Physics 40: Laboratory Thirteen

Tuesday, May 12, 2020

Goals:

Reminder on eigenvectors and eigenvalues.
Eigenvalues in Classical Mechanics (normal modes).
Eigenvalues in Quantum Mechanics (spin-1/2 matrices).

[1]

On the course website you will find five files needed for today’s lab:

jacobi.c	the subroutine for diagonalizing a matrix
nrutil.c	a file containing some utilities used by jacobi.c
nrutil.h	a header file for nrutil.c
jacobi_test.c	the main module which reads in the matrix and calls jacobi.c
input.txt	sample input file

Here’s how to use them:

First, compile the different pieces:

```
gcc -o jacobi_test.o -c jacobi_test.c
gcc -o jacobi.o -c jacobi.c
gcc -o nrutil.o -c nrutil.c
```

The ’-o’ tells the compiler to create object files with the indicated names.

Then put everything together:

```
gcc -o jacobi_test jacobi_test.o jacobi.o nrutil.o -lm
```

Then run the program, which is called jacobi_test (again, the ’-o’ told the compiler to give it this special name instead of the default ’a.out’)

```
jacobi_test
```

You can either input stuff from the screen: first the dimension of the matrix, then the entries in each row, with each entry separated by a space and the rows separated by a carriage return, or you can enter the same information into a file, and then tell the program to look in the file for the input via the command ’< input.txt’. I believe it is easier to use the latter approach.

```
jacobi_test < input.txt
```

Here’s what should happen:

```
 $ jacobi_test
dimension of the matrix: 5
enter a 5 x 5 matrix (separated by space):
3 0 1 0 .5
0 4 1 0 .1
1 1 5 .4 .2
0 0 .4 2 1
.5 .1 .2 1 3

eigen problem for matrix A:
  3.000  0.000  1.000  0.000  0.500
  0.000  4.000  1.000  0.000  0.100
  1.000  1.000  5.000  0.400  0.200
  0.000  0.000  0.400  2.000  1.000
  0.500  0.100  0.200  1.000  3.000

number of Jacobi applied: 49
eigenvalues:
  2.523  3.834  5.982  1.280  3.382

eigenvectors:
  0.820 -0.391  0.305  0.234 -0.168
  0.257  0.689  0.426  0.071  0.523
 -0.376 -0.062  0.827 -0.140 -0.388
 -0.346 -0.301  0.124  0.804  0.357
 -0.030 -0.528  0.162 -0.523  0.648
```

[HW7-1] Diagonalize the matrix

$$k = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 1 \\ 0 & 1 & 4 \end{bmatrix}$$

"by hand". Now use "jacobi" and verify you get the same answers.

[HW7-2] The matrices associated with the measurement of the components of the spin of an electron in the x, y, z directions are:

$$S_x = \frac{\hbar}{2} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \qquad S_y = \frac{\hbar}{2} \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix} \qquad S_z = \frac{\hbar}{2} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Here $\hbar = 1.055 \times 10^{-34}$ is Planck's constant. What values can you get if you are in the lab and you measure S_x? How about S_y and S_z?

[HW7-3] Write a code which prints the dynamical matrix for a chain of N masses connected by springs k to a file. Write it in a way that "jacobi" can read it in: each line contains all the matrix elements in that row. Have the parameters N and k be inputs to the code.

One approach is to follow the steps:
Declare some big matrix, eg dynmatrix[128][128].
Fill the entire matrix with zeros (use two nested loops).
Set the diagonal entries dynmatrix[i][i] to $2k$ (just one loop needed).
Set the elements dynmatrix[i][i+1] to -k (just one loop needed).
Set the elements dynmatrix[i][i-1] to -k (just one loop needed).
In the last two steps you will need to be careful with the first and last rows!

[HW7-4] Run your code from HW7-3 for $k = 3$ and $N = 8$ to get the dynamical matrix into a file. Then run "jacobi" to diagonalize. What eigenvalues do you get?