Paul Buttles

10/30/2020

PHYS 331, HW09

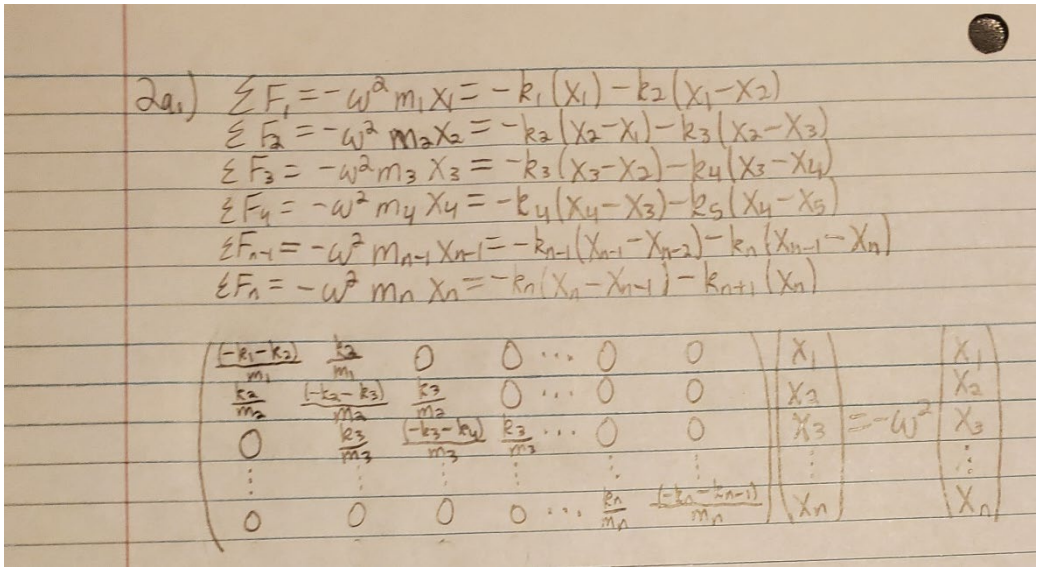
1b.) Aw1 and lambda1*w1 are equal to each other, which is to be expected as the eigenvalue equation they form sets them equal to each other.

1c.) The lambda matrix is what we expected. The off-diagonal elements are all on the order of magnitude of 10^-16 or less in both the real and imaginary parts, which makes them effectively 0+0i within machine precision. The diagonal elements match the calculated eigenvalues, for example the Lii element of L corresponds to the ith eigenvalue. This is effectively the same as taking the column vector of eigenvalues and multiplying them by the identity matrix, pasting the values along the diagonal. Thus this is what we expected.

1ECb.) When tested in python, the eigenvalues of the random Hermitian matrix were seen to be real for a 5x5 Hermitian matrix. Eigenvalues returned from a finite Hermitian matrix of size nxn are expected to be real. The determinant of a Hermitian matrix of size nxn is decomposed into 2x2 determinants. By properties of Hermitian matrices, the diagonals must be real, and the off-diagonals can be real or complex. If they are both complex, multiplying them makes them real. If they are both real, they are obviously real when multiplied. If one is real and one is complex, the output is complex, but because the Hermitian matrix is symmetric with its complex conjugate, the complex output of this determinant will produce the complex conjugate of this output elsewhere in the matrix, thus canceling them out. As a result, the Hermitian matrix is reduced to a series of real valued determinants, and hence the whole determinant is real. Subtracting eigenvalues along the diagonal does not change this, as the real-valued diagonal must continue to be real-valued in order to produce the determinant we have proven to be real. Thus this implies that whatever is subtracted from the diagonals must be real. Hence it is proven that the eigenvalues of Hermitian matrix are real.

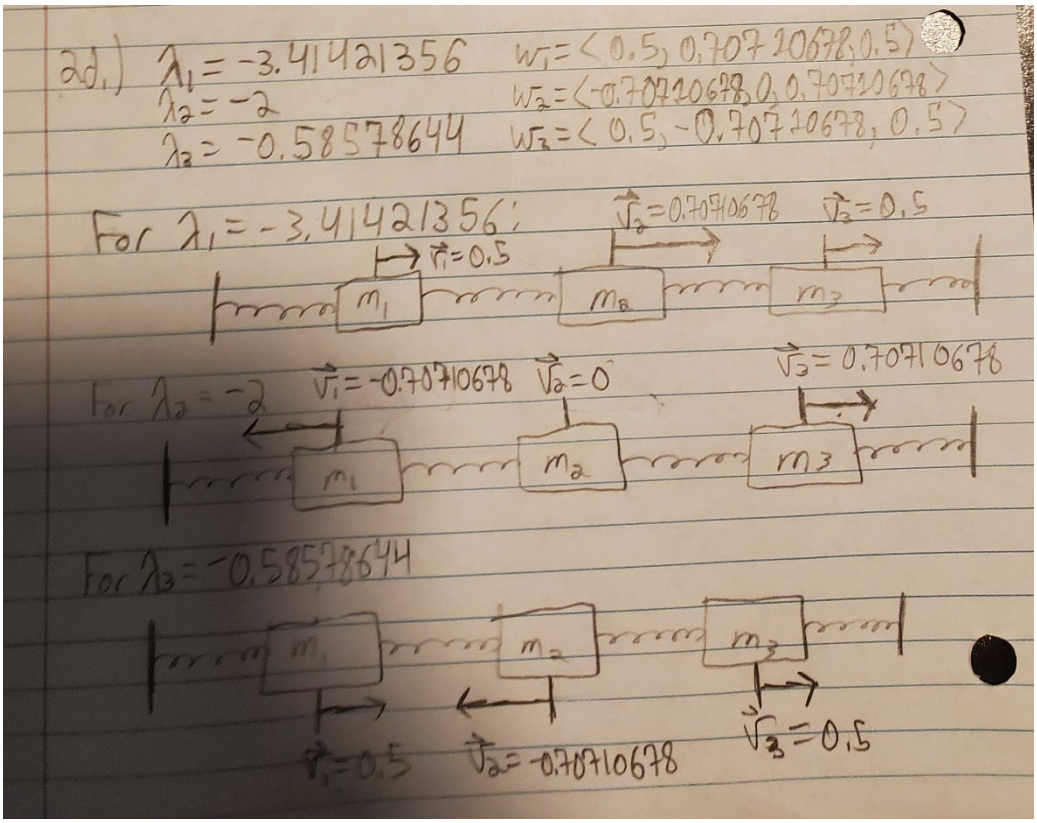
1ECc.) The first two eigenvectors of a 5x5 Hermitian matrix were seen to be orthogonal as their dot product is 0 within machine precision. This is not true for all eigenvectors, but it is a special property of symmetric and Hermitian matrices that their eigenvectors are always orthogonal. The proof for why this is, is a bit outside my scope, but suffice it to say that all eigenvectors of a matrix are linearly independent, and certain special cases like symmetric matrices and Hermitian matrices are orthogonal as well.

2a.)



2c.) The eigenvalues and eigenvectors are to be expected, as the calculated values are equal to the solutions we analytically solved for in class. The eigenvalues we solved for are from an equation of the form Ax=lambda*x, but in this case our lambda was replaced with -omega^2. Thus the eigenvalues we solved for are equivalent to -omega^2. So the oscillatory frequency of the masses is equal to sqrt(-lambda) for the lambdas we found. The eigenvectors correspond to the velocities of the masses. For example, w1 = (0.70710687, -0.70710687) corresponds to the masses swinging in equal magnitude and opposite direction, while w2 = (0.70710687, 0.70710687) corresponds to them swinging together in equal magnitude and direction.

2d.) For lambda_1 = -3.41421356, w1 = < 5.00000000e-01, 7.07106781e-01, 5.00000000e-01 >
For lambda_2 = -2, w2 = < -7.07106781e-01, -4.05405432e-16, 7.07106781e-01 >
For lambda_3 = -0.58578644, w3 = < 5.00000000e-01, -7.07106781e-01, 5.00000000e-01 >
Or more simply, w1 = < 0.5, 0.70710678, 0.5 >, w2 = < -0.70710678, 0, 0.70710678 >,
w3 = < 0.5, -0.70710678, 0.5 >



2e.) The eigenvalues are the frequencies at which the molecules in the lattice can oscillate. There is a distinct gap in the middle of the histogram, a region of eigenvalues which the molecules cannot occupy. Thus they cannot oscillate at this frequency, creating an energy band gap. For a lower value of k, the "springs" are less stiff, corresponding to the molecules oscillating more easily and thus at a wider range of frequencies. This reduces the energy band gap as they are now able to move at frequencies previously unattainable with "stiffer springs".

3a.)

3a.) $y''(x) + (2x+3)y'(x) + 6xy = x$

$y(0) = 1$

$y'(0) = 1$

Let $y_1 = y$ and $y_2 = y'$.

$y_1 = y$

$y_2 = y'$

$y'(x) = y_2$

$y''(x) = y_2'$

$$y_1'(x) = y_2(x)$$
$$y_2'(x) = x - 6xy_1(x) - (2x+3)y_2(x)$$

Because $y_1 = y$ and $y(0) = 1$, $y_1(0) = 1$

Because $y_2 = y'$ and $y'(0) = 1$, $y_2(0) = 1$

$$y_1(0) = 1$$
$$y_2(0) = 1$$

3c.) When h=1, only five points are returned from the numerical integration for each function. The middle three values are roughly on the same order of magnitude as the points in the other graphs, and the left hand endpoint also happens to be pretty close, but the righthand endpoint is several orders of magnitude off from the others. Runge-Kutta 4 is good for computational efficient numerical integration of ODEs, but it relies on a good choice of h, particularly in the way it handles truncation error (it doesn't). As a result, this endpoint ends up being very skewed for a poor choice of h, and the function doesn't do anything about it, resulting in a plot that is neither accurate nor useful. Better choices of h fix this, producing the curve we might expect.

The first curve (blue) is y1(x), and y1(x)=y, so this curve is the function y that we sought to find. The second curve (orange) is y2(x), and y2(x)=y', so this curve is the first derivative of the function y that we sought to find. The choice of h=1 is not good and produces a large truncation error, as discussed before. For h=0.1, it is immediately evident that this is a better choice. Because of the somewhat coarse mesh size, the curve is slightly jagged, but is essentially exactly what one might expect to see. Bumping this up to h=0.01, the curve smooths out, producing a nice quality curve that is fit to be used as a model. Bumping this up again to h=0.001 produces virtually no noticeable difference, as RK4 is very computationally efficient so it zeros in on an adequate model quite quickly, making increasingly fine values for h superfluous to finding a good model.