Homework: Practice with Matlab

January 11, 2018

Here are a few exercises I developed to help you gain facility with some of the Matlab operations we will use this semester. Work in Matlab using the Command Window as we did in class.

1. Using a single colon (:), define a row vector `v` as $(1, 2, 3, 4, 5)$.
2. Using *two* colons define a row vector `v` as $(1, 2, 3, 4, 5)$. Yes, it's the same vector, but I'd like you to remember there is more than one way to accomplish this task.
3. Using the `linspace` command define a row vector `v` as $(1, 2, 3, 4, 5)$.
4. Use the `size` command, ask Matlab what the dimensions of `v` are. Do they make sense?
5. To create a column vector you can begin with a row vector and take its transpose. The command for this, when dealing with real vectors is `'`. So the transpose of `v` is `v'`. Try it to see if it works. Use `size` to confirm everything makes sense.
6. Matlab works nicely with complex numbers, and knows what `i` means. To test this out, take the square root of `-1`. Use Matlab's built-in help to figure out how to compute a square root, and then apply it to `-1`. Remember that complex numbers are expressed as $a + i \cdot b$ where a and b are both real, and $i = \sqrt{-1}$.
7. Create a new row vector `w`, this time setting its elements to be $(-2, -1012)$. Then redefine `w` by taking its square root. Use Matlab's powerful vector language to avoid taking the square root of each element individually. Instead, take the square root of the whole vector at once. Notice that it's fine to have `w` appear on both sides of `=` in an assignment. If I say `z=3` it gives `z` the value 3. If I then say `z = 2*z` then `z` is redefined as 6, and Matlab loses its memory of `z` being 3.
8. What happens when you do `w'`? What does this imply about the true action of `'`? Be sure to take into account your previous result in question 5. For comparison, try `w.'` (notice the dot before the apostrophe). How was the operation `'` modified by the addition of the dot before the apostrophe?

9. You already know how important $\int \psi^\star(x)\psi(x)\,dx$ is in the quantum theory. We need to be able to do something similar with vectors, and the inner product is the elegant solution to the problem.

   Matlab's multiplication symbol, `*`, defaults to meaning the *linear algebra* version of multiplication: scalar `*` scalar yields the familiar scalar multiplication result; row vector `*` column vector yields the inner product of the two vectors; matrix `*` column vector yields a column vector; and matrix `*` matrix gives another matrix.

   We're trying to duplicate with vectors the actions in $\int \psi^\star(x)\psi(x)\,dx$. In this expression, for each x, we take the value of $\psi(x)$ and multiply it by the value of $\psi^\star(x)$. We then add up all the multiplication results and get a number. Recall that the inner product of vectors involves multiplying corresponding entries and adding them up to give a scalar. This is exactly what we need! For this to work, though, the left vector must be a row, and the right vector must be a column. We also need to take that complex conjugation into account.

   This will work most smoothly if we follow the convention that quantum states are, by default, represented by *column* vectors. In other words, the vector version of $\psi(x)$ is a *column* vector $\vec{\psi}_x$. Then $\int \psi^\star(x)\psi(x)\,dx$ can be, in vector form, the inner product of $\vec{\psi}_x^\star$ and $\vec{\psi}_x$. In linear algebra notation we write

$$\langle \vec{\psi}_x, \vec{\psi}_x \rangle \tag{1}$$

   This notation implies that the vector to the left of the comma is both complex conjugated and transposed. If you've only worked with real vectors before you will not have encountered the complex conjugation aspect. We will need to keep careful track of it.

   Fortunately, Matlab is set up to make this easy. Recall that `z'` means take the complex conjugate transpose of the vector `z`—all in one step.

   Try this out with your previously defined vectors `w` and `v`. Take their inner product with `w` on the left and `v` on the right but first make sure that you still have `v` and `w` defined as $(1, 2, 3, 4, 5)$ and $(\sqrt{2}i, i, 0, 1, \sqrt{2})$, respectively. We set them up as rows originally, but we really want to be working with column vectors as our starting points. So step one is to transpose the two vectors without doing any conjugation. You may need to review the distinction between `'` and `.'` in Matlab. Once you have `v` and `w` as columns, take their inner product with `w` on the left and `v` on the right. The Matlab command to do this should be satisfyingly concise, and the result should be the complex valued scalar $11.0711 + 3.4142i$.