Supplementary Notes for lecture 12 for McGill Phys 357.

1. Raising/Lowering Operators and Defective Matrices

Let's say we have a multi-state system, and we've ordered the states by some quantum number m of an operator. For spin-1/2 systems, the allowed values of m would be $(-1/2, 1/2)$, for spin-1 they would be $(-1, 0, 1)$ and so on. If the basis we're working in are the eigenvectors of the operator, with the highest values of m first, then the raising/lowering operators are trivial to write down. Up to constants, the raising has to be:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

If you multiply by a vector on the right, then the first value (corresponding to the highest value of m) goes away, and all the remaining values move up by one spot, with a zero in the final value. If we put the ones along the diagonal, we'd get the same thing out we started with, but because we shifted them up by one spot in the matrix, we end up shifting up the values in our vector by one. In math:

$$\mathrm{A}_{+} = (a, b, c, d, e) \rightarrow (b, c, d, e, 0)$$

If we're already at the maximum m, then we can't raise that state to a higher value, so it goes away. Similarly, there is no state below the smallest value of m, so when we raise our state, we need to get a zero for the smallest m. If you start with a state $(0, 0, 1, 0, 0)$ say, then we get out $(0, 1, 0, 0, 0)$, and we can see that if we apply the raising operator to our starting pure state, the output is a pure state with the next higher m. The lowering operator is the same, but now we want to shift the ones down by one from the diagonal. That leaves us with

$$\mathrm{A}_{-} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

and (a, b, c, d, e) goes to $(0, a, b, c, d)$. Depending on how the raising/lowering operators are defined, there may be multiplicative constants (*e.g.* the usual definitions of the angular momentum raising/lowering operators include factors of $\hbar$ and numerical constants). The basic idea, that you have an operator with numbers one above or one below the diagonal, does not change.

1.1. Defective Matrices and SVD

The raising/lowering operators are a bit peculiar from a linear algbera viewpoint. If we have a triangular matrix, the eigenvalues are just the diagonal elements since the characteristic equation is the product of the diagonal[1]. We can see that the raising/lowering operators are triangular, because everything below/above the diagonal is zero. Of course, the diagonal is also zero, so all of our eigenvalues are zero. You might think "if all the eigenvalues are zero, the matrix A must be zero" because A times any eigenvector gives zero. Of course, we can see that not all vectors go away, so we must have vectors that *aren't* eigenvectors. That means we can't write the raising/lowering operators as

$$\mathrm{A} = \mathrm{V}\Lambda\mathrm{V}^T$$

. We do get the right number of eigenvalues, but the corresponding eigenvectors don't form a set of basis vectors. These matrices are called *defective*, and much of our intuition about linear algebra doesn't apply.

In this sort of situation, the singular value decomposition (SVD) of a matrix can be very useful. SVD factors a matrix A like this:

$$\mathrm{A} = \mathrm{USV}^{T?}$$

where U and V are orthogonal (if A is square), and S is diagonal. If A is complex, then U and V are unitary, while S is still real[2]. The values of S are the singular values, and behave similar to eigenvalues. We note that the SVD reduces to eigendecomposition for square symmetric/Hermitian matrices. The question mark on the transpose reflects that some packages take the transpose of V, while others don't. You should check before blindly using (or not) the tranpose. Python does not use the transpose, so we have A = USV. Happily, the SVD exists for all matrices, even when the eigendecomposition fails. When we call SVD on a raising/lowering operator, something interesting happens. We find there are $n-1$ singular values of 1 and, one singular value with a value of 0. The columns of U are just the columns of V shifted up by one (unless you're using python, in which case it's the rows of V).

What this is telling us is that if we multiply a vector by the SVD, V^T picks out the amplitude of the state for each of the pure states and moves them to the next state, except for the state with the zero singular value. For the simple case of working in your own basis, this is overkill. However, if someone gave you the matrix for the raising operator for a different operator, you could figure out the pure states of that operator just by taking the SVD of the raising operator. We can go even further, and order the pure states, because we know that each time we apply it, the states move up by one. We can find the state with the largest quantum number, because that corresponds to the zero singular value (this behavior is due to the fact that the raising operator has to kill off the state with the largest quantum number). So then I look for the state that turns into the highest state,

[1]If you aren't familiar with this, you can show it using the determinant. One way of calculating a determinant is to take every element in a column (or row), block out the row/column of that element, and take the determinant of the sub-matrix. The total determinant is then the sum over the row of the row times the sub-matrix determinant, with some signs. For a triangular matrix, we only have one non-zero element in the first column, then when we block out that row/column, we're left with a single non-zero entry in the first row/column of the sub-matrix. That process continues until the determinant is the product of the diagonal elements. If we subtract λ from the diagonal, we get a zero whenever λ is equal to an element along the diagonal, so the eigenvalues are just the diagonal entries.

[2]If A is rectangular, then we can zero-pad S to be rectangular, and then U and V are unitary. The more usual thing to do, though, is keep S square and just keep the corresponding columns of U and V. Is say, A has more rows than columns, then V will be unitary and U will have orthogonal columns, but not be fully unitary

which much be the state with the next-highest quantum number. If you keep multiplying by A, then you can rank-order the states by watching how many multiplies of A are needed to zero out a state. And of course, everything works with lowering operators as well, only it's the bottom-most state that disappears first.

In summary, raising/lowering operators are straightforward to write down if you're working in the basis of the operator that is getting raised/lowered. We won't always have that luxury, though, and in that case SVD can tell us a lot about the behavior of the operator that our usual eigenvalue decomposition can't.