Armaan Sethi
Phys 331

1.

a) The elimination phase of the Gauss elimination simplifies the linear systems in the matrix in order to solve it. It is very similar to the elimination method to solving algebraic systems (when you add or subtract two algebraic equations in order to solve the equation).

b) The solutions are the same, comments in code. (Before the Sakai announcement I compared with numpys linalg solution since my solution was not agreeing with the solution given.) The datatype must be float for the function to work.

c) You cannot have a zero on the diagonal for gaussian elimination, so I had to pivot rows 1 and 2 before passing it to the function. (Since we were unable to modify the function to support pivoting)

2.

a) In the code.

b) In the code.

3.

a)

$$\begin{bmatrix} 4 & -2 & 1 \\ -3 & -1 & 4 \\ 1 & -1 & 3 \end{bmatrix} \begin{matrix} \\ 3/4 \\ -1/4 \end{matrix} \qquad \begin{matrix} 1 \\ -3/4 \\ 1/4 \end{matrix}$$

Part A

1 mass

$$\frac{-k_1 - k_2}{m_A} x_A$$

2 mass

$$\Sigma F_A = -K_1 x_A - k_2 (x_A - x_B) = m_A \ddot{x}_A$$

$$\Sigma F_B = -k_2 (x_B - x_A) - k_3 x_B = m_B \ddot{x}_B$$

$$\begin{bmatrix} -\left(\frac{k_1 + k_2}{m_a}\right) & \left(\frac{k_2}{m_a}\right) \\ \left(\frac{k_2}{m_b}\right) & -\left(\frac{k_2 + k_3}{m_b}\right) \end{bmatrix} \begin{bmatrix} \ddot{x}_a \\ \ddot{x}_b \end{bmatrix} = -\omega^2 \begin{bmatrix} x_A \\ x_B \end{bmatrix}$$

3 mass

$$\Sigma F_A = -K_1 x_A - k_2 (x_A - x_B) = m_A \ddot{x}_A$$

$$\Sigma F_B = -K_2 (x_B - x_A) - K_3 (x_B - x_C) = m_B \ddot{x}_B$$

$$-k_2 x_B + k_2 x_A \qquad -k_3 x_B + k_3 x_C$$

$$\Sigma F_C = -K_3 (x_B - x_C) - K_4 x_C = m_C \ddot{x}_C$$

$$-k_3 x_b + k_3 x_c \qquad -k_4 x_c$$

Tridiagonal matrix

$$\begin{bmatrix} -\left(\frac{k_1 + k_2}{m_a}\right) & \left(\frac{k_2}{m_a}\right) & 0 \\ \left(\frac{k_2}{m_B}\right) & -\left(\frac{k_2 + k_3}{m_b}\right) & \left(\frac{k_3}{m_b}\right) \\ 0 & \left(\frac{k_3}{m_c}\right) & \left(-\frac{k_3 + k_4}{m_c}\right) \end{bmatrix} \begin{bmatrix} x_a \\ x_B \\ x_C \end{bmatrix}$$

4 mass

$$\sum F_A = -k_1 x_A + k_2(x_B - x_A)$$
$$\sum F_B = -k_2(x_B - x_A) + k_3(x_C - x_B)$$
$$\sum F_C = -k_3(x_C - x_B) + k_4(x_D - x_C)$$
$$\sum F_D = -k_4(x_D - x_C) + \underbrace{k_5}_{\to 0} x_D$$

$$\downarrow i \begin{bmatrix} (-k_1 - k_2)/m_A & k_2 & 0 & 0 \\ k_2 & (-k_2 - k_3) & k_3 & 0 \\ 0 & k_3 & -k_3 - k_4 & k_4 \\ 0 & 0 & k_4 & -k_4 + k_5 \end{bmatrix} \begin{array}{l} /m_A \\ /m_B \\ /m_C \\ /m_D \end{array} \begin{bmatrix} x_A \\ x_B \\ x_C \\ x_D \end{bmatrix}$$

b) In the code

c) In the code

d)In the code

Eigenvalues are : (-3.41421356, -2, -0.58578644)

Angular Velocities [1.8477590650225726j, 1.414213562373095j, 0.7653668647301795j]

e) The width of the gap of eigenvalues that do not exist in the system is much greater in the first example (when the masses are 1 and 1.5) then the second example (when the masses are 1 and 1.2)

Extra Credit:

Eigenvalues seem to appear completely symmetrical around the band gap.