problem as a linear programming problem by assigning pre-emptive priority levels to (1) limiting emissions (minimizing positive deviation from 120), (2) preserving interindustry relationships (minimizing negative deviation from achieving the target levels of final demand $\begin{bmatrix} 2 & 2 & 3 \end{bmatrix}'$, and (3) maximizing employment (minimizing negative deviation from the target of 10.49) with objective function weights of 1000, 100, and 100, respectively. The resulting tableau for using `LINPROG` is:

```
      MGP
 0     0     0     1000  0  0  0  0  0 0 0 10 100 100 100 ¯1   0
 4.2   7     9.1    ¯1   0  0  0  0  0 1 0  0   0   0   0  0 120
 7.6   2.6   0.5     0 ¯1  0  0  0  0 0 1  0   0   0   0  0  79.24
 0.73  0.33  0.63    0  0 ¯1  0  0  0 0 0  1   0   0   0  0  10.49
 0.96 ¯0.23 ¯0.38    0  0  0 ¯1  0  0 0 0  0   1   0   0  0   2
¯0.33  0.48 ¯0.47    0  0  0  0 ¯1  0 0 0  0   0   1   0  0   2
 0     0     0.9     0  0  0  0  0 ¯1 0 0  0   0   0   1  0   3
```

Using LINPROG, the result is r found by

```
      r←LINPROG MGP
      r
0 85.714004 5.5991113 10.493795 2.5304576 0 0 0 0 0 0 0 8.1376571 1.3455079 0 0 0.72258814
```

Recall the first value is the code indicating that status of the solution (0 indicating an optimal solution), the second values is the final value of the objective function, and the next three elements (in this case) are the values of $\mathbf{x}^*$, the goal programming solution

```
      r[3 4 5]
5.5991113 10.493795 2.5304576
```

The next six values comprise the values of the negative deviational variables, all of which are 0 indicating no under achievement of the targets, and the final six values are the values of the positive deviational variables:

```
      ¯6↑r
0 8.1376571 1.3455079 0 0 0.72258814
```

This shows that the highest priority of limiting emissions was tight against its constraint value, the values of energy consumption and employment were short of their targets by 8.138 and 1.346, respectively, and of the values of total final demand (the last three values of the vector of results) sector three's final demand was short of its target by 0.723. We can verify this by computing the total outputs corresponding to the shortfalls in final demand, which should be the same as the optimal solution.

```
      L+.×2 2 3-¯3↑r
5.5991113 10.493795 2.5304576
```

15.5 The Contemporary Approach to Hypothetical Extraction

From Chapter 7, recall that the hypothetical extraction approach was originally conceived to quantify how much the total output of an *n*-sector economy would change (decrease) if a particular sector, say the *j*th, were completely removed from that economy. Problem 7.7 showed this approach, modeled in an input-output context by deleting row and column *j* from the **A** matrix. Recall that, for convenience, we adopted an alternative notation in which $\mathbf{A}^{-j}$ is defined as the $(n \times n)$ coefficients matrix in which row and column *j* have been nullified (replaced by zeros; sector *j*'s "removal"), with $\mathbf{L}^{-j}$ as its associated Leontief inverse and $\mathbf{f}^{-j}$ as the final demand vector with $f_j = 0$. Then the output vector for this reduced economy is given by $\mathbf{x}^{-j} = \mathbf{L}^{-j}\mathbf{f}^{-j}$ and, hence, the result in terms of reduction in total outputs is $\Delta_j = \mathbf{i}'(\mathbf{x} - \mathbf{x}^{-j})$ and the percentage decrease is $\overline{\Delta}_j = 100[(\mathbf{i}'\mathbf{x} - \mathbf{i}'\overline{\mathbf{x}}_{(j)})/\mathbf{i}'\mathbf{x}]$.

In Chapter 7 we showed Temursho's much simpler and more contemporary approach to computing the results for hypothetical extraction which eliminates many of the tedious calculations as the following: $\Delta_j = (\frac{1}{l_{jj}})\mathbf{i}'\mathbf{L}\mathbf{e}_j\mathbf{e}_j'\mathbf{x}$ where $\mathbf{e}_j$ is the *j*th column of the $(n \times n)$ identity matrix—all zeros except for a 1 in location *j*. Note that $\mathbf{e}_j\mathbf{e}_j'$ results in an $(n \times n)$ matrix of all 0's except for a 1 on the main diagonal at location *j*.

We can reconfigure Problem 7.7 to use Temursho's approach with the following dyadic APL Function `HEXTRACT` which takes a matrix of technical coefficients A as the left argument and the corresponding vector of total outputs x as the right argument. The function applies Temurhso's formula and returns, as the explicit result, a matrix where each column includes the index of the hypothetically extracted sector, the reduction of the total outputs, and the percentage reduction of total outputs.

```
[  0]  R←A HEXTRACT x;L;f;T;j;n;IJ;Aj;fj
[  1] ⍝ Temursho's Hypothetical Extraction
[  2]  L←LINV A ⋄ f←x-A+.×x ⋄ n←⍴x ⋄ T←n⍴0 ⋄ j←1
[  3] L1:Aj←A ⋄ fj←f ⋄ IJ←(2⍴n)⍴0
[  4]  fj[j]←0 ⋄ Aj[j;]←0 ⋄ Aj[;j]←0 ⋄ IJ[j;j]←1
[  5]  T[j]←(÷L[j;j])×+⌿L+.×IJ+.×x
[  6]  →(n≥j←j+1)/L1
[  7]  R←(3,n)⍴(⍳n),T,100×T÷+/x
```

Applying `HEXTRACT` to the data in Problem 7.7 we can reproduce the results much more efficiently as

```
          A HEXTRACT x
    1              2              3             4              5              6              7
547449.46      591737.58      2632898.3     6814684.2      5036387.3      12526318       4307032.5
    2.3950914      2.5888519      11.51893      29.814244      22.034195      54.802643      18.843267
```

We can illustrate the use of `HEXTRACT` with a larger set of input-output data. For example, we apply it to the 52-sector Washington State input-output table for 2012 used in Problem 15.3 (the technical coefficients matrix was defined as `A1` and the vector of total outputs as `x1`), so, in APL as with the earlier example,

```
      R1←A1 HEXTRACT x1
```

For convenience we can use another APL primitive function known as "grade down" (designated with the character ⍒) which is a monadic function for sorting data that returns the indices of a vector that would put the elements of that vector in descending order, as in

```
      q←3 6 8 5 2
      ⍒q
3 2 4 1 5
      q[⍒q]
8 6 5 3 2
```

As an aside, there is a companion primitive function “grade up” (designated with the character ⍋) which is a monadic function that returns the indices of a vector that would put the elements of that vector in ascending order.

We can use “grade down” to modify the result of HEXTRACT applied to the Washington State table to return the ten sectors with the highest values of hypothetically extracted total output in descending order (sector 10 is construction and sector 24 is aircraft manufacturing):

```
      R1←A1 HEXTRACT x1
      10 3↑⍉R1[;⍒R1[2;]]
10 90869.998 15.052685
24 44180.124  7.3184716
37 39871.259  6.6047047
40 38940.138  6.4504638
52 37815.12   6.2641038
29 37639.707  6.2350465
31 35825.854  5.9345803
16 30135.748  4.9920098
11 28672.03   4.7495438
46 26884.182  4.4533854
```

References

Input-Output Analysis: Foundations and Extensions

Input-Output Analysis Computational Workbook, Workbook to Accompany Miller, Ronald E. and Peter D. Blair. 2021. *Input-Output Analysis: Foundations and Extensions. Third Edition.* Cambridge, UK: Cambridge University Press. http://cambridge.com/millerandblair.

Miller, Ronald E. and Peter D. Blair. 1985. *Input-Output Analysis: Foundations and Extensions.* Englewood Cliffs, NJ: Prentice-Hall.

Miller, Ronald E. and Peter D. Blair. 2009. *Input-Output Analysis: Foundations and Extensions. Second Edition.* Cambridge, UK: Cambridge University Press.

Miller, Ronald E. and Peter D. Blair. 2021. *Input-Output Analysis: Foundations and Extensions. Third Edition.* Cambridge, UK: Cambridge University Press.

Selected General Scientific Software Suitable for Input-Output Analysis

GAUSS. 2021 (Version 21). Aptech Systems, Inc., Higley, AZ. https://store.aptech.com/gauss-platform-category.html.

IMSL Numerical Libraries. 2021. Perforce Software, Minneapolis, MN. https://www.imsl.com.

MAPLE. 2020. Maplesoft, Waterloo, ON, Canada. https://www.maplesoft.com/products/Maple.

MATHEMATICA. 2021. Wolfram, Champaign, IL. https://www.wolfram.com/mathematica.

MATLAB. 2021. Mathworks, Natick, MA. https://www.mathworks.com/products/matlab.htm.

SAS (statistical analysis system). 2021. SAS Institute, Inc., Raleigh, NC. https://www.sas.com/en_us/home.html.

SciLab. 2021. ESI Group, Rungis, France. https://www.scilab.org.

STATA (statistical software for data science). 2021 (Version 17). StataCorp, College Station, Texas. https://www.stata.com.

Selected Specialized Software for Input-Output Analysis

ICIO. Belotti, Federico, Alessandro Borin, Michele Mancini. 2021. "ICIO: Stata module for Economic Analysis with Inter-Country Input-Output tables," Statistical Software Components S458463, Boston College, Department of Economics. https://ideas.repec.org/c/boc/bocode/s458463.html.

IO-Snap (Input-Output - State and National Analysis Program). EconAlyze, Morgantown, WVA. https://www.io-snap.com.

IOT. 2015. Shehata, Emad Abd Elmessih and Sahra Khaleel A. Mickaiel. "IOT: Stata module to estimate Leontief Input-Output Table," Statistical Software Components S458024, Boston College, Department of Economics. https://ideas.repec.org/c/boc/bocode/s458024.html.

IRIOS (Interregional Input Output Software). 2000. Department of Economics and Econometrics, Faculty of Economics and Business, University of Groningen, Groningen, The Netherlands. https://www.rug.nl/research/reg/research/irios/?lang=en.

REAL I-O (generic toolbox of input-output analysis). 2020. Regional Economics Applications Laboratory (REAL), The University of Illinois at Urbana-Champaign. http://www.real.illinois.edu/realio; Python Module for Input-Output Analysis (PyIO), 2009. http://www.real.illinois.edu/pyio.

WIOD. Dietzenbacher, Erik, Bart Los, Robert Stehrer, Marcel Timmer, and Gaaitzen de Vries. 2013. "The Construction of the World Input-Output Tables in the WIOD Project, *Economic Systems Research*, **25**, 71-98. World Input-Output Database (WIOD). http://www.wiod.org/home.

APL, A Programming Language

APL2. 2021. Log-On Software (formerly offered by IBM), Bedford, NH. https://log-on.com/apl2-solution-page.

Iverson, Kenneth E. 1962. *A Programming Language*. New York: John Wiley & Sons. ISBN 978-0-471-43014-8. https://www.jsoftware.com/papers/APL.htm, Retrieved 15 July 2021.

Legrand, Bernard. 2009. *Mastering Dyalog APL: A Complete Introduction to Dyalog APL.* Alton, Hampsire, UK: Dyalog, Limited. ISBN: 978-0-9564638-0-7, www.dyalog.com.

Other References

Lahr, Michael L. and Benjamin H. Stevens. 2002. "A Study of the Role of Regionalization in the Generation of Aggregation Error in Regional Input-Output Models," *Journal of Regional Science,* **42**, 477–507.

Leontief, Wassily. 1951. "Input-Output Economics," *Scientific American*, **185**, 15–21.

Morgenstern, Oskar (ed.). 1954. *Economic Activity Analysis.* New York: Wiley.

Polenske, Karen R. 1980. *The U.S. Multiregional Input-Output Accounts and Model.* Lexington, MA: Lexington Books (D. C. Heath and Co.).

Yamano, Norihiko and Joaquim J. M. Guilhoto. 2020. *CO_2 Emissions Embodied in International Trade and Domestic Final Demand, Using the OECD Inter-Country Input-Output Database*, Organization for Economic Cooperation and Development (OECD), Directorate for Science, Technology and Innovation (STI). https://www.oecd.org/sti/ind/carbondioxideemissionsembodiedininternationaltrade.htm.

Appendix: Additional Sampling of Basic APL Features

Configuring Dyalog APL for IOA

For non-commercial use, the latest version of Dyalog APL, one of the most accessible versions of APL, can be downloaded from the corporate website, http://www.dyalog.com. The system is available for various operating systems (Windows, MacOS, Linux, and others). Upon selection of the relevant operating system, the relevant version is downloaded as a compressed zip file which, when decompressed, will include the application that installs the APL system, the font, and a number of supporting files and APL libraries. A full introduction to the Dyalog APL implementation of the language is provided in Legrand (2009) available at https://www.dyalog.com/mastering-dyalog-apl.htm.

APL Fonts

Many fonts are available for working in APL and APL fonts can be used in other applications as well, as in, for example, this workbook, in which most of the text is written in the Times New Roman 12-point font while APL expressions and results are written in the Dyalog APL Unicode Font. Some APL font is used by default in most modern APL systems.

The APL Keyboard

Interacting with an APL interpreter begins with a standard personal computer keyboard, but various key combinations enable access to the special characters designating primitive functions, operators and other APL-specific features.

The specific key combinations that produce APL characters vary somewhat with APL systems, but, as an example, in one of the most commonly used APL systems today (Dyalog APL), the key combination of CTRL (or CTRL+SHIFT) and other keys designate most APL characters, as illustrated in the keyboard map shown below as Figure A1. Note, in the map for each key, the *left lower character* is the standard lower case character; the *upper left character* is the standard upper case character; the *lower right character* (produced by depressing CTRL and that key simultaneously) are the most common APL characters; and the *upper right character* (produced by depressing CTRL, SHIFT, and that key simultaneously) are additional, usually less commonly-used APL characters if they are defined.

So, for example, depressing the upper right key to the left of the backspace key, produces the = character, depressing SHIFT plus that key simultaneously produces the + character, depressing the CTRL plus the that key simultaneously produces the ÷ character, and depressing CTRL plus SHIFT plus that key simultaneously produces the ⌹ character. Dyalog APL also provides a “ribbon” accompanying the empty workspace when initially loaded that can be used to generate APL characters instead of using a keyboard combination by moving the cursor to the relevant character on the ribbon and depressing the left mouse button.

Figure A1. The APL Keyboard. Source: Legrand (2009), p. 36.

System Commands, Functions, and Variables

There are a variety of ways in which APL users interact with the system environment from within APL. Mechanisms for controlling specific features of the environment in which APL functions execute, or to query information about this environment are possible through a collection of so-called *system interfaces* of which there are three types: (1) *system commands*, several of which have already been introduced, (2) system functions, and (3) system variables.

System Commands

A number of the most important system commands have already been introduced. A complete description of available commands for Dyalog APL is included in Legrand (2009), especially Appendix 8. Additional system commands worth noting here (others are identified later) are described in the following.

APL initially loads in a *clear* workspace, i.e., one in which not no global variables or user-defined functions (or other objects discussed later) are yet defined—the system commands, `)vars` and `)fns` , will return no entries.

`)CLEAR` will return APL to an empty workspace.

`)WSID` will return the name of the *active* workspace or if the active workspace is an empty workspace will return the message "`is CLEAR WS`". If an argument is provided then `)WSID` will change the name of the active workspace to the specified name, e.g., `)WSID WORK`, which change the name of the active workspace to `WORK`.

`)SAVE` will save the active workspace in the workspace library under the current name or under a new name if one is specified as an argument, e.g., `)SAVE WORK2`.

`)LIB` lists the names of saved workspaces.

`)LOAD` replaces the active workspace with a copy of one from the library specified by the argument, e.g., `)LOAD WORK2`.

`)ERASE` will delete specified objects, e.g., global variables or user-defined functions, from the active workspace.

`)COPY` copies all or selected objects from a saved workspace into the active workspace, e.g., `)COPY WORK2` copies all of the objects in the workspace `WORK2` into the active workspace or `)COPY WORK2 FUN1 FUN2 VAR1 VAR2` copies the functions `FUN1` and `FUN2` and the variables `VAR1` and `VAR2` from the workspace `WORK2` into the active workspace.

`)OFF` quits APL.

`)ED` invokes the built-in APL function editor for the function specified as the right argument, e.g., `)ED FUN1` will open the APL function editor in a separate window for the editing function `FUN1`. If `FUN1` is being created anew, it will open the editor to a new line `[0]` indicating only the name `FUN1`. If the specified function already exists the entire collection of statements comprising the function will be opened in the editor. The editor itself has many word-processing-like features. Exiting the editor saves the edited function under the name specified in line `[0]`.

`)SI` known as *state indicator* returns the names of all functions whose operations have been suspended due to an error or other condition and the line numbers where the function is suspended. When a function is suspended all objects are suspended, including, for example, all local variables within the suspended function at the time of the suspension are accessible.

`)RESET` resets the state indicator and removes suspensions on all variables and functions.

System Functions

The system commands just described can only be entered at the keyboard as a user interacts with the APL interpreter. One other important limitation of system commands is that when they require specifying a folder path and/or workspace name as an argument when the path or name contains blanks, the entire file specification must be enclosed between a pair of double quotes since, unless specified within double quotes, spaces are interpreted as delimiters between objects. So, for example, the command to load a workspace located at the file location, c:\my documents\November work2, would be written as `)LOAD “c:\my documents\November work2”` rather than `)LOAD c:\my documents\November work2`.

System functions provide many of the same features system commands and many more as well, but can be executed within an APL function. System functions begin with the character `⎕` known as *quad*. For example, `⎕LOAD, ⎕WSID, ⎕CLEAR, ⎕SAVE`, all accomplish the same as their system command counterparts, but their arguments are specified as character arrays. For example, `⎕LOAD ‘WORK2’` would load a copy of the saved APL library workspace WORK2. The system function `⎕CY` is the analog to `)COPY` but with the format `OBJECTS ⎕CY 'wsid'` where `OBJECTS` is a character array specifying a list of objects and `wsid` is the workspace containing those objects. Several other useful system functions, detailed in Legrand (2009) along with many others as well, include:

`⎕LX` known as *latent expression* executes an APL expression specified as the argument when the workspace is loaded, e.g., specifying `⎕LX←'FUN1'`, would execute the user-defined function FUN1 when the workspace is initially loaded.

`⎕EX` known as *expunge* erases the objects listed in a character array as the argument.

`⎕TS` known as *time stamp* chronicles the exact date and time.

⎕PP known as print precision specifies the maximum number of significant digits used to display numeric values when no particular format is specified; the argument is an integer between 1 and 17 with the default set to 10.

System Variables

Many characteristics of an APL session are stored in *system variables*. All system variables have default values that can be changes for specific needs. Some of the most commonly used system variables are the following.

`⎕IO,`known as *index origin,* specifies whether index sequences between with 0 or 1 (the default is 1). For example with `⎕IO←1`, the expression `⍳3` would yield the vector `1 2 3` but with `⎕IO←0` would yield the vector `0 1 2.` For the defined vector `R←1 2 3,` With `⎕IO←0`, the expression `R[0]` would yield `1` but with `⎕IO←1` the result would be an index error.

`⎕PW`, known as *page width*, specifies the display width used in the current APL workspace. The default is 200 characters, but can be re-specified by assigning a new value, e.g., `⎕PW←400.`

`⎕CT`, known as *comparison tolerance*, specifies the threshold for equality between two numbers, e.g., to compensate for inaccuracies due to the limited precision of numbers. That is APL considers two numbers to be equal if the difference between them is within a specified tolerance which in Dyalog APL can be any value between 0 and `16*¯8` (2.3283064E¯10).

`⎕RL`, known as *random link*, is the initial seed value for the random number generator built into APL. The default value for a clear workspace is 16807 (, i.e., `7*5`), but changes with any use of the random number generator. Hence, if an expression using pseudo-random values is executed immediately after APL has been started, the expression will yield the same result on each such occasion, which can be helpful for reproducing the same experimental conditions, but it can also be a disadvantage in some circumstances as well. The APL random number generator is invoked with the monadic or dyadic function *roll* designated with the character `?`, discussed in the next section.

There many other additional system interface features (see Legrand, 2009, for a complete collection and descriptions).

Additional Useful Functions and Operators

Primitive Functions and Operators

In this workbook we have used only a small number of the primitive functions and operators built into modern APL systems that are especially useful for IOA. Other sources, such as

LeGrand (2009), provide comprehensive descriptions of the additional primitive functions and operators available in modern APL implementations. Below is a sampling:

? The monadic form is known as *roll*, which is APL's built in pseudo-random number generator. The argument is an array of positive integer values and the result is a pseudo-random value between the index origin (the setting of ⎕IO) and the value(s) in the array. Hence the array of pseudo-random integers is produced *with* replacement. The dyadic version, known as *deal*, produces as many pseudo-random integer values as specified by the left argument, all between the index origin and the value of the right argument and all different. That is the array of pseudo- random integers is generated *without* replacement. Also recall that the initial random number generator seed is specified by ⎕RL.

| The monadic version of this primitive function, known as *magnitude*, returns the absolute value of the argument. The dyadic version, know as *residue*, produces the remainder of right argument divided by the left argument. Note that this is the opposite of the division function ÷ which computes the left argument divided by the right argument.

⌽ ⍉ ⊖ This family of primitive functions, known as *rotate*, *transpose*, and *reverse*, are used to pivot arrays on an axis, as their names suggest.

Additional User-Defined Functions Helpful for IOA

Throughout this volume most user defined functions were listed when they were introduced. Some, however, involved concepts beyond the scope of the text or this volume, such as computational methods for computing matrix determinants (beyond the method of cofactors), linear programming, computing eigenvalues, or functions that involve more advanced APL features not covered in this volume. Those functions are listed here.

<u>DETER, Computing the Determinant of a Square Matrix</u>

```
[  0]  ANS←DETER M;I;N;K;A;eps
[  1] ⍝ 3/18/77 EVALUATES DETERMINANT OF SQ MX
[  2] ⍝THE DET. IS SET TO 0 WHEN A DIAGONAL ELEMENT BECOMES <eps.
[  3]  eps←1E¯10
[  4]  ANS←¯1+I←1
[  5]  ⍎(2≠⍴⍴M)/'→0,0↑⎕←''NOT A MATRIX'''
[  6]  ⍎(≠/⍴M)/'→0,0↑⎕←''NOT A SQUARE MATRIX'''
[  7]  ⍎(0=N←''⍴⍴M)/'→0,0↑⎕←''ORDER IS 0'''
[  8] L:→(I=K←(I-1)+A⍳⌈/A←|(I-1)↓M[;I])/SKP
[  9]  M[I,K;]←M[K,I;]
[ 10]  M[I;]←-M[I;]
[ 11] SKP:→(eps>|M[I;I])/0
[ 12]  M[I+⍳N-I;]←M[I+⍳N-I;]-M[(N-I)⍴I;]×⍉(N,N-I)⍴(I↓M[;I])÷M[I;I]
[ 13]  →(N≥I←I+1)/L
[ 14]  ANS←1 1⍉M
[ 15]  ⍎(75<+/10⍟|ANS)/'ANS←⌊/⍳0,0↑⎕←''VALUE EXCEEDS COMPUTER CAPABILITY.'''
[ 16]  ANS←×/ANS
```

EIG, Computing the Dominant Eigenvalue of a Square Matrix

```
[0]  R←EIG A;B;P;LAM;n
[1] ⍝Principal Eigenvalue and Eigenvector
[2]  R←R÷+/R←+/B←A+(2⍴n)⍴1,(n←1↑⍴A)⍴0
[3] L1:P←R
[4]  R←R÷+/(R←B+.×R)
[5]  →(1E¯13≤+/|R-P)/L1
[6]  LAM←(A+.×R)÷R
[7]  R←(1↑LAM),R
```

LINPROG Linear Programming via the Revised Simplex Method

```
[0]R←LINPROGD;u;nr;I;i;j;na;ns;nc;nt;x;n;no;J;V;L;pr;MX;MO;BM;nn;nm;si;S;B;C;A;M;N;Z0;Z;b;z;rnd;k
[ 1] ⍝--Linear Programming via Simplex Method (Agnew code)
[ 2] ⍝RETURNS solution code, k, final val of obj fun, OBJ, and first optimal soln,
[ 3] ⍝   and PRIMAL[1;] if it exists; other results in global variables PRIMAL, DUAL and OBJ
[ 4] ⍝Solution Code: k (Explicit Result) is 0 for opt soln; 1 no soln exists; 2 unbounded;
[ 5] ⍝    and 3 iteration limit)
[ 6] ⍝⍝--usesh direct definition of rnd function
[ 7] ⍝--D is Partitioned Array of Input Data
[ 8] ⍝------C,u,0
[ 9] ⍝------A,S,b
[10] ⍝-----C is row vector of objective function coefficients (nm←⍴C)
[11] ⍝-----u is scalar u=1 for maximization u=¯1 for minimization
[12] ⍝-----A is matrix of constraint equation coefficients (nn×nm)
[13] ⍝-----S is col vector of signs: ¯1 for ≤; 0 for =; 1 for ≥' for constraint equations;
[14] ⍝       (nn←⍴S=⍴b) the no. of const eqns
[15] ⍝-----b is col vector of rhs terms of constraint equations (nn)
[16] ⍝--BM is pos number signifcantly larger than ⌈/|C
[17] ⍝--MAXITER is max number of simplex iterations
[18] ⍝--MAXOPT is max number of alternative optimal solutions
[19] ⍝--pr is rounding paramter (typically 10 or 12), called by rnd function
[20] ⍝----SPECIFY CONSTANTS AND EXTRACT DATA FROM D
[21]  pr←12 ⋄ MX←3000 ⋄ MO←100 ⋄ BM←1000000
[22]  nn←(1↑⍴D)-1 ⋄ nm←(¯1↑⍴D)-2 ⋄ C←,D[1;⍳nm] ⋄ A←D[1+⍳nn;⍳nm] ⋄ b←,D[1+⍳nn;nm+2] ⋄ u←D[1;nm+1]
[23]  S←nn⍴' ' ⋄ si←,D[1+⍳nn;nm+1] ⋄ S[(si=¯1)/⍳nn]←'≤' ⋄ S[(si=0)/⍳nn]←'=' ⋄ S[(si=1)/⍳nn]←'≥'
[24] ⍝--Z is pivot criterion vector (Z0 is initial vector)
[25] ⍝--M is augmented constraint matrix (the tableau)
[26] ⍝-----OUTPUTS
[27] ⍝---------k (Explicit Result) is 0 for opt soln;
[28] ⍝1 no soln exists; 2 unbounded; and 3 iteration limit)
[29] ⍝---------GLOBAL VARIABLE OTUPUTS
[30] ⍝------------PRIMAL is provisional primal solution
[31] ⍝------------DUAL is provisional dual solution
[32] ⍝------------OBJ is optimal value of objective function
[33]  rnd←{(10*-⍺)×⌊0.5+(10*⍺)×⍵}
[34]  I←(⍳nr)∘.=⍳nr←⍴b ⋄ b[i]←b[i←(b<0)/⍳nr] ⋄ A[i;]←-A[i;]
[35]  S[j,k]←((⍴j←(S[i]='≥')/i)⍴'≤'),(⍴k←(S[i]='≤')/i)⍴'≥'
[36]  M←A,((S='≤')/I),(-(S='≥')/I),((S≠'≤')/I),b
[37]  Z0←(u×C),((ns←+/S≠'=')⍴0),((na←+/S≠'≤')⍴-BM),0
[38]  Z←pr rnd-Z0+BM×+⌿(S≠'≤')⌿M
[39]  B←(nc+⍳+/S='≤'),(nc←⍴C)+ns+⍳na
[40]  B←B[J←⍋((S='≤')/⍳nr),((S='≥')/⍳nr),(S='=')/⍳nr]
[41]  PRIMAL←(0,nc)⍴0 ⋄ DUAL←(0,nr)⍴0
[42]  nt←nc+ns+na+n←no←k←1
[43] L1:→(0>V←⌊/¯1↓z←(pr-2)rnd Z[N←(~(⍳nt)∊B)/⍳nt])/L3
[44]  →(∨/0<,M[(B∊(-na)↑⍳nt-1)/⍳nr;nt])/END,x←(i←B≤nc)/,M[;nt]
[45]  →((V=0)∧1=∨/(x←((⍳nc)∊i)\x[⍋i←i/B])∧.=⍉PRIMAL)/L2
[46]  PRIMAL←PRIMAL,[1]x
[47]  i←(nc+⍳ns),nc+ns+(+/S='≥')+⍳+/S='='
```

```
[48]  DUAL←DUAL,[1]pr rnd((i∊N)\(Z+Z0)[(N∊i)/N])[J]
[49] L2:→((V>0)∨MO<no←no+1)/END,(k←0),OBJ←u×¯1↑z
[50] L3:L←0<⌈/̸(M[;N])[;j←(V=¯1↓z)/⍳nt-nr+1]
[51]  →((0>V),(∨/L),1)/L4,L5,END
[52] L4:→(∧/L)/L5
[53]  →END,k←2
[54] L5:j←j[?⍴j←L/j]
[55]  V←⌊/x←(,M[L;nt])÷,M[L←(0<,M[;N[j]])/⍳nr;N[j]]
[56]  B[i←i[?⍴i←(V=x)/L]]←N[j]
[57]  M←pr rnd M-(,M[;N[j]])∘.×V←(,M[i;])÷,M[i;N[j]]
[58]  Z←pr rnd Z-Z[N[j]]×,M[i;]←pr rnd V
[59]  →(MX≥n←n+1)/L1
[60]  →END,k←3
[61] END:R←k,OBJ
[62]  ⍎(k=0)/'R←R,PRIMAL[1;]'
```

<u>PINV, Matrix Inverse by Successive Partitioning</u>

```
[0]  R←NM PINV MAT;N;M;X;ALPHA;BETA;DELTA;GAMMA;DELTAI;FORM;A;B;C;D
[1] ⍝FUNCTION TO SOLVE INV BY PARTITIONING
[2] ⍝CREATES GLOBAL A,B,C,D
[3]  N←NM[1] ⋄ M←NM[2]
[4]  ⍎(0==/⍴MAT)/'→0,0⍴⎕←''*** MAT MUST BE A SQUARE MATRIX ***'''
[5]  ⍎((N+M)≠1↑⍴MAT)/'→0,0⍴⎕←''*** N+M MUST SUM TO NO OF ROWS IN MAT ***'''
[6]  ⍎(0=DETER2 MAT)/'→0,0⍴⎕←''*** MAT IS SINGULAR ***'''
[7]  ALPHA←MAT[⍳N;⍳N] ⋄ BETA←MAT[⍳N;N+⍳M] ⋄ GAMMA←MAT[N+⍳M;⍳N] ⋄ DELTA←MAT[N+⍳M;N+⍳M]
[8]  'PARTITIONED MATRIX:'
[9]  X←¯1↑⍴⎕←(FORM⍕ALPHA),' ','|',(FORM←10 3)⍕BETA
[10]  X⍴'-'
[11]  (FORM⍕GAMMA),' ','|',FORM⍕DELTA
[12]  ⍎(0=DETER2 DELTA)/'→0,0⍴⎕←''*** DELTA IS A SINGULAR MATRIX ***'''
[13]  DELTAI←⌹DELTA ⋄ X←⎕EX'DELTA'
[14] ⍝CREATE INV PARTITION A
[15]  A←DELTAI+.×GAMMA ⋄ A←BETA+.×A ⋄ A←ALPHA-A ⋄ X←⎕EX'ALPHA' ⋄ A←⌹A
[16] ⍝CREATE INV PARTITION B
[17]  B←BETA+.×DELTAI ⋄ X←⎕EX'BETA' ⋄ B←A+.×B ⋄ B←-B
[18] ⍝CREATE INV PARTITION C
[19]  C←GAMMA+.×A ⋄ C←DELTAI+.×C ⋄ C←-C
[20] ⍝CREATE INV PARTITION D
[21]  D←GAMMA+.×B ⋄ D←((M,M)⍴1,M⍴0)-D ⋄ D←DELTAI+.×D ⋄ X←⎕EX 2 6⍴'GAMMA DELTAI'
[22]  'PARTITIONED INVERSE IS:'
[23]  X←¯1↑⍴⎕←(FORM⍕A),' ','|',FORM⍕B
[24]  X⍴'-'
[25]  (FORM⍕C),' ','|',FORM⍕D
[26]  R←(A,B),[1]C,D
```

Overview of Additional APL Features

As mentioned in the introduction to this volume, the APL features used in this workbook only scratch the surface of those available in modern implementations of APL, as detailed in Legrand (2009) and many other sources. Additional primitive functions for sorting and searching, nested arrays for combining various data types, direct definition functions (writing a function within a function), more advanced data interface mechanisms, more sophisticated methods for identifying and tracing errors, defining local *namespaces* within APL workspaces, recursive programming features, and more advanced and powerful primitive operators and functions are all built in to modern APL implementations.