Numerical Methods for Chemical Engineering

Kenneth J. Beers, Massachusetts Institute of Technology

Supplemental Material for text published by Cambridge University Press, 2006

www.cambridge.org/978052189714

Linear Algebra: Proof of Euler's Formula

We now prove the very useful and important ***Euler formula***,

$$e^{i\theta} = \cos\theta + i\sin\theta \qquad \textbf{(EQ 1)}$$

First, let us review some properties of the real-valued exponential function,

$$f(x) = e^x \qquad x \in \Re \qquad f(x) \in \Re \qquad \textbf{(EQ 2)}$$

This exponential function has the property that $df/dx = f(x)$ as $\frac{d}{dx}e^x = e^x$.

We now wish to extend this definition to find a complex valued function

$$f(z) = e^z \qquad z \in C \qquad f(z) \in C \qquad \textbf{(EQ 3)}$$

such that

1. $f(z)$ is single-valued and df/dz is defined

2. $df/dz = f$, *i.e.* $\frac{d}{dz}e^z = e^z$

3. $e^z \to e^x$, as $z = x + iy$ and $y = Im\{z\} \to 0$

Since $z = a + ib$, to determine when condition (1) is satisfied, we need to define what we mean by df/dz. For $x \in R, f(x) \in R$, the definition of df/dx as a limit is familiar,

$$\left.\frac{df}{dx}\right|_{x^+} = \lim_{\Delta x \to 0} \frac{f(x + \Delta x) - f(x)}{\Delta x} \qquad \left.\frac{df}{dx}\right|_{x^-} = \lim_{\Delta x \to 0} \frac{f(x) - f(x - \Delta x)}{\Delta x} \qquad \textbf{(EQ 4)}$$

If both limits exist and return the same finite value, this value is taken to be the derivative df/dx at x; otherwise, $f(x)$ is not differentiable at x.

For $z \in C, f(z) \in C$, we define the derivative $\frac{df}{dz}$ at a point $\hat{z}$ to be

$$\left.\frac{df}{dz}\right|_{\hat{z}} = \lim_{\Delta z \to 0} \frac{f(\hat{z} + \Delta z) - f(\hat{z})}{\Delta z} \qquad \textbf{(EQ 5)}$$

where

$$\hat{z} = \hat{a} + i\hat{b} \qquad \Delta z = \Delta a + i\Delta b \qquad \textbf{(EQ 6)}$$

For this derivative to exist, we must obtain the same finite limiting value along all paths $\Delta z \to 0$. It is sufficient to show that we obtain the same limiting value along the two paths (I) and (II) in Figure 1.

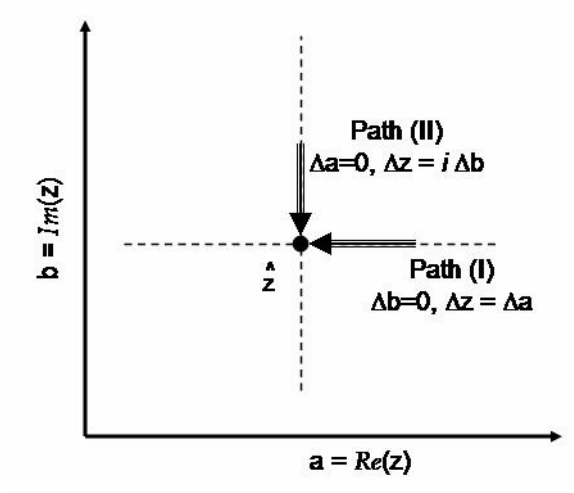

FIGURE 1. Paths (I) and (II) in the complex plane used to prove that *df/dz* exists at a point

If $z = a + ib$, let us write $f(z)$ as

$$f(z) = u(a, b) + iw(a, b) \qquad \textbf{(EQ 7)}$$

where *u* and *w* are both real-valued, differentiable functions.

Along path (I), the path of constant imaginary part of *z*,

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(I)} = \lim_{\Delta a \to 0} \frac{[u(\hat{a} + \Delta a, \hat{b}) + iw(\hat{a} + \Delta a, \hat{b})] - [u(\hat{a}, \hat{b}) + iw(\hat{a}, \hat{b})]}{\Delta a}$$

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(I)} = \lim_{\Delta a \to 0} \left\{ \frac{u(\hat{a} + \Delta a, \hat{b}) - u(\hat{a}, \hat{b})}{\Delta a} + i\, \frac{w(\hat{a} + \Delta a, \hat{b}) - w(\hat{a}, \hat{b})}{\Delta a} \right\} \qquad \textbf{(EQ 8)}$$

Therefore, in the limit $\Delta a \to 0$,

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(I)} = \frac{\partial u}{\partial a} + i\frac{\partial w}{\partial a} \qquad \textbf{(EQ 9)}$$

Along path (II), the path of constant real part of z,

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(II)} = \lim_{\Delta b \to 0} \frac{[u(\hat{a}, \hat{b} + \Delta b) + iw(\hat{a}, \hat{b} + \Delta b)] - [u(\hat{a}, \hat{b}) + iw(\hat{a}, \hat{b})]}{i\Delta b}$$

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(II)} = \lim_{\Delta b \to 0} \left\{ \frac{1}{i} \frac{u(\hat{a}, \hat{b} + \Delta b) - u(\hat{a}, \hat{b})}{\Delta b} + \frac{i}{i} \frac{w(\hat{a}, \hat{b} + \Delta \hat{b}) - w(\hat{a}, \hat{b})}{\Delta b} \right\} \quad \textbf{(EQ 10)}$$

so that when $\Delta b \to 0$,

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(II)} = -i \frac{\partial u}{\partial b} + \frac{\partial w}{\partial b} \quad \textbf{(EQ 11)}$$

For $\left.\frac{df}{dz}\right|_{\hat{z}}$ to exist, these two limiting values of the derivative must be equal,

$$\left.\frac{df}{dz}\right|_{\hat{z}}^{(I)} = \left.\frac{df}{dz}\right|_{\hat{z}}^{(II)} \Rightarrow \frac{\partial u}{\partial a} + i \frac{\partial w}{\partial a} = -i \frac{\partial u}{\partial b} + \frac{\partial w}{\partial b} \quad \textbf{(EQ 12)}$$

This yields the ***Cauchy-Riemann equations*** that must be satisfied for $\frac{df}{dz}$ to exist, *i.e.* for $f(z)$ to be ***analytic***,

$$\frac{\partial u}{\partial a} = \frac{\partial w}{\partial b} \qquad \frac{\partial w}{\partial a} = -\frac{\partial u}{\partial b} \quad \textbf{(EQ 13)}$$

To find an analytic exponential function for complex numbers, we write $f(z) = e^z$ as

$$e^z = u(a, b) + iw(a, b) \quad \textbf{(EQ 14)}$$

and find the real functions $u(a, b), w(a, b)$ that satisfy the Cauchy-Riemann equations and the defining equation for the exponential function,

$$\frac{d}{dz} e^z = e^z = u + iw \quad \textbf{(EQ 15)}$$

Along path (I),

$$\left.\frac{de^z}{dz}\right|^{(I)} = \frac{\partial u}{\partial a} + i\frac{\partial w}{\partial a} = u + iw \qquad \textbf{(EQ 16)}$$

which yields the two conditions

$$\frac{\partial u}{\partial a} = u \qquad \frac{\partial w}{\partial a} = w \qquad \textbf{(EQ 17)}$$

The first is satisfied if

$$u = e^a g(b) \qquad \textbf{(EQ 18)}$$

Using the second Cauchy-Riemann equation, the second condition in (EQ 17) becomes

$$w = \frac{\partial w}{\partial a} = -\frac{\partial u}{\partial b} \qquad \textbf{(EQ 19)}$$

Differentiation with respect to *b* yields

$$\frac{\partial w}{\partial b} = -\frac{\partial^2 u}{\partial b^2} \qquad \textbf{(EQ 20)}$$

Using the first Cauchy-Riemann equation, we obtain

$$\frac{\partial u}{\partial a} = -\frac{\partial^2 u}{\partial b^2} \qquad \textbf{(EQ 21)}$$

We then use the real part of the path (I) limit, $u = \partial u/\partial a$, to obtain

$$\frac{\partial^2 u}{\partial b^2} = -u \qquad \textbf{(EQ 22)}$$

Substituting $u = e^a g(b)$ yields

$$e^a \frac{d^2 g}{db^2} = -e^a g \qquad \Rightarrow \qquad \frac{d^2 g}{db^2} = -g \tag{EQ 23}$$

This differential equation has the general solution

$$g(b) = c_1 \cos b + c_2 \sin b \tag{EQ 24}$$

therefore

$$u(a, b) = e^a g(b) = e^a [c_1 \cos b + c_2 \sin b] \tag{EQ 25}$$

We next use (EQ 19) to obtain

$$\begin{aligned} w &= -\frac{\partial u}{\partial b} = -e^a [-c_1 \sin b + c_2 \cos b] \\ w(a, b) &= e^a [c_1 \sin b - c_2 \cos b] \end{aligned} \tag{EQ 26}$$

Therefore, to satisfy the Cauchy-Riemann equations and possess the properties of the exponential function, we must have

$$e^z = u(a, b) + iw(a, b) = e^a [c_1 \cos b + c_2 \sin b] + ie^a [c_1 \sin b - c_2 \cos b] \tag{EQ 27}$$

Finally, we must have $\lim_{b \to 0} e^z = e^a$, so with $\cos 0 = 1, \sin 0 = 0$ we have

$$\lim_{b \to 0} e^z = e^a [c_1] + ie^a [-c_2] \tag{EQ 28}$$

requiring $c_1 = 1$ and $c_2 = 0$; so that for $z = a + ib$,

$$e^z = e^a [\cos b + i \sin b] \tag{EQ 29}$$

For $a = 0, b = \theta$, this yields the ***Euler formula***,

$$e^{i\theta} = \cos\theta + i \sin\theta \tag{EQ 30}$$

Q.E.D.

Linear Algebra: Floating point real number representation and round-off errors

In the algorithm for partial pivoting, we perform a pivot not just when the diagonal element is zero, but whenever the diagonal element is not greater than all values in the column below. This may seem like wasted effort; however, there is a good reason to do so. Pivoting in this case makes the elimination algorithm more stable with respect to the round-off errors that occur whenever we store real numbers in the memory of a digital computer.

Floating point number representation

If we were to look at the memory within a computer, we would find data represented digitally as a sequence of 0s and 1s

$$|01001011|00001101|00000001|00010110|$$

Each individual 0 or 1 location in memory is called a ***bit***, and each set of eight consecutive bits is called a ***byte***. To store a real number (such as 3.273) in memory, we need to represent it in such a binary format. This is done typically using ***floating point notation***. Let *f* be a real number that we want to store in memory. To do so, we must approximate it as some ***representation value*** $\tilde{f} \approx f$ that we can express in binary format as

$$\tilde{f} = \pm[d_1 \times 2^{e-1} + d_2 \times 2^{e-2} + \ldots + d_t \times 2^{e-t}] \quad \textbf{(EQ 31)}$$

t is a positive integer fixed in the representation system. Each d_i takes a value of 0 or 1, and is stored in digital memory as a single bit. *e* is a integer exponent in the range

$$L \le e \le U \quad \textbf{(EQ 32)}$$

that is also stored in binary format, *e.g.* by allocating a byte,

$$|e| = e_0 \times 2^0 + e_1 \times 2^1 + e_2 \times 2^2 + e_3 \times 2^3 + e_4 \times 2^4 + e_5 \times 2^5 + e_6 \times 2^6 \quad \textbf{(EQ 33)}$$

with the sign of e determined by the eighth bit e_7 (say if $e_7 = 0$, $e \geq 0$; else $e \leq 0$). To represent the integer $e = 3$, we would have

$$\underset{e_7}{0}\,\underset{e_6}{0}\,\underset{e_5}{0}\,\underset{e_4}{0}\,\underset{e_3}{0}\,\underset{e_2}{0}\,\underset{e_1}{1}\,\underset{e_0}{1} = 3$$

The upper bound $e = U$ occurs when $e_0 = e_1 = e_2 = \ldots = e_6 = 1$,

$$U = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 = \sum_{k=0}^{6} 2^k = 2^7 - 1 \qquad \textbf{(EQ 34)}$$

In general, $U \approx 2^{\kappa - 1}$ and $L \approx -2^{\kappa - 1}$, where κ is the number of bits allocated to store each e. The values of t and κ are fixed in the definition of our system of binary representation. For each real number f, the ***floating-point representation*** $\tilde{f}$ consists of the bits allocated to store the values of $d_1, d_2, \ldots, d_t$, e, and the sign of f. To maximize the efficiency of storing data, we may adjust e so that $d_1 = 1$, in which case a binary representation with $t = \kappa = 8$ is

$$\tilde{f} = \underset{+/-}{0}\ \underset{d_1 = 1}{\underline{\quad}}\ \underset{d_2}{0}\,\underset{d_3}{1}\,\underset{d_4}{0}\,\underset{d_5}{1}\,\underset{d_6}{1}\,\underset{d_7}{0}\,\underset{d_8}{0} \qquad \underset{+/-}{1}\ \underset{e_6}{0}\,\underset{e_5}{0}\,\underset{e_4}{1}\,\underset{e_3}{0}\,\underset{e_2}{1}\,\underset{e_1}{0}\,\underset{e_0}{0} \qquad \textbf{(EQ 35)}$$

What real number does this represent? First, we compute the exponent integer,

$$e = -[2^4 + 2^2] = -20 \qquad \textbf{(EQ 36)}$$

In decimal notation,

$$\tilde{f} = +[2^{e-1} + 2^{e-3} + 2^{e-5} + 2^{e-6}] \approx 6.4075 \times 10^{-7} \qquad \textbf{(EQ 37)}$$

What are the smallest and largest magnitude real numbers that can be represented in memory for a given t and κ?

The smallest magnitude m of a real number that can be represented occurs when only d_1 is non-zero and $e = L$,

$$m = \frac{0}{+/-}\ \underbrace{\frac{}{d_1 = 1}\ \frac{0}{d_2}\frac{0}{d_3}\frac{0}{d_4}\frac{0}{d_5}\frac{0}{d_6}\frac{0}{d_7}\frac{0}{d_8}}\qquad \frac{1}{+/-}\ \frac{1}{e_6}\frac{1}{e_5}\frac{1}{e_4}\frac{1}{e_3}\frac{1}{e_2}\frac{1}{e_1}\frac{1}{e_0} \tag{EQ 38}$$

so that

$$m = 2^{L-1} \tag{EQ 39}$$

The largest magnitude M that can be represented occurs when all $d_i = 1$ and $e = U$,

$$M = \frac{0}{+/-}\ \underbrace{\frac{1}{d_1 = 1}\ \frac{1}{d_2}\frac{1}{d_3}\frac{1}{d_4}\frac{1}{d_5}\frac{1}{d_6}\frac{1}{d_7}\frac{1}{d_8}}\qquad \frac{0}{+/-}\ \frac{1}{e_6}\frac{1}{e_5}\frac{1}{e_4}\frac{1}{e_3}\frac{1}{e_2}\frac{1}{e_1}\frac{1}{e_0} \tag{EQ 40}$$

so that

$$M = \sum_{j=1}^{t} 2^{U-j} = 2^U \sum_{j=1}^{t} 2^{-j} = 2^{U-1} \sum_{j=1}^{t} \left(\frac{1}{2}\right)^j = 2^U[1 - 2^{-t}] \tag{EQ 41}$$

In the last equality we have used the identity for a ***geometric progression***,

$$\sum_{j=1}^{N} x^j = \frac{x^N - 1}{x - 1} \qquad x \neq 1 \tag{EQ 42}$$

For given κ and t, we can store representation values in the range

$$m \leq |\tilde{f}| \leq M \tag{EQ 43}$$

It is common to represent real numbers with $|\tilde{f}| < m$ by "0" and those with $|\tilde{f}| > M$ by "NaN", *i.e.* "Not a Number". We can increase the breadth of this range by increasing the values of κ and t; however, the number of bits, $\kappa + t + 1$, necessary to store each real number also increases. A common system is ***32-bit single precision*** with $\kappa = 8$, $t = 23$ and $U = -L = 128$,

$m \approx 1.5 \times 10^{-39}$, $M \approx 3.4 \times 10^{38}$. In ***64-bit double precision***, the memory allocation is doubled with $\kappa = 16$, $t = 52$. It is important to note that the actual representation system labeled as single or double precision varies from one machine to another.

Propagation of round-off errors

An important point is that when we wish to store a real number f in memory, in general it cannot be exactly represented by a finite set of bits in floating point notation. We must therefore approximate it by some value $\tilde{f}$ that can be represented in memory,

$$\tilde{f} = \pm[d_1 \times 2^{e-1} + d_2 \times 2^{e-2} + \ldots + d_t \times 2^{e-t}] \quad \textbf{(EQ 44)}$$

The difference between the "true" value f and the representation value $\tilde{f}$ is called the ***round-off error***, $rd(f)$

$$rd(f) = f - \tilde{f} \quad \textbf{(EQ 45)}$$

For binary representation of a number f with $m \le |f| \le M$, the magnitude of the round-off error is

$$rd(f) \sim 2^{e-t} = 2^{-t} \times 2^{e} = (eps) \times 2^{e} \quad \textbf{(EQ 46)}$$

where we define the ***machine precision*** (MATLAB command **eps**) as

$$eps = 2^{-t} \quad \textbf{(EQ 47)}$$

We write the round-off error as

$$rd(f) = r_f \times eps \times 2^{e} \quad \textbf{(EQ 48)}$$

where r_f is some number on the order of one. We write the representation value $\tilde{f}$ as

$$\tilde{f} = m_f \times 2^e \quad \text{(EQ 49)}$$

where the ***mantissa***,

$$m_f = d_1 \times 2^{-1} + d_2 \times 2^{-2} + \ldots + d_t \times 2^{-t} \quad \text{(EQ 50)}$$

is also of order one. We then characterize the ***relative round-off error*** in our representation of f as

$$\frac{rd(f)}{\tilde{f}} = \frac{r_f \times eps \times 2^e}{m_f \times 2^e} = \frac{r_f}{m_f} \times eps \quad \text{(EQ 51)}$$

As r_f / m_f is generally about on the order of one, *eps* determines the relative amount of error when representing real numbers in floating point notation. If $eps \ll 1$, as is usually the case, $rd(f) \ll \tilde{f}$; that is, when storing a real number in memory, we can represent it to high relative precision, although there is in general some finite amount, though small, of round-off error.

*Even though relative round-off errors are initially quite small, we may perform a sequence of operations on the set of representation values that store the data for our problem that increase (magnify) the initial round-off errors until they **do** become significant.*

For example, let us take the difference between two similar, but large, numbers,

$$f = 3.0000001 \times 10^6 \qquad g = 3.0000009 \times 10^6 \quad \text{(EQ 52)}$$

We see that $|f - g| \ll |f|, |g|$. Let us write each real number as its representation plus some round off error,

$$f = \tilde{f} + rd(f) \qquad g = \tilde{g} + rd(g) \quad \text{(EQ 53)}$$

Likewise the difference is

$$f - g = [\tilde{f} - \tilde{g}] + [rd(f) - rd(g)] \quad \text{(EQ 54)}$$

Thus, the representation error in the value of $f-g$ after forming it from the difference of the representations of f and g is

$$crd(f-g) = rd(f) - rd(g) \qquad \textbf{(EQ 55)}$$

Here $crd(f)$ denotes the ***cumulative round-off error*** of a quantity f, which takes into account all sources of round-off error that feed into its representation $\tilde{f}$ through the operations that we have performed to date. Let us write

$$\begin{aligned} \tilde{f} &= m_f \times 2^{e_f} \qquad & \tilde{g} &= m_g \times 2^{e_g} \\ rd(f) &= r_f \times eps \times 2^{e_f} \qquad & rd(g) &= r_g \times eps \times 2^{e_g} \end{aligned} \qquad \textbf{(EQ 56)}$$

The relative cumulative round-off error of $f-g$ is then

$$\frac{crd(f-g)}{\tilde{f}-\tilde{g}} = eps \times \left[\frac{r_f \times 2^{e_f} - r_g \times 2^{e_g}}{m_f \times 2^{e_f} - m_g \times 2^{e_g}} \right] \qquad \textbf{(EQ 57)}$$

For the case of $f = 3.0000001 \times 10^6$, $g = 3.0000009 \times 10^6$; that is, two numbers that are very similar, we have $e_f = e_g = e$, and

$$\frac{crd(f-g)}{\tilde{f}-\tilde{g}} = eps \times \left[\frac{r_f - r_g}{m_f - m_g} \right] \qquad \textbf{(EQ 58)}$$

In general, the difference $r_f - r_g$ is expected to on the order of one. However, if f and g are very close, we do know that $|m_f - m_g| \ll 1$, and thus

$$\left| \frac{crd(f-g)}{\tilde{f}-\tilde{g}} \right| \gg max\left\{ \left| \frac{rd(f)}{\tilde{f}} \right|, \left| \frac{rd(g)}{\tilde{g}} \right| \right\} \qquad \textbf{(EQ 59)}$$

When subtracting two real numbers f and g that are very close in value, the relative representation error for the difference $f-g$, computed from the representations of f and g, may be far larger than the original round-off errors in either f or g.

Such a subtraction therefore has poor ***error propagation***, and the accuracy of the stored representation values becomes less each time this operation is performed. We wish to design our algorithms so that the cumulative round-off errors do not grow larger, and ideally decay to zero. Once the relative cumulative errors grow to the order of one, the algorithm may crash as the representation values stored in memory are highly inaccurate.

Favorable error propagation with partial pivoting

To design algorithms with favorable error propagation, we must consider what happens to cumulative round-off error when performing the operation

$$a \leftarrow a - \lambda b \tag{EQ 60}$$

As we really perform the operation on the floating point representations, the values stored in memory transform as

$$\tilde{a} \leftarrow rd(\tilde{a} - \tilde{\lambda}\tilde{b}) \;=\; \tilde{a} - \tilde{\lambda}\tilde{b} + e_{new} \tag{EQ 61}$$

e_{new} is new round-off error introduced when storing the result in memory,

$$e_{new} \sim eps \times (\tilde{a} - \tilde{\lambda}\tilde{b})_{old} \tag{EQ 62}$$

The cumulative round-off error of a transforms during $a \leftarrow a - \lambda b$ as

$$\begin{aligned} a - \tilde{a} &\leftarrow a - \tilde{a} - \lambda b + \tilde{\lambda}\tilde{b} - e_{new} \\ crd(a) &\leftarrow crd(a) - \lambda b + \tilde{\lambda}\tilde{b} - e_{new} \end{aligned} \tag{EQ 63}$$

Assuming the best case that $\lambda \approx \tilde{\lambda}$, we have

$$crd(a) \leftarrow crd(a) - \lambda \times crd(b) - e_{new} \tag{EQ 64}$$

We note that $crd(b)$ and e_{new} may be positive or negative, and so the fact that they are subtracted from $crd(a)$ is irrelevant.

What is important, is that if $|\lambda| > 1$, *the contribution from* $crd(b)$ *to the new value of* $crd(a)$ *is amplified, so that we have poor error propagation.*

Therefore, when designing algorithms that include scalar operations of the form $a \leftarrow a - \lambda b$, we should ensure that $|\lambda| < 1$ and so make the algorithm stable with respect to the propagation of error.

We now can see why we perform partial pivoting during Gaussian elimination, even if the diagonal element in the column is non-zero. To eliminate the value at the (j, i) position, we perform a number of scalar operations

$$a_{jk} \leftarrow a_{jk} - \lambda_{ji} a_{ik} \qquad \lambda_{ji} = a_{ji} / a_{ii} \qquad \textbf{(EQ 65)}$$

If we first perform a partial pivot, if needed, to ensure that a_{ii} has the largest magnitude of any element at or below the diagonal in column #*i*, then all $|\lambda_{ji}| \leq 1$.

Therefore, partial pivoting produces a Gaussian elimination algorithm that is numerically stable with respect to the propagation of error, at least as long as the elements of A are of comparable magnitude.

We add this qualification, because if the elements of A are dramatically different, especially across any given row, we then have scalar operations involving numbers of vastly different magnitudes that are not favorable for the propagation of error. If we use ***full pivoting***, in which we also swap columns, we can further improve the error propagation properties. For most systems, however, partial pivoting is sufficient.

Linear Algebra: Computing the FLOPs necessary to perform Gaussian elimination and backward substitution

Let us now determine the exact numbers of FLOPs required for elimination, v_{elim}, and backward substitution, v_{sub}.

First, consider elimination. Let ν_1 be the number of FLOPs required to put all zeros below the diagonal in the first column. To place a zero at (2,1), we perform a row operation involving $2N+1$ FLOPS,

$$\begin{aligned} 1 \text{ FLOP} &\Leftrightarrow \text{calculate } \lambda_{21} \\ 2\times(N-1) \text{ FLOPs} &\Leftrightarrow \text{calculate } a_{22}^{(2,1)}, a_{23}^{(2,1)}, \ldots, a_{2N}^{(2,1)} \\ 2 \text{ FLOPs} &\Leftrightarrow \text{calculate } b_2^{(2,1)} \end{aligned} \qquad \textbf{(EQ 66)}$$

To place all zeros in the remainder of column 1, we need to perform another $(N-2)$ row operation to place zeros at $(3,1),(4,1),\ldots,(N,1)$. Each row operation requires $2N+1$ FLOPs. The number of FLOPS required to place zeros below the diagonal in the first column is

$$\nu_1 = [N-1]\times[2N+1] \qquad \textbf{(EQ 67)}$$

To place zeros below the diagonal in the second column, we perform $N-2$ row operations, each of which requires two fewer FLOPs than those for the first column. The total number of FLOPs required to eliminate the lower triangular elements in the second column is

$$\nu_2 = [N-2]\times[2N-1] \qquad \textbf{(EQ 68)}$$

In general, to place zeros below the diagonal in column #*i*, we must perform $N-i$ row operations, each involving $2(N-i)+1$ FLOPs, so that the number of FLOPs required for column #*i* is

$$\nu_i = [N-i]\times[2(N-i)+1] \qquad \textbf{(EQ 69)}$$

The total number of FLOPs required to put the system in triangular form is

$$\nu_{elim} = \sum_{i=1}^{N-1}\nu_i = \sum_{i=1}^{N-1}[N-i]\times[2(N-i)+1] \qquad \textbf{(EQ 70)}$$

In the limit $N \gg 1$, we can approximate this sum by an integral,

$$v_{elim} = \sum_{i=1}^{N-1} [N-i] \times [2(N-i)+1] \approx \int_{1}^{(N-1)} [N-x] \times [2(N-x)+1] dx$$

$$v_{elim} \approx \int_{1}^{(N-1)} [N-x] \times [2(N-x)] dx \approx \int_{0}^{N} [N-x] \times [2(N-x)] dx \qquad \textbf{(EQ 71)}$$

Using a change of variable $y = N - x$, we have

$$v_{elim} \approx \int_{N}^{0} [y] \times [2y](-dy) = 2\int_{0}^{N} y^2 dy = \frac{2}{3}N^3 \qquad \textbf{(EQ 72)}$$

The number of FLOPs required for backward substitution is

$$v_{sub} = \sum_{i=1}^{N} 2(N-i+1) \approx 2\int_{1}^{N} (N-x+1) dx \approx 2\int_{0}^{N} (N-x) dx = N^2 \qquad \textbf{(EQ 73)}$$

Therefore, for large systems of equations, when the number of calculations required to solve the system is a concern, $v_{elim} \gg v_{sub}$.

For large systems, the CPU time required to perform backward substitution is trivially small compared to the effort required to transform the system into upper triangular form. The overall number of FLOPs required to solve the system by Gaussian elimination is essentially $2N^3/3$.

Linear Algebra: More on matrix determinants

We now provide a more detailed discussion of matrix determinants, and in particular provide some reasons why the proposed functional form is a reasonable one. In addition, proofs of some of the properties of determinants and a demonstration of expansion by minors are provided.

We define the ***determinant*** of a square $N \times N$ matrix A as

$$det(A) = |A| = \begin{cases} c \neq 0, \text{ if } K_A = \mathbf{0} \\ 0, \text{ if } \exists \boldsymbol{w} \in K_A, \boldsymbol{w} \neq \mathbf{0} \end{cases} \quad \text{(EQ 74)}$$

If $det(A) \neq 0$; that is, if A is ***nonsingular***, then $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution for all $\underline{b} \in \Re^N$. Otherwise, if $det(A) = 0$; that is, if A is ***singular***, then no unique solution to $A\boldsymbol{x} = \boldsymbol{b}$ exists (either there are no solutions or an infinite number of them).

We wish to find a formula to compute from the matrix A the scalar value of the determinant that satisfies the above definition. To do so, we first identify a number of characteristics that this determinant function must have to be a reliable measure of singularity.

Characteristic # 1

Let us we multiply any equation in our linear system, say the second, by some non-zero real scalar $c \neq 0$,

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \Rightarrow \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ ca_{21} & ca_{22} & \dots & ca_{2N} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \qquad \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_N \end{bmatrix} \Rightarrow \begin{bmatrix} b_1 \\ cb_2 \\ \vdots \\ b_N \end{bmatrix} \quad \text{(EQ 75)}$$

If $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution, certainly the transformed, equivalent one does as well. Such a transformation should not change whether the determinant is zero or not.

Characteristic #2

Since the existence of a solution to $A\boldsymbol{x} = \boldsymbol{b}$ does not depend upon the order in which we write the equations, we should be able to swap any rows in the matrix A without affecting whether the determinant is zero or not.

Characteristic #3

Similarly, we should be able to modify the system $A\mathbf{x} = \mathbf{b}$ by relabeling the unknowns; for example, by exchanging $x_j \Leftrightarrow x_k$ which corresponds to swapping the j^{th} and k^{th} columns of *A*. This should not affect whether or not the determinant is zero.

Characteristic #4

We can perform any elementary row operation $k \leftarrow k + c \times j$ with $c \neq 0$ to the augmented matrix $(A, \mathbf{b})$ to obtain a new equivalent system $(A', \mathbf{b}')$. If $det(A) \neq 0$, then $det(A') \neq 0$ and if $det(A) = 0$, then $det(A') = 0$.

Characteristic #5

If the matrix product $C = AB$ is to be nonsingular, *i.e.* there is a unique solution to $C\mathbf{x} = \mathbf{b}$; then there must be a unique $\mathbf{v}$ such that $A\mathbf{v} = \mathbf{b}$ and also a unique $\mathbf{x}$ such that $B\mathbf{x} = \mathbf{v}$. Thus, for $C = AB$ to be nonsingular, both *A* and *B* must be nonsingular.

Characteristic #6

If any two rows of *A* are identical, or simply differ by a scalar factor *c*, the equations that they represent are dependent and it is possible to convert the system into an equivalent one with a row containing only zeros. We therefore must have $det(A) = 0$.

Characteristic #7

Similarly, if any two columns of *A*, say the j^{th} and the k^{th}, are multiples of each other, *i.e.* $a_{mj} = ca_{mk}$, we can write any equation #*m* in the system as,

$$\begin{aligned} a_{m1}x_1 + \ldots + a_{mj}x_j + \ldots + a_{mk}x_k + \ldots + a_{mN}x_N &= b_m \\ a_{m1}x_1 + \ldots + ca_{mk}x_j + \ldots + a_{mk}x_k + \ldots + a_{mN}x_N &= b_m \\ a_{m1}x_1 + \ldots + (0)x_j + \ldots + a_{mk}(x_k + cx_j) + \ldots + a_{mN}x_N &= b_m \end{aligned} \qquad \textbf{(EQ 76)}$$

We thus can make any change to $\boldsymbol{x}$ of the form

$$x_j \leftarrow x_j + \Delta x_j \qquad x_k \leftarrow x_k - c\Delta x_j \qquad x_m \leftarrow x_m,\ m \neq j, k \qquad \textbf{(EQ 77)}$$

without changing the value of $A\boldsymbol{x}$. Therefore, there can be no unique solution to $A\boldsymbol{x} = \boldsymbol{b}$ and we must have $det(A) = 0$.

We have listed some properties that $det(A)$ must have to be a reliable measure of singularity, and now show that the definition in the text satisfies them,

$$det(A) = \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1, i_2, \ldots, i_N}a_{i_1,1}a_{i_2,2}\ldots a_{i_N,N} \qquad \textbf{(EQ 78)}$$

Expansion by minors of a 3 x 3 **matrix**

Using (EQ 78), we now demonstrate expansion by minors for the determinant of a 3×3 matrix,

$$det(A) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = \sum_{i_1=1}^{3}\sum_{i_2=1}^{3}\sum_{i_3=1}^{3}\varepsilon_{i_1, i_2, i_3}a_{i_1,1}a_{i_2,2}a_{i_3,3} \qquad \textbf{(EQ 79)}$$

We rearrange the equation for the 3×3 matrix by splitting the summation over i_1,

$$det(A) = \sum_{i_2=1}^{3}\sum_{i_3=1}^{3}\varepsilon_{1, i_2, i_3}a_{11}a_{i_2,2}a_{i_3,3} + \sum_{\substack{i_1=1 \\ i_1\neq1}}^{3}\sum_{i_2=1}^{3}\sum_{i_3=1}^{3}\varepsilon_{i_1, i_2, i_3}a_{i_1,1}a_{i_2,2}a_{i_3,3} \qquad \textbf{(EQ 80)}$$

If either $i_2 = 1$ or $i_3 = 1$, then $\varepsilon_{1, i_2, i_3} = 0$, so the first sum is

$$\sum_{i_2=1}^{3}\sum_{i_3=1}^{3}\varepsilon_{1,i_2,i_3}a_{i_2,2}a_{i_3,3} = \varepsilon_{1,2,3}a_{22}a_{33} + \varepsilon_{1,3,2}a_{32}a_{23} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} \quad \textbf{(EQ 81)}$$

We now split the summation over i_2,

$$det(A) = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + a_{12}\sum_{\substack{i_1=1 \\ i_1\neq 1}}^{3}\sum_{i_3=1}^{3}\varepsilon_{i_1,1,i_3}a_{i_1,1}a_{i_3,3} + \sum_{\substack{i_1=1 \\ i_1\neq 1}}^{3}\sum_{\substack{i_2=1 \\ i_2\neq 1}}^{3}\sum_{i_3=1}^{3}\varepsilon_{i_1,i_2,i_3}a_{i_1,1}a_{i_2,2}a_{i_3,3} \quad \textbf{(EQ 82)}$$

The sum multiplying a_{12} is

$$\sum_{\substack{i_1=1 \\ i_1\neq 1}}^{3}\sum_{i_3=1}^{3}\varepsilon_{i_1,1,i_3}a_{i_1,1}a_{i_3,3} = \varepsilon_{2,1,3}a_{21}a_{33} + \varepsilon_{3,1,2}a_{31}a_{23} = -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} \quad \textbf{(EQ 83)}$$

Finally, we split the summation over i_3,

$$det(A) = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\sum_{\substack{i_1=1 \\ i_1\neq 1}}^{3}\sum_{\substack{i_2=1 \\ i_2\neq 1}}^{3}\varepsilon_{i_1,i_2,1}a_{i_1,1}a_{i_2,2} + \sum_{\substack{i_1=1 \\ i_1\neq 1}}^{3}\sum_{\substack{i_2=1 \\ i_2\neq 1}}^{3}\sum_{\substack{i_3=1 \\ i_3\neq 1}}^{3}\varepsilon_{i_1,i_2,i_3}a_{i_1,1}a_{i_2,2}a_{i_3,3} \quad \textbf{(EQ 84)}$$

In the final sum, we note that if none of $\{i_1, i_2, i_3\}$ equals one, then at least two of them must be equal, and $\varepsilon_{i_1,i_2,i_3} = 0$ for every term in the sum. The sum multiplying a_{13} is

$$\sum_{\substack{i_1 = 1 \\ i_1 \neq 1}}^{3} \sum_{\substack{i_2 = 1 \\ i_2 \neq 1}}^{3} \varepsilon_{i_1, i_2, 1} a_{i_1, 1} a_{i_2, 2} = \varepsilon_{2, 3, 1} a_{21} a_{32} + \varepsilon_{3, 2, 1} a_{31} a_{22} = \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \quad \text{(EQ 85)}$$

Therefore, the determinant of a 3×3 matrix can be written as an expansion in minors,

$$det(A) = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \quad \text{(EQ 86)}$$

General properties of the determinant function

We now provide proofs for the general properties of determinants. From these properties, it is shown that the formula for the determinant proposed above satisfies the desired characteristics that we have identified.

Property I

The determinant of a $N \times N$ real matrix A equals that of its transpose, A^T.

Proof:

The determinant of A^T, with $a^T_{jk} = a_{kj}$, is

$$\begin{aligned} det(A^T) &= \sum_{i_1 = 1}^{N} \sum_{i_2 = 1}^{N} \cdots \sum_{i_N = 1}^{N} \varepsilon_{i_1, i_2, \ldots, i_N} a^T_{i_1, 1} a^T_{i_2, 2} \cdots a^T_{i_N, N} \\ det(A^T) &= \sum_{i_1 = 1}^{N} \sum_{i_2 = 1}^{N} \cdots \sum_{i_N = 1}^{N} \varepsilon_{i_1, i_2, \ldots, i_N} a_{1, i_1} a_{2, i_2} \cdots a_{N, i_N} \end{aligned} \quad \text{(EQ 87)}$$

For every ordered set $(i_1, i_2, \ldots, i_N)$ that is a permutation of $(1, 2, \ldots, N)$, there exists another permutation $(j_1, j_2, \ldots, j_N)$ such that

$$a_{1,i_1}a_{2,i_2}\ldots a_{N,i_N} = a_{j_1,1}a_{j_2,2}\ldots a_{j_N,N} \qquad \textbf{(EQ 88)}$$

Let us reorder the product $a_{1,i_1}a_{2,i_2}\ldots a_{N,i_N}$ by a sequence of exchanges, to put first the element from column 1, next the element from column 2, *etc.*, to obtain finally $a_{j_1,1}a_{j_2,2}\ldots a_{j_N,N}$. During this sequence of exchanges, the row indices are transformed as $(1, 2, \ldots, N) \rightarrow (j_1, j_2, \ldots, j_N)$ and the column indices are transformed as $(i_1, i_2, \ldots, i_N) \rightarrow (1, 2, \ldots, N)$. Therefore, both $(j_1, j_2, \ldots, j_N)$ and $(i_1, i_2, \ldots, i_N)$ are of the same parity and

$$\varepsilon_{i_1, i_2, \ldots, i_N} = \varepsilon_{j_1, j_2, \ldots, j_N} \qquad \textbf{(EQ 89)}$$

Thus, we can write the determinant of A^T as

$$det(A^T) = \sum_{j_1=1}^{N}\sum_{j_2=1}^{N}\ldots\sum_{j_N=1}^{N}\varepsilon_{j_1,j_2,\ldots,j_N}a_{j_1,1}a_{j_2,2}\ldots a_{j_N,N} = det(A) \qquad \textbf{(EQ 90)}$$

Q.E.D.

Property II

If every element in a row (column) of A is zero, $det(A) = 0$.

Proof:

Let every element in column #m of A be zero. In the formula for $det(A)$,

$$det(A) = \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\ldots\sum_{i_N=1}^{N}\varepsilon_{i_1,i_2,\ldots,i_N}a_{i_1,1}a_{i_2,2}\ldots a_{i_m,m}\ldots a_{i_N,N} \qquad \textbf{(EQ 91)}$$

every term in the summation contains in its product $a_{i_1,1}a_{i_2,2}\ldots a_{i_m,m}\ldots a_{i_N,N}$ an element of column #m that is zero. Therefore, every term in the summation is zero, and $det(A) = 0$. From property I, $det(A^T) = det(A)$, we see that

if every element in row #*m* of *A* is zero, every element in column #*m* of A^T will be zero, and $det(A^T) = 0 = det(A)$.

Q.E.D.

Property III

If every element in a row (column) of a matrix *A* is multiplied by a scalar *c* to form a matrix *B*, then $det(B) = c \times det(A)$.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mN} \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \qquad B = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ \vdots & \vdots & & \vdots \\ (ca_{m1}) & (ca_{m2}) & \dots & (ca_{mN}) \\ \vdots & \vdots & & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \quad \textbf{(EQ 92)}$$

Property IV

If two rows (columns) of *A* are interchanged to form a matrix *B*, $det(B) = -det(A)$.

Proof:

Let us interchange columns #*r* and #*s*, with $r < s$,

$$A = \begin{bmatrix} a_{11} & \dots & a_{1r} & \dots & a_{1s} & \dots & a_{1N} \\ a_{21} & \dots & a_{2r} & \dots & a_{2s} & \dots & a_{2N} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{N1} & \dots & a_{Nr} & \dots & a_{Ns} & \dots & a_{NN} \end{bmatrix} \quad B = \begin{bmatrix} a_{11} & \dots & a_{1s} & \dots & a_{1r} & \dots & a_{1N} \\ a_{21} & \dots & a_{2s} & \dots & a_{2r} & \dots & a_{2N} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{N1} & \dots & a_{Ns} & \dots & a_{Nr} & \dots & a_{NN} \end{bmatrix} \quad \textbf{(EQ 93)}$$

The determinant of *B* is

$$det(B) = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \dots \sum_{i_N=1}^{N} \varepsilon_{i_1, \dots, i_r, \dots, i_s, \dots, i_N} a_{i_1,1} \dots a_{i_r,s} \dots a_{i_s,r} \dots a_{i_N,N} \quad \textbf{(EQ 94)}$$

If we exchange i_r and i_s in the permutation,

$$\varepsilon_{i_1, \ldots, i_r, \ldots, i_s, \ldots, i_N} = -\varepsilon_{i_1, \ldots, i_s, \ldots, i_r, \ldots, i_N} \quad \textbf{(EQ 95)}$$

so that we way write $det(B)$ as

$$det(B) = -\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1, \ldots, i_s, \ldots, i_r, \ldots, i_N} a_{i_1,1}\ldots a_{i_r,s}\ldots a_{i_s,r}\ldots a_{i_N,N} \quad \textbf{(EQ 96)}$$

We now relabel the dummy indices, switching everywhere $i_r \Leftrightarrow i_s$,

$$det(B) = -\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1, \ldots, i_r, \ldots, i_s, \ldots, i_N} a_{i_1,1}\ldots a_{i_s,s}\ldots a_{i_r,r}\ldots a_{i_N,N} \quad \textbf{(EQ 97)}$$

Therefore, $det(B) = -det(A)$. That the same property holds for exchanging two rows follows from $det(A^T) = det(A)$.

Q.E.D.

Property V

If two rows (columns) of *A* are the same, then $det(A) = 0$.

Proof:

Let *B* be the matrix that is obtained by switching the two identical rows or columns of *A*. By property IV, $det(B) = -det(A)$, but since the two exchanged rows (columns) are identical, $B = A$, and $det(B) = det(A)$. Therefore, we must have $det(A) = 0$.

Q.E.D.

Property VI

If $\boldsymbol{a}^{(m)}$ is the row vector for row #*m* of matrix *A*, and we decompose this row vector as

$$\boldsymbol{a}^{(m)} = \boldsymbol{b}^{(m)} + \boldsymbol{d}^{(m)} \qquad \textbf{(EQ 98)}$$

to form the matrices

$$A = \begin{bmatrix} \text{-----}\boldsymbol{a}^{(1)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{a}^{(m)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{a}^{(N)}\text{-----} \end{bmatrix} \quad B = \begin{bmatrix} \text{-----}\boldsymbol{a}^{(1)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{b}^{(m)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{a}^{(N)}\text{-----} \end{bmatrix} \quad D = \begin{bmatrix} \text{-----}\boldsymbol{a}^{(1)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{d}^{(m)}\text{-----} \\ \vdots \\ \text{-----}\boldsymbol{a}^{(N)}\text{-----} \end{bmatrix} \qquad \textbf{(EQ 99)}$$

then

$$det(A) = det(B) + det(D) \qquad \textbf{(EQ 100)}$$

Proof:

We write the determinant of *A* as

$$\begin{aligned} det(A) &= \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1,i_2,\ldots,i_N}a_{i_1,1}a_{i_2,2}\ldots(b_{i_m,m}+d_{i_m,m})\ldots a_{i_N,N} \\ &= \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1,i_2,\ldots,i_N}a_{i_1,1}a_{i_2,2}\ldots(b_{i_m,m})\ldots a_{i_N,N} \\ &\quad + \sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\cdots\sum_{i_N=1}^{N}\varepsilon_{i_1,i_2,\ldots,i_N}a_{i_1,1}a_{i_2,2}\ldots(d_{i_m,m})\ldots a_{i_N,N} \\ det(A) &= det(B) + det(D) \end{aligned} \qquad \textbf{(EQ 101)}$$

Property VII

If a matrix B is obtained from A by adding c times one row (column) of A to another row (column) of A, then $det(B) = det(A)$. That is, elementary row operations do not change the value of the determinant.

Proof:

Let us define the following matrices in terms of their row vectors,

$$A = \begin{bmatrix} ---\boldsymbol{a}^{(1)}--- \\ \vdots \\ ---\boldsymbol{a}^{(j)}--- \\ \vdots \\ ---\boldsymbol{a}^{(k)}--- \\ \vdots \\ ---\boldsymbol{a}^{(N)}--- \end{bmatrix} \qquad B = \begin{bmatrix} ---\boldsymbol{a}^{(1)}--- \\ \vdots \\ ---\boldsymbol{a}^{(j)}--- \\ \vdots \\ ---[\boldsymbol{a}^{(k)} + c\boldsymbol{a}^{(j)}]--- \\ \vdots \\ ---\boldsymbol{a}^{(N)}--- \end{bmatrix}$$

$$D = \begin{bmatrix} ---\boldsymbol{a}^{(1)}--- \\ \vdots \\ ---\boldsymbol{a}^{(j)}--- \\ \vdots \\ ---[c\boldsymbol{a}^{(j)}]--- \\ \vdots \\ ---\boldsymbol{a}^{(N)}--- \end{bmatrix} \qquad E = \begin{bmatrix} ---\boldsymbol{a}^{(1)}--- \\ \vdots \\ ---\boldsymbol{a}^{(j)}--- \\ \vdots \\ ---\boldsymbol{a}^{(j)}--- \\ \vdots \\ ---\boldsymbol{a}^{(N)}--- \end{bmatrix} \qquad \text{(EQ 102)}$$

By property IV, $det(B) = det(A) + det(D)$. By property III, $det(D) = c \times det(E)$, so that $det(B) = det(A) + c \times det(E)$. But, since two rows of E are the same, $det(E) = 0$. Therefore, $det(B) = det(A)$.

Q.E.D.

Property VIII

$$det(AB) = det(A) \times det(B) \qquad \textbf{(EQ 103)}$$

Proof:

We demonstrate this property for 2×2 matrices. Expansion by minors may be used to show that this property extends to matrices of higher dimension. For the 2×2 matrices

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \qquad C = AB \qquad \textbf{(EQ 104)}$$

we use $c_{ij} = a_{i1}b_{1j} + a_{i2}b_{2j}$ to write $det(C) = c_{11}c_{22} - c_{21}c_{12}$ as

$$\begin{aligned} det(C) = {} & (a_{11}b_{11} + a_{12}b_{21})(a_{21}b_{12} + a_{22}b_{22}) \\ & - (a_{21}b_{11} + a_{22}b_{21})(a_{11}b_{12} + a_{12}b_{22}) \end{aligned} \qquad \textbf{(EQ 105)}$$

Multiplying the sums yields

$$\begin{aligned} det(C) = {} & (a_{11}a_{21})(b_{11}b_{12}) + (a_{11}a_{22})(b_{11}b_{22}) \\ & + (a_{12}a_{21})(b_{21}b_{12}) + (a_{12}a_{22})(b_{21}b_{22}) \\ & - (a_{21}a_{11})(b_{11}b_{12}) - (a_{21}a_{12})(b_{11}b_{22}) \\ & - (a_{22}a_{11})(b_{21}b_{12}) - (a_{22}a_{12})(b_{21}b_{22}) \end{aligned} \qquad \textbf{(EQ 106)}$$

Collecting terms gives

$$\begin{aligned} det(C) = {} & (a_{11}a_{22})(b_{11}b_{22} - b_{21}b_{12}) \\ & + (a_{11}a_{21})(b_{11}b_{12} - b_{11}b_{12}) \\ & + (a_{12}a_{21})(b_{21}b_{12} - b_{11}b_{22}) \\ & + (a_{12}a_{22})(b_{21}b_{22} - b_{21}b_{22}) \end{aligned} \qquad \textbf{(EQ 107)}$$

Canceling out the zero terms and collecting the remainder yields

$$det(C) = (a_{11}a_{22} - a_{12}a_{21})(b_{11}b_{22} - b_{12}b_{22})$$
$$det(C) = det(AB) = det(A) \times det(B) \quad \textbf{(EQ 108)}$$

Property IX

If A is upper triangular or lower triangular, $det(A)$ equals the product of the elements on the principal diagonal, $det(A) = a_{11} \times a_{22} \times \ldots \times a_{NN}$.

Proof:

Let us consider a lower triangular matrix

$$L = \begin{bmatrix} L_{11} & & & \\ L_{21} & L_{22} & & \\ \vdots & \vdots & \ddots & \\ L_{N1} & L_{N2} & \ldots & L_{NN} \end{bmatrix} \quad \textbf{(EQ 109)}$$

whose determinant is

$$det(L) = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \ldots \sum_{i_N=1}^{N} \varepsilon_{i_1, i_2, \ldots, i_N} L_{i_1, 1} L_{i_2, 2} \ldots L_{i_N, N} \quad \textbf{(EQ 110)}$$

Among the terms in this summation, there is one with $i_1 = 1$, $i_2 = 2$, *etc.* with $\varepsilon_{i_1, i_2, \ldots, i_N} = \varepsilon_{1, 2, \ldots, N} = 1$. All other terms in the summation must have at least one off-diagonal element. For any term with no two indices alike (and thus $\varepsilon_{i_1, i_2, \ldots, i_N} \neq 0$), if there is an element from below the diagonal in the product, there also must be one from above the diagonal. Since all elements above the diagonal are zero, the only non-zero product of elements contributing to $det(L)$ is that containing the diagonal elements, and

$$det(L) = L_{11} \times L_{22} \times \ldots \times L_{NN} \quad \textbf{(EQ 111)}$$

Similar reasoning holds for an upper triangular matrix.

Q.E.D.

The following table shows how the desired characteristics that we have required for $det(A)$ follow from the properties stated above.

characteristic # 1 $\Leftarrow$ property III
characteristic # 2 $\Leftarrow$ property IV
characteristic # 3 $\Leftarrow$ property IV
characteristic # 4 $\Leftarrow$ property VII
characteristic # 5 $\Leftarrow$ property VIII
characteristic # 6 $\Leftarrow$ properties II, V
characteristic # 7 $\Leftarrow$ properties II, V

Nonlinear algebraic systems: Formal convergence properties of Newton's method for systems of multiple equations

As is the case with a single equation, Newton's method for multiple equations converges quadratically when the current estimate is in the close vicinity of the solution; however, the proof is somewhat more involved. Let $\boldsymbol{x}^{[k]}$ be our current estimate of the solution $\boldsymbol{x}_s$, with $\boldsymbol{f}(\boldsymbol{x}_s) = \boldsymbol{0}$. Let us assume that $\boldsymbol{x}^{[k]}$ and $\boldsymbol{x}_s$ are both within some convex region Ω, and that within Ω, $\boldsymbol{f}(\boldsymbol{x})$ is continuously differentiable. By ***convex***, we mean that Ω has the property that any line segment connecting two points in Ω lies completely in Ω. An example of a convex region would be a hypersphere of some specified radius about $\boldsymbol{x}_s$ that contains $\boldsymbol{x}^{[k]}$.

We define the current error vector as

$$\boldsymbol{\varepsilon}^{[k]} = \boldsymbol{x}^{[k]} - \boldsymbol{x}_s \qquad \boldsymbol{x}^{[k]} = \boldsymbol{x}_s + \boldsymbol{\varepsilon}^{[k]} \qquad \textbf{(EQ 112)}$$

Because $\boldsymbol{f}(\boldsymbol{x})$ is continuously differentiable in Ω, we can define the Jacobian matrix for all points along the line segment connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}^{[k]}$; that is, for every $s \in [0, 1]$, we can compute the Jacobian elements

$$J_{mn}(\boldsymbol{x}_s + s\boldsymbol{\varepsilon}^{[k]}) = \left.\frac{\partial f_m}{\partial x_n}\right|_{\boldsymbol{x}_s + s\boldsymbol{\varepsilon}^{[k]}} \qquad \textbf{(EQ 113)}$$

From the Jacobian, we compute the derivative of each component of the function vector with respect to s, the fractional path length along the line segment running from $\boldsymbol{x}_s$ at $s = 0$ to $\boldsymbol{x}^{[k]}$ at $s = 1$. Using the chain rule

$$\frac{df_i}{ds} = \frac{\partial f_i}{\partial x_1}\frac{dx_1}{ds} + \frac{\partial f_i}{\partial x_2}\frac{dx_2}{ds} + \ldots + \frac{\partial f_i}{\partial x_N}\frac{dx_N}{ds} \qquad \textbf{(EQ 114)}$$

and the fact that along the line segment $\boldsymbol{x} = \boldsymbol{x}_s + s\boldsymbol{\varepsilon}^{[k]}$,

$$\frac{dx_j}{ds} = \varepsilon_j^{[k]} \qquad \textbf{(EQ 115)}$$

we have

$$\frac{df_i}{ds} = J_{i1}\varepsilon_1^{[k]} + J_{i2}\varepsilon_2^{[k]} + \ldots + J_{iN}\varepsilon_N^{[k]} = (J\boldsymbol{\varepsilon}^{[k]})_i \qquad \textbf{(EQ 116)}$$

The variation in f_i over the line segment connecting $\boldsymbol{x}_s$ and $\boldsymbol{x}^{[k]}$ is obtained from the path integral

$$f_i(\boldsymbol{x}^{[k]}) - f_i(\boldsymbol{x}_s) = \int_0^1 \left.\frac{df_i}{ds}\right|_{\boldsymbol{x}_s + s\boldsymbol{\varepsilon}^{[k]}} ds = \int_0^1 [J(\boldsymbol{x}_s + s\boldsymbol{\varepsilon}^{[k]})\boldsymbol{\varepsilon}^{[k]}]_i ds \qquad \textbf{(EQ 117)}$$

Since at the solution, $f_i(\boldsymbol{x}_s) = 0$, this yields

$$\underline{f}(\boldsymbol{x}^{[k]}) = \int_0^1 J(\boldsymbol{x}_s + s\varepsilon^{[k]})\varepsilon^{[k]} ds \qquad \textbf{(EQ 118)}$$

In Newton's method, we update the estimate of the solution by the rule

$$J^{[k]}\Delta\boldsymbol{x}^{[k]} = -\underline{f}(\boldsymbol{x}^{[k]}) \qquad \textbf{(EQ 119)}$$

Therefore, this relation becomes

$$-J^{[k]}\Delta\boldsymbol{x}^{[k]} = \int_0^1 J(\boldsymbol{x}_s + s\varepsilon^{[k]})\varepsilon^{[k]} ds \qquad \textbf{(EQ 120)}$$

The update vector is related to the error vectors by

$$\Delta\boldsymbol{x}^{[k]} = \boldsymbol{x}^{[k+1]} - \boldsymbol{x}^{[k]} = \varepsilon^{[k+1]} - \varepsilon^{[k]} \qquad \textbf{(EQ 121)}$$

therefore,

$$\begin{aligned} -J^{[k]}[\varepsilon^{[k+1]} - \varepsilon^{[k]}] &= \int_0^1 J(\boldsymbol{x}_s + s\varepsilon^{[k]})\varepsilon^{[k]} ds \\ -J^{[k]}\varepsilon^{[k+1]} &= \int_0^1 [J(\boldsymbol{x}_s + s\varepsilon^{[k]}) - J^{[k]}]\varepsilon^{[k]} ds \\ -J^{[k]}\varepsilon^{[k+1]} &= \left\{ \int_0^1 [J(\boldsymbol{x}_s + s\varepsilon^{[k]}) - J^{[k]}] ds \right\} \varepsilon^{[k]} \end{aligned} \qquad \textbf{(EQ 122)}$$

Assuming that the Jacobian matrix is non-singular at $\boldsymbol{x}^{[k]}$,

$$-\varepsilon^{[k+1]} = \left\{ (J^{[k]})^{-1} \int_0^1 [J(\boldsymbol{x}_s + s\varepsilon^{[k]}) - J^{[k]}] ds \right\} \varepsilon^{[k]} \qquad \textbf{(EQ 123)}$$

As the quantity within the {} on the right is merely a matrix, we use the concept of a matrix norm,

$$\|A\| \equiv \max_{\boldsymbol{v} \neq \boldsymbol{0}} \frac{\|A\boldsymbol{v}\|}{\|\boldsymbol{v}\|} \qquad \textbf{(EQ 124)}$$

and the identities

$$\begin{aligned} \|A\boldsymbol{v}\| &\leq \|A\| \|\boldsymbol{v}\| \\ \|AB\boldsymbol{v}\| \leq \|A\| \ \|B\boldsymbol{v}\| &\leq \|A\| \ \|B\| \ \|\boldsymbol{v}\| \end{aligned} \qquad \textbf{(EQ 125)}$$

to write

$$\left\|\varepsilon^{[k+1]}\right\| \leq \left\|\left(J^{[k]}\right)^{-1}\right\| \left\| \int_0^1 [J(\boldsymbol{x}_s + s\varepsilon^{[k]}) - J^{[k]}]ds \right\| \left\|\varepsilon^{[k]}\right\| \qquad \textbf{(EQ 126)}$$

We now make the additional assumption that not only is $J(\boldsymbol{x})$ continuous in Ω, but that it is also ***Lipschitz continuous***; that is, for some finite L,

$$\|J(\boldsymbol{x}) - J(\boldsymbol{y})\| \leq L \|\boldsymbol{x} - \boldsymbol{y}\| \qquad \forall \boldsymbol{x}, \boldsymbol{y} \in \Omega \qquad \textbf{(EQ 127)}$$

so that,

$$\left\| \int_0^1 [J(\boldsymbol{x}_s + s\varepsilon^{[k]}) - J^{[k]}]ds \right\| \leq L|s| \left\|\varepsilon^{[k]}\right\| \qquad \textbf{(EQ 128)}$$

We therefore find that

$$\left\|\varepsilon^{[k+1]}\right\| \leq L|s| \left\|\left(J^{[k]}\right)^{-1}\right\| \left\|\varepsilon^{[k]}\right\|^2 \qquad \textbf{(EQ 129)}$$

This establishes quadratic convergence, with rapid reduction of the error within a few iterations when the current estimate is sufficiently close to the solution. Although we have proved that such a quadratic region exists, in practice, we may have to be very close to the solution to observe it.

Matrix eigenvalue analysis: Proof that the trace equals the sum of eigenvalues

We now show that the trace of a real matrix A equals the sum of its eigenvalues,

$$tr(A) = a_{11} + a_{22} + \ldots + a_{NN} = \lambda_1 + \lambda_2 + \ldots + \lambda_N \quad \textbf{(EQ 130)}$$

We first use the property $det(A) = (-1)^N det(-A)$ to write

$$det(A - \lambda I) = (-1)^N det(\lambda I - A) \quad \textbf{(EQ 131)}$$

We now write this determinant explicitly,

$$det(A - \lambda I) = (-1)^N \sum_{i_1 = 1}^{N} \sum_{i_2 = 1}^{N} \ldots \sum_{i_N = 1}^{N} (\varepsilon_{i_1, i_2, \ldots, i_N})(\lambda \delta_{i_1, 1} - a_{i_1, 1}) \ldots (\lambda \delta_{iN, N} - a_{iN, N}) \quad \textbf{(EQ 132)}$$

We isolate the contribution from the term with $i_1 = 1, i_2 = 2, \ldots, i_N = N$,

$$det(A - \lambda I) = (-1)^N [\varepsilon_{1, 2, \ldots, N}(\lambda - a_{11})(\lambda - a_{22}) \ldots (\lambda - a_{NN}) + R(\lambda)] \quad \textbf{(EQ 133)}$$

$R(\lambda)$ is some polynomial of at most degree $N - 1$. As $\varepsilon_{1, 2, \ldots, N} = 1$,

$$det(A - \lambda I) = (-1)^N [(\lambda - a_{11})(\lambda - a_{22}) \ldots (\lambda - a_{NN}) + R(\lambda)] \quad \textbf{(EQ 134)}$$

Actually, $R(\lambda)$ must be a polynomial of at most degree $N - 2$, because for any possible term contributing a non-zero amount to $R(\lambda)$, no two $\{i_1, i_2, \ldots, i_N\}$ be identical. Therefore, any off-diagonal element of $(A - \lambda I)$ in such a term must be paired with another off-diagonal element. Thus, there can be no term in $R(\lambda)$ proportional to λ^{N-1}, as such a term must have only one off-diagonal element. As the degree of $R(\lambda)$ is then at most $N - 2$, we write it as $R_{N-2}(\lambda)$. We next write

$$(\lambda - a_{11})(\lambda - a_{22})\ldots(\lambda - a_{NN}) = \lambda^N - (a_{11} + a_{22} + \ldots + a_{NN})\lambda^{N-1} + S_{N-2}(\lambda) \quad \textbf{(EQ 135)}$$

where $S_{N-2}(\lambda)$ is some polynomial of degree $N-2$. (EQ 134) then becomes,

$$det(A - \lambda I) = (-1)^N[\lambda^N - (a_{11} + a_{22} + \ldots + a_{NN})\lambda^{N-1} + S_{N-2}(\lambda) + R_{N-2}(\lambda)] \quad \textbf{(EQ 136)}$$

If $\lambda_1, \ldots, \lambda_N$ are the roots of the characteristic polynomial, we can write

$$det(A - \lambda I) = (-1)^N det(\lambda I - A) = (-1)^N[(\lambda - \lambda_1)(\lambda - \lambda_2)\ldots(\lambda - \lambda_N)]$$
$$det(A - \lambda I) = (-1)^N[\lambda^N - (\lambda_1 + \lambda_2 + \ldots + \lambda_N)\lambda^{N-1} + T_{N-2}(\lambda)] \quad \textbf{(EQ 137)}$$

where $T_{N-2}(\lambda)$ is again some polynomial of degree $N-2$.

Since (EQ 136) and (EQ 137) both represent the same polynomial, comparing the coefficient that multiplies λ^{N-1} shows that the trace is equal to the sum of the eigenvalues,

$$tr(A) = \lambda_1 + \lambda_2 + \ldots + \lambda_N \quad \textbf{(EQ 138)}$$

Q.E.D.

Matrix eigenvalue analysis: Proof of Gershgorin's theorem

We now prove Gershgorin's theorem, starting first with the following:

Theorem:

If *B* is an arbitrary $N \times N$ matrix, then for all eigenvalues λ of a $N \times N$ matrix *A*,

$$1 \le \left\| (\lambda I - B)^{-1}(A - B) \right\| \le \left\| (\lambda I - B)^{-1} \right\| \left\| A - B \right\| \qquad \textbf{(EQ 139)}$$

unless λ is also an eigenvalue of *B*.

Proof:

Let $\boldsymbol{w}$ be an eigenvector for λ, $A\boldsymbol{w} = \lambda\boldsymbol{w}$, then

$$(A - B)\boldsymbol{w} = (\lambda I - B)\boldsymbol{w} \qquad \textbf{(EQ 140)}$$

If λ is not an eigenvalue of *B*, $det(\lambda I - B) \neq 0$, and we may write

$$(\lambda I - B)^{-1}(A - B)\boldsymbol{w} = \boldsymbol{w} \qquad \textbf{(EQ 141)}$$

We now take the norm of both sides,

$$\left\| (\lambda I - B)^{-1}(A - B)\boldsymbol{w} \right\| = \left\| \boldsymbol{w} \right\| \qquad \textbf{(EQ 142)}$$

and apply the definition of a matrix norm,

$$\frac{\left\| (\lambda I - B)^{-1}(A - B)\boldsymbol{w} \right\|}{\left\| \boldsymbol{w} \right\|} = 1 \le \left\| (\lambda I - B)^{-1}(A - B) \right\| \qquad \textbf{(EQ 143)}$$

We now use the general property $\|AB\| \le \|A\|\|B\|$ to obtain (EQ 139).

Q.E.D.

We use this theorem to prove Gershgorin's theorem by selecting

$$B = diag(A) = \begin{bmatrix} a_{11} & & & \\ & a_{22} & & \\ & & . & \\ & & & a_{NN} \end{bmatrix} \qquad \textbf{(EQ 144)}$$

for which

$$\lambda I - B = \begin{bmatrix} \lambda - a_{11} & & & \\ & \lambda - a_{22} & & \\ & & . & \\ & & & \lambda - a_{NN} \end{bmatrix} \quad \textbf{(EQ 145)}$$

Inverting this diagonal matrix,

$$(\lambda I - B)^{-1} = \begin{bmatrix} (\lambda - a_{11})^{-1} & & & \\ & (\lambda - a_{22})^{-1} & & \\ & & . & \\ & & & (\lambda - a_{NN})^{-1} \end{bmatrix} \quad \textbf{(EQ 146)}$$

and noting that

$$(A - B) = \begin{bmatrix} 0 & a_{12} & \dots & a_{1N} \\ a_{21} & 0 & \dots & a_{2N} \\ : & : & . & : \\ a_{N1} & a_{N2} & \dots & 0 \end{bmatrix} \quad \textbf{(EQ 147)}$$

we obtain by matrix multiplication

$$(\lambda I - B)^{-1}(A - B) = \begin{bmatrix} 0 & (\lambda - a_{11})^{-1} a_{12} & (\lambda - a_{11})^{-1} a_{13} & \dots & (\lambda - a_{11})^{-1} a_{1N} \\ (\lambda - a_{22})^{-1} a_{21} & 0 & (\lambda - a_{22})^{-1} a_{23} & \dots & (\lambda - a_{22})^{-1} a_{2N} \\ : & : & : & & : \\ (\lambda - a_{NN})^{-1} a_{N1} & (\lambda - a_{NN})^{-1} a_{N2} & (\lambda - a_{NN})^{-1} a_{N3} & \dots & 0 \end{bmatrix} \quad \textbf{(EQ 148)}$$

Selecting the infinity norm for $\|\boldsymbol{v}\|$, the corresponding matrix norm is

$$\|A\|_\infty = \max_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\max_j\{|(A\boldsymbol{v})_j|\}}{\max_j\{|v_j|\}} = \max_{\boldsymbol{v}\neq\boldsymbol{0}} \frac{\max_j\left\{\left|\sum_{k=1}^{N} a_{jk}v_k\right|\right\}}{\max_j\{|v_j|\}} = \max_j\left\{\sum_{k=1}^{N}|a_{jk}|\right\} \quad \textbf{(EQ 149)}$$

where in the last equality we have chosen $\boldsymbol{v} = \begin{bmatrix}\pm c & \pm c & \ldots & \pm c\end{bmatrix}^T$ such that for each $j = 1, 2, \ldots, N$, $a_{jk}v_k > 0$. We now apply this matrix norm to $(\lambda I - B)^{-1}(A - B)$, that has the elements

$$[(\lambda I - B)^{-1}(A - B)]_{jk} = \begin{cases} 0, \text{ if } j = k \\ (\lambda - a_{jj})^{-1}a_{jk}, \text{ if } j \neq k \end{cases} \quad \textbf{(EQ 150)}$$

to yield

$$\left\|(\lambda I - B)^{-1}(A - B)\right\|_\infty = \max_j\left\{\sum_{k=1}^{N}(\lambda - a_{jj})^{-1}a_{jk}\right\} = \max_j\left\{\sum_{\substack{k=1\\k\neq j}}^{N}\frac{|a_{jk}|}{|\lambda - a_{jj}|}\right\} \quad \textbf{(EQ 151)}$$

From (EQ 139), $\left\|(\lambda I - B)^{-1}(A - B)\right\| \geq 1$, we have

$$\max_j\left\{\sum_{\substack{k=1\\k\neq j}}^{N}\frac{|a_{jk}|}{|\lambda - a_{jj}|}\right\} \geq 1 \quad \textbf{(EQ 152)}$$

Thus, for each eigenvalue λ of A, there must be some $j \in [1, N]$ such that

$$|\lambda - a_{jj}|^{-1}\sum_{\substack{k=1\\k\neq j}}^{N}|a_{jk}| \geq 1 \quad \textbf{(EQ 153)}$$

This proves Gershgorin's theorem,

$$|\lambda - a_{jj}| \leq \sum_{\substack{k=1 \\ k \neq j}}^{N} |a_{jk}| \qquad \text{for some } j \in [1, N] \qquad \textbf{(EQ 154)}$$

Q.E.D.

Matrix eigenvalue analysis: Applications in Quantum Mechanics

In chapter 3 of the text, we demonstrate how eigenvalue analysis is used to compute the energy and wavefunction of a single-electron system in quantum mechanics. This section provides some further background discussion of the role of eigenvalue analysis in quantum mechanics.

Consider the simple case of an electron moving in a potential energy field $V(\boldsymbol{r})$ at a momentum $\boldsymbol{p} = m_e \boldsymbol{v}$. The classical expression for the energy of the electron is

$$E_{\text{class}} = \frac{1}{2m_e}(\boldsymbol{p} \cdot \boldsymbol{p}) + V(\boldsymbol{r}) \qquad \textbf{(EQ 155)}$$

This formula assumes that we can specify exactly the position and momentum of the electron, and Newton's second law of motion, $\frac{d\boldsymbol{p}}{dt} = -\nabla V$, assumes that these exactly-defined quantities vary continuously with time. At the beginning of the 20$^{\text{th}}$ century, experiments showed that this was not the case and guided the development of quantum mechanics in the 1920's.

The results can be summed up by considering the case of electrons passing through slits or holes in a barrier (Figure 2). We have two slits in a barrier, and consider a beam of electrons arriving from the left. The barrier stops the electrons unless they pass through one of the openings. If we were to place a screen to the right of the barrier and count the number of electrons that hit

the screen at each location, we expect from classical mechanics to see peaks only directly behind each slit (*left*). But, in the experiment, the density of observed hits on the screen shows an interference pattern more to be expected from the behavior of waves (*right*). This pattern appears even if the flux of electrons is so small that only one electron passes through the system at a time; therefore, this behavior is inherent in the nature of the electron itself. Further experiments show that a particle moving at a momentum p has some of the properties of a wave with the wavelength given by the ***de Broglie relation*** $\lambda = h/p$ where h is ***Planck's constant*** - a fundamental physical constant.

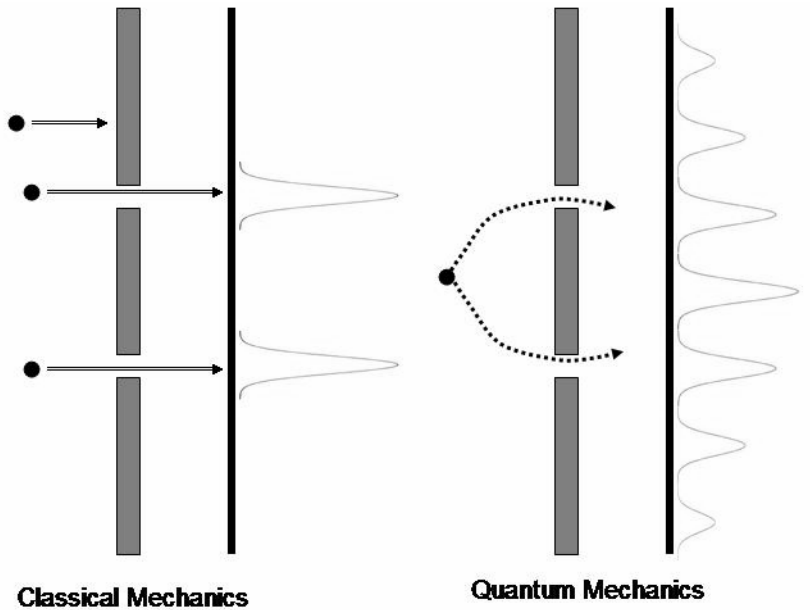

FIGURE 2. Passage of electrons through a hole in a barrier. At left, the expectation of classical mechanics. At right the experimentally-observed interference pattern for single-electron passage.

The prescribed procedure for obtaining the quantum mechanical description of the electron is to define a complex-valued wavefunction $\Psi(\boldsymbol{r}, t)$ such that

the probability that the electron is observed in a differential volume dV around the position $\boldsymbol{r}$ at time t is

$$|\Psi(\boldsymbol{r}, t)|^2 dV \qquad \textbf{(EQ 156)}$$

Thus, the wavefunction must satisfy the normalization,

$$\int_{\Re^3} |\Psi(\boldsymbol{r}, t)|^2 dV = 1 \qquad \textbf{(EQ 157)}$$

We now have the problem of finding a replacement for Newton's second law of motion that describes the time-evolution of this wavefunction. While the particle has some of the properties of a wave, it does tend to be located in a compact region of space somewhere, outside of which the wavefunction decays to zero, as it must for (EQ 157) to be satisfied. Thus, we expect the wavefunction to look like a wavepacket (Figure 3).

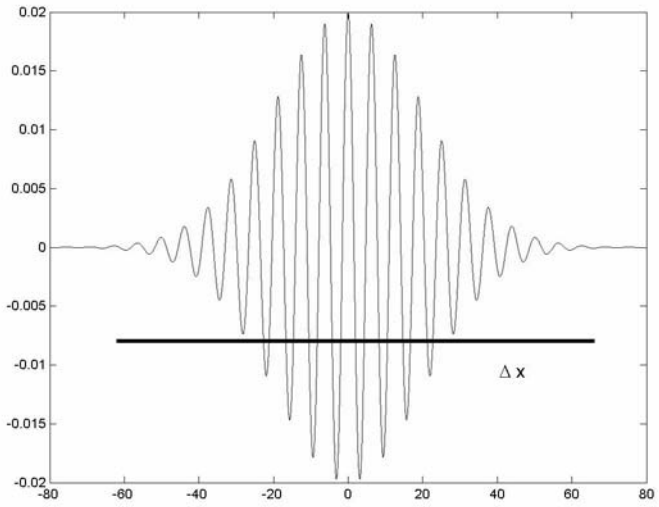

FIGURE 3. The wavefunction of a particle has wavelike properties but is somewhat localized in space, thus resembling a wavepacket.

How do we obtain the momentum of a particle from its wavefunction? From de Broglie's relation, $p = h/\lambda$, so we need to measure the wavelength. If the width of the region of observable non-zero Ψ is Δx, let N be the observed number of cycles of the wave in this region. As $\Delta x = N\lambda$, we have an observed momentum of

$$p \approx \frac{Nh}{\Delta x} \qquad \textbf{(EQ 158)}$$

But, it is clear that we cannot obtain a perfect observation of N since the amplitude goes to zero near the edges of the observation region, thus we have an observation error of at best $\Delta N = \pm 1$. This yields an uncertainty in the observed momentum of $\Delta p \approx h/(\Delta x)$, so that the uncertainties in the observed position and in the observed momentum satisfy the ***Heisenberg uncertainty relation***,

$$\Delta x \Delta p > \sim h \qquad \textbf{(EQ 159)}$$

We now consider how to perform this observation mathematically. Consider the case of a wavefunction with the spatial modulation,

$$\Psi(x, t) \propto e^{ikx} = \cos(kx) + i\sin(kx) \qquad \textbf{(EQ 160)}$$

The wavelength is specified exactly, $\lambda = 2\pi/k$, and so the momentum is known exactly, $p = \bar{h}k$, $\bar{h} = h/(2\pi)$. As $\Delta p = 0$, Heisenberg uncertainty requires $\Delta x = \infty$ as indeed it is for this non-localized wavefunction. Note that this is somewhat of a hypothetical case, as particles like electrons have finite Δx and Δp.

What mathematical operation could we define to extract the momentum value from such a state? We note that since the first derivative is

$$\frac{de^{ikx}}{dx} = ike^{ikx} = i(p/\bar{h})e^{ikx} \qquad \textbf{(EQ 161)}$$

we see that e^{ikx} satisfies the equation,

$$-i\bar{h}\frac{de^{ikx}}{dx} = pe^{ikx} \qquad \textbf{(EQ 162)}$$

That is, the momentum is exacted from the wavefunction Ψ by finding the eigenvalue of a differential ***linear momentum operator*** $\hat{p}$,

$$\hat{p}\psi = p\Psi \qquad \hat{p} = -i\bar{h}\frac{d}{dx} \qquad \textbf{(EQ 163)}$$

This operator provides a mathematical tool to extract the momentum of a particle from its wavefunction, and provides a tie-in to eigenvalue analysis.

The general idea behind (non-relativistic) quantum mechanics is to write down first the classical expression for any observable quantity $A(\boldsymbol{r}, \boldsymbol{p})$ that depends upon the position and momentum of the particle. In this expression, we replace each momentum component p_j with the corresponding momentum operator $\hat{p}_j = -i\bar{h}\frac{\partial}{\partial x_j}$ to obtain a differential operator $\hat{A}$. Generally, before we make the measurement there is some uncertainty in the value of A, but our process of observation unavoidably changes the wavefunction of the particle during measurement. Let us say that we measure a value of A equal to α. Then, following the observation, the value of $A = \alpha$ is specified exactly, so that the wavefunction after measurement must be an eigenfunction of $\hat{A}$ satisfying

$$\hat{A}\Psi = \alpha\Psi \qquad \textbf{(EQ 164)}$$

We shall find that constraints on the wavefunction after measurement such as $\int|\Psi(\boldsymbol{r}, t)|^2 dV = 1$ or boundary conditions on Ψ may result in the measurement process being able to return only one of a countable number of discrete values $\alpha_1, \alpha_2, \ldots$.

Now, after the measurement the wavefunction will evolve with time according to some unspecified dynamics until we have a new interaction with the surroundings to measure some new observable. We generally describe the dynamic evolution of the wavefunction with respect to observations of the

energy. A particle with classical energy $E_{\text{class}} = \frac{1}{2m_e}(\boldsymbol{p} \cdot \boldsymbol{p}) + V(\boldsymbol{r})$ has a wavefunction $\Psi(\boldsymbol{r}, t)$ governed by the ***time-dependent Schrödinger equation***

$$\hat{H}\Psi = i\bar{h}\frac{\partial \Psi}{\partial t} \qquad \textbf{(EQ 165)}$$

The ***Hamiltonian operator*** $\hat{H}$ is obtained from E_{class} by making the canonical substitution for $\boldsymbol{p}$, the corresponding ***momentum operator***

$$\hat{\boldsymbol{p}} = -i\bar{h}\nabla \qquad \textbf{(EQ 166)}$$

For the case of a single electron in an external field the Hamiltonian is

$$\hat{H} = \frac{1}{2m_e}[-i\bar{h}\nabla] \cdot [-i\bar{h}\nabla] + V(\boldsymbol{r})$$
$$\hat{H} = -\frac{\bar{h}^2}{2m_e}\nabla^2 + V(\boldsymbol{r}) \qquad \textbf{(EQ 167)}$$

and the wavefunction follows the partial differential equation

$$-\frac{\bar{h}^2}{2m_e}\nabla^2\Psi(\boldsymbol{r}, t) + V(\boldsymbol{r})\Psi(\boldsymbol{r}, t) = i\bar{h}\frac{\partial \Psi}{\partial t} \qquad \textbf{(EQ 168)}$$

If the electron were interacting with nothing but the external potential $V(\boldsymbol{r})$, from classical mechanics its energy would be constant at a value E. *What is the corresponding situation for a wavefunction in quantum mechanics?* In a stationary state of constant energy, the wavefunction can be factored as

$$\Psi(\boldsymbol{r}, t) = \psi(\boldsymbol{r})g(t) \qquad \textbf{(EQ 169)}$$

Substitution into the time-dependent Schrödinger equation yields

$$-\frac{\bar{h}^2}{2m_e}[\nabla^2\psi]g + V(\boldsymbol{r})\psi g = i\bar{h}\psi\frac{dg}{dt} \qquad \textbf{(EQ 170)}$$

Upon division by the product $\psi(\boldsymbol{r})g(t)$, we find the equation to be separable,

$$-\frac{\bar{h}^2}{2m_e}\frac{1}{\psi}[\nabla^2\psi] + V(\boldsymbol{r}) = i\bar{h}\frac{1}{g}\frac{dg}{dt} = E \qquad \textbf{(EQ 171)}$$

E must be a constant since the left function does not depend upon time and the right one does not depend upon position. Since the classical situation corresponds to constant energy, we identify E as the energy of the electron, that must satisfy the ***time-independent Schrödinger equation***

$$-\frac{\bar{h}^2}{2m_e}\nabla^2\psi(\boldsymbol{r}) + V(\boldsymbol{r})\psi(\boldsymbol{r}) = \hat{H}\psi(\boldsymbol{r}) = E\psi(\boldsymbol{r}) \qquad \textbf{(EQ 172)}$$

When an electron is in a stationary energy state with $\Psi(\boldsymbol{r}, t) = \psi(\boldsymbol{r})g(t)$ and $\hat{H}\psi(\boldsymbol{r}) = E\psi(\boldsymbol{r})$, the time-dependent Schrödinger equation yields

$$i\bar{h}\frac{dg}{dt} = Eg \qquad \Rightarrow \qquad g(t) = e^{-iEt/\bar{h}} = \cos(Et/\bar{h}) - i\sin(Et/\bar{h}) \qquad \textbf{(EQ 173)}$$

Thus, although the observed energy in such a stationary state is constant, the wavefunction is not.

> According to the rules of quantum mechanics, whenever we observe the energy of a particle, we obtain a value that satisfies the eigenvalue equation $\hat{H}\psi = E\psi$ for some ψ that satisfies all constraints on the allowable wavefunction.

While above we have explained the rules of quantum mechanics for a single particle, the approach is generally valid to systems of several interacting particles. Let a system of N particles have a wavefunction $\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)$ such that the probability of finding particle 1 within a volume $d\boldsymbol{r}_1$ around $\boldsymbol{r}_1$, particle 2 within $d\boldsymbol{r}_2$ around $\boldsymbol{r}_2$, *etc.* is $|\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)|^2 d\boldsymbol{r}_1 d\boldsymbol{r}_2 \ldots d\boldsymbol{r}_N$. This must be normalized to

$$\int_{\Re^3}\int_{\Re^3}\cdots\int_{\Re^3}|\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)|^2 d\boldsymbol{r}_1 d\boldsymbol{r}_2 \ldots d\boldsymbol{r}_N = 1 \qquad \textbf{(EQ 174)}$$

The total classical energy of the multi-particle system is

$$E_{\text{class}} = \sum_{j=1}^{N} \frac{\boldsymbol{p}_j \cdot \boldsymbol{p}_j}{2m_j} + V(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) \qquad \textbf{(EQ 175)}$$

Making the replacement of momentum operators yields the time-independent Schrödinger equation for the multi-particle system

$$\hat{H}\psi = -\sum_{j=1}^{N} \frac{\bar{h}^2}{2m_j}\nabla_j^2\psi + V(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)\psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) = E\psi \qquad \textbf{(EQ 176)}$$

Unfortunately, this is a $3N$ dimensional partial differential equation, and it is infeasible to solve it for the exact wavefunction, even numerically, for $N > 1$. As we see in Chapter 5, solving even 3-D PDE's is numerically challenging, so that solving 30-D PDE's when $N = 10$ is out of the question. This has necessitated the use of approximate methods for many-electron systems such as density function theory and the Hartree-Fock self-consistent field theory that are beyond the scope of this text.

Energy states of a Hydrogen-like atom

Consider the case of an electron orbiting an atomic nucleus of atomic number Z. The potential field acting on the electron is spherically-symmetric, $V(r) = -Zq_e^2/r$. Schrödinger's equation, written in spherical coordinates, is

$$-\frac{\bar{h}^2}{2m_e}\left\{\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial\psi}{\partial r}\right) + \frac{1}{r^2}\hat{\Lambda}\psi\right\} - \frac{Zq_e^2}{r}\psi(r, \theta, \phi) = E\psi(r, \theta, \phi) \qquad \textbf{(EQ 177)}$$

where

$$\hat{\Lambda}\psi = \frac{\partial^2\psi}{\partial\theta^2} + (\cot\theta)\frac{\partial\psi}{\partial\theta} + \frac{1}{(\sin\theta)^2}\frac{\partial^2\psi}{\partial\phi^2} \qquad \textbf{(EQ 178)}$$

It may be shown analytically that the eigenfunctions satisfy

$$\psi^{[n,\,l,\,m]}(r,\theta,\phi) \;=\; R^{[n,\,l]}(r)\,Y^{[l,\,m]}(\theta,\phi) \qquad \textbf{(EQ 179)}$$

where the indices are

$$\begin{aligned} n &= 0, 1, 2, \ldots \\ l &= 0, 1, \ldots, n-1 \\ m &\in [-l, -l+1, \ldots, 0, \ldots, l-1, l] \end{aligned} \qquad \textbf{(EQ 180)}$$

If $l = 0$, Ψ is called an *s*-orbital; if $l = 1$, a *p*-orbital; if $l = 2$ a *d*-orbital; if $l = 3$, a *f*-orbital. These discrete allowable eigenvalues arise from the restriction that the wavefunction needs to be normalized to one.

The angular part of the eigenfunction is

$$Y^{[l,\,m]}(\theta,\phi) \;=\; \left[\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}\right]^{1/2} P_l^{|m|}(\cos\theta)\,e^{im\varphi} \qquad \textbf{(EQ 181)}$$

where the $P_l^{|m|}(\cos\theta)$ are ***associated Legendre functions***,

$$\begin{aligned} P_l^{|m|}(w) \;&=\; (1-w^2)^{|m|/2}\frac{d^{|m|}}{dw^{|m|}}P_l(w) \qquad P_l^0(w) \;=\; P_l(w) \\ P_l(w) \;&=\; \frac{1}{2^l l!}\frac{d^l}{dw^l}(w^2-1)^l \end{aligned} \qquad \textbf{(EQ 182)}$$

These eigenfunctions, $Y^{[l,\,m]}(\theta,\phi)$ are known as ***spherical harmonics***, and prove useful in describing orientational distributions in spherical coordinates because they satisfy the orthonormality relations

$$\int_0^{\pi}\left\{\int_0^{2\pi}[Y^{[k,\,n]}(\theta,\phi)]^*\,[Y^{[l,\,m]}(\theta,\phi)]\sin\theta\,d\theta\right\}d\phi \;=\; \delta_{kl}\delta_{nm} \qquad \textbf{(EQ 183)}$$

and thus we may write any “orientational” function as

$$f(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} f_{lm}Y^{[l,m]}(\theta,\phi)$$
$$f_{lm} = \int_0^{\pi}\left\{\int_0^{2\pi}[Y^{[l,m]}(\theta,\phi)]^* f(\theta,\phi)\sin\theta d\theta\right\}d\phi \qquad \textbf{(EQ 184)}$$

These $Y^{[l,m]}(\theta,\phi)$ have the eigenfunction properties

$$\hat{\Lambda}Y^{[l,m]}(\theta,\phi) = -l(l+1)Y^{[l,m]}(\theta,\phi) \qquad \textbf{(EQ 185)}$$

As $e^{im\phi} = \cos(m\phi) + i\sin(m\varphi)$ and $\hat{\Lambda}Y^{[l,m]}(\theta,\phi) = \hat{\Lambda}Y^{[l,-m]}(\theta,\phi)$, we can write

$$Y^{[l,m]}(\theta,\phi) = \left[\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}\right]^{1/2}P_l^{|m|}(\cos\theta)S_m(\phi) \qquad \textbf{(EQ 186)}$$

where

$$S_m(\phi) = e^{im\varphi} \qquad \text{or} \qquad S_m(\phi) = \begin{cases}\cos(m\phi), m \geq 0\\ \sin(m\varphi), m < 0\end{cases} \qquad \textbf{(EQ 187)}$$

The radial eigenfunctions are

$$R^{[n,l]}(r) = \exp\left\{-\frac{Zq_e^2m_e}{n\bar{h}^2}(r)\right\}r^l\sum_{j=0}^{n-l-1}a_j^{[n,l]}r^j \qquad \textbf{(EQ 188)}$$

where the coefficients satisfy the recursion formula

$$\frac{a_{j+1}^{[n,l]}}{a_j^{[n,l]}} = \frac{2Zq_e^2m_e}{n\bar{h}^2}\frac{(j+l+1-n)}{(j+1)(j+2l+2)} \qquad \textbf{(EQ 189)}$$

and $a_0^{[n,l]}$ is chosen such that $R^{[n,l]}(r)$ satisfies the normalization

$$\int_0^\infty \left|R^{[n,\,l]}(r)\right|^2 r^2 dr \;=\; 1 \qquad \textbf{(EQ 190)}$$

The energy eigenvalues depend upon the principal quantum number alone

$$E^{[n,\,l,\,m]} \;=\; -\frac{Z^2 q_e^4 m_e}{\bar{h}^2 n^2} \qquad \textbf{(EQ 191)}$$

Here, we have simply presented the analytical results; however, one could numerically compute these energy eigenstates by using the basis-set expansion method described above. To compute the necessary integrals numerically, *e.g.* using **triplequad** in MATLAB, it may be helpful to use a renormalization to avoid division by zero,

$$V(\underline{r}) \;=\; -\frac{Z q_e^2}{r+\varepsilon} \qquad \varepsilon \to 0^+ \qquad \textbf{(EQ 192)}$$

Initial value problems: Gaussian Quadrature

We now prove the theorems for Gaussian quadrature. Consult the section in the text on Gaussian quadrature for definitions of any unfamiliar terms or notation.

Theorem GQ1: For the interval $[a,\,b]$ and the weight function $w(x) \geq 0$, there exists a set of orthogonal polynomials of leading coefficient one, $p_j(x) \in {}_1\Pi_j$, $j \;=\; 0,\,1,\,2,\,\ldots$, such that $\langle p_j | p_k \rangle \;=\; 0$ for $j \neq k$. These polynomials are defined uniquely by the recursion formula

$$p_{j+1}(x) \;=\; (x-\delta_{j+1})p_j(x) - \gamma_{j+1}^2 p_{j-1}(x) \qquad \textbf{(EQ 193)}$$

where

$$p_{-1}(x) = 0 \qquad p_0(x) = 1$$
$$\delta_{j+1} = \langle xp_j | p_j \rangle / \langle p_j | p_j \rangle \qquad j = 1, 2, \ldots$$
$$\gamma_{j+1}^2 = \begin{cases} 0, & j = 0 \\ \langle p_j | p_j \rangle / \langle p_{j-1} | p_{j-1} \rangle, & j = 1, 2, \ldots \end{cases} \qquad \text{(EQ 194)}$$

Proof:

We prove this theorem by induction; that is, we suppose that we have already constructed an orthogonal basis set $\{p_0, p_1, \ldots, p_j\}$ for Π_j and then show that there exists a unique polynomial $p_{j+1} \in {}_1\Pi_{j+1}$ that is orthogonal to all previous orthogonal polynomials,

$$\langle p_j | p_k \rangle = 0 \qquad k = 0, 1, \ldots, j \qquad \text{(EQ 195)}$$

and that satisfies the recursion formulas above. If we can show this, then starting from $p_0(x) = 1 \in {}_1\Pi_0$, we can construct the set of orthogonal polynomials recursively one-by-one.

To do so, we first note that it is possible to write **<u>any</u>** polynomial of exact degree $j+1$ and a leading coefficient of one as

$$p_{j+1}(x) = (x - \delta_{j+1}) p_j(x) + c_{j-1} p_{j-1}(x) + c_{j-2} p_{j-2}(x) + \ldots + c_0 p_0(x)$$
$$p_k(x) \in {}_1\Pi_k \qquad k \le j+1 \qquad \text{(EQ 196)}$$

for some set of scalar coefficients $\{\delta_{j+1}, c_{j-1}, c_{j-2}, \ldots, c_0\}$. We take the dot product of this equation with $p_j(x)$,

$$\langle p_{j+1} | p_j \rangle = \langle xp_j | p_j \rangle - \delta_{j+1} \langle p_j | p_j \rangle + c_{j-1} \langle p_{j-1} | p_j \rangle + c_{j-2} \langle p_{j-2} | p_j \rangle + \ldots + c_0 \langle p_0 | p_j \rangle \qquad \text{(EQ 197)}$$

Since we have assumed that the set $\{p_0, p_1, \ldots, p_j\}$ is mutually orthogonal,

$$\langle p_{j-1} | p_j \rangle = \langle p_{j-2} | p_j \rangle = \ldots = \langle p_0 | p_j \rangle = 0 \qquad \text{(EQ 198)}$$

we can enforce orthogonality of $p_{j+1}(x)$ and $p_j(x)$ by setting

$$\langle p_{j+1}|p_j\rangle = \langle xp_j|p_j\rangle - \delta_{j+1}\langle p_j|p_j\rangle = 0 \quad \Rightarrow \quad \delta_{j+1} = \langle xp_j|p_j\rangle / \langle p_j|p_j\rangle \qquad \textbf{(EQ 199)}$$

Next, we take the scalar product of $p_{j+1}(x)$ with $p_{j-1}(x)$, force this scalar product to be zero, and again invoke orthogonality of the set $\{p_0, p_1, \ldots, p_j\}$,

$$\langle p_{j+1}|p_{j-1}\rangle = \langle xp_j|p_{j-1}\rangle + c_{j-1}\langle p_{j-1}|p_{j-1}\rangle = 0$$
$$-c_{j-1} = \frac{\langle xp_j|p_{j-1}\rangle}{\langle p_{j-1}|p_{j-1}\rangle} \qquad \textbf{(EQ 200)}$$

If we can $c_{j-2} = \ldots = c_0 = 0$ then $p_{j+1}(x)$ is orthogonal to all of $p_{j-2}(x)$, $p_{j-3}(x)$, ..., $p_0(x)$ as well. This yields the recursion formula

$$p_{j+1}(x) = (x - \delta_{j+1})p_j(x) + c_{j-1}p_{j-1}(x) \qquad \textbf{(EQ 201)}$$

that we put in the form of the theorem by writing

$$p_j(x) = (x - \delta_j)p_{j-1}(x) + c_{j-2}p_{j-2}(x) \qquad \textbf{(EQ 202)}$$

and taking the scalar product with $p_j(x)$,

$$\langle p_j|p_j\rangle = \langle xp_{j-1}|p_j\rangle - \delta_j\langle p_{j-1}|p_j\rangle + c_{j-2}\langle p_{j-2}|p_j\rangle \qquad \textbf{(EQ 203)}$$

Using $\langle p_{j-1}|p_j\rangle = \langle p_{j-2}|p_j\rangle = 0$ and $\langle xp_{j-1}|p_j\rangle = \langle xp_j|p_{j-1}\rangle$, we obtain

$$\langle xp_j|p_{j-1}\rangle = \langle p_j|p_j\rangle \qquad \textbf{(EQ 204)}$$

The two-term recursion formula for the orthogonal polynomials is then

$$p_{j+1}(x) = (x - \delta_{j+1})p_j(x) - \gamma_{j+1}^2 p_{j-1}(x) \qquad \textbf{(EQ 205)}$$

where

$$\delta_{j+1} = \langle xp_j|p_j\rangle / \langle p_j|p_j\rangle \qquad \gamma_{j+1}^2 = -c_{j-1} = \frac{\langle p_j|p_j\rangle}{\langle p_{j-1}|p_{j-1}\rangle} \geq 0 \qquad \textbf{(EQ 206)}$$

To start the recursion properly, we set $p_{-1}(x) = 0$ and set $\gamma_0^2 = 0$.

Q.E.D.

Theorem GQ2: The N roots $\{x_1, x_2, \ldots, x_N\}$ of the orthogonal polynomial $p_N(x)$ are real, distinct and lie in (a,b), *i.e.* $a < x_1 < x_2 < \ldots < x_N < b$. The roots of $p_N(x)$ may be computed by the eigenvalue technique of Chapter 3.

Proof:

Let us assume that only $d \le N$ of the roots of $p_N(x)$ are distinct, and write the polynomial in its factored form as

$$p_N(x) = (x - x_1)^{\nu_1}(x - x_2)^{\nu_2}\ldots(x - x_d)^{\nu_d} \qquad \textbf{(EQ 207)}$$

Let us assume that we have ordered the roots such that the first $m \le d \le N$ are real, have odd multiplicity (*i.e.* $\nu_1, \ldots, \nu_m$ are all odd), and lie within (a, b), and that the remaining $d - m$ distinct roots either are of even multiplicity or lie outside of (a, b). Therefore, within $[a, b]$, $p_N(x)$ changes sign only at the first m roots.

Let us now define the polynomial

$$q(x) = \prod_{k=1}^{m} (x - x_k)^{\nu_k} \qquad \textbf{(EQ 208)}$$

We then write the product $p_N(x)q(x)$ in factored form as

$$p_N(x)q(x) = \left[\prod_{k=1}^{d}(x-x_k)^{\nu_d}\right]\left[\prod_{k=1}^{m}(x-x_k)^{\nu_k}\right]$$
$$p_N(x)q(x) = \left[\prod_{k=1}^{m}(x-x_k)^{2\nu_k}\right]\left[\prod_{k=m+1}^{d}(x-x_k)^{\nu_k}\right] \qquad \textbf{(EQ 209)}$$

We see that $p_N(x)q(x)$ has no real roots within (a,b) of odd multiplicity and therefore that $p_N(x)q(x)$ changes sign nowhere within $[a,b]$. As $w(x) \geq 0$, we then have

$$\langle p_N|q\rangle = \int_a^b w(x)p_N(x)q(x)dx \neq 0 \qquad \textbf{(EQ 210)}$$

But, since we have constructed $p_N(x)$ to be orthogonal to every polynomial of degree less than N, we see that the degree of $q(x)$ must be N. $p_N(x)$ therefore has N distinct real roots within (a,b).

Q.E.D.

Theorem GQ3: The $N \times N$ matrix

$$A = \begin{bmatrix} p_0(x_1) & p_0(x_2) & \cdots & p_0(x_N) \\ p_1(x_1) & p_1(x_2) & \cdots & p_1(x_N) \\ \vdots & \vdots & & \vdots \\ p_{N-1}(x_1) & p_{N-1}(x_2) & \cdots & p_{N-1}(x_N) \end{bmatrix} \qquad \textbf{(EQ 211)}$$

is non-singular for any mutually distinct arguments $x_1, x_2, \ldots, x_N$.

Proof:

We present a proof by contradiction. Let us assume that A were singular. Then, there would exist a vector $\underline{c}^T = \begin{bmatrix} c_0 & c_1 & \cdots & c_{N-1} \end{bmatrix} \neq \underline{0}$ such that

$$\underline{c}^T A = \begin{bmatrix} c_0 & c_1 & \cdots & c_{N-1} \end{bmatrix} \begin{bmatrix} p_0(x_1) & p_0(x_2) & \cdots & p_0(x_N) \\ p_1(x_1) & p_1(x_2) & \cdots & p_1(x_N) \\ \vdots & \vdots & & \vdots \\ p_{N-1}(x_1) & p_{N-1}(x_2) & \cdots & p_{N-1}(x_N) \end{bmatrix} = \underline{0} \qquad \textbf{(EQ 212)}$$

This yields the N conditions for the points $\{x_1, x_2, \ldots, x_N\}$, assumed distinct,

$$\sum_{k=0}^{N-1} c_k p_k(x_j) = 0 \qquad j = 1, 2, \ldots, x_N \qquad \textbf{(EQ 213)}$$

Thus, the polynomial

$$q(x) = \sum_{j=0}^{N-1} c_j p_j(x) \qquad \textbf{(EQ 214)}$$

has at least N distinct roots; however, we see that it must be of degree $N-1$ or less. These two conditions can only be met if $q(x) = 0$.

Let m be the largest index with $c_m \neq 0$. Then, we can write

$$q(x) = 0 = c_m p_m(x) + \sum_{j=0}^{m-1} c_j p_j(x) \qquad \textbf{(EQ 215)}$$

Therefore, if A were singular, we could write

$$p_m(x) = \frac{1}{c_m} \sum_{j=0}^{m-1} c_j p_j(x) \qquad \textbf{(EQ 216)}$$

However, the left hand side is a polynomial of degree m, but the right side is of a lesser degree. Therefore, this equation cannot be valid and we have

introduced a contradiction. For any mutually distinct $\{x_1, x_2, \ldots, x_N\}$, A must be non-singular.

Q.E.D.

Theorem GQ4: Let $\{x_1, x_2, \ldots, x_N\}$ be the N distinct roots of $p_N(x)$ in the open interval (a, b). Let $\{w_1, w_2, \ldots, w_N\}$ be the solution of the nonsingular set of linear equations

$$\sum_{k=1}^{N} p_j(x_k) w_k = \begin{cases} \langle p_0 | p_0 \rangle, & \text{if } j = 0 \\ 0, & \text{if } j = 1, 2, \ldots, N-1 \end{cases} \qquad \textbf{(EQ 217)}$$

Then $w_j > 0$ for all $j = 1, 2, \ldots, N$, and for all $p(x) \in \Pi_{2N-1}$,

$$\int_a^b w(x) p(x) dx = \sum_{j=1}^{N} w_j p(x_j) \qquad \textbf{(EQ 218)}$$

That is, if we place the support points at the zeros of the orthogonal polynomial $p_N(x)$, then even though with only N support points, we can form an exact representation of polynomials up to degree $N-1$, we obtain the **exact** value of the definite integral for any polynomials of degree $2N-1$ or less.

Placing the support points at the zeros of $p_N(x)$ provides optimal accuracy.

Proof:

We have shown in theorem GQ2 that the set of linear equations for the weights is nonsingular and has therefore a unique solution. We now consider an arbitrary polynomial $p(x) \in \Pi_{2N-1}$ that we write as

$$p(x) = p_N(x) q(x) + r(x) \qquad q(x), r(x) \in \Pi_{N-1} \qquad \textbf{(EQ 219)}$$

We next write $q(x)$ and $r(x)$,

$$q(x) = \sum_{k=0}^{N-1} q_k p_k(x) \qquad r(x) = \sum_{k=0}^{N-1} r_k p_k(x) \tag{EQ 220}$$

and evaluate the definite integral of $p(x) \in \Pi_{2N-1}$,

$$\int_a^b w(x)p(x)dx = \int_a^b w(x)[p_N(x)q(x) + r(x)]dx = \langle p_N|q\rangle + \int_a^b w(x)r(x)dx \tag{EQ 221}$$

First, we see that since $q(x) \in \Pi_{N-1}$, from the orthogonality of $\{p_0(x), p_1(x), \ldots, p_N(x)\}$,

$$\langle p_N|q\rangle = \sum_{k=0}^{N-1} q_k\langle p_N|p_k\rangle = \sum_{k=0}^{N-1} q_k(0) = 0 \tag{EQ 222}$$

Also, as $p_0(x) = 1$, we can write

$$\int_a^b w(x)r(x)dx = \int_a^b w(x)p_0(x)r(x)dx = \langle p_0|r\rangle \tag{EQ 223}$$

Again, from the orthogonality of $\{p_0(x), p_1(x), \ldots, p_N(x)\}$,

$$\langle p_0|r\rangle = \sum_{k=0}^{N-1} r_k\langle p_0|p_k\rangle = \sum_{k=0}^{N-1} r_k[\langle p_0|p_0\rangle\delta_{0k}] = r_0\langle p_0|p_0\rangle \tag{EQ 224}$$

Therefore, we find that the definite integral of $p(x) \in \Pi_{2N-1}$ is

$$\int_a^b w(x)p(x)dx = r_0\langle p_0|p_0\rangle \tag{EQ 225}$$

Next, we note that at every root $\{x_1, x_2, \ldots, x_N\}$ of $p_N(x)$,

$$p(x_j) = p_N(x_j)q(x_j) + r(x_j) = (0)q(x_j) + r(x_j) = r(x_j) \tag{EQ 226}$$

so that

$$\sum_{j=1}^{N} w_j p(x_j) = \sum_{j=1}^{N} w_j r(x_j) \qquad \textbf{(EQ 227)}$$

We now express each $r(x_j)$ as an expansion in orthogonal polynomials to write

$$\sum_{j=1}^{N} w_j p(x_j) = \sum_{j=1}^{N} w_j \sum_{k=0}^{N-1} r_k p_k(x_j) = \sum_{k=0}^{N-1} r_k \left[\sum_{j=1}^{N} p_k(x_j) w_j \right] \qquad \textbf{(EQ 228)}$$

Thus, if the set of weights are obtained by solving the linear system

$$\sum_{j=1}^{N} p_k(x_j) w_j = \begin{cases} \langle p_0 | p_0 \rangle, & \text{if } j = 0 \\ 0, & \text{if } j = 1, 2, \ldots, N-1 \end{cases} \qquad \textbf{(EQ 229)}$$

then

$$\sum_{j=1}^{N} w_j p(x_j) = r_0 \langle p_0 | p_0 \rangle = \int_a^b w(x) p(x) dx \qquad \textbf{(EQ 230)}$$

for any $p(x) \in \Pi_{2N-1}$.

For proof of the statements in the final paragraph of the theorem, the reader is referred to p. 153 of Stoer and Burlisch, Introduction to Numerical Analysis, Springer-Verlag, 1993.

Q.E.D.

Initial value problems: Parameter continuation and homotopy

In our previous discussion of nonlinear algebraic equations, we have seen how one can improve the robustness of Newton's method by first starting from a set of parameters Θ_0 for which the system $\boldsymbol{f}(\boldsymbol{x};\Theta_0) = \boldsymbol{0}$ is easy to solve and then modify the parameters by small increments to obtain the final, desired solution with parameter vector Θ_1. If we use as an initial guess at each step the result from the previous parameter value, we hope to start Newton's method in the close vicinity of a solution where it performs more robustly. In this section, we see how to frame this ***homotopy method*** in a more robust and efficient form as an initial value problem.

We start with the parameter vector Θ_0 and solve $\boldsymbol{f}(\boldsymbol{x};\Theta_0) = \boldsymbol{0}$ for the solution $\boldsymbol{x}_{s0}$. We then change the parameter vector from Θ_0 to the final value Θ_1 over some path $\Theta(\lambda)$ where

$$\Theta(\lambda=0) = \Theta_0 \qquad \Theta(\lambda=1) = \Theta_1 \tag{EQ 231}$$

The simplest choice is the linear path

$$\Theta(\lambda) = (1-\lambda)\Theta_0 + \lambda\Theta_1 \tag{EQ 232}$$

We generate a solution path $\boldsymbol{x}_s(\lambda)$ where

$$\boldsymbol{f}(\boldsymbol{x}_s(\lambda);\Theta(\lambda)) = \boldsymbol{0} \tag{EQ 233}$$

As a first approach, we may think to integrate over λ,

$$\boldsymbol{x}_s(1) - \boldsymbol{x}_s(0) = \int_0^1 \left(\frac{d\boldsymbol{x}_s}{d\lambda}\right) d\lambda \tag{EQ 234}$$

To obtain an expression for $d\boldsymbol{x}_s/d\lambda$, we use the expansion for small $\delta\lambda$,

$$\mathbf{0} = \boldsymbol{f}(\boldsymbol{x}_s(\lambda + \delta\lambda);\Theta(\lambda + \delta\lambda))$$

$$\mathbf{0} = \boldsymbol{f}(\boldsymbol{x}_s(\lambda);\Theta(\lambda)) + \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\right)\left(\frac{d\boldsymbol{x}_s}{d\lambda}\right)\delta\lambda + \left(\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\right)\left(\frac{d\Theta}{d\lambda}\right)\delta\lambda \quad \text{(EQ 235)}$$

As along the solution path

$$\boldsymbol{f}(\boldsymbol{x}_s(\lambda + \delta\lambda);\Theta(\lambda + \delta\lambda)) = \boldsymbol{f}(\boldsymbol{x}_s(\lambda);\Theta(\lambda)) = \mathbf{0} \quad \text{(EQ 236)}$$

this yields the equation for $d\boldsymbol{x}_s/d\lambda$,

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\right)\left(\frac{d\boldsymbol{x}_s}{d\lambda}\right) + \left(\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\right)\left(\frac{d\Theta}{d\lambda}\right) = \mathbf{0} \quad \text{(EQ 237)}$$

We can compute (analytically or with finite differences) the partial derivative matrices for the function, and for a linear path

$$\frac{d\Theta}{d\lambda} = \Theta_1 - \Theta_0 \quad \text{(EQ 238)}$$

Solving for $d\boldsymbol{x}_s/d\lambda$ yields

$$\frac{d\boldsymbol{x}_s}{d\lambda} = -\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\right)^{-1}\left(\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\right)\left(\frac{d\Theta}{d\lambda}\right) \quad \text{(EQ 239)}$$

We see that this works fine as long as the Jacobian of the function, $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}$, is non-singular. Unfortunately, the Jacobian may well become singular, for example at a ***turning point*** where $d\boldsymbol{x}_s/d\lambda$ diverges to infinity as the solution curve becomes vertical in $(\boldsymbol{x}_s, \lambda)$ space (Figure 4.16 in the text). The homotopy method presented above will fail in such situations.

To avoid this problem, it is more robust to parameterize the solution curve not in terms of λ, but rather in terms of the arc-length s along this curve, a method known as ***arc-length continuation***. As we move along the solution

curve in (x_s, λ) space by an amount δx_s and $\delta\lambda$ (Figure 5), the change in the path length is

$$(\delta s)^2 = (\delta\lambda)^2 + \delta x_s \cdot \delta x_s \qquad \textbf{(EQ 240)}$$

Dividing by $(\delta s)^2$ and letting $\delta s \to 0$ yields the following arc-length condition that must be satisfied by the functions $x_s(s)$ and $\lambda(s)$,

$$\frac{dx_s}{ds} \cdot \frac{dx_s}{ds} + \left(\frac{d\lambda}{ds}\right)^2 = 1 \qquad \textbf{(EQ 241)}$$

We now derive a simple predictor-corrector method for moving along this solution path, that being parameterized in terms of the arc-length, can step through turning points. As a ***predictor***, we use the explicit Euler method to integrate from s_k to $s_{k+1} = s_k + \Delta s$,

$$\begin{aligned} x_s^{[0]}(s_k + \Delta s) &= x_s(s_k) + \left(\left.\frac{dx_s}{ds}\right|_{s_k}\right)\Delta s \\ \lambda^{[0]}(s_k + \Delta s) &= \lambda(s_k) + \left(\left.\frac{d\lambda}{ds}\right|_{s_k}\right)\Delta s \end{aligned} \qquad \textbf{(EQ 242)}$$

We use the superscript $^{[0]}$ to denote that these are only our initial predictions of the new values that we will use later as initial guesses for a ***corrector*** stage calculation that ensures that we remain on the true solution path. To obtain the derivatives of the solution path variables with respect to arc-length, we use the expansion,

$$\begin{aligned} \mathbf{0} &= f(x_s(s + \delta s);\Theta(s + \delta s)) \\ \mathbf{0} &= f(x_s(s);\Theta(s)) + \left(\frac{\partial f}{\partial x^T}\right)\left(\frac{dx_s}{ds}\right)\delta s + \left(\frac{\partial f}{\partial \Theta^T}\right)\left(\frac{d\Theta}{d\lambda}\right)\left(\frac{d\lambda}{ds}\right)\delta s \end{aligned} \qquad \textbf{(EQ 243)}$$

As along the solution path,

$$f(x_s(s + \delta s);\Theta(s + \delta s)) = f(x_s(s);\Theta(s)) = \mathbf{0} \qquad \textbf{(EQ 244)}$$

we have the linear system

$$\left(\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\right|_{s_k}\right)\left(\frac{d\boldsymbol{x}_s}{ds}\right)+\left(\left.\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\frac{d\Theta}{d\lambda}\right|_{s_k}\right)\left(\frac{d\lambda}{ds}\right)=\boldsymbol{0} \qquad \text{(EQ 245)}$$

subject to the quadratic constraint

$$\frac{d\boldsymbol{x}_s}{ds}\cdot\frac{d\boldsymbol{x}_s}{ds}+\left(\frac{d\lambda}{ds}\right)^2=1 \qquad \text{(EQ 246)}$$

We write this more compactly by defining the vectors

$$\boldsymbol{z}=\begin{bmatrix}\boldsymbol{x}_s\\ \lambda\end{bmatrix} \qquad \dot{\boldsymbol{z}}=\begin{bmatrix}d\boldsymbol{x}_s/ds\\ d\lambda/ds\end{bmatrix} \qquad \text{(EQ 247)}$$

to obtain

$$\left[\left(\left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\right|_{s_k}\right)\left(\left.\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\frac{d\Theta}{d\lambda}\right|_{s_k}\right)\right]\dot{\boldsymbol{z}}=\boldsymbol{0} \qquad \dot{\boldsymbol{z}}\cdot\dot{\boldsymbol{z}}=1 \qquad \text{(EQ 248)}$$

To solve this equation, we generate some vector $\boldsymbol{c}$ such that $\boldsymbol{c}$ is not perpendicular to $\dot{\boldsymbol{z}}$; a randomly-generated vector should work fine because it is highly unlikely to ever be perpendicular to $\dot{\boldsymbol{z}}$. We can then remove the nonlinearity by replacing the unit length condition $\dot{\boldsymbol{z}}\cdot\dot{\boldsymbol{z}}=1$ with the equivalent normalization,

$$\dot{\boldsymbol{z}}=\pm\frac{\boldsymbol{v}}{|\boldsymbol{v}|} \qquad \boldsymbol{c}^T\boldsymbol{v}=1 \qquad \text{(EQ 249)}$$

where we choose the plus or minus sign to enforce that we are moving in a consistent direction in $(\boldsymbol{x}_s, \lambda)$ space,

$$\dot{\boldsymbol{z}}\cdot\begin{bmatrix}\boldsymbol{x}_s(s_k)-\boldsymbol{x}_s(s_{k-1})\\ \lambda(s_k)-\lambda(s_{k-1})\end{bmatrix}\geq 0 \qquad \text{(EQ 250)}$$

We then solve for $\boldsymbol{v}$ from the linear system

$$\begin{bmatrix} \left(\dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\bigg|_{s_k}\right) & \left(\dfrac{\partial \boldsymbol{f}}{\partial \Theta^T}\dfrac{d\Theta}{d\lambda}\bigg|_{s_k}\right) \\ \text{------} \quad \boldsymbol{c}^T \quad \text{------} \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} \boldsymbol{0} \\ 1 \end{bmatrix} \tag{EQ 251}$$

The predictor estimate of $(\boldsymbol{x}_s, \lambda)$ along the solution curve obtained after increasing the arc-length by an amount Δs is then

$$\begin{bmatrix} \boldsymbol{x}_s^{[0]}(s_{k+1}) \\ \lambda^{[0]}(s_{k+1}) \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_s(s_k) \\ \lambda(s_k) \end{bmatrix} + \dot{\boldsymbol{z}}(\Delta s) \qquad \dot{\boldsymbol{z}} = \pm\frac{\boldsymbol{v}}{|\boldsymbol{v}|} \tag{EQ 252}$$

We next use this prediction as an initial guess in an iterative corrector stage that solves the nonlinear system

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}_s(s_{k+1}); \Theta(s_{k+1})) &= \boldsymbol{0} \\ |\boldsymbol{x}_s(s_{k+1}) - \boldsymbol{x}_s(s_k)|^2 + |\lambda(s_{k+1}) - \lambda(s_k)|^2 &= (\Delta s)^2 \end{aligned} \tag{EQ 253}$$

Applying Newton's method to the first of (EQ 253) yields the iterative rule

$$\begin{aligned} \boldsymbol{x}_s^{[m+1]}(s_{k+1}) &= \boldsymbol{x}_s^{[m]}(s_{k+1}) + \Delta \boldsymbol{x}_s^{[m]} \\ \lambda^{[m+1]}(s_{k+1}) &= \lambda^{[m]}(s_{k+1}) + \Delta\lambda^{[m]} \end{aligned} \tag{EQ 254}$$

where the updates must satisfy the N linear equations

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\bigg|_{[m]}\right)\Delta \boldsymbol{x}_s^{[m]} + \left(\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\frac{d\Theta}{d\lambda}\bigg|_{[m]}\right)\Delta\lambda^{[m]} = -\boldsymbol{f}(\boldsymbol{x}_s^{[m]}(s_{k+1}); \Theta^{[m]}(s_{k+1})) \tag{EQ 255}$$

Rather than apply Newton's method to the second arc-length equation, we note that for small Δs, any change in $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{x}_s & \lambda \end{bmatrix}^T$ that maintains a constant arc-length will be perpendicular to the vector $\dot{\boldsymbol{z}}$ computed by the predictor. For small Δs, we make the algorithm more efficient by reusing the deriva-

tive matrices computed at the predictor stage, to yield an update equation that is solved quickly by LU factorization,

$$\begin{bmatrix} \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}^T}\Big|_{s_k}\right) & \left(\frac{\partial \boldsymbol{f}}{\partial \Theta^T}\frac{d\Theta}{d\lambda}\Big|_{s_k}\right) \\ \text{------} \quad \dot{\boldsymbol{z}}^T & \text{------} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{x}_s^{[m]} \\ \Delta\lambda^{[m]} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{f}(\boldsymbol{x}_s^{[m]}(s_{k+1});\Theta^{[m]}(s_{k+1})) \\ 0 \end{bmatrix} \quad \textbf{(EQ 256)}$$

We stop the Newton iterations when we reach the solution curve,

$$\left\| \boldsymbol{f}(\boldsymbol{x}_s^{[m]}(s_{k+1});\Theta^{[m]}(s_{k+1})) \right\| \leq \delta_{tol} \quad \textbf{(EQ 257)}$$

We continue stepping along the curve by increasing the arc length until we reach $\lambda = 1$.

This basic arc-length continuation method is the foundation of more advanced algorithms for examining the dependence of the solution of a nonlinear algebraic system upon the system parameters. A popular free-domain continuation software package is AUTO that also identifies ***bifurcation points***, *i.e.* choices of parameters at which Jacobian of the function is singular at the solution. For more on bifurcation analysis, and its application to nonlinear dynamics, consult R. Seydel, Practical Bifurcation and Stability Analysis: From equilibrium to chaos. 2nd edition. Springer-Verlag, 1994.

Numerical optimization: The nonlinear simplex method

We now describe the simplex algorithm for finding the minimum of a cost function $F(\boldsymbol{x})$ step-by-step.

Step I. Form the initial simplex

From the initial guess $\boldsymbol{x}^{[0]}$, generate N additional points $\{\boldsymbol{x}^{[1]}, \boldsymbol{x}^{[2]}, \ldots, \boldsymbol{x}^{[N]}\}$,

$$\boldsymbol{x}^{[j]} = \boldsymbol{x}^{[0]} + l_j \boldsymbol{e}^{[j]} \qquad j = 1, 2, \ldots, N \qquad e_m^{[j]} = \delta_{jm} \tag{EQ 258}$$

The l_j are chosen so that the $\left|F(\boldsymbol{x}^{[j]}) - F(\boldsymbol{x}^{[0]})\right|$ are of comparable magnitude.

Step II. Check for convergence

When the simplex tightly bounds the local minimum, the cost function values of all vertices will be nearly identical. The center of the simplex

$$\bar{\boldsymbol{x}} = \frac{1}{N+1}\sum_{j=0}^{N} \boldsymbol{x}^{[j]} \tag{EQ 259}$$

is an acceptable estimate of the minimum if the following criterion is met for some small $\delta_{tol} > 0$,

$$\frac{1}{N}\sum_{j=0}^{N} \left|F(\boldsymbol{x}^{[j]}) - F(\boldsymbol{x}^{[0]})\right|^2 \leq \delta_{tol} \tag{EQ 260}$$

If this criterion is satisfied, execution stops; else a new iteration begins at Step III.

Step III. Rank the vertices by their cost function values

Obtain the values of the following integers,

M = index of vertex with largest cost function

M_2 = index of vertex with second-to-largest cost function

m = index of vertex with smallest cost function

For example, if in the 2-D case $F(\boldsymbol{x}^{[2]}) < F(\boldsymbol{x}^{[0]}) < F(\boldsymbol{x}^{[1]})$, then

$$M = 1 \qquad M_2 = 0 \qquad m = 2 \tag{EQ 261}$$

Step IV. Attempt to shift the simplex towards a region of lower cost function

If we move the simplex away from the vertex $x^{[M]}$ of highest $F(x)$, we expect to move it towards $x_{\min}$. Thus, we compute the centroid of vertex M,

$$x_c^{[M]} \equiv \frac{1}{N} \sum_{\substack{j=0 \\ j \neq M}}^{N} x^{[j]} \qquad \textbf{(EQ 262)}$$

and "reflect" $x^{[M]}$ respect to $x_c^{[M]}$ using some scalar $\alpha > 0$ (say, $\alpha = 1$),

$$x^{(r)} \equiv x_c^{[M]} + \alpha(x_c^{[M]} - x^{[M]}) \qquad \textbf{(EQ 263)}$$

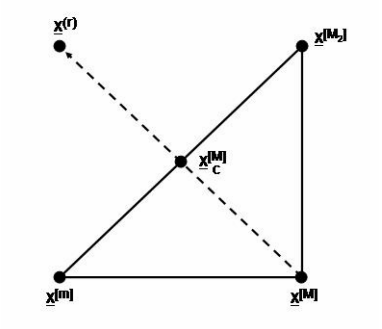

FIGURE 4. Reflection step in simplex method

We now must decide whether replacing $x^{[M]}$ with $x^{(r)}$ will yield a simplex that is more likely to contain $x_{\min}$. Of course, without knowing $x_{\min}$, it is impossible to know if the new simplex will contain it; however, we consider the replacement acceptable if,

$$F(x^{[m]}) \leq F(x^{(r)}) \leq F(x^{[M_2]}) \qquad \textbf{(EQ 264)}$$

This condition is not met in the following cases:

Case 1. $F(\boldsymbol{x}^{(r)}) < F(\boldsymbol{x}^{[m]})$

Here, we have been lucky and have moved the simplex in a direction that gives us a new "best" vertex. This suggests that we could do even better if we were to continue in this direction. We therefore find a point along the extension of the search direction,

$$\boldsymbol{x}^{(e)} = \boldsymbol{x}_c^{[M]} + \beta(\boldsymbol{x}^{(r)} - \boldsymbol{x}_c^{[M]}) \qquad \textbf{(EQ 265)}$$

where $\beta > 1$ (say $\beta = 2$). Then, if $F(\boldsymbol{x}^{(e)}) < F(\boldsymbol{x}^{[m]})$, we form our new simplex by replacing $\boldsymbol{x}^{[M]}$ with $\boldsymbol{x}^{(e)}$. Otherwise, we already know that $F(\boldsymbol{x}^{(r)}) < F(\boldsymbol{x}^{[m]})$, so we replace $\boldsymbol{x}^{[M]}$ with $\boldsymbol{x}^{(r)}$. In either case, we have significantly improved the quality of the simplex.

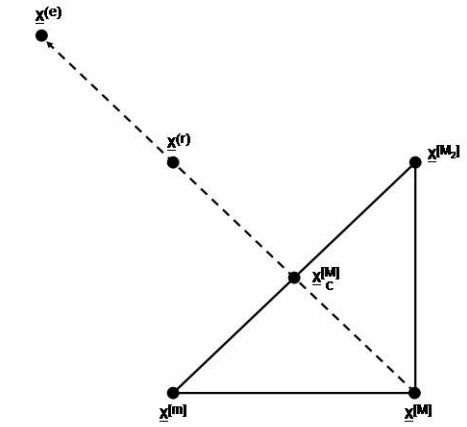

FIGURE 5. The extension step of the simplex method when $F(\boldsymbol{x}^{(r)}) < F(\boldsymbol{x}^{[m]})$

Case 2. $F(\boldsymbol{x}^{(r)}) > F(\boldsymbol{x}^{[M_2]})$

In this case, we try to remedy the situation by contracting from whichever of $\boldsymbol{x}^{(r)}$ or $\boldsymbol{x}^{[M]}$ is lower in $F(\boldsymbol{x})$, using some $0 < \gamma < 1$ (say, $\gamma = 1/2$),

$$\boldsymbol{x}^{(co)} = \begin{cases} \boldsymbol{x}_c^{[M]} - \gamma(\boldsymbol{x}_c^{[M]} - \boldsymbol{x}^{(r)}), \text{ if } F(\boldsymbol{x}^{(r)}) < F(\boldsymbol{x}^{[M]}) \\ \boldsymbol{x}_c^{[M]} - \gamma(\boldsymbol{x}_c^{[M]} - \boldsymbol{x}^{[M]}), \text{ if } F(\boldsymbol{x}^{(r)}) \geq F(\boldsymbol{x}^{[M]}) \end{cases} \quad \textbf{(EQ 266)}$$

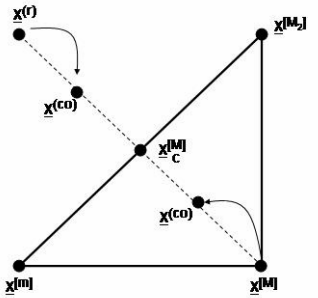

FIGURE 6. Contraction is performed when $F(\boldsymbol{x}^{(r)}) > F(\boldsymbol{x}^{[M_2]})$

We now identify two possibilities. First, if $F(\boldsymbol{x}^{(co)}) < F(\boldsymbol{x}^{[M]})$ and $F(\boldsymbol{x}^{(co)}) < F(\boldsymbol{x}^{(r)})$, the search direction that we had been using can reduce $F(\boldsymbol{x})$, but when doing the reflection, we just happened to step too far. If so, we replace $\boldsymbol{x}^{[M]}$ with $\boldsymbol{x}^{(co)}$ to form a new, improved simplex.

Alternatively, if $F(\boldsymbol{x}^{(co)}) \geq F(\boldsymbol{x}^{[M]})$ or $F(\boldsymbol{x}^{(co)}) \geq F(\boldsymbol{x}^{(r)})$, then it appears that our choice of a search direction was incorrect. When $F(\boldsymbol{x})$ is approximately linear over the simplex region, reflection should succeed in finding a point of lower cost function. Therefore, the appropriate action in this case is to make the simplex smaller, so that the curvature of $F(\boldsymbol{x})$ over the simplex is reduced. We thus do not accept any of the vertex substitutions suggested above, and execute instead step V of the algorithm.

Step V. If $F(\boldsymbol{x}^{(co)}) \geq F(\boldsymbol{x}^{[M]})$ or $F(\boldsymbol{x}^{(co)}) \geq F(\boldsymbol{x}^{(r)})$, reduce simplex size by general contraction

If step IV fails, we reduce the simplex size, as the failure of the reflection, expansion, and contraction steps is due to the curvature of $F(\boldsymbol{x})$ over the region of the simplex. We thus perform a general contraction in which all vertices are contracted towards that of smallest cost function by halving the edge lengths. For each $j \neq m$,

$$\boldsymbol{x}^{[j]} \leftarrow \boldsymbol{x}^{[m]} + \frac{1}{2}(\boldsymbol{x}^{[j]} - \boldsymbol{x}^{[m]}) \qquad \textbf{(EQ 267)}$$

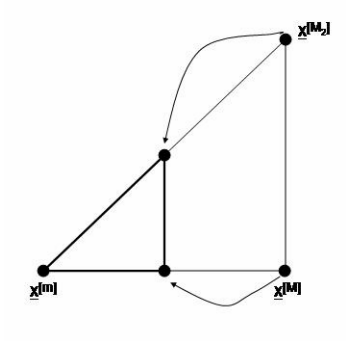

FIGURE 7. The simplex method performs a general contraction if needed to make a local linear approximation of the cost function more valid.

After completion of step IV (and optionally of step V), the algorithm returns to step II to check again for convergence. When the simplex encloses $\boldsymbol{x}_{\min}$, step IV is likely to fail, and the simplex will shrink by general contraction until it binds tightly the local minimum, at which point execution stops. In Matlab, the simplex method is invoked as the optimization toolkit routine **fminsearch**.

Boundary value problems: Modeling a microfluidic H-filter using FEMLAB™

We now consider the use of FEMLAB, a software package (www.comsol.com) built on top of MATLAB, to solve a BVP using the finite element method. In this section, we use FEMLAB to model a microfluidic H-filter. Microfluidics is the study of systems that transport fluids in micron-sized channels, in which the small dimensions ensure laminar flow (unlike macroscopic fluid systems that are nearly always turbulent). Microfluidic systems are of technological interest as microreactors and in a number of biotechnological applications such as “DNA on a chip”.

While microfluidic systems allow one to perform analyses on small volumes of liquids (useful in DNA applications), the laminar flow characteristics associated with their small size scale also yields interesting phenomena not significant on the macroscale. As an example, consider a H-filter for separating dilute components based on differences in the diffusivities (Figure 8). The dimensions of the channels are assumed to be on the micron scale in the plane of the figure, but very large in the out of plane dimension.

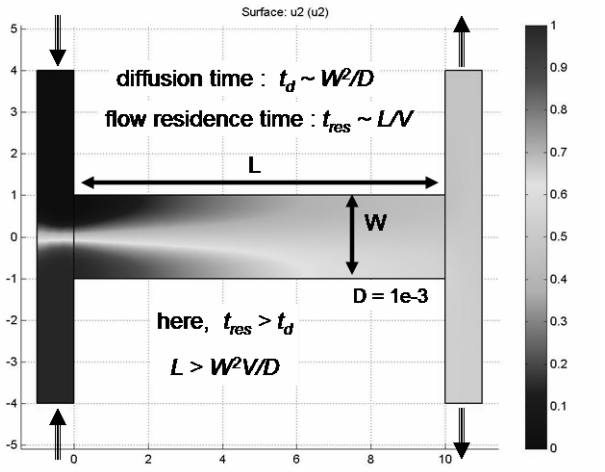

FIGURE 8. Microfluidic H-filter

Two fluids flow in on the left, one containing a pure fluid (*upper left*) and one (*lower left*) containing one or more species of suspended particles or dissolved solute. When these two streams meet, they flow from left to right across the middle of the filter in two neighboring layers, due to the laminar nature of the flow. In this section, the upper half is comprised initially of pure fluid and the bottom half initially contains solute and fluid at the concentrations found when entering the H-region at the lower left. As the fluid moves across the filter, the dilute components in the bottom half diffuse into the upper half, but since the flow is laminar, there is no turbulent mixing. Mass transfer occurs only through the relatively slow process of molecular diffusion. Thus, mixing will be significant only if the characteristic time for diffusion across the filter, $t_d \sim W^2/D$ is smaller than the time necessary for the fluid to transverse the filer, $t_{res} \sim L/V$.

If the bottom fluid stream contains two types of solutes of different sizes and diffusivities, *e.g.* large suspended particles and small molecules, the H-filter section can be designed such that only the faster-diffusing solute has

sufficient time to mix. This microfluidic device thus can be used to separate solutes based on their relative diffusivities.

In this section, we demonstrate the use of FEMLAB to model the transfer of a component from one stream to the other in a H-filter using the finite element method. We will go through the solution process step-by-step to see how the software is used.

1. Start the FEMLAB program

2. In the Model Navigator, press the button "Multiphysics". You will see on the left a scroll window with various types of PDE's. Select first "Incompressible Navier-Stokes" and hit the >> button to add this equation to your model. You will see on the right at bottom, a list of the fields that this equation set models. They are u (x-direction velocity), v (y-direction velocity) and p (pressure). Next, scroll down the left window and select "Convection-diffusion equation", that adds another field u2 (the concentration of our dilute component). Hit OK to accept these model equations.

3. You are now shown a window with a grid in which you can specify the geometry of your 2-D BVP. We first modify the grid axes to meet the dimensions of our computational domain. Select from the pull-down menu "Options->Axes/Grid Settings". Set "Xmin" to the value -2, "Xmax" to 12, "Ymin" to -5, and "Ymax" to 5. Then, hit the "Grid" button, unclick "Auto" and enter in the box "X" a value of 1 and a value of 1 in the "Y" box. Then, hit OK.

4. Next, draw the H-filer, with a channel width of 1.0 in the vertical sections and 2.0 in the horizontal section. Click on the button in the upper left that has a rectangle (with no central dot) and then click on the grid at the point (-1,4). Drag the cursor to the point (-4,0) and click again to add the left-most vertical channel. Next, add the central channel by hitting the same rectangle button at the upper-left, and start from (0,1) to (10,-1). Then, we add the right vertical channel by drawing a rectangle from (10,4) to (11,-4).

5. We now fuse the three rectangles into a single computational domain to create a composite geometric object and remove the internal boundary partitions. Select "Draw -> Create Composite Object..." and, holding the shift

key, click on R1, R2, and R3 in the "object selection" window to highlight all three. Then, hit the "Apply" button to fuse all three into a single composite object and hit "OK" to close the menu.

6. Next, we modify some parameters that we will wish to vary later when studying the behavior of our model. First, we set the average velocity in the vertical channel. Select "Options -> Add/Edit Constants", and in the "Name" field enter v_avg. Immediately below, enter a value of 0.01 and hit the "Set" button. Next, in the "Name" field type D (this is the diffusion constant) and enter a value of 1e-3. Then, input the centerline *x*-coordinate of left vertical channel by setting xc1 to -0.5. Hit the "Apply" button and then "OK" to close the window.

7. We now start to specify the problem, starting first with the boundary conditions. Select "Multiphysics" from the pull-down menu, and select "incompressible Navier-Stokes" to make this equation set active. Then, select "Boundary -> Boundary Mode" to begin entering the boundary condition data for this equation set.

8. Double-click on the left-most vertical line and you will see a "Boundary Settings" window appear that shows that no-slip boundary conditions are selected for this boundary section. This is the default BC of the Navier-Stokes equation. This is OK for all boundary sections except the inlet and outlet sections. Double-click on the outlet segment at the extreme upper-right and select "straight-out" boundary conditions for this section. Do the same for the outlet at the lower-right. You should see that these sections are now of a different color to denote visually that they employ a different BC.

Next, set the inlet BC. First, start with the inlet at the lower-left. We know that the flow in this region has a positive *y*-direction velocity that is parabolic, reaches a maximum at the center-line and is zero at the boundaries. The *x*-direction velocity here is zero. Therefore, in the "Boundary Settings" window, hit the top-button and type in the "Inflow, y velocity" window the expression:

```
2*v_avg*(1 - ( (x - xc1)/0.5 )^2)
```

For the inlet at the upper-left, the velocity is in the negative *y* direction, and so in the "Inflow, y velocity" window enter

```
-2*v_avg*(1 - ( (x - xc1)/0.5 )^2)
```

This completes the specifications of the boundary conditions for the Navier-Stokes equation.

9. Now, select "Multiphysics -> Convection-Diffusion Equation" to make this equation active. If we now click on the left-most vertical line, we find that the default is a zero-value Dirichlet BC. For all vertical lines (the solid walls of the vertical channels) and the upper and lower walls of the central horizontal channel, change the BC to "Neumann" type to make these walls impermeable (i.e. no flux in or out). The lines should change from solid to dashed.

Next, at the two outlets at the upper and lower right, also select "Neumann" BC, because after we have separated the stream, there is no further change in the concentration field as we move downstream (if we place the outlets far enough downstream). Therefore, the derivative of the concentration field in the *y*-direction will be zero at this point.

For the inlets, at the upper-left, retain the zero-value Dirichlet BC. For the lower-left inlet, set the inlet concentration to 1 by entering this value in the "r" window. We have now completed the specification of all boundary conditions for this problem.

10. We next move to specifying the coefficients of the differential equation, selecting first "Multiphysics -> incompressible Navier Stokes". Select "Subdomain -> Subdomain Mode" to begin entering the PDE coefficients. We want to use the same coefficients for all domains, so hold the shift-key and click on each subdomain (the two vertical and one horizontal rectangles) to highlight them all. Then select "Subdomain -> Subdomain Settings" to open the window to set the coefficients in the Navier-Stokes equation. We will not change anything, but if we wanted to change the viscosity or density, or add a volumetric body force (*e.g.* from gravity) we could do so. Hit the "OK" button to exit.

11. Next, select "Multiphysics -> Convection-Diffusion Equation" to enter the PDE coefficients for this equation. Again, highlight all boundaries and select "Subdomain -> Subdomain Settings". We see a number of coefficients, and the form of the PDE is written in the window near the upper left corner. Since we want a steady-state solution, we set d_a to 0, set the coefficient c to be our user-defined diffusion constant by entering D in this window, and then remove the source term by setting f to zero. The row vector β contains the coefficients that multiply the first derivatives in this equation. These should take the values of the *x* and *y* direction velocities, so in the β window, enter u v. Then, hit the "OK" button to accept the results.

12. We have now completely specified the geometry, boundary conditions, and the differential equations that define the BVP, and are ready to solve it using the finite element method. First, we select "File -> Save As -> Model.mat-file" and save our model as H_filter_FEMLAB.mat.

13. We next generate the mesh for the system. We select "Mesh -> Mesh Mode" to place an initial mesh. We then select "Mesh -> Refine Mesh" to add more nodes. Finally, add more nodes to where the vertical and horizontal elements meet by highlighting each section with the mouse and then selecting "Mesh -> Refine Selected Elements". Finally, select "Mesh -> Jiggle Mesh" to move the nodes so as to make the angles of each triangular more equal (this improves numerical performance). Select "File -> Save" to save the model with the generated mesh.

14. Finally, we solve the system by selecting "Solve -> Solve Problem". This assembles the equation and uses a nonlinear algebraic solver to find the steady-state solution. Alternatively, we can select "Solve -> Parameters" to modify the method used to solve numerically the BVP.

15. From the contour plot of the concentration field, we see that there is significant mixing of the two streams along the two channels. If we modify the diffusion constant by selecting "Options -> Add/Edit Constants" and change the value of D to 1e-4, we can immediately compute a new solution by selecting "Solve -> Restart", to renew the calculation using the previous result as an initial guess. We see that there remains considerable mixing at the exit due to the presence of a stagnation region where the fluid resides for

a comparatively long time. Also, we see that we should extend the outlet region further downstream so that the Neumann BC is rigorously valid; however, for simplicity, we do not make this change.

16. If we further reduce the value of D to 1e-5, we find oscillations in the solution. This is due to the convection-dominated nature of the problem, and the large local Peclet numbers. Essentially, we are solving the problem using an approach very similar to the central difference scheme (CDS), and there is no upwind differencing. We see from the colorbar that there are regions in which the concentration unphysically becomes negative in the numerical solution.

17. To avoid these spurious oscillations, we can add streamline diffusion by selecting "Solver Parameters -> Streamline Diffusion" (below we set "Scale" to 0.1). In addition, we can very effectively reduce the magnitude of the oscillations by refining the mesh at the interface along the middle of the horizontal channel, where the gradients are most steep. In "Mesh -> Mesh Mode", we use the mouse to highlight the elements near the boundary, and then select "Mesh -> Refine Selected Elements" to add more nodes in this region. After again selecting "Mesh -> Jiggle Mesh", we select "Post -> Get Solution" to interpolate from the present solution the estimated values at the new nodes. We then solve the problem with the refined mesh by selecting "Solve -> Restart". The oscillations are still present, but are now very small.

Boundary value problems: Modeling natural convection in FEMLAB™

We are all familiar with the concept that hot air rises. This is because fluids (generally) become less dense as their temperature raises, and so the downward gravitational force per volume acting on hot fluid is less than that acting on cold fluid. This process is known as ***natural convection***. Here, we use FEMLAB™ to model this phenomenon in a simple convection cell.

Consider a rectangular cell in 2-D, with solid walls enclosing a region of fluid (see figure below). The top and bottom walls are made of a solid that is a good thermal insulator. The right and left walls are made of a solid that is a good thermal conductor, and the temperatures of these walls are maintained at constant values. The temperature of the right wall, T_1, is higher than that of the left wall, T_0, so that we expect the velocity and temperature fields as shown in the figure below. Here, gravity acts in the -*y* direction.

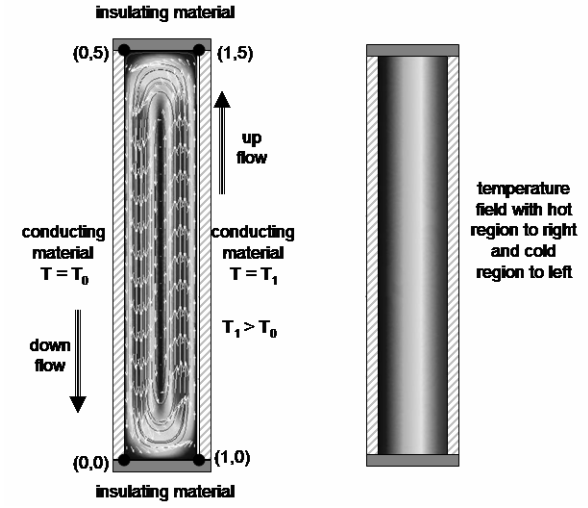

FIGURE 9. Simple convection cell in which a horizontal temperature gradient induces through free-convection vertical flow in which the hot fluid at right rises and the cool fluid at left falls

To model this system, we will use a modification of the Navier-Stokes equations. Remember that these equations describe the flow behavior of a fluid that is Newtonian and that has a constant density in the flow. Here, of course, the density will change, but we expect that the magnitude of the density variations is small compared to the average value. Therefore, in our equations, we write the temperature-dependent density to be

$$\rho(T) = \bar{\rho} - \bar{\rho}\bar{\beta}(T - \bar{T}) \qquad \textbf{(EQ 268)}$$

where $\bar{T} = \frac{1}{2}(T_1 + T_0)$ is a reference temperature and $\bar{\rho} = \rho(\bar{T})$. The thermal expansion coefficient of the fluid is $\beta = \frac{1}{\bar{V}}\left(\frac{\partial V}{\partial \bar{T}}\right)_p$.

If we neglect the effect of small density variations in the convection term (the ***Boussinesq approximation***), we have the modified Navier-Stokes equations of motion and continuity,

$$\bar{\rho}\frac{D\boldsymbol{v}}{Dt} = -\nabla P + \mu\nabla^2\boldsymbol{v} + \bar{\rho}\boldsymbol{g}\bar{\beta}(T - \bar{T})$$
$$\nabla \cdot \boldsymbol{v} = 0 \qquad \textbf{(EQ 269)}$$

where $\boldsymbol{g}$ is the acceleration vector due to gravity, and the dynamic pressure is defined as $P = p + \bar{\rho}gh$, where h is the elevation in the gravitational field. The acceleration of a "particle" of fluid as it is convected through the system is

$$\frac{D\boldsymbol{v}}{Dt} = \frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla\boldsymbol{v} \qquad \textbf{(EQ 270)}$$

In addition to these equations, we must also model the transport of thermal energy as heat through the system. The temperature field for this problem is governed by the PDE

$$\bar{\rho}\hat{C}_p\frac{DT}{Dt} = k\nabla^2 T + \mu\Phi_v \qquad \textbf{(EQ 271)}$$

where $\hat{C}_p$ is the specific heat of the fluid, k is the thermal conductivity of the fluid, and the volumetric rate at which heat is generated due to the viscous dissipation of energy (remember viscosity is like internal friction within the fluid) is

$$\mu\Phi_v = \mu[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^2]:\nabla\boldsymbol{v} \qquad \textbf{(EQ 272)}$$

For slow flows, as in this example, this term, which is second-order in the velocity gradients, may be neglected (the ***free convection*** limit).

In addition to these PDE's, we need to specify the boundary conditions. For all solid walls, we employ no-slip boundary conditions. For the temperature field, we impose Dirichlet BC's of known temperature values at the left and right walls, and at the top and bottom insulating walls we use the Neumann BC's that the component of the temperature gradient normal to the wall is zero.

We can decrease the number of unknowns in the problem statement by defining a reference length l_0, *e.g.* the distance between the plates, and a reference velocity v_0. Using these values, we can define the dimensionless quantities

$$\tilde{x} = x/l_0 \qquad \tilde{y} = y/l_0 \qquad \tilde{z} = z/l_0 \qquad \tilde{t} = v_0 t/l_0$$
$$\tilde{\boldsymbol{v}} = \boldsymbol{v}/v_0 \qquad \tilde{P} = P/(\bar{\rho} v_0^2) \qquad \tilde{T} = \frac{T - T_0}{T_1 - T_0} \qquad \text{(EQ 273)}$$
$$\tilde{\nabla} = l_0 \nabla \qquad \frac{D}{D\tilde{t}} = \left(\frac{l_0}{v_0}\right)\frac{D}{Dt}$$

in which case the governing equations take the dimensionless forms

$$\frac{D\tilde{\boldsymbol{v}}}{D\tilde{t}} = -\tilde{\nabla}\tilde{P} + (Pr)\tilde{\nabla}^2\tilde{\boldsymbol{v}} - (Gr)(Pr)^2(\boldsymbol{g}/g)(\tilde{T} - 1/2)$$
$$\tilde{\nabla}\cdot\tilde{\boldsymbol{v}} = 0 \qquad \text{(EQ 274)}$$
$$\frac{D\tilde{T}}{D\tilde{t}} = \tilde{\nabla}^2\tilde{T}$$

The only two free dimensionless parameters are the ***Prandtl number***

$$Pr = \frac{\mu \hat{C}_p}{k} \qquad \text{(EQ 275)}$$

and the ***Grashof number***

$$Gr = \frac{g\bar{\beta}(T_1 - T_0)l_0^3}{\nu^2} \qquad \nu = \frac{\mu}{\rho} \qquad \text{(EQ 276)}$$

We see that the Prandtl number is a characteristic of the fluid. Typical values at room temperature are

$$Pr_{\text{air}} = 0.7 \qquad Pr_{\text{water}} = 6.0 \qquad Pr_{\text{Hg metal}} = 0.029 \qquad \textbf{(EQ 277)}$$

The Grashof number is a dimensionless measure of the ability of the importance of the temperature gradient on the flow behavior.

To solve this problem numerically using FEMLAB^{TM}, choose in the initial Multiphysics menu an incompressible Navier-Stokes mode and a convection-diffusion mode for the field T. Draw a rectangular domain with lower-left corner (0,0) and upper-right corner (1,5). Taking l_0 to be the horizontal distance between plates, we see that this choice of a distance of 1 is appropriate for solving the dimensionless form of the equations.

Next, switch to Boundary Mode, and accept the default no-slip BC's for the Navier-Stokes' mode on each wall. For the convection-diffusion mode, select the Neumann BC for the top and bottom walls. You switch between modes by selecting the appropriate one under the Multiphysics pull-down menu. For the left wall, the default Dirichlet BC that T = 0 is appropriate for the dimensionless temperature field. For the right wall, we want to switch r = 1 to set in the Dirichlet BC T = 1 (see the notation at the top of the window for the meaning of each coefficient in the Neumann and Dirichlet BC's).

Now that the boundary conditions are set, we move to the specification of the governing equations. First, in the Options->Add/Edit constants tab, define values of the parameters Pr = 1 and Gr = 0. Then, in the Navier-Stokes mode, we set the value of the viscosity to Pr (its "value" in the dimensionless equations), and for the *y*-component of the body force, enter Gr*Pr^2*(T-0.5). Next, we switch to the convection-diffusion mode. In the top of the Subdomain Settings window, we see the form of this differential equation. We get the correct form of the equation for the dimensionless temperature field by setting the parameters:

da = 1, c = 1, β = [u v], f = 0

Next, form an appropriate finite element mesh for the domain, and use the stationary linear solver to find the steady-state solution. Here, because Gr = 0, there is no flow. Then, set Gr = 1e-3 to induce a circulation pattern due to free convection. If the stationary non-linear solver has problems converging, you might try switching to the time-dependent solver (the fldaspk option with an end time of 10 should get you to steady-state).

Probability theory and stochastic simulation: Statistical mechanics and the Boltzmann distribution

In chapter 7 of the text, we apply probability theory to study physical phenomena using ***statistical mechanics***, the microscopic theory underpinning classical thermodynamics. Here, we provide a cursory introduction to this vital subject including a derivation of the Boltzmann distribution.

Let us say that we have a physical system whose state is defined by a vector of parameters $\boldsymbol{q}$. We assume that each component of $\boldsymbol{q}$ takes one of a countable number of discrete values (as happens in quantum mechanics). The total energy of the system is $E(\boldsymbol{q})$. Let $\Omega_s(E)$ be the total number of microstates $\boldsymbol{q}$ of the system that have an energy value $E(\boldsymbol{q}) = E$,

$$\Omega_s(E) = \sum_{\underline{q}} \delta_{E(\boldsymbol{q}),\,E} \qquad \delta_{E(\boldsymbol{q}),\,E} = \begin{cases} 1, & \text{if } E(\boldsymbol{q}) = E \\ 0, & \text{if } E(\boldsymbol{q}) \neq E \end{cases} \qquad \textbf{(EQ 278)}$$

We define the entropy of the system as

$$S(N, V, E) = k_b \ln \Omega_s(E) \qquad \textbf{(EQ 279)}$$

with k_b being the Boltzmann constant (the ideal gas constant divided by Avagadro's number).

At equilibrium for an isolated system (constant energy *E*, constant volume *V*, constant number of atoms *N*), all microstates are equally likely to be observed.

What is the value of some property $A(\boldsymbol{q})$ *that we observe in this case?*

The number of microstates that have energy *E* and a value $A(\boldsymbol{q}) = A$ is

$$\Omega_s(E, A) = \sum_{\boldsymbol{q}} \delta_{E(\boldsymbol{q}),\,E}\, \delta_{A(\boldsymbol{q}),\,A} \qquad \textbf{(EQ 280)}$$

We define the corresponding *A*-dependent entropy

$$S(N, V, E, A) = k_b \ln \Omega_s(E, A) \qquad \textbf{(EQ 281)}$$

Let $A_{\max}$ be the value of *A* that maximizes $\Omega_s(E, A)$, and thus also maximizes $S(N, V, E, A)$. Typically, for a large microscopic system containing as many as a mole, 6.23×10^{23}, of atoms, we have

$$\Omega_s(E, A_{\max}) \gg \Omega_s(E, A') \qquad \textbf{(EQ 282)}$$

for any A' in which we do not have $A'/A_{max} \approx 1$.

As an example of this property, consider a system in which an ideal gas of *n* non-interacting particles is enclosed in a container of fixed volume *V*. Let us partition this system into two halves of equal volume $V/2$, and let the number of atoms in each half be n_1 and $n_2 = n - n_1$ (Figure 10).

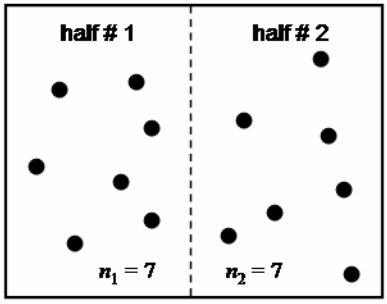

FIGURE 10. Example of a box containing ideal gas particles, counting numbers of particles on each half of the box

We want to compute the average of n_1 in the thermodynamic limit as $n \to \infty$. Since each atom has an equal chance of being in either half at the time of measurement, the probability of observing a particular value of n_1 is given by the binomial distribution

$$Pr(n, n_1) = \binom{n}{n_1}\left(\frac{1}{2}\right)^{n_1}\left(\frac{1}{2}\right)^{n-n_1} = \left[\frac{n!}{n_1!(n-n_1)!}\right]\left(\frac{1}{2}\right)^{n} \quad \textbf{(EQ 283)}$$

For fixed *n*, let the fraction of atoms found in the first half of the system be $\varphi_1 = n_1/n$. The probability distribution of observing a particular value of this quantity, for fixed *n*, is then

$$Pr(\varphi_1; n) = Pr(n, n_1 = n\varphi_1) \quad \textbf{(EQ 284)}$$

Using the ***gamma function***

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt \quad \textbf{(EQ 285)}$$

that satisfies the property

$$\Gamma(n+1) = n! \quad \textbf{(EQ 286)}$$

we write the distribution for φ_1, treated as a continuous variable, as

$$Pr(\varphi_1; n) \propto \frac{n!}{(n\varphi_1)![n(1-\varphi_1)]!} = \frac{\Gamma(n+1)}{\Gamma(n\varphi_1+1)\Gamma(n(1-\varphi_1)+1)} \quad \textbf{(EQ 287)}$$

so that

$$Pr(\varphi_1; n) = \frac{[\Gamma(n\varphi_1+1)\Gamma(n(1-\varphi_1)+1)]^{-1}}{\int_0^1 [\Gamma(n\varphi_1+1)\Gamma(n(1-\varphi_1)+1)]^{-1} d\varphi_1} \quad \textbf{(EQ 288)}$$

Figure 11 plots this distribution of φ_1 for various values of n. We see that as n increases, this distribution becomes more tightly peaked around the maximum value at $\varphi_1 = 1/2$. Thus, in the thermodynamic limit, $n \to \infty$, we essentially only observe the maximum value and $\langle \varphi_1 \rangle = 1/2$, $\langle n_1 \rangle = n/2$.

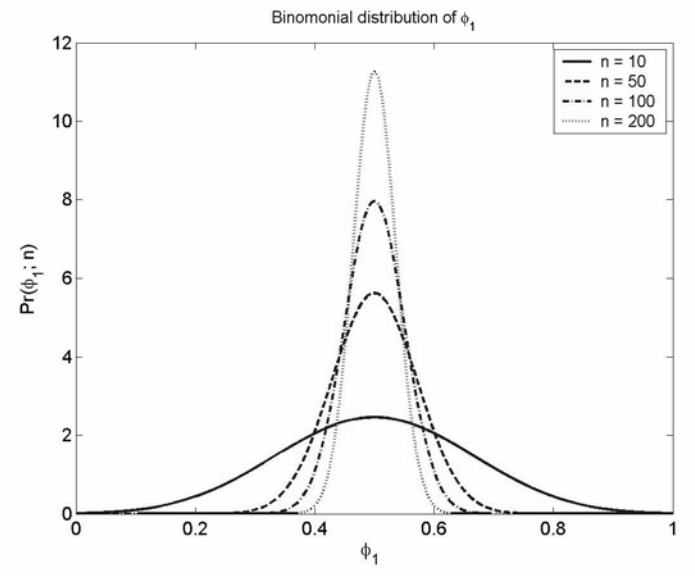

FIGURE 11. Distribution of the fraction φ_1 of n non-interacting atoms observed in a particular half of an enclosed domain. As n increases, the distribution becomes progressively more peaked around the maximum value at $\varphi_1 = 1/2$. This value is effectively the only one observed in the thermodynamic limit $n \to \infty$.

In general, the expectation value of A in the thermodynamic limit is (to a close approximation) $A_{\max}$, as at a constant value of E,

$$\langle A \rangle = \frac{\sum_{\boldsymbol{q}} A(\boldsymbol{q}) \delta_{E(\boldsymbol{q}),E}\, \delta_{A(\boldsymbol{q}),A}}{\sum_{\boldsymbol{q}} \delta_{E(\boldsymbol{q}),E}\, \delta_{A(\boldsymbol{q}),A}} = \frac{A_{\max}\Omega_s(E, A_{\max}) + \sum_{A(\boldsymbol{q}) \neq A_{\max}} A(\boldsymbol{q})\Omega_s(E, A)}{\Omega_s(E, A_{\max}) + \sum_{A(\boldsymbol{q}) \neq A_{\max}} \Omega_s(E, A)} \quad \textbf{(EQ 289)}$$

If whenever we do not have $A'/A_{max} \approx 1$, $\Omega_s(E, A_{\max}) \gg \Omega_s(E, A')$, then

$$\langle A \rangle = A_{\max} \left\{ \frac{1 + \sum_{A(\boldsymbol{q}) \neq A_{\max}} \frac{A(\boldsymbol{q})}{A_{\max}} \frac{\Omega_s(E, A)}{\Omega_s(E, A_{\max})}}{1 + \sum_{A(\boldsymbol{q}) \neq A_{\max}} \frac{\Omega_s(E, A)}{\Omega_s(E, A_{\max})}} \right\} \approx A_{\max} \qquad \textbf{(EQ 290)}$$

Thus, we observe the ***second law of thermodynamics***, that in an isolated system, the properties change so as to maximize entropy. For such a isolated (microcanonical) system, every microstate $\boldsymbol{q}$ such that $E(\boldsymbol{q}) = E$ is equally probable to be observed.

We now ask, what is the probability distribution $p(\boldsymbol{q})$ of microstates when the system is no longer isolated, but rather exchanges heat with its surroundings such that the temperature is constant at a value T?

We still assume that the system remains at a constant volume *V* and a constant number of atoms *N*. Thus, this situation is called the NVT, or ***canonical***, ensemble of microstates. To treat this situation, we couple the system to a thermal bath, such that the bath and the system exchange heat continually and are always at the same temperature (Figure 12). But, the only exchange between the two is heat, there is no transfer of mass or change in volume of the system or of the bath. The total composite system, the system of interest and the bath, is then isolated so that they cannot exchange any energy, mass, or volume with the outside.

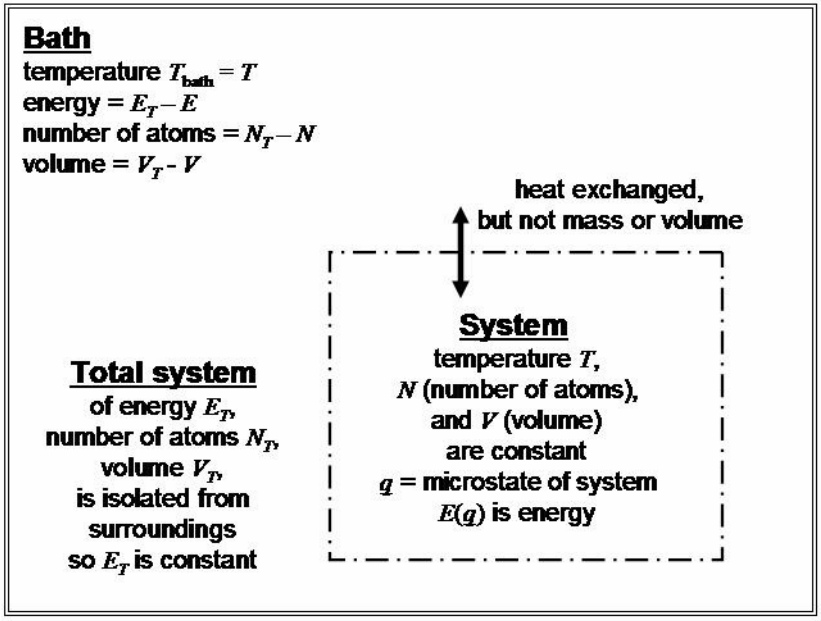

FIGURE 12. Composite system with thermal bath for NVT ensemble

Let the composite system have a total energy E_T, held constant. For a system microstate of energy $E(\boldsymbol{q}) = E$, the bath must then have an energy $E_T - E$. Let $\Omega_B(E_T - E)$ be the number of microstates of the bath with this energy. For each of these $\Omega_B(E_T - E)$ number of bath microstates, there are a number $\Omega_s(E)$ of possible microstates of the system, so that the total number of microscopic ways of arranging the system with an energy E of the system and an energy $E_T - E$ of the bath is

$$\Omega(E, N_T, V_T, E_T) = \Omega_s(E) \times \Omega_B(E_T - E) \quad \textbf{(EQ 291)}$$

Taking the log of both sides and multiplying by k_b, we obtain the total entropy of the isolated composite system and bath,

$$S(E, N_T, V_T, E_T) = S(N, V, E) + S_B(N_B, V_B, E - E_T) \quad \textbf{(EQ 292)}$$

N_B and V_B and the number of atoms and volume of the bath and the corresponding values for the total composite system are $N_T = N + N_B$ and $V_T = V + V_B$. By the second law of thermodynamics, heat is transferred between the system and bath so as to maximize the total entropy of the isolated composite system. When the bath is much larger than the system, $S_B(N_B, V_B, E - E_T) \gg S(N, V, E)$, and we say that the system evolves such as to maximize the number of microstates of the bath $\Omega_B(E_T - E)$. Thus, the distribution of the system microstates is

$$p(\boldsymbol{q}) = \frac{\Omega_B(E_T - E(\boldsymbol{q}))}{\sum_{\boldsymbol{q}} \Omega_B(E_T - E(\boldsymbol{q}))} \quad \textbf{(EQ 293)}$$

Since for a very large bath, $E_T \gg E(\boldsymbol{q})$, we use the expansion at $E(\boldsymbol{q}) = 0$,

$$\ln\Omega_B(E_T - E(\underline{q})) \approx \ln\Omega_B(E_T) - E(\boldsymbol{q}) \times \frac{\partial}{\partial E}\ln\Omega_B(E)\bigg|_{E_T} \quad \textbf{(EQ 294)}$$

Now, using our definition of entropy, and the result of classical thermodynamics $(\partial S/\partial E)|_{V,N} = T^{-1}$, we have

$$\frac{\partial}{\partial E}\ln\Omega_B(E)\bigg|_{E_T} = \frac{1}{k_b}\left(\frac{\partial S}{\partial E}\right)\bigg|_{E_T} = \frac{1}{k_b T} \quad \textbf{(EQ 295)}$$

and

$$\ln\Omega_B(E_T - E(\underline{q})) \approx \ln\Omega_B(E_T) - \frac{E(\underline{q})}{k_b T} \quad \textbf{(EQ 296)}$$

Substituting this result into $p(\boldsymbol{q})$ yields the famous ***Boltzmann distribution***

$$p(\boldsymbol{q}) = \frac{\exp\left[-\frac{E(\boldsymbol{q})}{k_b T}\right]}{\sum_{\boldsymbol{q}} \exp\left[-\frac{E(\boldsymbol{q})}{k_b T}\right]} \quad \textbf{(EQ 297)}$$

For a more detailed description of this subject, and a derivation of probability distributions in other ensembles, consult D.A. McQuarrie, Statistical Mechanics, Harper-Collins, 1973 and D. Frenkel and B. Smit, Understanding Molecular Simulation, 2nd. ed., Academic Press, 2002.

Statistics and parameter estimation: Corrective treatment of (near) singular X^TX

We consider here an example of least squares analysis in which singularity of X^TX is nearly guaranteed. Often, in business and engineering, one wishes to improve the performance of a system for which it is difficult to derive a theoretical model. In ***Response Surface Methodology (RSM)***, one fits a general (*e.g.* polynomial) model to the data, and then optimizes this model to generate the best design of the system. For a system with the *n* known system parameters $\{\alpha_1, \alpha_2, ..., \alpha_n\}$, we wish to fit the polynomial model

$$y = c_0 + \sum_{j=1}^{n} c_j \alpha_j + \sum_{j=1}^{n} \sum_{l=1}^{n} c_{jl} \alpha_j \alpha_l + \ldots \quad \textbf{(EQ 298)}$$

We define the vector of predictor variables

$$\boldsymbol{x} = \left[1 \ \alpha_1 \ \alpha_2 \ \ldots \ \alpha_n \ (\alpha_1^2) \ (\alpha_1\alpha_2) \ \ldots \ (\alpha_1\alpha_n) \ (\alpha_2\alpha_1) \ \ldots \ (\alpha_n^2) \ \ldots\right] \quad \textbf{(EQ 299)}$$

and the parameter vector

$$\theta = \left[c_0 \ c_1 \ c_2 \ \ldots \ c_n \ c_{11} \ c_{12} \ \ldots \ c_{1n} \ c_{21} \ \ldots \ c_{nn} \ \ldots\right]^T \quad \textbf{(EQ 300)}$$

Truncation of this polynomial yields a linear least squares problem that can be solved, in principle, using the methods discussed above. In practice, as one fits more and more parameters, X^TX often becomes singular. We describe here how one may drop the unnecessary, or unpredictable, parameters from the model using eigenvalue analysis of X^TX.

With $X^TX = V\Lambda V^T$, we note that the covariance matrix of θ is

$$cov(\theta) = \sigma^2 V\Lambda^{-1}V^T \qquad \textbf{(EQ 301)}$$

and that the least squares parameter estimate is

$$\theta_{LS} = V\Lambda^{-1}V^TX^T\boldsymbol{y} = \sum_{j=1}^{P}\lambda_j^{-1}[\boldsymbol{v}^{[j]}\cdot(X^T\boldsymbol{y})]\boldsymbol{v}^{[j]} \qquad \textbf{(EQ 302)}$$

The presence of (near) zero eigenvalues of X^TX makes the variance of the least squares parameter estimate very large or infinite. In this example, these (near) zero eigenvalues are due to the presence of parameters in our model that cannot be estimated accurately given the data at hand. We would do better if we did not attempt to estimate these parameters at all (thus obtaining a smaller covariance for the subset of parameters that we **do** have sufficient data to estimate). To do so, we form the ***pseudo-inverse*** of Λ, where for some chosen $\lambda_{\min} \geq 0$,

$$\tilde{\Lambda}^{-1} = \begin{bmatrix} \tilde{\lambda}_1^{-1} & & & \\ & \tilde{\lambda}_2^{-1} & & \\ & & \ddots & \\ & & & \tilde{\lambda}_{M+1}^{-1} \end{bmatrix} \qquad \tilde{\lambda}_j^{-1} = \begin{cases} \lambda_j^{-1}, \lambda_j > \lambda_{\min} \\ 0, \lambda_j \leq \lambda_{\min} \end{cases} \qquad \textbf{(EQ 303)}$$

This modification discards any attempt to estimate the component of θ in the eigenvector direction $\boldsymbol{v}^{[j]}$ if λ_j is too small. In RSM, we first consider fitting a polynomial with a very large set of parameters; however, a significant number of these will probably not be able to be estimated with sufficient accuracy from the data at hand. From eigenvalue analysis, we identify

and discard the linear combinations of parameters that we cannot fit, and write the best fit of parameters as

$$\theta_M = \sum_{j=1}^{P} \tilde{\lambda}_j^{-1} [\boldsymbol{v}^{[j]} \cdot (X^T \boldsymbol{y})] \boldsymbol{v}^{[j]} \qquad \textbf{(EQ 304)}$$

Since we will want to use this model in an optimization to find the "best" design of the system, we wish to decrease the number of independent model variables by retaining only the linear combinations of predictor variables that correspond to the significant eigenvectors (those that account for most of the information content in the data set). If the eigenvalues are ordered by decreasing magnitude, with

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_Q > \lambda_{\min} \qquad \lambda_{\min} \geq \lambda_{Q+1} \geq \ldots \geq \lambda_P \qquad \textbf{(EQ 305)}$$

then

$$\theta_M = \sum_{j=1}^{Q} \lambda_j^{-1} [\boldsymbol{v}^{[j]} \cdot (X^T \boldsymbol{y})] \boldsymbol{v}^{[j]} \qquad \textbf{(EQ 306)}$$

We streamline the predicting of responses by discarding those linear combinations of predictor variables outside of $span\{\underline{v}^{[1]}, \ldots, \underline{v}^{[Q]}\}$ that correspond to the linear combinations of parameters that we have not estimated. As the eigenvectors are orthonormal, the transformed predictor vector is

$$\boldsymbol{z}^{[k]} = \left[(\boldsymbol{v}^{[1]} \cdot \boldsymbol{x}^{[k]}) \; (\boldsymbol{v}^{[2]} \cdot \boldsymbol{x}^{[k]}) \; \ldots \; (\boldsymbol{v}^{[Q]} \cdot \boldsymbol{x}^{[k]}) \right] \in \Re^Q \qquad \textbf{(EQ 307)}$$

The reduced model is

$$\hat{y}^{[k]} = \gamma_1 z_1^{[k]} + \gamma_2 z_2^{[k]} + \ldots + \gamma_p z_p^{[k]} \qquad \textbf{(EQ 308)}$$

where the parameters are computed by linear least squares,

$$\gamma = (Z^T Z)^{-1} Z^T \boldsymbol{y} \qquad Z = \begin{bmatrix} \text{---}\ \boldsymbol{z}^{[1]}\ \text{---} \\ \text{---}\ \boldsymbol{z}^{[2]}\ \text{---} \\ \vdots \\ \text{---}\ \boldsymbol{z}^{[p]}\ \text{---} \end{bmatrix} \qquad \textbf{(EQ 309)}$$

We now find the optimal design by varying the original set of parameters $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, and for each choice of these parameters, compute the reduced predictor vector $\boldsymbol{z}$ and the corresponding value of the response $\boldsymbol{y}$ until a set $\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ yielding an optimal response is identified.

Statistics and parameter estimation: Additional results based on frequentist arguments

In the Bayesian paradigm, statistical conclusions are based on the posterior density $p(\theta, \sigma|\boldsymbol{y})$, treating θ and σ as random variables without any explicit reference to their deterministic "true" values. In this section, we discuss some results from analysis based on frequentist reasoning; that is, we ask:

What are the statistical properties our of estimates if we were to redo the same set of experiments over again many times?

Of course, since we can only base our conclusions on the data that we **have** collected, properties based on repeating the same set of experiments again some time(s) in the future do not affect the procedure that we follow in making estimates or testing hypotheses. They can, however, provide additional insight into the long-term performance of our estimation techniques. We would like, if we were to continue collecting data *an infinitum*, that our estimates eventually would converge to the "true" values.

Let $z(\boldsymbol{y})$ be some statistic extracted from the response data of a set of experiments. For example, $z(\boldsymbol{y})$ could be the least squares parameter estimate $\theta_{LS}(\boldsymbol{y})$ or it could be the sample variance $s^2(\boldsymbol{y})$. Each time we redo the set of

experiments and observe a different set $\boldsymbol{y}^{\langle n \rangle}$ of responses due to random error, we compute a different statistic $z^{\langle n \rangle} = z(\boldsymbol{y}^{\langle n \rangle})$ from the response data. In frequentist analysis, we are concerned with the expectation of this statistic over very many repeats of the set of experiments,

$$E_{\text{freq}}[z] = \lim_{N_{\text{rpt}} \to \infty} \frac{1}{N_{\text{rpt}}} \sum_{n=1}^{N_{\text{rpt}}} z(\boldsymbol{y}^{\langle n \rangle}) \qquad \textbf{(EQ 310)}$$

In our frequentist analysis, we postulate that the model, if a valid descriptor of the response that follows from specified predictor values, has a "true" set of parameters, to which the random error is added,

$$y = f(\boldsymbol{x};\theta^{(\text{true})}) + \varepsilon \qquad \textbf{(EQ 311)}$$

Of particular concern is the question, if we were to continue collecting data by repeating the same set of experiments, would our estimates converge to the "true" values? That is, when is it the case that

$$E_{\text{freq}}[\theta_{LS}] = \theta^{(\text{true})} \qquad \textbf{(EQ 312)}$$

The importance of the zero-mean Gauss-Markov condition

Among the Gauss-Markov conditions, the most important one is the requirement that the means of the errors in each experiment are zero, $E(\varepsilon) = \mathbf{0}$. Let us consider the particular case of a linear model,

$$\underline{y} = X\theta^{(true)} + \varepsilon \qquad \textbf{(EQ 313)}$$

Let us say that we repeat this set of experiments many times, using the same design matrix X but we obtain different response vectors $\boldsymbol{y}$ due to the randomness of ε. The averaged response over many repeated data sets is

$$E_{\text{freq}}(\boldsymbol{y}) = E_{\text{freq}}(X\theta^{(true)} + \varepsilon) = X\theta^{(true)} + E(\varepsilon) = X\theta^{(true)} \qquad \textbf{(EQ 314)}$$

Thus, the random error "averages out" if $E(\varepsilon) = \mathbf{0}$. Due to the random error, each set of experiments yields a different least squares estimate

$$\theta_{LS} = (X^T X)^{-1} X^T \boldsymbol{y} \qquad \textbf{(EQ 315)}$$

The averaged θ_{LS} over these many repeated data sets tends towards

$$E_{\text{freq}}(\theta_{LS}) = E[(X^T X)^{-1} X^T \boldsymbol{y}] = (X^T X)^{-1} X^T E(\boldsymbol{y}) = (X^T X)^{-1} X^T (X\theta^{(true)})$$
$$E_{\text{freq}}(\theta_{LS}) = I\theta^{(true)} = \theta^{(true)} \qquad \textbf{(EQ 316)}$$

As long as $E(\varepsilon) = \mathbf{0}$, the least squares estimate θ_{LS} is an ***unbiased estimate*** of $\theta^{(true)}$; that is, $E_{\text{freq}}(\theta_{LS}) = \theta^{(true)}$.

The sample variance is an unbiased estimate of the error variance for a linear model

Again, let us assume that the Gauss-Markov conditions hold,

$$E(\varepsilon^{[k]}) = 0 \qquad cov(\varepsilon^{[k]}, \varepsilon^{[j]}) = \delta_{kj}\sigma^2 \qquad \textbf{(EQ 317)}$$

After fitting the coefficients of a linear model, we have the following information available to estimate σ^2:

1. The design matrix X and the measured response vector $\boldsymbol{y}$

2. The least squares parameter estimate, $\theta_{LS} = (X^T X)^{-1} X^T \boldsymbol{y}$

3. The vector of model predictions using θ_{LS}

$$\hat{\boldsymbol{y}}(\theta_{LS}) = \begin{bmatrix} \hat{y}^{[1]}(\theta_{LS}) \\ \vdots \\ \hat{y}^{[N]}(\theta_{LS}) \end{bmatrix} = X\theta_{LS} \qquad \textbf{(EQ 318)}$$

4. The ***residual errors***

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{y}(\theta_{LS}) \qquad \textbf{(EQ 319)}$$

Note that since $\theta_{LS} \neq \theta^{(true)}$, $\hat{y}^{[k]}(\theta_{LS})$ is **not** the same as the model prediction obtained with the "true" parameters $\hat{y}^{[k]}(\theta^{(true)})$. Therefore, the residual errors that we know are **not** equal to the random errors added to the "true" model (that remain unknown).

$$\varepsilon^{[k]} \neq e^{[k]} \qquad \varepsilon^{[k]} \equiv y^{[k]} - \hat{y}^{[k]}(\theta^{(true)}) \qquad e^{[k]} \equiv y^{[k]} - \hat{y}^{[k]}(\theta_{LS}) \qquad \textbf{(EQ 320)}$$

We now examine how the known $\boldsymbol{e}$ is related to the unknown $\boldsymbol{\varepsilon}$ for a linear model, where

$$\boldsymbol{e} = \boldsymbol{y} - X\theta_{LS} \qquad \varepsilon = \boldsymbol{y} - X\theta^{(true)} \qquad \textbf{(EQ 321)}$$

In the first equation, we substitute for θ_{LS},

$$\boldsymbol{e} = \boldsymbol{y} - X\theta_{LS} = \boldsymbol{y} - X[(X^T X)^{-1} X^T \boldsymbol{y}] = [I - X(X^T X)^{-1} X^T]\boldsymbol{y} \qquad \textbf{(EQ 322)}$$

We simplify this equation by defining the matrix

$$A = I - X(X^T X)^{-1} X^T \qquad \textbf{(EQ 323)}$$

to obtain

$$\boldsymbol{e} = A\boldsymbol{y} \qquad \textbf{(EQ 324)}$$

A is a real, symmetric matrix, so that all of its eigenvalues are real and its eigenvectors form a complete orthonormal basis set.

Using $\boldsymbol{y} = X\theta^{(true)} + \varepsilon$, we have

$$\boldsymbol{e} = A\boldsymbol{y} = A[X\theta^{(true)} + \varepsilon] = AX\theta^{(true)} + A\varepsilon \qquad \textbf{(EQ 325)}$$

We next note that,

$$AX = [I - X(X^T X)^{-1} X^T]X = X - X(X^T X)^{-1}(X^T X) = X - X = 0 \qquad \textbf{(EQ 326)}$$

so that

$$AX\theta^{(true)} = \mathbf{0} \qquad \boldsymbol{e} = A\varepsilon \qquad \textbf{(EQ 327)}$$

First, note that since $E(\boldsymbol{e}) = AE(\varepsilon)$, a simple check to see whether our assumption $E(\varepsilon) = \mathbf{0}$ is violated is to examine whether $E(\varepsilon) = \mathbf{0}$. That is, when making plots of the residual errors against the response vector or any of the predictors, we should see no trend in the plot, but rather find scattered errors around zero with no apparent widening or narrowing.

This relationship $\boldsymbol{e} = A\varepsilon$ provides a method to estimate the variance σ^2 of the measurement error (unknown) from the vector of residual errors (known). We write the sum of squared residual errors, evaluated at θ_{LS}, as,

$$S(\theta_{LS}) = \sum_{k=1}^{N} [y^{[k]} - \hat{y}^{[k]}(\theta_{LS})]^2 = \sum_{k=1}^{N} (e^{[k]})^2 = \boldsymbol{e}^T \boldsymbol{e} \qquad \textbf{(EQ 328)}$$

Using the relation $\boldsymbol{e} = A\varepsilon$, we obtain

$$S(\theta_{LS}) = \boldsymbol{e}^T \boldsymbol{e} = (A\varepsilon)^T (A\varepsilon) = \varepsilon^T A^T A\varepsilon \qquad \textbf{(EQ 329)}$$

As $A = I - X(X^T X)^{-1} X^T$ is symmetric,

$$A^T = I - X(X^T X)^{T(-1)} X^T = I - X(X^T X)^{-1} X^T = A \qquad \textbf{(EQ 330)}$$

we have

$$A^T A = AA = A[I - X(X^T X)^{-1} X^T] = A - AX(X^T X)^{-1} X^T \qquad \textbf{(EQ 331)}$$

We have shown that $AX = 0$, so that the second term is zero. Thus, A is idempotent, $A^T A = AA = A$, and we have,

$$S(\theta_{LS}) = \varepsilon^T A^T A \varepsilon = \varepsilon^T A \varepsilon = \sum_{k=1}^{N}\sum_{l=1}^{N} \varepsilon^{[k]} A_{kl} \varepsilon^{[l]} \quad \textbf{(EQ 332)}$$

We now take the frequentist expectation of this equation,

$$E_{\text{freq}}\{S(\theta_{LS})\} = \sum_{k=1}^{N}\sum_{l=1}^{N} E_{\text{freq}}\{\varepsilon^{[k]} A_{kl} \varepsilon^{[l]}\}$$

$$E_{\text{freq}}\{S(\theta_{LS})\} = \sum_{k=1}^{N} A_{kk} E\{(\varepsilon^{[k]})^2\} + \sum_{k=1}^{N}\sum_{\substack{l=1 \\ l \neq k}}^{N} A_{kl} E\{\varepsilon^{[k]}\varepsilon^{[l]}\} \quad \textbf{(EQ 333)}$$

Using the Gauss-Markov assumption $E\{\varepsilon^{[k]}\varepsilon^{[l]}\} = \sigma^2 \delta_{kl}$, we have

$$E_{\text{freq}}\{S(\theta_{LS})\} = \sum_{k=1}^{N} A_{kk}\sigma^2 = \sigma^2 tr(A) \quad \textbf{(EQ 334)}$$

It is proved below that for an experimental design with N experiments and P fitted parameters,

$$tr(A) = \nu = N - \dim(\theta) = N - P \quad \textbf{(EQ 335)}$$

so that the ***sample variance*** s^2, an unbiased estimate of the random error variance σ^2, is

$$s^2 = \frac{1}{\nu} S(\theta_{LS}) = \frac{1}{\nu}\sum_{k=1}^{N} [y^{[k]} - \hat{y}^{[k]}(\theta_{LS})]^2 \qquad E_{\text{freq}}[s^2] = \sigma^2 \quad \textbf{(EQ 336)}$$

While we have derived this relation for a linear model, we can extend this as a definition of the sample variance to nonlinear models as well.

Proof that $tr(A) = \nu = N - \dim(\theta)$**:**

To show that $tr(A) = \nu = N - \dim(\theta)$, we write A explicitly

$$A = I - B \qquad B = X(X^T X)^{-1} X^T \tag{EQ 337}$$

so that

$$tr(A) = \sum_{k=1}^{N} A_{kk} = \sum_{k=1}^{N} (I-B)_{kk} = \sum_{k=1}^{N} (1) - \sum_{k=1}^{N} (B)_{kk} = N - tr(B) \tag{EQ 338}$$

As B is idempotent,

$$BB = [X(X^T X)^{-1} X^T][X(X^T X)^{-1} X^T] = X(X^T X)^{-1} X^T = B \tag{EQ 339}$$

and symmetric positive-semidefinite,

$$\begin{aligned} B^T &= [X(X^T X)^{-1} X^T]^T = X(X^T X)^{T(-1)} X^T = B \\ \boldsymbol{v}^T B \boldsymbol{v} &= \boldsymbol{v}^T [X(X^T X)^{T(-1)} X^T] \boldsymbol{v} = (X^T \boldsymbol{v})^T [(X^T X)^{T(-1)}](X^T \boldsymbol{v}) \geq 0 \end{aligned} \tag{EQ 340}$$

all eigenvalues of B must take a value of either 0 or 1, since if

$$B\boldsymbol{w}^{[j]} = \lambda_j \boldsymbol{w}^{[j]} \tag{EQ 341}$$

and we write an arbitrary vector $\boldsymbol{v}$ as

$$\boldsymbol{v} = \sum_{j=1}^{N} c_j \boldsymbol{w}^{[j]} \tag{EQ 342}$$

then

$$B\boldsymbol{v} = \sum_{j=1}^{N} c_j \lambda_j \boldsymbol{w}^{[j]} \qquad B^2 \boldsymbol{v} = \sum_{j=1}^{N} c_j \lambda_j^2 \boldsymbol{w}^{[j]} \tag{EQ 343}$$

Now, as $BB = B$, for all $\boldsymbol{v}$, $B\boldsymbol{v} = B^2 \boldsymbol{v}$, so that for every eigenvalue of B,

$$\lambda_j = \lambda_j^2 \qquad \Rightarrow \qquad \lambda_j = 0 \text{ or } 1 \tag{EQ 344}$$

As the trace of a matrix equals the sum of its eigenvalues,

$$tr(B) = \lambda_1 + \lambda_2 + \ldots + \lambda_N \tag{EQ 345}$$

$tr(B)$ equals the number of non-zero eigenvalues of B. $B = X(X^TX)^{-1}X^T$ is determined completely by the design matrix of dimension $N \times P$. Therefore, the number of non-zero eigenvalues of *B* equals the number of linearly independent columns of the design matrix. Assuming that no two of the *P* columns of *X* are dependent and that $N \geq P$; that is, there are at least as many experiments as parameters to be fitted,

$$tr(B) = P = \dim(\theta) \tag{EQ 346}$$

and

$$tr(A) = N - \dim(\theta) \tag{EQ 347}$$

Q.E.D.