Numerical Methods in Physics with Python, 2nd edition, Online Appendix

ALEX GEZERLIS

Appendix D Matrix odds and ends

D

Big book, big evil.

Callimachus

Here we provide some matrix-related material that was too extensive for the print version of chapter 4 in the book. Specifically, we extend section 4.2 (which was limited to mathematical derivations) with "experimental" investigations of matrix perturbations and the effect they have on linear-system and eigenproblem solutions. Similarly, as a preamble to the Project in section 4.6, we provide the theory behind the matrix representation of interacting spin-half particles in quantum mechanics.

D.1 Matrix perturbation theory by example

We now turn to a discussion of practical error estimation in work with matrices. In the spirit of chapter 2, this will entail us finding worst-case (pessimistic) error bounds. Note that this will not amount to a detailed error analysis of specific methods, say, for the solution of linear systems of equations. Instead, we will provide some general derivations and examples of when a problem is "well-conditioned", typically by using matrix perturbation theory (i.e., by checking what happens if there are uncertainties in the input data). An explicit analysis of specific methods (like the pioneering work by Wilkinson in the 1960s on Gaussian elimination) ends up showing that rounding errors are equivalent to perturbations of the input data, so in essence this is precisely what we will be probing.

Thus, in what follows, after some preliminary comments, examples, and definitions, we will investigate quantitatively how linear systems, eigenvalues, and eigenvectors depend on the input data. We will be examining in each case the simplest scenario but, hopefully, this will be enough to help you grasp the big picture. The present section will introduce a large number of examples and discuss their properties: to streamline the presentation, we don't show the analytical manipulations or Python code that is used to produce specific numbers. Once you are comfortable with the concepts at play, you can use Python programs like those introduced in the following sections (or the functionality contained in `numpy.linalg`) to verify our numerical results.

D.1.1 From *a posteriori* to *a priori* Estimates

Let's study a specific 2 × 2 linear system, namely $\mathbf{Ax} = \mathbf{b}$ for the case where:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{cc|c} 0.2161 & 0.1441 & 0.1440 \\ 1.2969 & 0.8648 & 0.8642 \end{array}\right) \tag{D.1}$$

This problem was introduced by W. Kahan (who was also responsible for many of the examples we studied in chapter 2) in Ref. 4. We are stating from the outset that this example is contrived. That being said, the misbehavior we are about to witness is not a phenomenon that happens only to experts who are looking for it. It is merely a more pronounced case of problematic behavior that does appear in the real world.

Simply put, there are two options on how to analyze errors: (a) an *a priori* analysis, in which case we try to see how easy/hard the problem is to solve before we begin solving it, and (b) an *a posteriori* analysis, where we have produced a solution, and attempt to see how good it is. Let's start with the latter option, namely an *a posteriori* approach.

Say you are provided with the following approximate solution to the problem in Eq. (D.1):

$$\tilde{\mathbf{x}}^T = \begin{pmatrix} 0.9911 & -0.4870 \end{pmatrix} \tag{D.2}$$

We are showing the transpose to save space on the page; we will keep doing this below. One way of testing how good a solution this is, is to evaluate the *residual vector*:

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} \tag{D.3}$$

Qualitatively, you can immediately grasp this vector's meaning: since the "true" solution $\mathbf{x}$ satisfies $\mathbf{Ax} = \mathbf{b}$, an approximate solution $\tilde{\mathbf{x}}$ should "almost" satisfy the same equation. Plugging in the matrices gives us for this case:

$$\mathbf{r}^T = \begin{pmatrix} -10^{-8} & 10^{-8} \end{pmatrix} \tag{D.4}$$

which might naturally lead you to the conclusion that our approximate solution $\tilde{\mathbf{x}}$ is pretty good, i.e., it may suffer from minor rounding-error issues (say, in the last digit or something) but other than that it's a done deal. Here's the thing, though: the exact solution to our problem is actually:

$$\mathbf{x}^T = \begin{pmatrix} 2 & -2 \end{pmatrix} \tag{D.5}$$

as you can easily see by substituting in the starting equation, $\mathbf{Ax} = \mathbf{b}$ (in other words, we get a zero residual vector for the exact solution). Thus, far from being only slightly off, our approximate "solution" $\tilde{\mathbf{x}}$ doesn't contain even a single correct significant figure.

With the disclaimer that there's much more that could be said at the *a posteriori* level, we now drop this line of attack and turn to an *a priori* analysis: could we have realized that solving the problem in Eq. (D.1) was difficult? How could we know that there's something pathological about it?

D.1.2 Magnitude of Determinant?

D.1.2.1 Example 1

In an attempt to see what is wrong with our example:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{cc|c} 0.2161 & 0.1441 & 0.1440 \\ 1.2969 & 0.8648 & 0.8642 \end{array}\right) \tag{D.6}$$

we start to make small perturbations to the input data. Imagine we didn't know the values of, say, the coefficients in **A** all that precisely. Would anything change then? Take:

$$\Delta\mathbf{A} = \begin{pmatrix} 0.0001 & 0 \\ 0 & 0 \end{pmatrix} \tag{D.7}$$

This is employing notation that is analogous to that in chapter 2: an absolute perturbation is $\Delta\mathbf{A}$. To be explicit, what we are now studying is the effect of a perturbation in **A** on our solution. In other words, we are now solving the linear system:

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} \tag{D.8}$$

where the constant vector **b** is kept fixed/unperturbed. For the specific case studied here:

$$(\mathbf{x} + \Delta\mathbf{x})^T = \begin{pmatrix} -2.3129409051813273 \times 10^{-4} & 9.996530588644692 \times 10^{-1} \end{pmatrix} \tag{D.9}$$

By any reasonable definition of the word, this is not a "small" effect. Our perturbation from Eq. (D.7) amounted to changing only one element of **A** by less than 0.1% and had a dramatic impact on the solution to our problem.

D.1.2.2 Example 2

You might be thinking that this is all a result of our example in Eq. (D.6) being contrived. OK, let's look at another linear system of equations:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 1 & 1.001 & 2.001 \end{array}\right) \tag{D.10}$$

The exact solution to this problem is:

$$\mathbf{x}^T = \begin{pmatrix} 1 & 1 \end{pmatrix} \tag{D.11}$$

as you can easily see by substituting in the starting equation, $\mathbf{Ax} = \mathbf{b}$.

We will now make a small perturbation to our coefficient matrix and see what happens. In other words, we will again solve Eq. (D.8). As before, we make a small change to the coefficient matrix, again of less than 0.1% in only one element (adding 0.001 to the bottom-right element of **A** this time). If you solve the new set of equations, you will find:

$$(\mathbf{x} + \Delta\mathbf{x})^T \approx \begin{pmatrix} 1.5 & 0.5 \end{pmatrix} \tag{D.12}$$

In this case, too, a change of a single element in the coefficient matrix by less than 0.1% led to a large effect (as much as 50%) on the solution vector.

We can, similarly, also perturb the constant vector **b**. In other words, we can try to solve:

$$\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} + \Delta\mathbf{b} \tag{D.13}$$

where **A** is left untouched. As usual, let's take a specific case, adding in 0.001 to the bottom element. This leads to a solution which is 100% different from the unperturbed one:

$$(\mathbf{x} + \Delta\mathbf{x})^T = \begin{pmatrix} 0 & 2 \end{pmatrix} \tag{D.14}$$

D.1.2.3 Example 3

Perturbations like $\Delta\mathbf{A}$ or $\Delta\mathbf{b}$ above may result from rounding error: in physics applications, the matrices **A** and **b** are often themselves the result of earlier calculations, i.e., not set in stone. They may also result from uncertainty in the input data. If you are thinking that small perturbations will always lead to dramatic consequences in the solution, know that this is *not* true: it's just a result of studying the specific cases of Eq. (D.6) and Eq. (D.10).

To see that this is, indeed, the case, let's look at a third example:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{cc|c} 2 & 1 & 2 \\ 1 & 2 & 7 \end{array}\right) \tag{D.15}$$

The exact solution to this problem is:

$$\mathbf{x}^T = \begin{pmatrix} -1 & 4 \end{pmatrix} \tag{D.16}$$

To probe the sensitivity of this problem to perturbations, we will make a slightly larger change to the coefficient matrix, this time of 1% in one element (adding 0.01 to the bottom-left one). If you solve the new set of equations, you will find:

$$(\mathbf{x} + \Delta\mathbf{x})^T = \begin{pmatrix} -1.003344481605351 & 4.006688963210702 \end{pmatrix} \tag{D.17}$$

Finally, here's a case where a small perturbation (1% change) in the coefficient matrix has a *small* effect (less than 0.5%) in the solution vector. Perhaps not all linear-system problems behave as strangely as the first two we studied in this section.

We, similarly, can perturb the constant vector **b**, by adding 0.01 to the top element, a change of less than 0.5%. This leads to:

$$(\mathbf{x} + \Delta\mathbf{x})^T = \begin{pmatrix} -0.9933333333333334 & 3.9966666666666666 \end{pmatrix} \tag{D.18}$$

which is a bit more than 0.5% different from our unperturbed solution. In this case, too, we see that our problem from Eq. (D.15) is much better behaved than the earlier ones.

What is making some problems behave poorly (i.e., be very sensitive to tiny perturbations) and others to behave better? One criterion that is sometimes mentioned in this context is: since (as we see in appendix C.2) a non-invertible/singular matrix has determinant 0, it is plausible that matrices that have determinants that are "close to 0" are close to being singular. Let's look at this potential criterion in more detail. For Example 1 we find $|\mathbf{A}| \approx -10^{-8}$, for Example 2 we find $|\mathbf{A}| = 0.001$, and for Example 3 we find $|\mathbf{A}| = 3$. Thus, at first sight, our criterion regarding the "smallness of the determinant" appears to be borne out by the facts: the examples that had small determinants were very sensitive to tiny perturbations, whereas the example with a larger determinant was not sensitive. (Don't stop reading here, though.)

D.1.3 Norms for Matrices and Vectors

D.1.3.1 Example 4

Consider the following question: what does "small determinant" mean? If the definition is "much less than 1", then one might counter-argue: what about the following matrix:

$$\mathbf{A} = \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.2 \end{pmatrix} \tag{D.19}$$

As you may have noticed, this is simply our matrix from Eq. (D.15), with each element multiplied by 0.1. Our new matrix has $|\mathbf{A}| = 0.03$, which is certainly smaller than 1. But here's the thing: if you also multiply each element in **b** by 0.1, you will find the same solution as in Eq. (D.16). (You shouldn't be surprised: this is simply the result of multiplying two equations by a constant.) What's more, the linear system of equations will be equally insensitive to perturbations in **A** or **b**. Obviously, our tentative definition that "small determinant means much less than 1" leaves a lot to be desired, since a determinant can be made as large/small as we wish by multiplying each element by a given number.

Intuitively, it makes sense to think of a "small determinant" as having something to do with the magnitude of the relevant matrix elements: in the case we just studied, the determinant was small, but so were the matrix elements involved. In contradistinction to this, our Example 2 above had matrix elements of order 1 but $|\mathbf{A}| = 0.001$, so it stands to reason that that determinant was "truly" small.

D.1.3.2 Definitions and Properties for Matrices

Let us provide our intuitions with quantitative backing. We will introduce the *matrix norm*, which measures the magnitude of **A**. There are several possible definitions of a norm, but we will employ one of two possibilities. First, we have the *Frobenius norm*:

$$||\mathbf{A}||_F = \sqrt{\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\left|A_{ij}\right|^2} \tag{D.20}$$

Note that double vertical lines are used to denote the norm. This is different from single vertical lines, used to denote the determinant of a matrix or the absolute value of a real number or the modulus of a complex number. Another popular definition is that of the *infinity norm*:

$$||\mathbf{A}||_\infty = \max_{0\le i\le n-1} \sum_{j=0}^{n-1} \left|A_{ij}\right| \tag{D.21}$$

which is also known as the *maximum row-sum norm*. As you can see, both of these definitions try to measure the magnitude of the various matrix elements. Other definitions choose different ways to accomplish this (e.g., maximum column sum).

Regardless of the specific definition employed, all matrix norms for square matrices obey the following properties:

$$\begin{aligned}
&||\mathbf{A}|| \ge 0\\
&||\mathbf{A}|| = 0 \text{ if and only if all } A_{ij} = 0\\
&||k\mathbf{A}|| = |k|\; ||\mathbf{A}||\\
&||\mathbf{A} + \mathbf{B}|| \le ||\mathbf{A}|| + ||\mathbf{B}||\\
&||\mathbf{AB}|| \le ||\mathbf{A}||\; ||\mathbf{B}||
\end{aligned} \tag{D.22}$$

Notice that a matrix norm is a number, not a matrix.[1] The fourth of these relations is known as the *triangle inequality* and should be familiar to you from other contexts.

We can now return to the question of when the determinant is “small”. A reasonable definition would be $|\det(\mathbf{A})| \ll ||\mathbf{A}||$, where we took the absolute value on the left-hand side and used the det notation to avoid any confusion. This new criterion has the advantage that it takes into account the magnitude of the matrix elements. Let’s test it out for the cases discussed above (employing the Frobenius norm, for the sake of concreteness):

- Example 1: $|\det(\mathbf{A})| \approx 10^{-8}$ and $||\mathbf{A}||_F \approx 1.58$, so $|\det(\mathbf{A})| \ll ||\mathbf{A}||_F$ holds.
- Example 2: $|\det(\mathbf{A})| \approx 0.001$ and $||\mathbf{A}||_F \approx 2.0$, so $|\det(\mathbf{A})| \ll ||\mathbf{A}||_F$ holds.
- Example 3: $|\det(\mathbf{A})| = 3$ and $||\mathbf{A}||_F \approx 3.16$, so $|\det(\mathbf{A})| \ll ||\mathbf{A}||_F$ does *not* hold.
- Example 4: $|\det(\mathbf{A})| = 0.03$ and $||\mathbf{A}||_F \approx 0.32$, so $|\det(\mathbf{A})| \ll ||\mathbf{A}||_F$ does not really hold.

These results seem to be consistent with what we had seen above: Examples 1 and 2 are near-singular, while Example 3 is not singular. For Example 4, this criterion claims that our matrix is not quite singular (though it’s getting there). Our introduction of the concept of the matrix norm seems to have served its purpose: a small determinant needs to be compared to the matrix norm, so Example 4 (despite having a small determinant) is not singular, given that its matrix elements are small, too.

D.1.3.3 Definitions for Vectors

In what follows, we’ll also make use of norms of column vectors, so we briefly go over two such definitions:

$$||\mathbf{x}||_E = \sqrt{\sum_{i=0}^{n-1} |x_i|^2}, \quad ||\mathbf{x}||_\infty = \max_{0\le i\le n-1} |x_i| \tag{D.23}$$

[1] This is analogous to a matrix determinant, which quantifies an entire matrix, but is not a matrix itself.

These are the *Euclidean norm* and the *infinity norm*, respectively. The latter is also known as the maximum-magnitude norm.

D.1.4 Condition Number for Linear Systems

D.1.4.1 Example 5

Unfortunately, our criterion $|\det(\mathbf{A})| \ll ||\mathbf{A}||$ is flawed (its appearance in textbooks notwithstanding).[2] We'll look at only two examples of how it can lead us astray. The first one is already implicit in what we saw above: take Example 3 and multiply the matrix elements by a small number. We have:

$$\mathbf{A} = \begin{pmatrix} 2\times 10^{-10} & 1\times 10^{-10} \\ 1\times 10^{-10} & 2\times 10^{-10} \end{pmatrix} \tag{D.24}$$

As advertised, this is simply our matrix from Eq. (D.15), with each element multiplied by 10^{-10}. Let's summarize where things stand for this case:

- Example 5: $|\det(\mathbf{A})| = 3\times 10^{-20}$ and $||\mathbf{A}||_F \approx 3.16\times 10^{-10}$, so $|\det(\mathbf{A})| \ll ||\mathbf{A}||_F$ holds.

But isn't this strange? Simply multiplying a set of equations by a small number cannot be enough to make the problem near-singular. Our intuition is borne out by a more detailed investigation: just like for Example 3, a 1% change in one element of **A** will have an effect of less than 0.5% on the solution-vector elements. Thus, for Example 5 the linear system of equations will be equally insensitive to perturbations in **A** or **b** as Example 3 was.

D.1.4.2 Example 6

We just saw a case where the determinant is much smaller than the norm, yet the problem is not sensitive to perturbations/near-singular. Now, let's turn to our next counter-example. We will find the reverse: a determinant that is much larger than the norm for a problem that *is* near-singular. Let's look at the following 8 × 8 problem:

$$\mathbf{A} = \begin{pmatrix} 2 & -2 & -2 & \dots & -2 \\ 0 & 2 & -2 & \dots & -2 \\ 0 & 0 & 2 & \dots & -2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 2 \end{pmatrix} \tag{D.25}$$

The corresponding results are:

- Example 6: $|\det(\mathbf{A})| = 256$ and $||\mathbf{A}||_F = 12$, so in this case $|\det(\mathbf{A})| \gg ||\mathbf{A}||_F$ holds (with a $\gg$, not a $\ll$).

[2] As Jonathan Swift put it in his 1731 *Verses on the Death of Dr. Swift*:
"Yet malice never was his aim;
He lash'd the vice, but spar'd the name."

OK, so the criterion is clearly not satisfied. Is this reason for concern? Well, try pairing this matrix with the following constant vector:

$$\mathbf{b}^T = \begin{pmatrix} +1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \end{pmatrix} \tag{D.26}$$

You should (analytically or not) find the following solution vector:

$$\mathbf{x}^T = \begin{pmatrix} -21 & -11 & -5 & -3 & -1 & -1 & 0 & -1/2 \end{pmatrix} \tag{D.27}$$

Let us now introduce a small perturbation (inspired by Ref. 6) of −0.01 in the bottom-left element of **A**. We find the following perturbed solution vector:

$$(\mathbf{x} + \Delta\mathbf{x})^T \approx \begin{pmatrix} -30.88 & -15.94 & -7.47 & -4.24 & -1.62 & -1.31 & -0.15 & -0.65 \end{pmatrix} \tag{D.28}$$

where we rounded for ease of viewing. Thus, we carried out a tiny change in 1 out of 64 matrix elements (with a magnitude that is 0.5% of a matrix element value) and ended up with a solution vector whose matrix elements are different by as much as 60%. In other words, this matrix is very sensitive to perturbations in the initial data.

To summarize, for Example 5 the criterion is satisfied but the matrix is not near-singular, whereas for Example 6 the criterion is not satisfied but the matrix is near-singular. These two examples should be enough to convince you that, when doing an *a priori* investigation into a linear system of equations, you should not bother with testing whether or not $|\det(\mathbf{A})| \ll \|\mathbf{A}\|$ holds.

D.1.4.3 Derivation

So where does this leave us? Just because we had a faulty criterion does not mean that a good one cannot be arrived at. In the present subsection, we will carry out an informal derivation that will point us toward a quantitative measure of ill-conditioning. (Spoiler alert: it does not involve the determinant.) This measure of the sensitivity of our problem to small changes in its elements will be called the *condition number*.

Let us start with the unperturbed problem:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{D.29}$$

and combine that with the case where **A** is slightly changed (as above), with **b** being held constant. Obviously, this will also impact the solution vector, as we saw above. The relevant equation is:

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{b} \tag{D.30}$$

Of course, you could have chosen to also perturb the elements of the constant vector **b** (either at the same time or separately). These scenarios are explored in problems 4.3 and 4.4.

Expanding out the parentheses in Eq. (D.30) and plugging in Eq. (D.29), we find:

$$\mathbf{A}\Delta\mathbf{x} = -\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) \tag{D.31}$$

Assuming **A** is non-singular (so you can invert it), you get:

$$\Delta\mathbf{x} = -\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x}) \tag{D.32}$$

Taking the norm on both sides we find:

$$||\Delta\mathbf{x}|| = ||\mathbf{A}^{-1}\Delta\mathbf{A}(\mathbf{x} + \Delta\mathbf{x})|| \leq ||\mathbf{A}^{-1}||\ ||\Delta\mathbf{A}||\ ||\mathbf{x} + \Delta\mathbf{x}|| \leq ||\mathbf{A}^{-1}||\ ||\Delta\mathbf{A}||\ ||\mathbf{x}|| \tag{D.33}$$

In the first step all we did was to take the absolute value of -1 (third property in Eq. (D.22)) and in the second step we simply applied the fifth property in Eq. (D.22), twice. In the third step we used the triangle inequality and dropped the second-order term. Using the non-negativity of norms (first property in Eq. (D.22)), we get:

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}||} \leq ||\mathbf{A}^{-1}||\ ||\Delta\mathbf{A}|| \tag{D.34}$$

Multiplying and dividing by a constant on the right-hand side gives us:

$$\frac{||\Delta\mathbf{x}||}{||\mathbf{x}||} \leq ||\mathbf{A}||\ ||\mathbf{A}^{-1}||\ \frac{||\Delta\mathbf{A}||}{||\mathbf{A}||} \tag{D.35}$$

In other words, if you know an error bound on $||\Delta\mathbf{A}||/||\mathbf{A}||$ then that translates to an error bound on $||\Delta\mathbf{x}||/||\mathbf{x}||$. The coefficient in front of $||\Delta\mathbf{A}||/||\mathbf{A}||$ determines if a small perturbation gets magnified when solving for **x** or not.

This derivation naturally leads us to the introduction of the following *condition number*:

$$\kappa(\mathbf{A}) = ||\mathbf{A}||\ ||\mathbf{A}^{-1}|| \tag{D.36}$$

A large condition number leads to an amplification of a small perturbation: we say we are dealing with an *ill-conditioned* problem. If the condition number is of order unity, then a small perturbation is not amplified, so we are dealing with a *well-conditioned* problem (the condition number is bounded below by unity). Qualitatively, this condition number tells us both how well- or ill-conditioned the solution of the linear problem $\mathbf{Ax} = \mathbf{b}$ is, as well as how well- or ill-conditioned the inversion of matrix **A** is. This dual role is not surprising: conceptually (though not in practice) solving a linear system is equivalent to inverting the matrix on the left-hand side. Obviously, the precise value of the condition number depends on which norm you are using (but we'll employ only the Frobenius norm here).

D.1.4.4 Examples

To get a feeling for the relevant magnitudes, let's evaluate the condition number for the six examples encountered above:

- Example 1: $\kappa(\mathbf{A}) = 249\,729\,267.388$, so the problem is (terribly) ill-conditioned.
- Example 2: $\kappa(\mathbf{A}) = 4002.001$, so the problem is ill-conditioned.

- Example 3: $\kappa(\mathbf{A}) \approx 3.33$, so the problem is well-conditioned.
- Example 4: $\kappa(\mathbf{A}) \approx 3.33$, so the problem is well-conditioned.
- Example 5: $\kappa(\mathbf{A}) \approx 3.33$, so the problem is well-conditioned.
- Example 6: $\kappa(\mathbf{A}) \approx 512.18$, so the problem is ill-conditioned.

We notice that Examples 3, 4, and 5 (which had quite different determinants) all have the same condition number: this makes perfect sense, since they *are* the same problem, just scaled with an overall factor. Our results for all six problems are consistent with what we discovered above by "experimentally" perturbing some elements in **A**: those matrices with a large condition number lead to a larger relative change being propagated onto the solution vector. What "large" actually means may depend on the specifics of your problem, but typically anything above 100 is large.

Thus, the condition number manages to quantify the sensitivity to perturbations *ahead of time*: you can tell that you will be sensitive to small perturbations even before you start solving the linear system of equations. Specifically, you might appreciate knowing the following *rule of thumb*: for a matrix with condition number 10^k, if you perturb the matrix elements in their t-th digits, then you will be perturbing the matrix elements of the inverse in their $(t - k)$-th digits; as noted earlier, these perturbations might not be some arbitrary change you are carrying out by hand, but the inevitable result of the solution method you are employing. As a matter of fact, the condition number also encapsulates how close we are to being singular: for an ill-conditioned matrix you can construct a small perturbation that will render the matrix singular. (Obviously, for a well-conditioned matrix you need a large perturbation to make the matrix singular.)

D.1.4.5 Example 7

Let's keep hammering away at the irrelevance of the determinant in connection with how well- or ill-conditioned a problem is. Look at the following 8×8 matrix:

$$\mathbf{A} = \begin{pmatrix} 0.1 & 0.1 & 0.1 & \dots & 0.1 \\ 0 & 0.1 & 0.1 & \dots & 0.1 \\ 0 & 0 & 0.1 & \dots & 0.1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0.1 \end{pmatrix} \tag{D.37}$$

If you are thinking in terms of determinants, you might confuse yourself into believing that having many similar elements leads to ill-conditioning (since determinants combine many + and − in sequence). This matrix should convince you that this conclusion is untrue:

- Example 7: $\kappa(\mathbf{A}) \approx 23.24$, so the problem is well-conditioned, despite the fact that $|\det(\mathbf{A})| = 10^{-8}$ and $\|\mathbf{A}\|_F = 0.6$.

As it so happens, in this case (of a triangular matrix with all the elements being identical), the +'s and −'s never show up in evaluating the determinant, since for a triangular matrix the determinant is simply the product of the elements on the diagonal.

D.1.4.6 Final Remarks

Both the $\kappa(\mathbf{A})$ we introduced and the judiciously chosen perturbations we showed for specific examples are probing the worst-case situation, i.e., they can be too pessimistic. This is not too troubling given our limited goals: we are merely providing some quantitative insight for when you should be careful. This does not mean that every single time you get a large $\kappa(\mathbf{A})$ you will be extremely sensitive to minor perturbations. As a matter of fact, in actual physical applications rounding or other errors typically accumulate not on an individual matrix element (as in the cases we explicitly tried out above), but in entire rows, entire columns, or the entire matrix itself. In other words, what we've been exploring has to some degree been a worst-case "artificial ill-conditioning" scenario, in order to teach you what to look out for. Depending on your needs, you might have to study other condition numbers.

Another qualitative point: to evaluate the condition number $\kappa(\mathbf{A})$ from Eq. (D.36) we need to first compute the matrix inverse $\mathbf{A}^{-1}$. This raises two issues: (a) wouldn't the computation of the inverse necessitate use of the same methods (such as those discussed in section 4.3 below) whose appropriateness we are trying to establish in the first place?, and (b) computing the inverse requires $O(n^3)$ operations (as we will discuss below) where the coefficient in front is actually more costly than the task we were faced with (solving a system of equations). But then it hardly seems reasonable to spend so much effort only to determine ahead of time how well you may end up doing at solving your problem. We won't go into details here, but both of these concerns are addressed by established practice: use an alternative method to *estimate* the condition number (within a factor of 10 or so) using only $O(n^2)$ operations. Thus, you can quickly get a rough estimate of how ill-conditioned your problem is going to be, before you start out on a detailed study.

D.1.5 Condition Number for Simple Eigenvalues

Having studied the sensitivity of a linear system to small perturbations, the natural place to go from there is to carry out an analogous study for the eigenvalue problem. As you might have expected, we begin with a few examples of matrices and some explicit perturbations added in "by hand". In this case, however, we are interested in solving not the linear system of equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, but the eigenvalue problem $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$. For now, let us focus on the effect of the perturbations on the evaluation of the eigenvalues, λ. We will study two examples that were proposed by J. H. Wilkinson (Ref. 7).

D.1.5.1 Example 8

Take:

$$\mathbf{A} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 3 & 2 & 1 \\ 0 & 2 & 2 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \tag{D.38}$$

You can evaluate its eigenvalues using the characteristic polynomial (as per section 4.1.2) or a more robust method from section 4.4 below. You will find:

$$\lambda_0 \approx 7.31274,\ \lambda_1 \approx 2.06663,\ \lambda_2 \approx 0.483879,\ \lambda_3 \approx 0.136748 \tag{D.39}$$

where we ordered the eigenvalues by decreasing magnitude.

If you now introduce a small change to one out of 16 matrix elements (adding 0.01 to the top right element) and recalculate the eigenvalues, you will find:

$$\lambda_0 + \Delta\lambda_0 \approx 7.31298,\ \lambda_1 + \Delta\lambda_1 \approx 2.06287,\ \lambda_2 + \Delta\lambda_2 \approx 0.499374,\ \lambda_3 + \Delta\lambda_3 \approx 0.124777 \tag{D.40}$$

It's easy to see that (for this example) the smaller the eigenvalue, the bigger the impact of our small perturbation. For λ_3 we go from 0.136748 to 0.124777, a change (in absolute terms) that is a bit larger than the perturbation in our matrix element.

At this point, having read the previous subsection, you may be thinking that this specific matrix may have a large condition number $\kappa(\mathbf{A})$, which would explain (or so you think) the sensitivity to small perturbations when computing eigenvalues. It turns out that this matrix has $\kappa(\mathbf{A}) \approx 126.744$, so indeed it is ill-conditioned (according to our somewhat arbitrary demarcation point of taking condition numbers above 100 as "large").

D.1.5.2 Example 9

Well, it's time to (once again) shoot down our tentative criterion. The following matrix:

$$\mathbf{A} = \begin{pmatrix} 4 & 4 & 0 & 0 \\ 0 & 3 & 4 & 0 \\ 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{D.41}$$

has the eigenvalues:

$$\lambda_0 = 4,\ \lambda_1 = 3,\ \lambda_2 = 2,\ \lambda_3 = 1 \tag{D.42}$$

This is a triangular matrix, so its eigenvalues are conveniently placed in its diagonal.

As per our previous tentative exploration, we note that this new matrix has a condition number $\kappa(\mathbf{A}) \approx 40.13$, so we expect it to be well-conditioned, or at least better-conditioned than the previous Example (don't take our word for it: solve the system that arises when you pick a "typical" constant vector $\mathbf{b}$). But when we introduce a tiny change to one out of 16 matrix elements (adding 0.005 to the bottom left one) and recalculate the eigenvalues (using the same approach), we find:

$$\lambda_0 + \Delta\lambda_0 \approx 4.04884,\ \lambda_1 + \Delta\lambda_1 \approx 2.81794,\ \lambda_2 + \Delta\lambda_2 \approx 2.18206,\ \lambda_3 + \Delta\lambda_3 \approx 0.951158 \tag{D.43}$$

Looking only at λ_2, we notice that it has changed from 2 to 2.18206, a change (in absolute terms) larger than 30 times the perturbation in our matrix element. Despite the fact that Example 9 has a smaller $\kappa(\mathbf{A})$ than Example 8 did, it appears to be more sensitive to small perturbations in its elements, as far as the computation of its eigenvalues is concerned.

D.1.5.3 Back to Example 3

It's starting to look like a matrix's eigenvalues can be sensitive to small perturbations, regardless of whether or not $\kappa(\mathbf{A})$ is large. Even so, you shouldn't walk away from this discussion with the impression that for any matrix you pick the eigenvalues will be highly sensitive to small perturbations in the matrix elements. To show this in action, let's return to our middle-of-the-road case, Example 3, Eq. (D.15). Its eigenvalues are:

$$\lambda_0 = 3, \ \lambda_1 = 1 \tag{D.44}$$

Making the small change of adding 0.01 to the bottom left element of the coefficient matrix, and then recalculating its eigenvalues we find them to be:

$$\lambda_0 + \Delta\lambda_0 \approx 3.005, \ \lambda_1 + \Delta\lambda_1 \approx 0.995 \tag{D.45}$$

namely an impact that is (in absolute terms) half the size of the perturbation. It's nice to see that a matrix can have eigenvalues that don't immediately start dramatically changing when you perturb individual matrix elements.

D.1.5.4 Back to Example 1

To (temporarily) complicate things even further, we now return to the matrix **A** in Example 1, Eq. (D.6); as you may recall, this was extremely ill-conditioned when solving a linear system of equations. Calculating its eigenvalues, we find:

$$\lambda_0 \approx 1.0809, \ \lambda_1 \approx -9.25155 \times 10^{-9} \tag{D.46}$$

Making the small change of adding 0.0001 to the top left element of the coefficient matrix and recalculating its eigenvalues, we find them to be:

$$\lambda_0 + \Delta\lambda_0 \approx 1.08092, \ \lambda_1 + \Delta\lambda_1 \approx 7.99967 \times 10^{-5} \tag{D.47}$$

We see that for λ_0 the change is five times *smaller* than the perturbation. As far λ_1 is concerned, while this eigenvalue changes quite a bit, it's worth observing that the change (in absolute terms) is still smaller than the perturbation. It looks like the worst-behaved matrix we've encountered so far is not too sensitive when it comes to the sensitivity (in absolute terms) of its eigenvalues to small changes in the input matrix elements.

D.1.5.5 Derivation

So where does this leave us? We saw that some matrices have eigenvalues that are sensitive to small perturbations, whereas others do not. We tried to use the same condition number as for the linear system problem, $\mathbf{Ax} = \mathbf{b}$, but were disappointed. In the present subsection, we will carry out an informal derivation that will point us toward a quantitative measure of conditioning *eigenvalues*. (Spoiler alert: it is not $\kappa(\mathbf{A})$.) This quantitative measure of the sensitivity of our problem to small changes in the input matrix elements will be called the *condition number for simple eigenvalues*; "simple" means we don't have repeated eigenvalues (this is done to streamline the presentation).

Let us start with the unperturbed problem:

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{D.48}$$

We are using explicit indices, so we can keep track of the different eigenvalues and the corresponding eigenvectors. By complete analogy to what we did in our derivation for the linear system in Eq. (D.30) above, the relevant perturbed equation now is:

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{v}_i + \Delta\mathbf{v}_i) = (\lambda_i + \Delta\lambda_i)(\mathbf{v}_i + \Delta\mathbf{v}_i) \tag{D.49}$$

Here we are carrying out an (absolute) perturbation of the matrix **A** and checking to see its impact on λ_i and on $\mathbf{v}_i$.[3]

So far (and for most of this chapter) we are keeping things general, i.e., we have not made an assumption that **A** is symmetric (which would have simplified things considerably). This is as it should be, since our Examples 8 and 9 above were clearly not symmetric and we're trying to derive a condition number that will help us understand *a priori* why they behave the way they do. We now realize that we've been calling the column vectors $\mathbf{v}_i$ that appear in Eq. (D.48) "eigenvectors" though, properly speaking, they should be called *right eigenvectors*. If we have access to "right eigenvectors", then it stands to reason that we can also introduce the *left eigenvectors* $\mathbf{u}_i$ as follows:

$$\mathbf{u}_i^T \mathbf{A} = \lambda_i \mathbf{u}_i^T \tag{D.50}$$

where more generally we should have been taking the Hermitian conjugate/conjugate-transpose, †, but this distinction won't matter to us, since in all our applications everything will be real-valued. Notice how this works: **A** is an $n \times n$ matrix whereas $\mathbf{v}_i$ and $\mathbf{u}_i$ are $n \times 1$ column vectors (so $\mathbf{u}_i^T$ is a $1 \times n$ row vector). Notice also a very simple way of evaluating left-eigenvectors if you already have a method to produce right eigenvectors: simply take the transpose of Eq. (D.50) to find:

$$\mathbf{A}^T \mathbf{u}_i = \lambda_i \mathbf{u}_i \tag{D.51}$$

Thus, the right eigenvectors of the transpose of a matrix give you the left eigenvectors of the matrix itself (remarkably, corresponding to the *same* eigenvalues, as you will show in Problem 4.7). Since we are not assuming that we're dealing with a symmetric matrix, in general $\mathbf{A} \neq \mathbf{A}^T$, so the left eigenvectors $\mathbf{u}_i$ are different from the right eigenvectors $\mathbf{v}_i$.

We will now use the last three boxed equations to derive an error bound on the magnitude of the change of an eigenvalue, $\Delta\lambda_i$. Start with Eq. (D.49) and expand the parentheses out. If you also take second-order changes (of the form $\Delta \times \Delta$) as being negligible, you find:

$$\mathbf{A}\Delta\mathbf{v}_i + \Delta\mathbf{A}\mathbf{v}_i = \lambda_i \Delta\mathbf{v}_i + \Delta\lambda_i \mathbf{v}_i \tag{D.52}$$

[3] Actually, right now we are only interested in the impact on λ_i, so we'll try to eliminate $\Delta\mathbf{v}_i$ here: we return to the sensitivity of eigenvectors in the following subsection.

where we also made use of Eq. (D.48) in order to cancel two terms. Note that dropping higher-order terms is legitimate under the assumption we are dealing with small perturbations and changes, and simply determines the validity of our results (i.e., they are valid "to first order"). Multiplying the last equation by $\mathbf{u}_i^T$ on the left, we get:

$$\mathbf{u}_i^T \mathbf{A}\Delta\mathbf{v}_i + \mathbf{u}_i^T \Delta\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{u}_i^T \Delta\mathbf{v}_i + \Delta\lambda_i \mathbf{u}_i^T \mathbf{v}_i \tag{D.53}$$

But two of these terms cancel, as per our definition in Eq. (D.50), so we are left with:

$$\mathbf{u}_i^T \Delta\mathbf{A}\mathbf{v}_i = \Delta\lambda_i \mathbf{u}_i^T \mathbf{v}_i \tag{D.54}$$

Taking the absolute value of both sides and solving for $|\Delta\lambda_i|$, we have:

$$|\Delta\lambda_i| = \frac{|\mathbf{u}_i^T \Delta\mathbf{A}\mathbf{v}_i|}{|\mathbf{u}_i^T \mathbf{v}_i|} \tag{D.55}$$

We realize that we can apply the Cauchy–Schwarz inequality to the numerator:

$$|\mathbf{u}_i^T \Delta\mathbf{A}\mathbf{v}_i| \le ||\mathbf{u}_i||\, ||\Delta\mathbf{A}||\, ||\mathbf{v}_i|| \tag{D.56}$$

This is very similar to the fifth property in Eq. (D.22), but here we're faced with an absolute value of a number on the left-hand side, not the norm of a matrix. We can now take the eigenvectors to be normalized such that $||\mathbf{u}_i|| = ||\mathbf{v}_i|| = 1$ (as is commonly done in standard libraries and we will also do in section 4.4 below).

This means that we have managed to produce an error bound on $|\Delta\lambda_i|$, as desired:

$$|\Delta\lambda_i| \le \frac{1}{|\mathbf{u}_i^T \mathbf{v}_i|}\, ||\Delta\mathbf{A}|| \tag{D.57}$$

But this is fully analogous to what we had found for the perturbations in the case of the linear system of equations. The coefficient in front of $||\Delta\mathbf{A}||$ determines whether or not a small perturbation gets amplified in a specific case. Thus, we are led to introduce a new *condition number for simple eigenvalues*, as promised:

$$\kappa_{ev}^{\lambda_i}(\mathbf{A}) = \frac{1}{|\mathbf{u}_i^T \mathbf{v}_i|} \tag{D.58}$$

where the subscript is there to remind us that this is a condition number for a specific problem: for the evaluation of eigenvalues. The superscript keeps track of which specific eigenvalue's sensitivity we are referring to. To calculate the condition number for a given eigenvalue you first have to calculate the product of the corresponding left- and right-eigenvectors. We examine below the expected magnitude of this new condition number.

Observe that we chose to study an *absolute* error bound, that is, a bound on $|\Delta\lambda_i|$ instead of one on $|\Delta\lambda_i|/|\lambda_i|$: this is reasonable, since an eigenvalue is zero if you're dealing with non-invertible matrices.[4] Perhaps you can now see why when discussing the examples of this section we focused on absolute changes in the eigenvalues.

[4] You can see this for yourself: the determinant of a matrix is equal to the product of its eigenvalues; when one of these is 0, the determinant is 0, so the matrix is non-invertible.

D.1.5.6 Examples

We saw above, while discussing Examples 8 and 9, that the linear-system condition number $\kappa(\mathbf{A})$ was not able to tell us how sensitive a specific eigenvalue is to small perturbations in the elements of matrix $\mathbf{A}$. We now turn to a discussion of the same examples, this time employing $\kappa_{ev}^{\lambda_i}(\mathbf{A})$, which was designed to quantify the sensitivity in the problem at hand. In order to keep things manageable, we will only focus on one eigenvalue for each example:

- Example 8: $\kappa_{ev}^{\lambda_3}(\mathbf{A}) \approx 2.82$
- Example 9: $\kappa_{ev}^{\lambda_2}(\mathbf{A}) \approx 37.11$
- Example 3: $\kappa_{ev}^{\lambda_1}(\mathbf{A}) = 1$
- Example 1: $\kappa_{ev}^{\lambda_1}(\mathbf{A}) \approx 1.46$

Examples 8 and 9 are similar and can therefore be discussed together: we find that the eigenvalue condition number is larger than 1, whether by a few times (Example 8) or by many times (Example 9).[5] As Eq. (D.57) clearly shows, $\kappa_{ev}^{\lambda_i}(\mathbf{A})$ tells us what to multiply the perturbation $\|\Delta\mathbf{A}\|$ by in order to produce the absolute change in the eigenvalue. Of course, Eq. (D.57) only gives us an upper bound, so it is possible that in specific cases the actual error is much smaller. For the cases of Examples 8 and 9, specifically, our trial-and-error approach above gave answers that are pretty similar to those we now find using the eigenvalue condition number $\kappa_{ev}^{\lambda_i}(\mathbf{A})$. Crucially, the latter is an *a priori* estimate which doesn't necessitate actual experimentation.

Turning now to Example 3: we find that the eigenvalue condition number is 1, namely that a small perturbation is *not* amplified for this case. We now realize that this result is a specific instance of a more general pattern: the matrix in Example 3 is symmetric. For symmetric matrices, as you can see from Eq. (D.51), the right eigenvectors are identical to the left eigenvectors. Thus, for normalized eigenvectors, such that $\|\mathbf{u}_i\| = \|\mathbf{v}_i\| = 1$, we find that $\kappa_{ev}^{\lambda_i}(\mathbf{A}) = 1$ always. This means that *for real symmetric matrices the eigenvalue problem is always well-conditioned*.[6]

Finally, our result for Example 1 shows us that a matrix for which the solution to the linear equation problem may be tremendously ill-conditioned, does not have to be ill-conditioned when it comes to the evaluation of eigenvalues. In other words, we need different condition numbers for different problems. As it so happens, for this specific example the condition number for the other eigenvalue also has the same value: $\kappa_{ev}^{\lambda_0}(\mathbf{A}) \approx 1.46$, showing that this eigenvalue is not more ill-conditioned than the other one: this is consistent with our experimental finding.

To summarize, an eigenvalue condition number that is close to 1 corresponds to an eigenvalue that is well-conditioned and an eigenvalue condition number that is much larger than 1 is ill-conditioned; as usual, the demarcation between the two cases is somewhat arbitrary.

[5] For Example 9 the denominator, $|\mathbf{u}_i^T\mathbf{v}_i|$, was very small, implying that the corresponding left and right eigenvectors are almost orthogonal.

[6] Incidentally, we now see that this condition number, too, is bounded below by unity.

D.1.6 Sensitivity of Eigenvectors

Having studied the sensitivity of a linear system solution and of eigenvalue evaluations to small perturbations, the obvious next step is to do the same for the eigenvectors. You might be forgiven for thinking that this problem has already been solved (didn't we just produce a new condition number for eigenvalues in the previous section?), but things are not that simple. To reiterate, we are interested in probing the sensitivity of the eigenvalue problem:

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{D.59}$$

to small perturbations. This time, we focus on the effect of the perturbations on the evaluation of the (right) eigenvectors, $\mathbf{v}_i$. We begin with a few examples of matrices and some explicit perturbations added in "by hand" and only then turn to a semi-formal derivation.

The pattern should be clear by now: we start with a mistaken assumption of how to quantify the sensitivity, then we find counter-examples, after this we proceed to provide a correct expression determining the sensitivity dependence, and at the end verify that our new concept applies to the earlier examples.

D.1.6.1 Back to Example 3

We first return to our middle-of-the-road case, Example 3, Eq. (D.15). We saw above that if we add 0.01 to the bottom left element then we get an impact on the eigenvalues that is (in absolute terms) half the size of the perturbation. We also saw that the eigenvalue condition number $\kappa_{ev}^{\lambda_1}(\mathbf{A}) = 1$, this being consistent with our experimental finding that the eigenvalues for this problem are not sensitive to small perturbations.

We now explicitly check the sensitivity of an eigenvector. Before the perturbation:

$$\mathbf{v}_1^T = \begin{pmatrix} -0.70710678 & 0.70710678 \end{pmatrix} \tag{D.60}$$

After the perturbation is applied, the corresponding eigenvector becomes:

$$(\mathbf{v}_1 + \Delta\mathbf{v}_1)^T = \begin{pmatrix} -0.70534562 & 0.70886357 \end{pmatrix} \tag{D.61}$$

The eigenvector components have changed in the third digit after the decimal point: this is a change that is smaller than the perturbation itself. Nothing unexpected going on here: the present example is well-conditioned as far as any linear algebra problem is concerned.

D.1.6.2 Back to Example 9

We turn to the matrix in Eq. (D.41), which had an eigenvalue $\lambda_2 = 2$ which changed to $\lambda_2 + \Delta\lambda_2 \approx 2.18206$ after we added 0.005 to the bottom left element. This was consistent with the magnitude of the eigenvalue condition number, $\kappa_{ev}^{\lambda_2}(\mathbf{A}) \approx 37.11$.

Let's look at the corresponding eigenvector. Before the perturbation, we have:

$$\mathbf{v}_2^T = \begin{pmatrix} 0.88888889 & -0.44444444 & 0.11111111 & 0 \end{pmatrix} \tag{D.62}$$

After the perturbation is applied, the corresponding eigenvector becomes:

$$(\mathbf{v}_2 + \Delta\mathbf{v}_2)^T = \begin{pmatrix} 0.90713923 & -0.41228175 & 0.0843057 & 0.00383712 \end{pmatrix} \tag{D.63}$$

The change in the eigenvector components seems to be a few times larger than the perturbation we applied. It appears that, in this case, the eigenvectors are not as sensitive to small perturbations as the eigenvalues were.

D.1.6.3 Example 10

Our finding on Example 9 is already starting to cast doubts on the appropriateness of using the eigenvalue condition number $\kappa_{ev}^{\lambda_i}(\mathbf{A})$ to quantify the sensitivity of eigenvectors to small perturbations. We will now examine a new example which will leave no doubt:

$$\mathbf{A} = \begin{pmatrix} 1.01 & 0.01 \\ 0 & 0.99 \end{pmatrix} \tag{D.64}$$

Its eigenvalues can be read off the diagonal:

$$\lambda_0 = 1.01, \ \lambda_1 = 0.99 \tag{D.65}$$

We now apply a not-so-small perturbation of adding 0.005 to the top right element:

$$\lambda_0 + \Delta\lambda_0 = 1.01, \ \lambda_1 + \Delta\lambda_1 = 0.99 \tag{D.66}$$

In other words, both eigenvalues remain unchanged. This is consistent with the eigenvalue condition number $\kappa_{ev}^{\lambda_i}(\mathbf{A})$ which comes out to be approximately 1.11803 for both eigenvalues. As far as the eigen*values* are concerned, this problem is perfectly well-conditioned.

We now examine one of the eigenvectors explicitly. Before the perturbation we have:

$$\mathbf{v}_1^T = \begin{pmatrix} -0.4472136 & 0.89442719 \end{pmatrix} \tag{D.67}$$

After the perturbation is applied, the corresponding eigenvector becomes:

$$(\mathbf{v}_1 + \Delta\mathbf{v}_1)^T = \begin{pmatrix} -0.6 & 0.8 \end{pmatrix} \tag{D.68}$$

We notice a dramatic impact on the eigenvector components, of more than an order of magnitude larger than the perturbation we applied (in absolute terms). In other words, the eigen*vector* problem is ill-conditioned, despite the fact that the corresponding eigenvalue problem was well-conditioned. It is obvious that something new is at play here, which is not captured by our earlier condition number. We will explain what's going on below, after we derive a formula on the perturbed eigenvector.

D.1.6.4 Example 11

It's starting to look like the eigenvalue problem condition number is not a good measure of the sensitivity of eigenvectors to small perturbations. We will now see an example where the same conclusion holds, but is arrived at in the opposite direction: the eigenvalue condition number is huge, but the eigenvectors are not too sensitive. In some ways (but not all), this situation is similar to what we encountered when discussing Example 9 above.

Take the following 3 × 3 matrix:

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 4.001 \end{pmatrix} \tag{D.69}$$

Its eigenvalues can, again, be read off the diagonal:

$$\lambda_0 = 4.001, \ \lambda_1 = 4, \ \lambda_2 = 1 \tag{D.70}$$

If we now add 0.005 to the bottom left element, then the new eigenvalues are:

$$\lambda_0 + \Delta\lambda_0 \approx 4.12933336, \ \lambda_1 + \Delta\lambda_1 \approx 3.87111014, \ \lambda_2 + \Delta\lambda_2 \approx 1.0005565 \tag{D.71}$$

The first and second eigenvalues are considerably impacted by our perturbation. This is consistent with the eigenvalue condition numbers for our matrix:

$$\kappa_{ev}^{\lambda_0}(\mathbf{A}) \approx 6009.19059687, \ \kappa_{ev}^{\lambda_1}(\mathbf{A}) \approx 6009.25224596, \ \kappa_{ev}^{\lambda_2}(\mathbf{A}) \approx 1.2069722023 \tag{D.72}$$

We see that we could have predicted *a priori* that the first and second eigenvalues are sensitive to perturbations, since the corresponding condition numbers are huge. Even so, it's worth noting that the change in these eigenvalues is less than 30 times larger than the perturbation itself: this is to be compared with Example 9, where the eigenvalue condition number was not even 40 but the eigenvalue changed by more than 30 times the magnitude of the perturbation.

We now examine one of the eigenvectors explicitly. Before the perturbation we have:

$$\mathbf{v}_0^T = \begin{pmatrix} 0.554687392 & 0.832058814 & 0.000166412 \end{pmatrix} \tag{D.73}$$

After the perturbation is applied, the corresponding eigenvector becomes:

$$(\mathbf{v}_0 + \Delta\mathbf{v}_0)^T = \begin{pmatrix} 0.552982 & 0.832915 & 0.0215447 \end{pmatrix} \tag{D.74}$$

The eigenvector seems to be largely oblivious to the perturbation: despite the large eigenvalue condition number and the sensitivity of the corresponding eigenvalue, the change in the eigenvector components is at most a few times larger than the perturbation we applied. As advertised, we have found another case where the eigenvectors are not as sensitive to small perturbations as the eigenvalues were. As we will see below, this is a somewhat special case, which will serve as a reminder of how simple arguments can lead us astray.

D.1.6.5 Derivation

We've seen that some matrices have eigenvectors that are sensitive to small perturbations, whereas others do not. We tried to use the same condition number as for the evaluation of the eigenvalues but were disappointed. We will now carry out an informal derivation that will point us toward a quantitative measure of conditioning *eigenvectors*. (Spoiler alert: it is not $\kappa_{ev}^{\lambda_i}(\mathbf{A})$.) This quantitative measure of the sensitivity of our problem to small changes in the input matrix elements will provide guidance regarding how to approach problems *a priori*.

To refresh your memory, we note that the problem we are solving is:

$$\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{D.75}$$

For simplicity, we are assuming we are dealing with distinct eigenvalues/linearly independent eigenvectors. Perturbing leads to Eq. (D.49):

$$(\mathbf{A} + \Delta\mathbf{A})(\mathbf{v}_i + \Delta\mathbf{v}_i) = (\lambda_i + \Delta\lambda_i)(\mathbf{v}_i + \Delta\mathbf{v}_i) \tag{D.76}$$

As you may recall, after a few manipulations we arrived at Eq. (D.52):

$$\mathbf{A}\Delta\mathbf{v}_i + \Delta\mathbf{A}\mathbf{v}_i = \lambda_i\Delta\mathbf{v}_i + \Delta\lambda_i\mathbf{v}_i \tag{D.77}$$

We now expand the perturbation in the eigenvector in terms of the other eigenvectors:

$$\Delta\mathbf{v}_i = \sum_{j\neq i} t_{ji}\mathbf{v}_j \tag{D.78}$$

where the coefficients t_{ji} are to be determined. We are employing here the linear independence of the eigenvectors. Note that this sum does not include a $j = i$ term: you can assume that if there existed a t_{ii} it could have been absorbed into our definition of what a perturbation for this eigenvector is.[7]

If we plug the last equation into the penultimate equation, we find:

$$\sum_{j\neq i}(\lambda_j - \lambda_i)t_{ji}\mathbf{v}_j + \Delta\mathbf{A}\mathbf{v}_i = \Delta\lambda_i\mathbf{v}_i \tag{D.79}$$

The λ_j arose because we also used our defining relation Eq. (D.75). We will now multiply our equation by the left eigenvector $\mathbf{u}_k^T$, keeping in mind that left and right eigenvectors for distinct eigenvalues are orthogonal to each other, $\mathbf{u}_k^T\mathbf{v}_i = 0$ for $k \neq i$. We find:

$$(\lambda_k - \lambda_i)t_{ki}\mathbf{u}_k^T\mathbf{v}_k + \mathbf{u}_k^T\Delta\mathbf{A}\mathbf{v}_i = 0 \tag{D.80}$$

We can solve this relation for t_{ki} and then plug the result into Eq. (D.78), thereby getting:

$$\Delta\mathbf{v}_i = \sum_{j\neq i}\frac{\mathbf{u}_j^T\Delta\mathbf{A}\mathbf{v}_i}{(\lambda_i - \lambda_j)\mathbf{u}_j^T\mathbf{v}_j}\mathbf{v}_j \tag{D.81}$$

This is our main result. Let's unpack it a little bit. First, we notice that (unlike our earlier results in condition-number derivations), the right-hand side contains a sum: the perturbation in one eigenvector contains contributions that are proportional to each of the other eigenvectors. Second, we observe that the numerator contains the perturbation in the input matrix, $\Delta\mathbf{A}$.[8] Third, and most significant, we see that our denominator contains two distinct contributions: (a) a $\mathbf{u}_j^T\mathbf{v}_j$ term, which is the same thing that appeared in our definition of

[7] $\mathbf{v}_i$ becomes $\mathbf{v}_i + \Delta\mathbf{v}_i$, so any term in $\Delta\mathbf{v}_i$ that is proportional to $\mathbf{v}_i$ simply adjusts the coefficient in front of $\mathbf{v}_i$.

[8] If we wanted to take the norm, the $\mathbf{u}_j^T$ and $\mathbf{v}_i$ in the numerator would disappear, since $||\mathbf{u}_i|| = ||\mathbf{v}_i|| = 1$.

the condition number for a simple eigenvalue in Eq. (D.58), and (b) a $\lambda_i - \lambda_j$ term, which encapsulates the separation between the eigenvalue λ_i and all other eigenvalues.

Thus, we have found that a perturbation in the input matrix will get amplified if, first, $\mathbf{u}_j^T \mathbf{v}_j$ is small or, second, *if any two eigenvalues are close*! In other words, the problem of evaluating eigen*vectors* may be ill-conditioned either because the eigen*value* problem for any of the eigenvalues is ill-conditioned, or because two (or more) eigenvalues are closely spaced. Intuitively, we already know that if two eigenvalues coincide then we cannot uniquely determine the eigenvectors, so our result can be thought of as a generalization of this to the case where two eigenvalues are close to each other.

D.1.6.6 Examples

We will now go over our earlier examples once again, this time armed with our main result in Eq. (D.81). Since this equation was derived specifically to quantify the effect of a perturbation on an eigenvector, we expect it to do much better than the eigenvalue condition number $\kappa_{ev}^{\lambda_i}(\mathbf{A})$. Let's examine Examples 3, 9, 10, and 11 in turn.

Example 3 is the easiest to discuss: we recall that the matrix was symmetric, implying that the eigenvalue evaluation was well-conditioned. That removes one possible source of eigenvector-evaluation issues. Since the eigenvalues were, in this case, also well removed from one another, there is no reason to expect any conditioning problems here, a conclusion that is consistent with our earlier experimental investigation.

Example 9, on the other hand, did exhibit sensitivity to perturbations. This, obviously, did not result from the eigenvalues being close to each other (as, in this case, they are well separated). We see that the sensitivity came from the ill-conditioning of some of its eigenvalues, in other words from the $\mathbf{u}_j^T \mathbf{v}_j$ in the denominator. From our earlier discussion we know that the left and right eigenvectors (corresponding to the same eigenvalue) can be near-orthogonal, thereby making the eigenvalue evaluation sensitive to small perturbations. Of course, Eq. (D.81) involves more than just one denominator, so in this case the overall effect is less dramatic than it was for the evaluation of the corresponding eigenvalue.

Example 10 exhibits precisely the opposite behavior: while the eigenvalues are well-conditioned, they happen to be very closely spaced. This is a quite extreme illustration of the impact closely spaced eigenvalues can have on the evaluation of the eigenvectors.[9] This is a smoking-gun case where the previous condition numbers ($\kappa(\mathbf{A})$ and $\kappa_{ev}^{\lambda_i}(\mathbf{A})$) do not raise any red flags, yet the eigenvector problem is extremely sensitive to minor variations in the input matrix.

Example 11 is trickier than all preceding examples and serves as a warning not to rush to judgement. Observe that two of the eigenvalues are quite ill-conditioned and the same two eigenvalues also happen to be very close to each other in value. Thus, naively applying our earlier rule of thumb (ill-conditioned eigenvalues or closely spaced eigenvalues can complicate the evaluation of an eigenvector) we would have expected this example to be a worst-case scenario, since both conditions are satisfied. However, our experimental investigation showed limited sensitivity of our eigenvector to external perturbations.

[9] Observe that before Example 10 all our examples were chosen to have well-separated eigenvalues.

Let's make matters even more confusing (for the moment) by examining the eigenvector $\mathbf{v}_2$ for the same matrix (Example 11): as you can see from Eq. (D.81), for this eigenvector we would get contributions from both the $\mathbf{u}_0^T\mathbf{v}_0$ and $\mathbf{u}_1^T\mathbf{v}_1$ denominators which are known to be very small. The relevant eigenvalue separations would be $\lambda_2 - \lambda_0$ and $\lambda_2 - \lambda_1$, which are nearly identical. We would thus expect a very large impact on $\mathbf{v}_2$ from a minor perturbation on the input matrix. Let's see what happens in practice. Before the perturbation we have:

$$\mathbf{v}_2^T = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \tag{D.82}$$

After the perturbation is applied, the corresponding eigenvector becomes:

$$(\mathbf{v}_2 + \Delta\mathbf{v}_2)^T = \begin{pmatrix} 0.999\,994\,75 & 0.002\,777\,87 & -0.001\,666\,41 \end{pmatrix} \tag{D.83}$$

This is a small effect which, at first sight, is inconsistent with the result we derived. What's going on? Remember that when we used our new trick in Eq. (D.78) we were employing the linear independence of the eigenvectors: this is OK when they are orthogonal. However, in the present case two of our eigenvectors are almost linearly dependent (stop reading and go check that $\mathbf{v}_0$ and $\mathbf{v}_1$ are nearly identical). Thus, $\Delta\mathbf{v}_2$ is made up of a large number times $\mathbf{v}_0$ plus a large number times $\mathbf{v}_1$: since the coefficients are nearly identical in magnitude and of opposite sign, the result is a small perturbation vector, which implies insensitivity to an external perturbation.[10]

D.2 Theory of interacting spin-half particles

We now turn to the prototypical eigenvalue problem in modern physics, the *time-independent Schrödinger equation*:

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{D.84}$$

where $\hat{H}$ is the Hamiltonian operator, $|\psi\rangle$ is a state vector (called a ket by Dirac) in a Hilbert space, and E is the energy. In a course on quantum mechanics you likely heard the terms "eigenstates" and "eigenenergies". At the time, it was pointed out to you that Eq. (D.84) is an eigenvalue equation: it contains the same state vector on the left-hand side as on the right-hand side and is therefore formally of the form of Eq. (4.124).

In practice, when solving Eq. (D.84) for a given physical problem, we typically get a differential equation, as we will see in chapter 8. In the present section, we limit ourselves to the case of one or more particles with spin-half, where there are no orbital degrees of freedom.[11] As we will see below, our problem maps onto a (reasonably) straightforward matrix form, where you don't have to worry about non-matrix features; that is, the $\hat{H}$ doesn't have

[10] If you understand our argument on $\Delta\mathbf{v}_2$ you should be able to produce an analogous argument for the case of $\Delta\mathbf{v}_0$, as you are asked to do in problem D.5, thereby explaining our earlier experimental finding.

[11] You should be able to generalize this to the case of spin-one once you've understood our approach.

a kinetic energy in it. Thus, the problem of spin-half particles becomes a direct application of the eigenproblem machinery we built earlier.

If you haven't taken a course on quantum mechanics (QM) yet, you can skim through the theory sections below and focus on the Python implementation that comes at the end. There you will notice that most of the code consists of setting up the problem, since the hard part (numerically evaluating eigenvalues of a matrix) has already been solved in previous sections. If you have taken a course on quantum mechanics, keep in mind that our approach here is slightly different from that given in a typical textbook on the subject. Generally, the calculations become too messy to carry out using paper-and-pencil; in contradistinction to this, we'll show below that once you've set up the appropriate framework, increasing the number of particles merely increases the dimensionality of your problem. Thus, we are able to attack head-on the setting of two or three spin-half interacting particles; it should be easy for the reader to generalize to the case of four (or more) particles with a minimum of complications (as you will find out when solving problems 4.50 and 4.51).

D.2.1 One Particle

We start with some basic concepts from the study of spin in quantum mechanics. In order to keep things manageable, we will assume you've encountered this material before, so the purpose of this section and of the following one is mainly to establish the notation.

D.2.1.1 Hilbert Space

As you may recall, spin may be thought of as an intrinsic angular momentum. In quantum mechanics you typically denote the spin angular momentum operator by $\hat{\mathbf{S}}$, this being made up of the three Cartesian components $\hat{S}_x$, $\hat{S}_y$, and $\hat{S}_z$. The two most important relations in this context are the ones for the square of the spin operator and for its z component:

$$\begin{aligned} \hat{S}^2|sm_s\rangle &= \hbar^2 s(s+1)|sm_s\rangle \\ \hat{S}_z|sm_s\rangle &= \hbar m_s|sm_s\rangle \end{aligned} \tag{D.85}$$

where $|sm_s\rangle$ is our notation for the spin eigenstates. Note that on the left-hand sides we have operators (in upper case) and on the right-hand sides we have eigenvalues (in lower case). More specifically, the latter are the spin s (in our case, $s = 1/2$) and the azimuthal quantum number m_s (which in our case can be either $m_s = +1/2$ or $m_s = -1/2$). The fact that there are only two possibilities for the value of m_s is a conclusion drawn from experimental measurements with particles like electrons, neutrons, and protons. As a result, we call this a two-state system.[12]

Since, as we just observed, we know that $s = 1/2$, we see that the first equation in Eq. (D.85) will always have $\hbar^2 3/4$ on the right-hand side. This means that the two eigenstates at play here are essentially labelled by the two possible values of the azimuthal quantum number, m_s. Thus, they are $|s = 1/2, m_s = +1/2\rangle$ and $|s = 1/2, m_s = -1/2\rangle$.

[12] Such two-state systems are heavily emphasized in *The Feynman Lectures on Physics, Vol. 3* (Ref. 3).

Instead of carrying around the general notation $|sm_s\rangle$, we can simply label our two eigenstates using the fact that the z-projection of the spin is either ↑ (spin-up) or ↓ (spin-down). Thus, we use the notation $|\zeta_\uparrow\rangle$ and $|\zeta_\downarrow\rangle$ (spin parallel and spin antiparallel to the z axis, respectively). The Greek letter here was picked in order to bring to mind the last letter of the English alphabet (so it should be easy to remember that $|\zeta_\uparrow\rangle$ is an eigenstate of the $\hat{S}_z$ operator). Using this notation, the second relation in Eq. (D.85) becomes:

$$\hat{S}_z |\zeta_\uparrow\rangle = \frac{\hbar}{2} |\zeta_\uparrow\rangle, \qquad \hat{S}_z |\zeta_\downarrow\rangle = -\frac{\hbar}{2} |\zeta_\downarrow\rangle \tag{D.86}$$

A further point: we can refer to either $|\zeta_\uparrow\rangle$ or $|\zeta_\downarrow\rangle$ using the notation $|\zeta_i\rangle$: here i is an index that covers all the possibilities, namely $i =\uparrow, \downarrow$.

An arbitrary spin state can be expressed as a linear superposition of our two basis states:

$$|\psi\rangle = \psi_\uparrow |\zeta_\uparrow\rangle + \psi_\downarrow |\zeta_\downarrow\rangle = \sum_{i=\uparrow,\downarrow} \psi_i |\zeta_i\rangle \tag{D.87}$$

where $\psi_\uparrow$ and $\psi_\downarrow$ are complex numbers. In the second equality we employed our new notation with the i index. It should be easy to see that:

$$\langle\zeta_\uparrow|\psi\rangle = \psi_\uparrow, \qquad \langle\zeta_\downarrow|\psi\rangle = \psi_\downarrow \tag{D.88}$$

where we used the fact that our two basis states are orthonormal.

D.2.1.2 Matrix Representation

We now turn to the matrix representation of spin-half particles. This is very convenient, since it involves 2 × 2 matrices for spin operators. You may have even heard that a 2 × 1 column vector (which represents a spin state vector) is called a *spinor*. However, this is putting the cart before the horse. Instead of starting from a result, let's start at the start: *the physics in quantum mechanics is contained in the inner products, or the matrix elements*, as opposed to the operators or the state vectors alone.

Let's try to form all the possible matrix elements, sandwiching $\hat{S}_z$ between the basis states: this leads to $\langle\zeta_i|\hat{S}_z|\zeta_j\rangle$, where we are employing our new notation, $|\zeta_i\rangle$ and $|\zeta_j\rangle$, where i and j take on the values ↑ and ↓. In other words, the azimuthal quantum numbers on the left and on the right can each take on the values $\pm 1/2$. This means that there are four possibilites (i.e., four matrix elements) in total. It then becomes natural to collect them into a 2 × 2 matrix. At this point, we have to be a bit careful with our notation, since in this chapter we are denoting matrices and vectors using bold symbols. Thus, we will group together all the matrix elements and denote the resulting matrix with a bold symbol, $\mathbf{S}_z$.

It may help to think in terms of the notation we introduce in appendix C.2: if you think of $\langle\zeta_i|\hat{S}_z|\zeta_j\rangle$ as the matrix element $(\mathbf{S}_z)_{ij}$, then the matrix made up of all these elements would be $\left\{(\mathbf{S}_z)_{ij}\right\}$.[13] In all, we have:

$$\mathbf{S}_z = \begin{pmatrix} \langle\zeta_\uparrow|\hat{S}_z|\zeta_\uparrow\rangle & \langle\zeta_\uparrow|\hat{S}_z|\zeta_\downarrow\rangle \\ \langle\zeta_\downarrow|\hat{S}_z|\zeta_\uparrow\rangle & \langle\zeta_\downarrow|\hat{S}_z|\zeta_\downarrow\rangle \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{D.89}$$

where we used, once again, the fact that our two basis states are orthonormal. Note that there is no operator (i.e., there is no hat) on the left-hand side, since we are dealing with a matrix containing inner products/matrix elements: we are not dealing with an operator, but with the effect the operator has in a specific basis (the one made up of $|\zeta_\uparrow\rangle$ and $|\zeta_\downarrow\rangle$).

A standard derivation using the raising and lowering operators $\hat{S}_+$ and $\hat{S}_-$ (not introduced here, but familiar to you, we hope) leads to corresponding results for the matrix elements of the $\hat{S}_x$ and $\hat{S}_y$ operators. These are:

$$\mathbf{S}_x = \begin{pmatrix} \langle\zeta_\uparrow|\hat{S}_x|\zeta_\uparrow\rangle & \langle\zeta_\uparrow|\hat{S}_x|\zeta_\downarrow\rangle \\ \langle\zeta_\downarrow|\hat{S}_x|\zeta_\uparrow\rangle & \langle\zeta_\downarrow|\hat{S}_x|\zeta_\downarrow\rangle \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{D.90}$$

and:

$$\mathbf{S}_y = \begin{pmatrix} \langle\zeta_\uparrow|\hat{S}_y|\zeta_\uparrow\rangle & \langle\zeta_\uparrow|\hat{S}_y|\zeta_\downarrow\rangle \\ \langle\zeta_\downarrow|\hat{S}_y|\zeta_\uparrow\rangle & \langle\zeta_\downarrow|\hat{S}_y|\zeta_\downarrow\rangle \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{D.91}$$

where you should note that we're always using $|\zeta_\uparrow\rangle$ and $|\zeta_\downarrow\rangle$ (i.e., the eigenstates of the z-component operator, $\hat{S}_z$) to sandwich the operator each time.

We're ready at this point to introduce the *Pauli spin matrices*; these are simply the above spin matrices with the prefactors removed:

$$\boldsymbol{\sigma}_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \boldsymbol{\sigma}_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{D.92}$$

You should probably memorize these matrices.

We now turn to the representation of the state vectors. Let's first approach this as a linear algebra problem: we need to diagonalize the 2 × 2 matrix $\mathbf{S}_z$. As you already know well after studying the present chapter, that implies finding the eigenvalues (which turn out to be $\pm\hbar/2$) and the eigenvectors, which we calculate to be:

$$\boldsymbol{\zeta}_\uparrow = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{\zeta}_\downarrow = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{D.93}$$

[13] We're being a bit sloppy here: the i and j in $\langle\zeta_i|\hat{S}_z|\zeta_j\rangle$ take on the values ↑ and ↓, whereas the i and j in $(\mathbf{S}_z)_{ij}$, being indices for a 2×2 matrix, take on the values 0 and 1. The correspondence between one meaning and the other is always implied.

Since the matrix we were diagonalizing was 2 × 2, it comes as no surprise that the eigenvectors are 2 × 1 column vectors. You should test your understanding by finding the eigenvectors corresponding to, say, $\mathbf{S}_y$.

As already noted, we are no longer dealing with operators and state vectors (no hats and no kets), but with matrices and column vectors, respectively. As a result, relations that in the Hilbert-space language involved actions on kets, now turn into relations involving matrices. For example, the equations from Eq. (D.86) translate to:

$$\mathbf{S}_z\boldsymbol{\zeta}_\uparrow = \frac{\hbar}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix}1\\ 0\end{pmatrix} = \frac{\hbar}{2}\boldsymbol{\zeta}_\uparrow \tag{D.94}$$

and:

$$\mathbf{S}_z\boldsymbol{\zeta}_\downarrow = \frac{\hbar}{2}\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\begin{pmatrix}0\\ 1\end{pmatrix} = -\frac{\hbar}{2}\begin{pmatrix}0\\ 1\end{pmatrix} = -\frac{\hbar}{2}\boldsymbol{\zeta}_\downarrow \tag{D.95}$$

where we carried out the matrix-vector multiplication in both cases.[14] As you can imagine, if we are ever faced with an expression like, say, $\mathbf{S}_z\mathbf{S}_z$, we have to carry out matrix–matrix multiplication.

We can combine our two eigenvectors to produce the matrix representation of an arbitrary spin state (just like in Eq. (D.87), $\psi_\uparrow$ and $\psi_\downarrow$ are complex numbers):

$$\boldsymbol{\psi} = \psi_\uparrow\boldsymbol{\zeta}_\uparrow + \psi_\downarrow\boldsymbol{\zeta}_\downarrow = \begin{pmatrix}\psi_\uparrow & \psi_\downarrow\end{pmatrix}^T \tag{D.96}$$

where we used Eq. (D.93) to get to the second equality. In our final result, we see the 2 × 1 column vector (called a *spinor* above) emerge organically. Recall that when we went from the operator $\hat{S}_z$ to the matrix $\mathbf{S}_z$, we simply grouped together all possibilities for $\langle\zeta_i|\hat{S}_z|\zeta_j\rangle$. This motivates a new way of looking at the 2 × 1 column vectors (which represent the state vectors): simply group together all the possibilities for $\langle\zeta_i|\psi\rangle$. Since i can take on two values, you end up with:

$$\boldsymbol{\psi} = \begin{pmatrix}\langle\zeta_\uparrow|\psi\rangle\\ \langle\zeta_\downarrow|\psi\rangle\end{pmatrix} = \begin{pmatrix}\psi_\uparrow\\ \psi_\downarrow\end{pmatrix} \tag{D.97}$$

The first equality is analogous to our definition in Eq. (D.89), while the second equality follows from Eq. (D.88).

To summarize, operators are represented by 2 × 2 matrices:

$$\begin{array}{c|cc} & |\zeta_\uparrow\rangle & |\zeta_\downarrow\rangle \\ \hline \langle\zeta_\uparrow| & \square & \square \\ \langle\zeta_\downarrow| & \square & \square \end{array} \tag{D.98}$$

where we have also labelled (outside the matrix) how we get each row and column: by sandwiching the operator with the bra on the left and the ket on the right, each time. (Examples of operator representations are given in Eqs. (D.89), (D.90), and (D.91).) Similarly, a state vector is represented by a 2 × 1 column vector. An example of a state vector representation is given in Eq. (D.97).

[14] There's nothing mysterious going on here: we're simply reiterating the fact that $\boldsymbol{\zeta}_\uparrow$ and $\boldsymbol{\zeta}_\downarrow$ are eigenvectors of the matrix $\mathbf{S}_z$ with the specificied eigenvalues.

D.2.1.3 Hamiltonian

We recall that what we were actually interested in all along was solving the Schrödinger equation, Eq. (D.84), $\hat{H}|\psi\rangle = E|\psi\rangle$. We are immediately faced with two questions: first, which Hamiltonian $\hat{H}$ should we use? Second, how do we translate this equation involving operators and kets into an equation involving matrices? Let's start from the second question. We take the Schrödinger equation, Eq. (D.84), and act with $\langle\zeta_\uparrow|$ on the left. We then introduce a resolution of the identity, $\hat{\mathcal{I}} = |\zeta_\uparrow\rangle\langle\zeta_\uparrow| + |\zeta_\downarrow\rangle\langle\zeta_\downarrow|$, to find:

$$\langle\zeta_\uparrow|\hat{H}|\zeta_\uparrow\rangle\langle\zeta_\uparrow|\psi\rangle + \langle\zeta_\uparrow|\hat{H}|\zeta_\downarrow\rangle\langle\zeta_\downarrow|\psi\rangle = E\langle\zeta_\uparrow|\psi\rangle \tag{D.99}$$

You may now repeat this exercise, this time acting with $\langle\zeta_\downarrow|$ on the left. At this point you are free to combine your two equations into matrix form, giving:

$$\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi} \tag{D.100}$$

Note how neatly this encompasses our earlier results. If you're still trying to understand what **H** looks like, just remember our general result about the representation of any operator, Eq. (D.98) – in other words, $\mathbf{H} = \{\langle\zeta_i|\hat{H}|\zeta_j\rangle\}$. You may have already encountered the rewriting of Eq. (D.84) in the form of Eq. (D.100): we hope that it now feels legitimate as a way of going from operators and kets to matrices and column vectors.[15]

Of course, even if we know what **H** looks like, we still need to answer our earlier question, namely deciding on which Hamiltonian $\hat{H}$ we should use. To do that, assume that our spin-half particle is interacting with an external magnetic field **B**. Associated with the spin angular momentum $\hat{\mathbf{S}}$ there will be a spin magnetic moment operator, $\hat{\boldsymbol{\mu}}$: since this operator needs to be a combination of the spin operators and the identity (and we know it has to be a vector operator), it follows that $\hat{\boldsymbol{\mu}}$ is proportional to $\hat{\mathbf{S}}$. It is customary to write the proportionality between the two operators as follows:

$$\hat{\boldsymbol{\mu}} = g\left(\frac{q}{2m}\right)\hat{\mathbf{S}} \tag{D.101}$$

where q is the electric charge of the particle and m is its mass. The proportionality constant is known as the *g-factor*: its value is roughly -2 for electrons and $+5.6$ for protons.

Since we have no orbital degrees of freedom, the Hamiltonian is simply made up of the interaction energy which, by analogy to the classical-physics case, is:

$$\hat{H} = -\hat{\boldsymbol{\mu}} \cdot \mathbf{B} = -\frac{gqB}{2m}\hat{S}_z \tag{D.102}$$

In the second step we took our z axis as pointing in the direction of the magnetic field. Combining our earlier point about how to go from operators to matrices, Eq. (D.98), with the explicit matrix representation of $\hat{S}_z$, Eq. (D.89), we find:

$$\mathbf{H} = -\frac{gqB\hbar}{4m}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{D.103}$$

[15] The same argument can help you see why, say, Eq. (D.94) is equivalent to Eq. (D.86).

This Hamiltonian is so simple that the matrix form of the Schrödinger equation (which we know from Eq. (D.100) is $\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi}$) can be solved analytically. Note that there are two reasons why this is such an easy problem: first, it's because of the small dimensionality (2×2) and second, it is because our Hamiltoninan matrix $\mathbf{H}$ is *diagonal*. We will soon discuss other cases, where the matrices are both larger and non-diagonal.

D.2.2 Two Particles

We went over things in great detail in the previous subsection (which dealt with a single spin-half particle) because we wanted to establish the notation and the concepts. We are now about to do something similar for the problem of two spin-half particles. As before, we start from the formulation involving operators and kets, then turn to the matrix representation, ending with the Hamiltonian for our problem.

D.2.2.1 Hilbert Space(s)

When dealing with a single spin-half particle, we saw that its state vectors were labelled $|sm_s\rangle$: this contained the possibilities $|s = 1/2, m_s = +1/2\rangle$ and $|s = 1/2, m_s = -1/2\rangle$. We then proceeded to use the alternative notation $|\zeta_\uparrow\rangle$ and $|\zeta_\downarrow\rangle$ for these two states, known collectively as $|\zeta_i\rangle$.

We now have to be especially careful about our notation: since we're dealing with two particles, we need some way of labelling them. Let's call them particle I and particle II, using Roman numerals: this will pay off later on, when we implement things in Python (we'll have enough indices to worry about). Thus, the first particle involves a vector space which is spanned by the two kets $\left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle$ and $\left|\zeta_\downarrow^{(\mathrm{I})}\right\rangle$: observe that we have employed superscripts and parentheses (within which we place the Roman numeral) to keep track of which particle we're talking about. If we wish to refer to either of the two states, we can use the notation $\left|\zeta_i^{(\mathrm{I})}\right\rangle$. Make sure you understand what's going on here: i is either ↑ or ↓, keeping track of the (eigenvalue of the) z-projection of the spin for the first particle. Similarly, the Hilbert space of the second particle is spanned by the two kets $\left|\zeta_\uparrow^{(\mathrm{II})}\right\rangle$ and $\left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle$, which can be compactly expressed as $\left|\zeta_j^{(\mathrm{II})}\right\rangle$, where we used a new index, j, since in general the second particle can be either ↑ or ↓, regardless of what the projection of the first particle spin was.

We now wish to start from these single-particle vector spaces and generalize to a two-particle space. To do this, we employ the concept of a *tensor product* (denoted by ⊗): this allows us to express the product between state vectors belonging to different Hilbert spaces (e.g., $\left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle$ and $\left|\zeta_\uparrow^{(\mathrm{II})}\right\rangle$). In short, the two-particle Hilbert space is a four-dimensional complex vector space which is spanned by the vectors:

$$\left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_\uparrow^{(\mathrm{II})}\right\rangle \equiv \left|\zeta_{\uparrow\uparrow}\right\rangle, \quad \left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle \equiv \left|\zeta_{\uparrow\downarrow}\right\rangle, \quad \left|\zeta_\downarrow^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_\uparrow^{(\mathrm{II})}\right\rangle \equiv \left|\zeta_{\downarrow\uparrow}\right\rangle, \quad \left|\zeta_\downarrow^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle \equiv \left|\zeta_{\downarrow\downarrow}\right\rangle \tag{D.104}$$

where we also took the opportunity to define a compact notation for the two-particle state vectors: in an expression like $\left|\zeta_{\uparrow\downarrow}\right\rangle$ it is implicit that the first arrow refers to particle I and the second arrow to particle II. Note that $\left|\zeta_{\uparrow\downarrow}\right\rangle$ doesn't have a superscript in parentheses, because it is *not* a one-particle state vector, but is made up of two one-particle state vectors.

We can compactly refer to any one of these four basis states using the notation $|\zeta_a\rangle$, where a is an index that covers all the possibilities, namely: $a =\uparrow\uparrow, \uparrow\downarrow, \downarrow\uparrow, \downarrow\downarrow$. Keep in mind that in the previous section we were using the notation $|\zeta_i\rangle$ (or $|\zeta_j\rangle$) to refer to single-particle states. In the present section one-particle states will always come with a parenthesized Roman numeral keeping track of which particle we're referring to. Here we are introducing the similar-yet-distinct notation $|\zeta_a\rangle$ (or perhaps also $|\zeta_b\rangle$) to keep track of two-particle states. We'll consistently pick letters from the start of the alphabet to denote two-particle indices. Thus, Eq. (D.104) can be compactly given in the following form:

$$\left|\zeta_i^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_j^{(\mathrm{II})}\right\rangle \equiv |\zeta_a\rangle \tag{D.105}$$

The left-hand side involves the one-particle states and the tensor product (and i, j indices), while the right-hand side has a two-particle state (and an a index). Depending on your learning style, you may wish to think of a as the ordered pair (i, j), which we would have called a tuple in Python (of course, this is an ordered pair of arrows, not numbers).

In terms of the Hilbert spaces themselves, we started from the space of the first particle ($\mathscr{H}^{(\mathrm{I})}$) and the space of the second particle ($\mathscr{H}^{(\mathrm{II})}$) and have produced the larger, two-particle Hilbert space $\mathscr{H}^{(\mathrm{I})} \otimes \mathscr{H}^{(\mathrm{II})}$. The four state vectors $\left|\zeta_i^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_j^{(\mathrm{II})}\right\rangle$ form the *product basis* of this Hilbert space $\mathscr{H}^{(\mathrm{I})} \otimes \mathscr{H}^{(\mathrm{II})}$.

Let us turn to the operators in the two-particle Hilbert space, focusing on the z-projection operator for concreteness. We already know the one-particle operator $\hat{S}_z^{(\mathrm{I})}$ which acts on the vector space of particle I and, similarly, the one-particle operator $\hat{S}_z^{(\mathrm{II})}$ which acts on the vector space of particle II. Each of these operators measures the z-projection of the spin for the respective particle. What we wish to do is come up with operators for the composite system. We do this by, again, employing the tensor product. For example:

$$\hat{S}_{\mathrm{I}z} = \hat{S}_z^{(\mathrm{I})} \otimes \hat{\mathcal{I}}^{(\mathrm{II})} \tag{D.106}$$

On the left-hand side we are introducing a new entity, $\hat{S}_{\mathrm{I}z}$, which is appropriate for the two-particle Hilbert space: note that it doesn't have a superscript in parentheses, because it is *not* a one-particle operator. Instead, it is made up of two one-particle operators, each of which knows how to act on its respective one-particle space. It should be easy to see why we have taken the tensor product with the identity operator $\hat{\mathcal{I}}$: the two-particle operator $\hat{S}_{\mathrm{I}z}$ measures the z component of the spin for particle I, so it does nothing to any particle-II ket it encounters. In complete analogy to this, the two-particle operator that measures the z component of the spin for particle II is:

$$\hat{S}_{\mathrm{II}z} = \hat{\mathcal{I}}^{(\mathrm{I})} \otimes \hat{S}_z^{(\mathrm{II})} \tag{D.107}$$

where we do nothing (i.e., have an identity) for particle I and take the tensor product with the appropriate operator for particle II.

Perhaps an example will help solidify your understanding of what's going on. Let's see what happens when a two-particle operator acts on a given two-particle state vector:

$$\begin{aligned}\hat{S}_{\mathrm{II}z}\left|\zeta_{\uparrow\downarrow}\right\rangle &= \left(\hat{I}^{(\mathrm{I})}\otimes\hat{S}_z^{(\mathrm{II})}\right)\left(\left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle\otimes\left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle\right) = \left(\hat{I}^{(\mathrm{I})}\left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle\right)\otimes\left(\hat{S}_z^{(\mathrm{II})}\left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle\right)\\ &= \left|\zeta_\uparrow^{(\mathrm{I})}\right\rangle\otimes\left(-\frac{\hbar}{2}\left|\zeta_\downarrow^{(\mathrm{II})}\right\rangle\right) = -\frac{\hbar}{2}\left|\zeta_{\uparrow\downarrow}\right\rangle\end{aligned} \tag{D.108}$$

In the first equality we used Eq. (D.107) and Eq. (D.104) for the operator and state vector, respectively, writing each in terms of a tensor product. In the second equality we acted with each operator on the appropriate state vector: the parenthesized superscripts help us keep track of which particle is which. In the third equality we applied our knowledge about the effect one-particle operators have on one-particle state vectors, specifically Eq. (D.86). In the fourth equality we re-identified the compact way of expressing the two-particle state vector (in the opposite direction from what was done on the first equality). The final result is not surprising, since we already knew that particle II was ↓, but it's nice to see that the different Hilbert spaces and operators work together to give the right answer (e.g., observe that the final answer is proportional to a two-particle state vector, as it should).

Finally, an arbitrary spin state can be expressed as a linear superposition:

$$|\psi\rangle = \psi_{\uparrow\uparrow}\left|\zeta_{\uparrow\uparrow}\right\rangle + \psi_{\uparrow\downarrow}\left|\zeta_{\uparrow\downarrow}\right\rangle + \psi_{\downarrow\uparrow}\left|\zeta_{\downarrow\uparrow}\right\rangle + \psi_{\downarrow\downarrow}\left|\zeta_{\downarrow\downarrow}\right\rangle = \sum_{a=\uparrow\uparrow,\uparrow\downarrow,\downarrow\uparrow,\downarrow\downarrow}\psi_a\left|\zeta_a\right\rangle \tag{D.109}$$

where $\psi_{\uparrow\uparrow}$, and so on, are complex numbers. Similarly to what we did in the one-particle case, in the second equality we show the superposition expressed as a sum. It's easy to see that as we get an increasing number of basis states, it is this second formulation that becomes more manageable (you just have to keep track of the possible values of the index).

At this stage, a QM textbook typically passes over into the *coupled representation*, where the total spin of the two-particle system is of interest. For the general problem of adding two angular momenta, this is where Clebsch–Gordan coefficients come into the picture. For the specific case of two spin-half particles, this leads to one spin-singlet state and a spin-triplet (made up of three states). In contradistinction to this, here we are interested in the *uncoupled representation*, where we consider the two-particle system as being made up of two individual particles. Below, we will show you how to build up the matrix representation of a two-particle operator using the matrix representation of one-particle operators: this will give us a tool that is then trivial to generalize to larger numbers of particles.

D.2.2.2 Matrix Representation

Turning to the matrix representation of two spin-half particles, you will not be surprised to hear that it involves 4×4 matrices for spin operators and 4×1 column vectors for the state vectors. We recall that the matrix representation of quantum mechanics translates to taking inner products, i.e., sandwiching operators between a bra and a ket. For the sake of concreteness, we will start our discussion from a given operator, $\hat{S}_{\mathrm{I}z}$, though eventually we will need to provide a prescription that gives the matrix representation of the other five relevant operators ($\hat{S}_{\text{I}x}$, $\hat{S}_{\text{I}y}$, as well as $\hat{S}_{\text{II}x}$, $\hat{S}_{\text{II}y}$, and $\hat{S}_{\text{II}z}$).

Let us try to form all the possible matrix elements, sandwiching $\hat{S}_{\text{I}z}$ between the basis states: using our latest notation from Eq. (D.105), this leads to $\langle\zeta_a|\,\hat{S}_{\text{I}z}\,|\zeta_b\rangle$.[16] Since each of a and b can take on four values, there are 16 possibilites (i.e., 16 matrix elements) in total. It then becomes natural to collect them into a 4×4 matrix, using a to keep track of rows and b for the columns. Once again, the notation of section C.2 may be helpful here: the entire matrix is generated by going over all the indices' values, namely $\{\langle\zeta_a|\,\hat{S}_{\text{I}z}\,|\zeta_b\rangle\}$.[17]

If you've never encountered this material before, you might want to pause at this point: a matrix has two indices (one for rows and one for columns), so in order to produce a matrix corresponding to the operator $\hat{S}_{\text{I}z}$ we needed to employ a single bra on the left and a single ket on the right. In other words, we moved away from keeping track of individual particles' quantum numbers and toward using two-particle states – this back-and-forth between one- and two-particle states is something we will return to below. Since we have four two-particle basis states, operators in a system made up of two spin-half particles are represented by 4×4 matrices as follows:

$$\begin{array}{ccccc} & \left|\zeta_{\uparrow\uparrow}\right\rangle & \left|\zeta_{\uparrow\downarrow}\right\rangle & \left|\zeta_{\downarrow\uparrow}\right\rangle & \left|\zeta_{\downarrow\downarrow}\right\rangle \\ \left\langle\zeta_{\uparrow\uparrow}\right| & \square & \square & \square & \square \\ \left\langle\zeta_{\uparrow\downarrow}\right| & \square & \square & \square & \square \\ \left\langle\zeta_{\downarrow\uparrow}\right| & \square & \square & \square & \square \\ \left\langle\zeta_{\downarrow\downarrow}\right| & \square & \square & \square & \square \end{array} \tag{D.110}$$

where we have also labelled (outside the matrix) how we get each row and column: by sandwiching the operator with the bra on the left and the ket on the right, each time.

We will now proceed to evaluate the matrix:

$$\mathbf{S}_{\text{I}z} = \left\{\langle\zeta_a|\,\hat{S}_{\text{I}z}\,|\zeta_b\rangle\right\} \tag{D.111}$$

in the most obvious way possible (we turn to a less obvious approach in the following subsection). In essence, we will make repeated use of our derivation in our earlier example, Eq. (D.108). In an identical fashion, one can show that, say:

$$\hat{S}_{\text{I}z}\left|\zeta_{\uparrow\uparrow}\right\rangle = \frac{\hbar}{2}\left|\zeta_{\uparrow\uparrow}\right\rangle \tag{D.112}$$

Here it is implied that we went through all the intermediate steps of using one-particle operators and one-particle states, and then zipped everything up again at the end. Acting with the bra $\left\langle\zeta_{\uparrow\uparrow}\right|$, we find:

$$\left\langle\zeta_{\uparrow\uparrow}\right|\hat{S}_{\text{I}z}\left|\zeta_{\uparrow\uparrow}\right\rangle = \frac{\hbar}{2} \tag{D.113}$$

where we assumed our basis vector is normalized. Since our basis states are orthonormal, had we used any other bra here, say $\left\langle\zeta_{\downarrow\downarrow}\right|$, we would have gotten 0. In other words, all other matrix elements on the same column are 0. Repeating this argument, we find:

[16] This a encapsulates two distinct azimuthal quantum numbers, one for each particle – the same holds for b.

[17] Remembering to distinguish between ket-indices with "arrow values" and matrix-indices with integer values.

$$\mathbf{S}_{Iz} = \frac{\hbar}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{D.114}$$

where, unsurprisingly, we find that the matrix is diagonal: the only way to get a non-zero entry is to use the same basis vector on the left and on the right. The only other feature of this matrix is that it measures the spin of the first particle (as it should): we get $+\hbar/2$ if the first particle is $\uparrow$ and $-\hbar/2$ if the first particle is $\downarrow$.

Without realizing it, we've evaluated the eigenvalues of the matrix (they're on the diagonal for a triangular/diagonal matrix). It's also a short step away to find the eigenvectors, now that we know the eigenvalues. For this 4×4 matrix, we find four eigenvectors, each of which is a 4×1 column vector:

$$\boldsymbol{\zeta}_{\uparrow\uparrow} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\zeta}_{\uparrow\downarrow} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, \quad \boldsymbol{\zeta}_{\downarrow\uparrow} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \boldsymbol{\zeta}_{\downarrow\downarrow} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \tag{D.115}$$

where we used the obvious notation for each eigenvector, by analogy to Eq. (D.93). As a result of this, an arbitrary state vector is represented by a 4×1 column vector:

$$\boldsymbol{\psi} = \psi_{\uparrow\uparrow}\boldsymbol{\zeta}_{\uparrow\uparrow} + \psi_{\uparrow\downarrow}\boldsymbol{\zeta}_{\uparrow\downarrow} + \psi_{\downarrow\uparrow}\boldsymbol{\zeta}_{\downarrow\uparrow} + \psi_{\downarrow\downarrow}\boldsymbol{\zeta}_{\downarrow\downarrow} = \begin{pmatrix} \psi_{\uparrow\uparrow} & \psi_{\uparrow\downarrow} & \psi_{\downarrow\uparrow} & \psi_{\downarrow\downarrow} \end{pmatrix}^T \tag{D.116}$$

D.2.2.3 Kronecker Product

Our approach, while good enough to get us going for the case of $\mathbf{S}_{Iz}$, has obvious limitations: for each new operator we need to evaluate 16 matrix elements. The example above was diagonal, so this task was considerably easier, but that won't always be the case. Similarly, if we were faced with a larger problem this way of doing things would quickly become prohibitive.[18] Thus, in the present section we will introduce a technique that can straightforwardly handle off-diagonalness and bigger matrices.[19]

Qualitatively, the main trick we will employ in this subsection is to focus on one-particle states and operators, in contradistinction to the previous subsection where we used two-particle basis states. Here we are in the fortunate situation of knowing what the answer should be for at least one case (that of $\mathbf{S}_{Iz}$), so we will be able to check if we got things right. Our starting point will be the same, namely Eq. (D.111), but soon thereafter things

[18] For example, three spin-half particles correspond to an 8×8 matrix, namely 64 matrix elements in total.

[19] Of course, applying it to two particles is overkill – "using a chain saw to trim your fingernails".

will start to take a different turn:

$$
\begin{aligned}
\mathbf{S}_{\mathrm{I}z} &= \left\{\langle\zeta_a|\,\hat{S}_{\mathrm{I}z}\,|\zeta_b\rangle\right\} = \left\{\left(\left\langle\zeta_i^{(\mathrm{I})}\right| \otimes \left\langle\zeta_j^{(\mathrm{II})}\right|\right)\left(\hat{S}_z^{(\mathrm{I})} \otimes \hat{\mathcal{I}}^{(\mathrm{II})}\right)\left(\left|\zeta_k^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_l^{(\mathrm{II})}\right\rangle\right)\right\} \\
&= \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{S}_z^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{\mathcal{I}}^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\} = \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{S}_z^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\right\} \otimes \left\{\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{\mathcal{I}}^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\} = \mathbf{S}_z \otimes \mathcal{I}
\end{aligned}
\tag{D.117}
$$

In the second equality we used the defining Eq. (D.105) and Eq. (D.106), which express our two-particle state vectors and operators in terms of corresponding one-particle entities. In the third equality we grouped together entities relating to each particle, separately: the tensor product has vanished, since we are now dealing only with matrix elements (i.e., complex numbers). We then notice that the four indices $ijkl$ appear in pairs: ik sandwiches one operator and jl sandwiches the other operator. This suggests that our 4×4 matrix (which is what the curly braces on the outside produce) is made up of 2×2 blocks. Thus, in the fourth equality we made the claim that we can go over all possible values of $ijkl$, two indices at a time (ik and jl), at the cost of having slightly changed the meaning of $\otimes$: in the second equality this was the tensor product, keeping state vectors and operators belonging to particles I and II separate: in the fourth equality, however, we are no longer dealing with state vectors or operators, but with 2×2 matrices.[20] In the fifth equality, we make this explicit: observe that there are no longer any particle labels, only the one-particle $\mathbf{S}_z$ matrix from Eq. (D.89), as well as a 2×2 identity matrix.

We now have to explain the meaning of this new $\otimes$ entity, which can combine matrices in this specific way. This is nothing other than the *Kronecker product*, which turns out to be not so new, of course, since it is merely a matrix version of the tensor product. Assume you're dealing with an $n \times n$ matrix $\mathbf{U}$ and a $p \times p$ matrix $\mathbf{V}$.[21] The most intuitive way of thinking of the Kronecker product $\mathbf{U} \otimes \mathbf{V}$ is as the $np \times np$ matrix that looks like this:

$$
\mathbf{W} = \mathbf{U} \otimes \mathbf{V} = \begin{pmatrix} U_{00}\mathbf{V} & U_{01}\mathbf{V} & \dots & U_{0,n-1}\mathbf{V} \\ U_{10}\mathbf{V} & U_{11}\mathbf{V} & \dots & U_{1,n-1}\mathbf{V} \\ \vdots & \vdots & \ddots & \vdots \\ U_{n-1,0}\mathbf{V} & U_{n-1,1}\mathbf{V} & \dots & U_{n-1,n-1}\mathbf{V} \end{pmatrix}
\tag{D.118}
$$

The presence of a $\mathbf{V}$ in each slot is to be interpreted as follows: to produce $\mathbf{U} \otimes \mathbf{V}$, take each element of $\mathbf{U}$, namely U_{ik}, and replace it by $U_{ik}\mathbf{V}$, which is a $p \times p$ matrix. (In total, there

[20] With that in mind, some authors use distinct notation for the Kronecker product, typically $\otimes_K$.

[21] We could also define the Kronecker product for non-square matrices, even for vectors. Incidentally, the Kronecker product is closely related (though not quite identical) to the outer product employed elsewhere in the book. As a simple example, you can run the following: `xs = np.arange(1,5); ys = np.arange(11,17); np.kron(xs,ys); np.outer(xs,ys); np.ravel(np.outer(xs,ys))`.

will be n^2 such $p \times p$ matrices.) Expanded out, this leads to the $np \times np$ matrix $\mathbf{W}$:

$$\begin{pmatrix}
U_{00}V_{00} & \dots & U_{00}V_{0,p-1} & U_{01}V_{00} & \dots & U_{01}V_{0,p-1} & \dots & U_{0,n-1}V_{0,p-1} \\
U_{00}V_{10} & \dots & U_{00}V_{1,p-1} & U_{01}V_{10} & \dots & U_{01}V_{1,p-1} & \dots & U_{0,n-1}V_{1,p-1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
U_{00}V_{p-1,0} & \dots & U_{00}V_{p-1,p-1} & U_{01}V_{p-1,0} & \dots & U_{01}V_{p-1,p-1} & \dots & U_{0,n-1}V_{p-1,p-1} \\
U_{10}V_{00} & \dots & U_{10}V_{0,p-1} & U_{11}V_{00} & \dots & U_{11}V_{0,p-1} & \dots & U_{1,n-1}V_{0,p-1} \\
U_{10}V_{10} & \dots & U_{10}V_{1,p-1} & U_{11}V_{10} & \dots & U_{11}V_{1,p-1} & \dots & U_{1,n-1}V_{1,p-1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
U_{10}V_{p-1,0} & \dots & U_{10}V_{p-1,p-1} & U_{11}V_{p-1,0} & \dots & U_{11}V_{p-1,p-1} & \dots & U_{1,n-1}V_{p-1,p-1} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
U_{n-1,0}V_{p-1,0} & \dots & U_{n-1,0}V_{p-1,p-1} & U_{n-1,1}V_{p-1,0} & \dots & U_{n-1,1}V_{p-1,p-1} & \dots & U_{n-1,n-1}V_{p-1,p-1}
\end{pmatrix} \tag{D.119}$$

You should spend some time making sure you understand that the last two matrices are showing exactly the same thing.

While this intuitive understanding is important, what we would eventually like to do is to implement the Kronecker product programmatically: in order to do that, we need an equation connecting the indices of the $\mathbf{U}$ and $\mathbf{V}$ matrix elements, on the one hand, with the indices of the $\mathbf{W}$ matrix, on the other. This is:

$$\begin{gathered}
W_{ab} = (\mathbf{U} \otimes \mathbf{V})_{ab} = U_{ik}V_{jl} \\
\text{where } a = pi + j, \quad b = pk + l
\end{gathered} \tag{D.120}$$

The original four indices take on the values:

$$i = 0, 1, \dots, n-1, \quad k = 0, 1, \dots, n-1, \quad j = 0, 1, \dots, p-1, \quad l = 0, 1, \dots, p-1 \tag{D.121}$$

As a result, the new indices take on the values:

$$a = 0, 1, \dots, np-1, \quad b = 0, 1, \dots, np-1 \tag{D.122}$$

You should spend some time thinking about our new equation: you will benefit from applying it by hand to one or two simple cases (say, the Kronecker product of a 2 × 2 matrix with a 3 × 3 matrix). Incidentally, the idea that the Kronecker product is the same thing as regular matrix multiplication is obligingly self-refuting (to borrow a memorable turn of phrase): a Kronecker product of a 2 × 2 matrix with 3 × 3 matrix leads to a 6 × 6 matrix, whereas the regular matrix multiplication of a 2 × 2 matrix with 3 × 3 matrix is not even possible, since the dimensions don't match.

Having introduced and explained the Kronecker product, we may now return to our derivation in Eq. (D.117). The third equality contains the product of the two matrix elements $(\mathbf{S}_z)_{ik}$ and $(\boldsymbol{\mathcal{I}})_{jl}$: armed with Eq. (D.120), we may now confidently identify $(\mathbf{S}_z)_{ik}(\boldsymbol{\mathcal{I}})_{jl}$ using the Kronecker product, namely $(\mathbf{S}_z \otimes \boldsymbol{\mathcal{I}})_{ab}$, thereby justifying our earlier claim. This means we can now continue our derivation at the point where Eq. (D.117) had left it off:

$$\mathbf{S}_{\mathrm{I}z} = \mathbf{S}_z \otimes \mathcal{I} \tag{D.123}$$

This is the matrix version of the operator relation Eq. (D.106), no longer involving any particle labels. It is perhaps not too late to try to avoid possible confusion: when comparing to other texts, you should keep in mind that $\mathbf{S}_z$ here has *nothing* to do with a total spin operator for the two-particle system: it is a 2 × 2 matrix corresponding to a one-particle operator. We can now plug in $\mathbf{S}_z$ from Eq. (D.89), as well as a 2 × 2 identity to find:

$$\mathbf{S}_{\mathrm{I}z} = \mathbf{S}_z \otimes \mathcal{I} = \frac{\hbar}{2}\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{D.124}$$

where, crucially, the last step applied our definition of the Kronecker product, Eq. (D.120).

To summarize what we've been doing in this subsection: in Eq. (D.117) we went from two-particle operators and state vectors to one-particle entities. Then, we identified one-particle matrix elements: collecting those together, we ended up with one-particle-space-dimensioned (i.e., small) matrices and took their Kronecker product. That led to a two-particle-space-dimensioned (i.e., larger) matrix. This $\mathbf{S}_{\mathrm{I}z}$ matrix turned out to be identical to what we had found in Eq. (D.114), but it is important to realize that here we didn't even have to think once about the effect of a specific spin operator on a specific ket: instead, we merely took the Kronecker product of a Pauli matrix with an identity matrix. In other words, we simply carried out a mathematical operation between two matrices.

D.2.2.4 Matrix Representation Continued

In short, we have encountered two ways of building up the 4 × 4 matrices we need to describe the system of two spin-half particles: first, using two-particle operators and state vectors explicitly, to produce matrix elements for all 16 cases. (Of course, to do that, we need to employ tensor products of one-particle operators and state vectors.) Second, we showed that the same answer could be arrived at via a Kronecker product between two 2×2 matrices. The second approach can now be used to find the answer for more complicated cases. For example, the matrix $\mathbf{S}_{\mathrm{II}x}$ can be computed as follows:

$$\begin{aligned}\mathbf{S}_{\mathrm{II}x} &= \left\{\langle\zeta_a|\,\hat{S}_{\mathrm{II}x}\,|\zeta_b\rangle\right\} = \left\{\left(\left\langle\zeta_i^{(\mathrm{I})}\right|\otimes\left\langle\zeta_j^{(\mathrm{II})}\right|\right)\left(\hat{\mathcal{I}}^{(\mathrm{I})}\otimes\hat{S}_x^{(\mathrm{II})}\right)\left(\left|\zeta_k^{(\mathrm{I})}\right\rangle\otimes\left|\zeta_l^{(\mathrm{II})}\right\rangle\right)\right\} \\ &= \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{\mathcal{I}}^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{S}_x^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\} = \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{\mathcal{I}}^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\right\}\otimes\left\{\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{S}_x^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\} \\ &= \mathcal{I}\otimes\mathbf{S}_x = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\otimes\frac{\hbar}{2}\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \frac{\hbar}{2}\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}\end{aligned} \tag{D.125}$$

The first several steps proceed by direct analogy to the steps in Eq. (D.117). The only differences are that (a) our operator is $\hat{S}_{\text{II}x} = \hat{\mathcal{I}}^{(\text{I})} \otimes \hat{S}_x^{(\text{II})}$ by analogy to Eq. (D.107), and (b) now we need to use the x one-particle matrix, from Eq. (D.90). If you wish to, you could check this result by doing things the hard way, namely by using two-particle states and explicitly evaluating all the matrix elements. (You'll find the same answer, unsurprisingly.)

Crucially, all the steps of the above process can be automated, so computing other matrices (e.g., $\mathbf{S}_{\text{I}y}$) is just as straightforward: all we're doing is taking the Kronecker product of an identity and a Pauli spin matrix (possibly not in that order). This is a purely mathematical task, which doesn't need to get caught up in the details of different spin operators: once you've determined the (one-particle) Pauli spin matrices, Eq. (D.92), you can straightforwardly arrive at the matrix representation of any two-particle operator; as we'll see below, the same also holds for problems involving three, four, and so on particles.

D.2.2.5 Interacting Spins

If you're following along so far, it shouldn't be too hard to see how to handle more complicated operators. The most obvious candidate is:

$$\hat{\mathbf{S}}_{\text{I}} \cdot \hat{\mathbf{S}}_{\text{II}} = \hat{S}_{\text{I}x}\hat{S}_{\text{II}x} + \hat{S}_{\text{I}y}\hat{S}_{\text{II}y} + \hat{S}_{\text{I}z}\hat{S}_{\text{II}z} \tag{D.126}$$

Let's see how to handle this operator, focusing on $\hat{S}_{\text{I}z}\hat{S}_{\text{II}z}$ for the moment.

As before, there are two ways of going about this: either we focus on 4 × 4 matrices (for the two-particle system directly), or we use Kronecker products between 2 × 2 matrices (corresponding to one-particle matrix elements). Let's start with the former approach, which involves two-particle matrices like $\mathbf{S}_{\text{I}z}$ and $\mathbf{S}_{\text{II}z}$: we assume that you know how to produce these (again, either the hard way or the easy way). Here's what we do in order to express our matrix elements in terms of 4 × 4 matrices:

$$\langle\zeta_a|\,\hat{S}_{\text{I}z}\hat{S}_{\text{II}z}\,|\zeta_b\rangle = \sum_{c=\uparrow\uparrow,\uparrow\downarrow,\downarrow\uparrow,\downarrow\downarrow} \langle\zeta_a|\,\hat{S}_{\text{I}z}\,|\zeta_c\rangle\,\langle\zeta_c|\,\hat{S}_{\text{II}z}\,|\zeta_b\rangle = \sum_c (\mathbf{S}_{\text{I}z})_{ac}\,(\mathbf{S}_{\text{II}z})_{cb} = (\mathbf{S}_{\text{I}z}\mathbf{S}_{\text{II}z})_{ab} \tag{D.127}$$

In the first equality we introduced a resolution of the identity. In the second equality we expressed our matrix elements using the notation that employs bold symbols. In the third equality we realized that we were faced with nothing other than a *matrix multiplication*, namely $C_{ij} = \sum_k A_{ik}B_{kj}$. Comparing the left-hand side with our result in the third equality, we see that we can build up the entire matrix $\{\langle\zeta_a|\,\hat{S}_{\text{I}z}\hat{S}_{\text{II}z}\,|\zeta_b\rangle\}$ by giving a and b all possible values. The same argument can be repeated for the x and y components. All in all, we have shown that the answer is arrived at if you multiply together the relevant 4 × 4 matrices and then sum the results up:

$$\mathbf{S}_{\text{I}\cdot\text{II}} = \left\{\langle\zeta_a|\,\hat{\mathbf{S}}_{\text{I}} \cdot \hat{\mathbf{S}}_{\text{II}}\,|\zeta_b\rangle\right\} = \mathbf{S}_{\text{I}x}\mathbf{S}_{\text{II}x} + \mathbf{S}_{\text{I}y}\mathbf{S}_{\text{II}y} + \mathbf{S}_{\text{I}z}\mathbf{S}_{\text{II}z} \tag{D.128}$$

where we also introduced a new symbol, $\mathbf{S}_{\text{I}\cdot\text{II}}$, to denote the 4 × 4 matrix corresponding to the dot product between two spin operators. As advertised, this is a result involving two-particle matrices; it is the formula which we will implement in our Python code below.

We now turn to our second approach to the operator $\hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{II}z}$, this time employing one-particle operators and state vectors. Let's start with expressing the product of the two two-particle operators in terms of one-particle operators:

$$\hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{II}z} = \left(\hat{S}_z^{(\mathrm{I})}\otimes\hat{I}^{(\mathrm{II})}\right)\left(\hat{I}^{(\mathrm{I})}\otimes\hat{S}_z^{(\mathrm{II})}\right) = \left(\hat{S}_z^{(\mathrm{I})}\hat{I}^{(\mathrm{I})}\right)\otimes\left(\hat{I}^{(\mathrm{II})}\hat{S}_z^{(\mathrm{II})}\right) = \hat{S}_z^{(\mathrm{I})}\otimes\hat{S}_z^{(\mathrm{II})} \tag{D.129}$$

In the first equality we used Eq. (D.106) and Eq. (D.107). In the second equality we grouped together operators acting on particle I and those acting on II. In the third equality we removed the identities, since they don't change anything. We see that the product of the two two-particle operators reduces itself to a tensor product between one-particle operators.

Let's now evaluate the matrix made up by sandwiching this operator. This derivation will be very similar in spirit to that in Eq. (D.125):

$$\begin{aligned}\left\{\langle\zeta_a|\,\hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{II}z}\,|\zeta_b\rangle\right\} &= \left\{\left(\left\langle\zeta_i^{(\mathrm{I})}\right|\otimes\left\langle\zeta_j^{(\mathrm{II})}\right|\right)\left(\hat{S}_z^{(\mathrm{I})}\otimes\hat{S}_z^{(\mathrm{II})}\right)\left(\left|\zeta_k^{(\mathrm{I})}\right\rangle\otimes\left|\zeta_l^{(\mathrm{II})}\right\rangle\right)\right\}\\ &= \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{S}_z^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{S}_z^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\}\\ &= \left\{\left\langle\zeta_i^{(\mathrm{I})}\right|\hat{S}_z^{(\mathrm{I})}\left|\zeta_k^{(\mathrm{I})}\right\rangle\right\}\otimes\left\{\left\langle\zeta_j^{(\mathrm{II})}\right|\hat{S}_z^{(\mathrm{II})}\left|\zeta_l^{(\mathrm{II})}\right\rangle\right\} = \mathbf{S}_z\otimes\mathbf{S}_z\end{aligned} \tag{D.130}$$

In the first equality we expressed two-particle entities in terms of one-particle entities. The equalities after that closely follow the steps in Eq. (D.125): our result is a Kronecker product between (multiples of) two Pauli spin matrices. Obviously, analogous relations hold for the x and y components. All in all, we have shown that you can get the desired 4×4 matrix by taking Kronecker products of 2×2 matrices and summing the results up:

$$\mathbf{S}_{\mathrm{I}\cdot\mathrm{II}} = \left\{\langle\zeta_a|\,\hat{\mathbf{S}}_{\mathrm{I}}\cdot\hat{\mathbf{S}}_{\mathrm{II}}\,|\zeta_b\rangle\right\} = \mathbf{S}_x\otimes\mathbf{S}_x + \mathbf{S}_y\otimes\mathbf{S}_y + \mathbf{S}_z\otimes\mathbf{S}_z \tag{D.131}$$

where we used the same symbol as above, $\mathbf{S}_{\mathrm{I}\cdot\mathrm{II}}$, to denote the 4×4 matrix corresponding to the dot product between two spin operators. As advertised, this is a result involving (Kronecker products of) one-particle matrices; it is the formula you are asked to implement in problem D.6. It should come as no surprise that Eq. (D.131) is equivalent to Eq. (D.128).

D.2.2.6 Hamiltonian

We end our discussion of two spin-half particles with the Schrödinger equation:

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{D.132}$$

You should work through the derivation that led to Eq. (D.100): you will realize that all the steps are still valid, the only difference being that now instead of dealing with a one-body $|\zeta_i\rangle$ we are faced with a two-body $|\zeta_a\rangle$ (i.e., we are still sandwiching and introducing a resolution of the identity). Thus, we arrive at the matrix form of the Schrödinger equation:

$$\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi} \tag{D.133}$$

where this time the **H** matrix is 4 × 4 and the state $\boldsymbol{\psi}$ is a 4 × 1 column vector.

Just like we did for the single-particle case, we now have to consider which Hamiltonian $\hat{H}$ to use. This time around, we first assume that each of particles I and II is interacting with an external magnetic field **B**. Each spin angular momentum ($\hat{\mathbf{S}}_\text{I}$ and $\hat{\mathbf{S}}_\text{II}$) will be associated with a spin magnetic moment operator ($\hat{\boldsymbol{\mu}}_\text{I}$ and $\hat{\boldsymbol{\mu}}_\text{II}$, respectively). Thus, there will be a contribution to the total energy coming from the interaction of each magnetic moment with the magnetic field ($-\hat{\boldsymbol{\mu}}_\text{I} \cdot \mathbf{B}$ and $-\hat{\boldsymbol{\mu}}_\text{II} \cdot \mathbf{B}$). As before, we are free to take our z axis as pointing in the direction of the magnetic field. In addition to the interaction with the magnetic field, the two particles may also be interacting with each other: this is why the previous subsection on the operator $\hat{\mathbf{S}}_\text{I} \cdot \hat{\mathbf{S}}_\text{II}$ was titled "interacting spins".

Putting all the pieces together, the Hamiltonian for the case of two spins is:

$$\hat{H} = -\frac{g_\text{I} q_\text{I} B}{2m_\text{I}}\hat{S}_{\text{I}z} - \frac{g_\text{II} q_\text{II} B}{2m_\text{II}}\hat{S}_{\text{II}z} + \gamma\hat{\mathbf{S}}_\text{I} \cdot \hat{\mathbf{S}}_\text{II} = -\omega_\text{I}\hat{S}_{\text{I}z} - \omega_\text{II}\hat{S}_{\text{II}z} + \gamma(\hat{S}_{\text{I}x}\hat{S}_{\text{II}x} + \hat{S}_{\text{I}y}\hat{S}_{\text{II}y} + \hat{S}_{\text{I}z}\hat{S}_{\text{II}z}) \tag{D.134}$$

The first two terms correspond to the interaction with the magnetic field and the third term to the interaction between the two spins. The first equality employs the dimensionless g-factor, charge, and mass for each particle, as well as the coupling constant γ for the two-spin interaction (appropriately, $\hbar^2\gamma$ has units of energy). The second equality lumps all the coefficient terms together in the form of ω_I and ω_II. We also took the opportunity to expand the dot product $\hat{\mathbf{S}}_\text{I} \cdot \hat{\mathbf{S}}_\text{II}$ as per Eq. (D.126).

All that's left is for us to build up the matrix **H**. By now you should feel pretty confident about taking matrix elements: what we need is $\mathbf{H} = \{\langle\zeta_a|\,\hat{H}\,|\zeta_b\rangle\}$. We get:

$$\mathbf{H} = -\omega_\text{I}\mathbf{S}_{\text{I}z} - \omega_\text{II}\mathbf{S}_{\text{II}z} + \gamma\left(\mathbf{S}_{\text{I}x}\mathbf{S}_{\text{II}x} + \mathbf{S}_{\text{I}y}\mathbf{S}_{\text{II}y} + \mathbf{S}_{\text{I}z}\mathbf{S}_{\text{II}z}\right) \tag{D.135}$$

where we also made use of Eq. (D.128). (As you may recall, you are asked to use the alternative expression for $\mathbf{S}_{\text{I}\cdot\text{II}}$, Eq. (D.131), in problem D.6.) We have succeeded in expressing **H** only in terms of 4 × 4 matrices (which, in their turn, can be computed using the techniques introduced in earlier subsections). At this point, solving $\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi}$ is simple: this is a matrix eigenvalue problem, like the ones we spent so much time solving in this chapter.

Equation (D.135) is deceptively simple, so it may be beneficial to explicitly write it out in 4 × 4 form. Obviously, this is no longer practical once you start dealing with larger numbers of particles, but the intuition you build at this stage will serve you well later on. We assume that you have already produced the six matrices $\mathbf{S}_{\text{I}x}$, $\mathbf{S}_{\text{II}x}$, and so on. We plug these matrices in to Eq. (D.135) to find:

$$\begin{aligned}\mathbf{H} &= -\frac{\hbar}{2}\begin{pmatrix}\omega_{\rm I}+\omega_{\rm II} & 0 & 0 & 0\\ 0 & \omega_{\rm I}-\omega_{\rm II} & 0 & 0\\ 0 & 0 & -\omega_{\rm I}+\omega_{\rm II} & 0\\ 0 & 0 & 0 & -\omega_{\rm I}-\omega_{\rm II}\end{pmatrix} + \gamma\frac{\hbar^2}{4}\begin{pmatrix}1 & 0 & 0 & 0\\ 0 & -1 & 2 & 0\\ 0 & 2 & -1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}\\ &= -\frac{\hbar}{2}\begin{pmatrix}\omega_{\rm I}+\omega_{\rm II}-\gamma\frac{\hbar}{2} & 0 & 0 & 0\\ 0 & \omega_{\rm I}-\omega_{\rm II}+\gamma\frac{\hbar}{2} & -\gamma\hbar & 0\\ 0 & -\gamma\hbar & -\omega_{\rm I}+\omega_{\rm II}+\gamma\frac{\hbar}{2} & 0\\ 0 & 0 & 0 & -\omega_{\rm I}-\omega_{\rm II}-\gamma\frac{\hbar}{2}\end{pmatrix}\end{aligned} \tag{D.136}$$

In the first line we grouped together the magnetic-field-related terms, before combining everything together in the second line. Note that something like $\hbar\omega_{\rm I}$ has units of energy.

We immediately see that the terms relating to the magnetic field are diagonal in this basis: this means that if we had non-interacting spins ($\gamma = 0$) then the problem would have been trivial/diagonal/already solved. Notice that we said "in this basis": obviously, this is referring to the eigenkets $|\zeta_a\rangle$ which, you should recall, were built up from the two individual spins in the uncoupled representation. As a matter of fact, if you take $\gamma = 0$ together with $\omega_{\rm I} = \omega_{\rm II}$, you're basically adding together these two spin projections: as a result, solving the eigenproblem for this case leads to the same four eigenvalues you may be familiar with from the *coupled* representation, corresponding to a spin-singlet and a spin-triplet (adding spins together *is* what the coupled representation does).

On the other hand, the interacting-spins term is not diagonal in our basis. As it so happens, if you had been working in the coupled representation, then this second term would have been diagonal[22] (but then the magnetic-field terms wouldn't have been diagonal). Thus, our Hamiltonian (which has the spins interacting both with the magnetic field and with each other) is an example of a situation where the eigenvalues and eigenvectors are harder to pin down; thus, an eigenproblem solver is helpful here.[23] Reiterating: if you solve for the eigenvectors (in our representation, the uncoupled one) for the general case, you will *not* find the column vectors $\boldsymbol{\zeta}_{\uparrow\uparrow}, \boldsymbol{\zeta}_{\uparrow\downarrow}, \boldsymbol{\zeta}_{\downarrow\uparrow}$, and $\boldsymbol{\zeta}_{\downarrow\downarrow}$ from Eq. (D.115): these were arrived at as eigenvectors of $\mathbf{S}_{{\rm I}z}$ (and they are also eigenvectors of $\mathbf{S}_{{\rm II}z}$) but they are not eigenvectors of our more general Hamiltonian. That's not a big deal: this is precisely why we are studying the solution of general eigenvalue problems in this chapter.

D.2.3 Three Particles

We are now (at last) ready to reap the benefits of the theoretical machinery we established in the previous sections. We will study the problem of three spin-half particles, interacting with a magnetic field and with each other. The matrix formulation of this problem gives rise to 8×8 matrices (so 64 matrix elements per matrix): since there are three particles and three Cartesian coordinates, we need to deal with at least nine matrices, each of which is 8×8.

[22] To see this, note that $\hat{\mathbf{S}}_{\rm I}\cdot\hat{\mathbf{S}}_{\rm II}$ can be trivially expressed in terms of $\hat{\mathbf{S}}_{\rm I}+\hat{\mathbf{S}}_{\rm II}$.

[23] Of course, this is still a small matrix, so you could still do everything by hand if you wanted to.

In other words, this is not a task that is comfortably carried out using paper and pencil, which is why it doesn't appear in QM textbooks traditionally. When the three-angular-momenta problem does appear in textbooks, it is typically in the context of the coupled representation: in this case the Clebsch–Gordan coefficients are generalized to entities like the Wigner 6j-symbols or the Racah W-coefficients; these are messy to calculate, so they are typically found tabulated. As we'll soon see, both in this section and in the next one, our framework is essentially no more laborious to implement for the case of three particles (or even for more particles) than the two-particle problem was. This is because we are side-stepping the whole issue of angular momentum coupling (or recoupling): we will, once again, be working in the uncoupled representation. At the end of this process, we can use our eigensolver to diagonalize any matrix we desire: that automatically will allow us to evaluate quantities like the total spin of the system (or its z projections).

Inevitably, the present subsection will be much shorter than the preceding ones: if you've been paying attention so far then you will recognize that we're merely repeating the same arguments; if you haven't been reading attentively, we urge you to first study the one- and two-particle sections more carefully.

As usual, we start from the formulation in terms of operators and state vectors. Let's call our three particles: particle I, particle II, and particle III, again using Roman numerals. The first particle lives in a vector space spanned by the two kets $\left|\zeta_{\uparrow}^{(\mathrm{I})}\right\rangle$ and $\left|\zeta_{\downarrow}^{(\mathrm{I})}\right\rangle$, compactly denoted by $\left|\zeta_{i}^{(\mathrm{I})}\right\rangle$. Similarly, a second particle ket is $\left|\zeta_{j}^{(\mathrm{II})}\right\rangle$ and for the third particle we have $\left|\zeta_{k}^{(\mathrm{III})}\right\rangle$. We wish to start from these single-particle vector spaces and generalize to a three-particle space. As before, we accomplish this using the concept of a tensor product:

$$\left|\zeta_{i}^{(\mathrm{I})}\right\rangle \otimes \left|\zeta_{j}^{(\mathrm{II})}\right\rangle \otimes \left|\zeta_{k}^{(\mathrm{III})}\right\rangle \equiv \left|\zeta_{\mu}\right\rangle \tag{D.137}$$

where μ is an index that covers all the three spin-projection possibilities, namely:

$$\mu = \uparrow\uparrow\uparrow,\quad \uparrow\uparrow\downarrow,\quad \uparrow\downarrow\uparrow,\quad \uparrow\downarrow\downarrow,\quad \downarrow\uparrow\uparrow,\quad \downarrow\uparrow\downarrow,\quad \downarrow\downarrow\uparrow,\quad \downarrow\downarrow\downarrow \tag{D.138}$$

We'll consistently pick Greek letters starting from μ to denote three-particle indices. Here it is implicit that the first arrow refers to particle I, the second arrow to particle II, and the third arrow to particle III. Note that an entity like $\left|\zeta_{\mu}\right\rangle$ (or like $|\zeta_{\uparrow\downarrow\downarrow}\rangle$) doesn't have a superscript in parentheses, because it is *not* a one-particle state vector, but is made up of three one-particle state vectors. You may wish to think of μ as the ordered triple (i, j, k).

Turning to how the operators look in the three-particle Hilbert space, we'll again build up our three-particle operators using tensor products between one-particle operators. Here's an example for the z projection of the first particle:

$$\hat{S}_{\mathrm{I}z} = \hat{S}_{z}^{(\mathrm{I})} \otimes \hat{\mathcal{I}}^{(\mathrm{II})} \otimes \hat{\mathcal{I}}^{(\mathrm{III})} \tag{D.139}$$

This is almost identical to Eq. (D.106): the only difference is that we have an extra tensor product and an extra identity operator at the end. As usual, the left-hand side doesn't have a superscript in parentheses, because it is *not* a one-particle operator (it's a three-particle operator).[24] Here's another example, namely the three-particle operator that measures the y component of the spin for particle II:

$$\hat{S}_{\mathrm{II}y} = \hat{\mathcal{I}}^{(\mathrm{I})} \otimes \hat{S}_y^{(\mathrm{II})} \otimes \hat{\mathcal{I}}^{(\mathrm{III})} \tag{D.140}$$

where we do nothing (i.e., have an identity) for particles I and III and take the tensor product with the appropriate operator for particle II.

Next up, the matrix representation. As advertised, this involves 8 × 8 matrices for spin operators (and therefore 8 × 1 column vectors for the state vectors). As usual, we are sandwiching operators between a bra and a ket: this time around, we have eight basis kets, as per Eq. (D.137): this is determined by the possible values of the μ index, see Eq. (D.138). For concreteness, we discuss the operator $\hat{S}_{\mathrm{I}z}$. We are interested in evaluating the matrix:

$$\mathbf{S}_{\mathrm{I}z} = \left\{ \langle \zeta_\mu | \hat{S}_{\mathrm{I}z} | \zeta_\nu \rangle \right\} \tag{D.141}$$

where we have used two different (Greek) indices on the left and on the right (each of which can take on eight values). As you may recall, this calculation may be carried out the hard way, explicitly evaluating each of the 64 possibilities. Instead, we will now take advantage of having introduced the Kronecker product. Repeating the derivation in Eq. (D.117) you will find out that the answer is simply:

$$\mathbf{S}_{\mathrm{I}z} = \mathbf{S}_z \otimes \boldsymbol{\mathcal{I}} \otimes \boldsymbol{\mathcal{I}} \tag{D.142}$$

where, as usual, we are now dealing with matrices so there are no more particle labels to worry about. We don't want to go too fast at this point, so let's take a moment to appreciate what this simple-looking result means. This is the first time we've encountered two Kronecker products in a row, so you may be wondering how to interpret such an operation. Luckily, the Kronecker product is *associative*:

$$(\mathbf{T} \otimes \mathbf{U}) \otimes \mathbf{V} = \mathbf{T} \otimes (\mathbf{U} \otimes \mathbf{V}) \tag{D.143}$$

meaning you simply carry out one Kronecker product after the other and it doesn't matter which Kronecker product you carry out first. (Even so, the Kronecker product is *not* commutative, $\mathbf{U} \otimes \mathbf{V} \neq \mathbf{V} \otimes \mathbf{U}$.) Thus, in the present case you could, similarly to what we did in Eq. (D.117), identify $(\mathbf{S}_z)_{ik}(\boldsymbol{\mathcal{I}})_{jl}$ using the Kronecker product, namely $(\mathbf{S}_z \otimes \boldsymbol{\mathcal{I}})_{ab}$. Crucially, you could then treat this resulting expression as just another matrix element, which would have the same role as U_{ik} in Eq. (D.120). You should think about this a little, keeping in mind that in that defining relation $\mathbf{U}$ and $\mathbf{V}$ did *not* have to have the same dimensions. This is precisely the situation we are faced with right now: in forming $\mathbf{S}_z \otimes \boldsymbol{\mathcal{I}} \otimes \boldsymbol{\mathcal{I}}$ we can first take one Kronecker product, producing a 4×4 matrix, and then take the Kronecker product of that matrix with the last 2 × 2 identity matrix: $(\mathbf{S}_z \otimes \boldsymbol{\mathcal{I}}) \otimes \boldsymbol{\mathcal{I}}$. That's what gives you an

[24] Note that, while $\hat{S}_z^{(\mathrm{I})}$ is always a one-particle operator, we are using the same symbol, $\hat{S}_{\mathrm{I}z}$, to denote a two- or a three-particle operator – you can figure out which one we mean based on the context.

8×8 matrix at the end. Obviously, the same arguments apply for any other operator/matrix pair, so we can produce analogous results, for example:

$$\mathbf{S}_{\mathrm{II}y} = \mathcal{I} \otimes \mathbf{S}_y \otimes \mathcal{I} \tag{D.144}$$

As you will discover when you print out this matrix using a Python program, it is starting to have some non-trivial structure. Obviously, now that we are dealing with 8×8 matrices, it is becoming harder to calculate (or even write out) things by hand.

One last stop before we discuss the three-particle Hamiltonian. The interaction between spins I and II will look identical to Eq. (D.126):

$$\hat{\mathbf{S}}_{\mathrm{I}} \cdot \hat{\mathbf{S}}_{\mathrm{II}} = \hat{S}_{\mathrm{I}x}\hat{S}_{\mathrm{II}x} + \hat{S}_{\mathrm{I}y}\hat{S}_{\mathrm{II}y} + \hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{II}z} \tag{D.145}$$

The derivation in Eq. (D.127) carries over in its essence, therefore:

$$\mathbf{S}_{\mathrm{I\cdot II}} = \left\{\langle\zeta_\mu|\hat{\mathbf{S}}_{\mathrm{I}} \cdot \hat{\mathbf{S}}_{\mathrm{II}}|\zeta_\nu\rangle\right\} = \mathbf{S}_{\mathrm{I}x}\mathbf{S}_{\mathrm{II}x} + \mathbf{S}_{\mathrm{I}y}\mathbf{S}_{\mathrm{II}y} + \mathbf{S}_{\mathrm{I}z}\mathbf{S}_{\mathrm{II}z} \tag{D.146}$$

This is basically identical to Eq. (D.128), but you have to keep in mind that now $\mathbf{S}_{\mathrm{I\cdot II}}$ and all the other matrices have dimensions 8×8, not 4×4.[25]

Let us conclude this section with a discussion of the Hamiltonian. The matrix form of the Schrödinger equation is still the same:

$$\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi} \tag{D.147}$$

where the $\mathbf{H}$ matrix is 8×8 and the state $\boldsymbol{\psi}$ is an 8×1 column vector. As far as the Hamiltonian operator $\hat{H}$ is concerned, we will again have interactions with a magnetic field and between spins. Since we now have three particles, there are more pairs one could form: in addition to having particles I and II interacting, we could also have particles I and III, and particles II and III. Thus, Eq. (D.134) is generalized to:

$$\begin{aligned}\hat{H} &= -\frac{g_{\mathrm{I}}q_{\mathrm{I}}B}{2m_{\mathrm{I}}}\hat{S}_{\mathrm{I}z} - \frac{g_{\mathrm{II}}q_{\mathrm{II}}B}{2m_{\mathrm{II}}}\hat{S}_{\mathrm{II}z} - \frac{g_{\mathrm{III}}q_{\mathrm{III}}B}{2m_{\mathrm{III}}}\hat{S}_{\mathrm{III}z} + \gamma\left(\hat{\mathbf{S}}_{\mathrm{I}} \cdot \hat{\mathbf{S}}_{\mathrm{II}} + \hat{\mathbf{S}}_{\mathrm{I}} \cdot \hat{\mathbf{S}}_{\mathrm{III}} + \hat{\mathbf{S}}_{\mathrm{II}} \cdot \hat{\mathbf{S}}_{\mathrm{III}}\right)\\ &= -\omega_{\mathrm{I}}\hat{S}_{\mathrm{I}z} - \omega_{\mathrm{II}}\hat{S}_{\mathrm{II}z} - \omega_{\mathrm{III}}\hat{S}_{\mathrm{III}z} + \gamma\left(\hat{S}_{\mathrm{I}x}\hat{S}_{\mathrm{II}x} + \hat{S}_{\mathrm{I}y}\hat{S}_{\mathrm{II}y} + \hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{II}z}\right)\\ &\quad + \gamma\left(\hat{S}_{\mathrm{I}x}\hat{S}_{\mathrm{III}x} + \hat{S}_{\mathrm{I}y}\hat{S}_{\mathrm{III}y} + \hat{S}_{\mathrm{I}z}\hat{S}_{\mathrm{III}z}\right) + \gamma\left(\hat{S}_{\mathrm{II}x}\hat{S}_{\mathrm{III}x} + \hat{S}_{\mathrm{II}y}\hat{S}_{\mathrm{III}y} + \hat{S}_{\mathrm{II}z}\hat{S}_{\mathrm{III}z}\right)\end{aligned} \tag{D.148}$$

The first three terms correspond to the interaction with the magnetic field and the remaining terms to the interaction within the spin pairs (we assumed, for simplicity, the same coupling constant for all pairs). In the second equality we expanded the dot products as per Eq. (D.145). We're now ready to build up the matrix $\mathbf{H} = \left\{\langle\zeta_\mu|\hat{H}|\zeta_\nu\rangle\right\}$. We get:

$$\mathbf{H} = -\omega_{\mathrm{I}}\mathbf{S}_{\mathrm{I}z} - \omega_{\mathrm{II}}\mathbf{S}_{\mathrm{II}z} - \omega_{\mathrm{III}}\mathbf{S}_{\mathrm{III}z} + \gamma\left(\mathbf{S}_{\mathrm{I\cdot II}} + \mathbf{S}_{\mathrm{I\cdot III}} + \mathbf{S}_{\mathrm{II\cdot III}}\right) \tag{D.149}$$

[25] The result in Eq. (D.131) doesn't translate to the three-particle case quite so cleanly: it has to be generalized.

where this time, in order to be concise, we chose not to expand $\mathbf{S}_{\text{I-II}}$ and its cousins from Eq. (D.146) and the corresponding relations. As was to be expected, all of the matrices involved here are 8 × 8: once again, the magnetic-field-related contributions are diagonal and the spin-interacting parts are non-diagonal. Yet again, that's not a problem because we know how to solve $\mathbf{H}\boldsymbol{\psi} = E\boldsymbol{\psi}$ for the general case.

As you will find out when you solve problem 4.48, our approach can handle the simplification of $\gamma = 0$ (i.e., non-interacting spins) together with $\omega_{\text{I}} = \omega_{\text{II}} = \omega_{\text{III}}$ (i.e., several copies of the same type of particle). This problem amounts to a version of standard angular momentum addition, this time applied to the case of three spin-half particles. This involves a diagonal matrix, so it's trivial to find the eigenvalues (the only labor involved is that required to produce the Hamiltonian matrix). On the other hand, similarly to what we saw for the case of two spin-half particles, if you switch off the magnetic field you get a non-diagonal problem (which is diagonal in the coupled representation – not employed here). Obviously, our situation consists of both the magnetic field and the spins interacting with each other, so we'll have to code this up in as general a fashion as possible (e.g., you might want to introduce a different interaction term in the future).

References

1 Byron, F. W. Jr, and Fuller, R. W. 1992. *Mathematics of Classical and Quantum Physics*. Dover.

2 Davio, M. 1981. Kronecker Products and Shuffle Algebra. *IEEE Trans Comput*, C-30, 116.

3 Feynman, R. P., Leighton, R. B., and Sands, M. L. 2010. *The Feynman Lectures on Physics*, Vol. 3. New millenium edn. Addison-Wesley.

4 Kahan, W. 1966. Numerical Linear Algebra. *Can Math Bul*, 7, 757.

5 Shankar, R. 1994. *Principles of Quantum Mechanics*. Second edn. Plenum Press.

6 Stewart, G. W., and Sun, J. 1990. *Matrix Perturbation Theory*. Academic Press.

7 Wilkinson, J. H. 1965. *The Algebraic Eigenvalue Problem*. Oxford University Press.

Problems

D.1 For the following linear system:

$$(\mathbf{A}|\mathbf{b}) = \left(\begin{array}{cc|c} 0.8647 & 0.5766 & 0.2885 \\ 0.4322 & 0.2822 & 0.1442 \end{array}\right) \tag{D.150}$$

evaluate the determinant, norm, condition number of $\mathbf{A}$, as well as the effect hand-picked small perturbations in the matrix elements have on the solution vector.

D.2 Search the NumPy documentation for further functions that produce or manipulate arrays. Use what you've learned to produce an $n \times n$ matrix of the form:

$$\begin{pmatrix} 1 & -1 & -1 & \dots & -1 \\ 0 & 1 & -1 & \dots & -1 \\ 0 & 0 & 1 & \dots & -1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \tag{D.151}$$

which is a generalization of what we called Example 6. Your code should be such that it should be trivial to change the n to a new value.

Now, take `n=16` and solve the system (with $\mathbf{b}$ made up of alternating +1's and −1's, as in Eq. (D.26)) with and without a small perturbation of −0.01 in the bottom-left element. Then, evaluate the condition number $\kappa(\mathbf{A})$.

D.3 Carry out an eigenvalue sensitivity analysis for the following 20×20 matrix:

$$\begin{pmatrix} 20 & 19 & 18 & 17 & \dots & 3 & 2 & 1 \\ 19 & 19 & 18 & 17 & \dots & 3 & 2 & 1 \\ 0 & 18 & 18 & 17 & \dots & 3 & 2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 1 \end{pmatrix} \tag{D.152}$$

which is a generalization of our Example 8 from the main text, Eq. (D.38). Specifically, evaluate the condition numbers for the largest (in absolute value) 11 real eigenvalues and discuss which seem to be ill-conditioned.

D.4 Find a matrix that is sensitive to perturbations for all three cases of: (a) linear system of equations, (b) eigenvalue, and (c) eigenvector. Explain how you would go about constructing such a matrix.

D.5 When adding 0.005 to the bottom-left element of Example 11, Eq. (D.69), we saw that eigenvector $\mathbf{v}_0$ didn't change very much. Now that you are armed with the general derivation of eigenvector sensitivity, you should explore this perturbation in more detail. After that, try out small perturbations in other matrix elements and repeat the study.

D.6 Study the case of two spins, as per `twospins.py`, implementing the Hamiltonian matrix of Eq. (D.135). This time, instead of getting the $\mathbf{S}_{\text{I-II}}$ term as per Eq. (D.128), you should use Eq. (D.131), which involves Kronecker products.

You should also write down the generalization of Eq. (D.131) for the case of three spins (no need to code this up).