Fall 2002. 10.34. Numerical Methods Applied to Chemical Engineering

Homework # 4. Nonlinear algebraic equations and matrix eigenvalue problems

Assigned Monday 10/6/02. Due Friday 10/11/02

Problem 1. Calculation of vibrational modes with eigenvalue analysis

In this homework, you will use a program that computes the normal (vibrational) modes of a 2D lattice (see figure below) using eigenvalue analysis. A MATLAB program will be posted on the web site that calculates each vibrational mode and natural frequency for this system. Another program will be provided that takes this information and makes a "movie" to animate a selected vibrational mode. You will be asked to use these programs to compare the natures of the high and low energy vibrational modes of this structure. ***No programming will be required!***

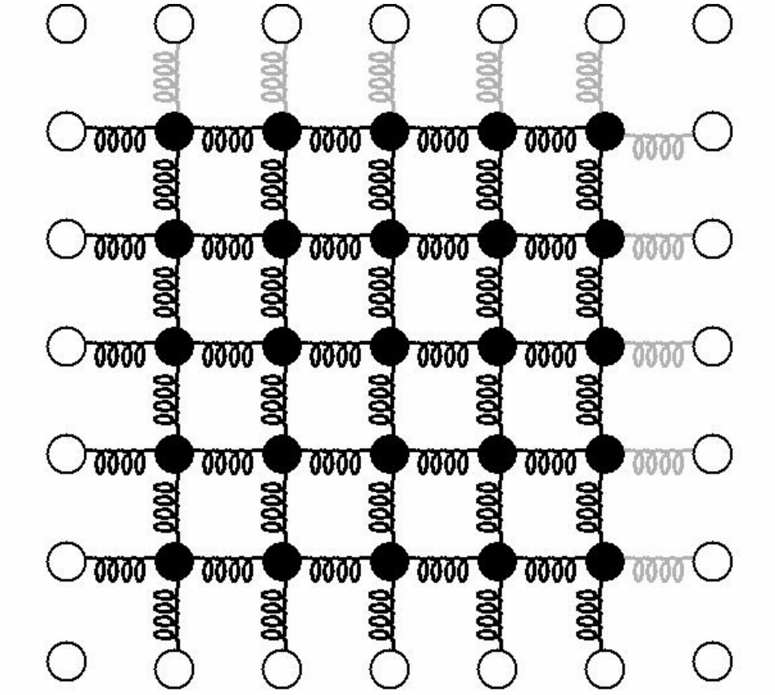

FIGURE 1. 2-D lattice of point mass-spring system

In this problem, we wish to investigate the vibrational modes of an 2-D lattice of point masses connected by springs. We assume that the lattice is infinite, but we only want to simulate a small section of *N* "rows" and *N* "columns" of points, shown in the figure above as the black, filled circles. To mimic the effect of having an infinitely large lattice of points, we employ periodic boundary conditions in which we assume that the system is bounded by perfect copies of itself on all sides (see figure below). When one of the "atoms" in our system moves to the left, its image also moves to the left. In the figure below, curved arrows link the atoms in the system (filled circles) to their images outside of the system (open circles). For further details on how the periodic boundary condition is employed, consult the attached MATLAB program.

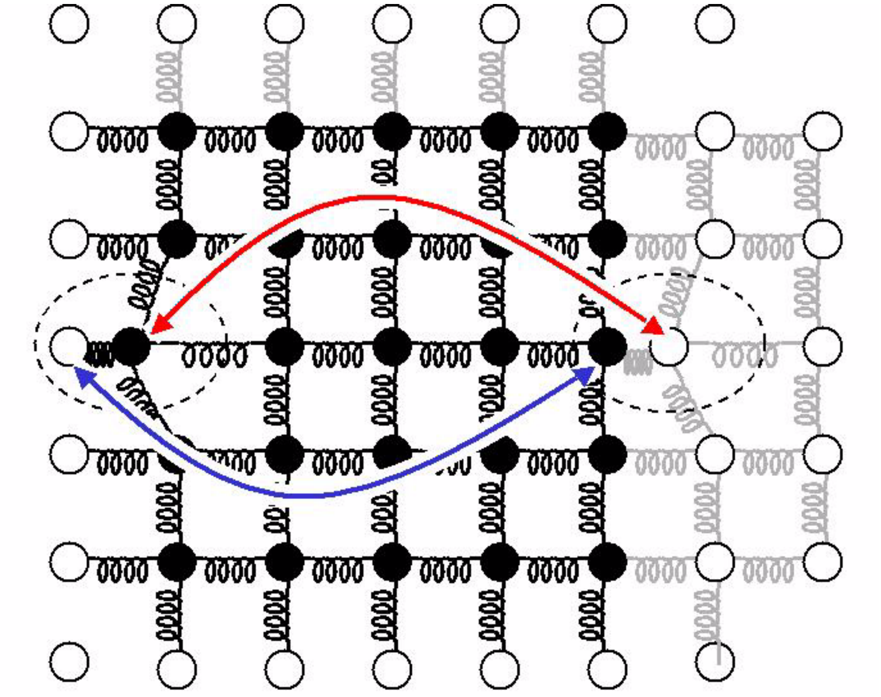

FIGURE 2. Linking of atoms and their images in period boundary conditions to mimic an infinite lattice

We now number the lattice sites from 1 to $S = N^2$ as shown in the figure below.

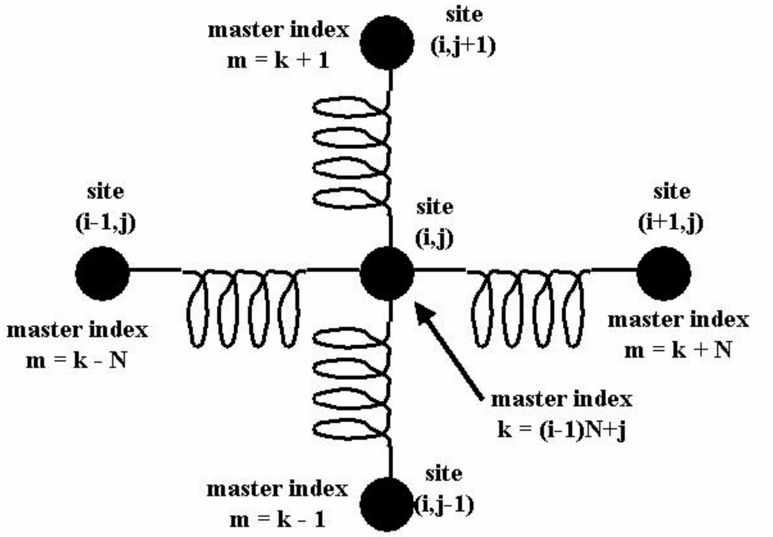

FIGURE 3. Numbering system for lattice sites, including identification of neighbor site bonding

While we will use the master indices for labelling purposes, it is easier to write the potential energy function for the system if we label the site in “column” *i* and “row” *j* as (i, j). The position of this site is

$$\underline{r}_{(i, j)} = \begin{bmatrix} x_{(i, j)} \\ y_{(i, j)} \end{bmatrix} \tag{EQ 1}$$

We write the total potential energy of the system as a sum of harmonic springs bonding neighboring lattice sites,

$$\begin{aligned} U(\underline{r}_{(1, 1)}, \ldots, \underline{r}_{(N, N)}) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{1}{2}K[(r^{(i,j)}_{(i-1,j)} - l_b)^2 + (r^{(i,j)}_{(i+1,j)} - l_b)^2 \\ + (r^{(i,j)}_{(i,j-1)} - l_b)^2 + (r^{(i,j)}_{(i,j+1)} - l_b)^2] \end{aligned} \tag{EQ 2}$$

A factor of $1/2$ is added before the summation to correct for the overcounting of each spring twice. The distance between site (i, j) and the image of site (m, n) with which it interacts is

$$r^{(i,j)}_{(m,n)} = [(\Delta x^{(i,j)}_{(m,n)})^2 + (\Delta y^{(i,j)}_{(m,n)})^2]^{1/2} \tag{EQ 3}$$

where we implement the periodic boundary conditions (figure 2) as

$$\begin{aligned} \Delta x^{(i,j)}_{(i-1,j)} &= \begin{cases} x_{(i,j)} - x_{(i-1,j)}, i \in [2, N] \\ x_{(1,j)} - x_{(N,j)} + Nl_b, i = 1 \end{cases} \\ \Delta x^{(i,j)}_{(i+1,j)} &= \begin{cases} x_{(i,j)} - x_{(i+1,j)}, i \in [1, N-1] \\ x_{(N,j)} - x_{(1,j)} - Nl_b, i = N \end{cases} \end{aligned} \tag{EQ 4}$$

$$\begin{aligned} \Delta y^{(i,j)}_{(i,j-1)} &= \begin{cases} y_{(i,j)} - y_{(i,j-1)}, j \in [2, N] \\ y_{(i,1)} - y_{(i,N)} + Nl_b, j = 1 \end{cases} \\ \Delta y^{(i,j)}_{(i,j+1)} &= \begin{cases} y_{(i,j)} - y_{(i,j+1)}, j \in [1, N-1] \\ y_{(i,N)} - y_{(i,1)} - Nl_b, j = N \end{cases} \end{aligned} \tag{EQ 5}$$

We subtract from each bond distance the preferred (zero-energy) value of l_b so that at mechanical equilibrium, all lattice sites are separated by this distance. The position in the minimum energy state of the (i, j) lattice site - that in “column” i and “row” j with master index $k = (i-1)N + j$ - is

$$\hat{\underline{r}}_{(i,j)} = \begin{bmatrix} \hat{x}_{(i,j)} \\ \hat{y}_{(i,j)} \end{bmatrix} = \begin{bmatrix} (i-1)l_b \\ (j-1)l_b \end{bmatrix} \tag{EQ 6}$$

We wish to compute the vibrational modes of the lattice and their natural frequencies. To do so, we pack all of the degrees of freedom of the system into a single vector of length $F = 2S$, $\underline{q}$. We also define a vector $\underline{f}$ of the forces on each degree of freedom that is the negative of the gradient of the potential energy.

$$\underline{q} = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ : \\ q_{F-1} \\ q_F \end{bmatrix} = \begin{bmatrix} x_{(1,1)} \\ y_{(1,1)} \\ x_{(2,1)} \\ y_{(2,2)} \\ : \\ x_{(N,N)} \\ y_{(N,N)} \end{bmatrix} \qquad \underline{f} = -\underline{\nabla} U = -\begin{bmatrix} \frac{\partial U}{\partial q_1} \\ \frac{\partial U}{\partial q_2} \\ \frac{\partial U}{\partial q_3} \\ \frac{\partial U}{\partial q_4} \\ : \\ \frac{\partial U}{\partial q_{F-1}} \\ \frac{\partial U}{\partial q_F} \end{bmatrix} = -\begin{bmatrix} \frac{\partial U}{\partial x_{(1,1)}} \\ \frac{\partial U}{\partial y_{(1,1)}} \\ \frac{\partial U}{\partial x_{(2,1)}} \\ \frac{\partial U}{\partial y_{(2,1)}} \\ : \\ \frac{\partial U}{\partial x_{(N,N)}} \\ \frac{\partial U}{\partial y_{(N,N)}} \end{bmatrix} \tag{EQ 7}$$

For the potential energy function above, the derivatives of the potential energy with respect to the lattice site positions are computed in a straight-forward, if tedious, manner.

$$\frac{\partial U}{\partial x_{(i,j)}} = \frac{K}{2}\frac{\partial}{\partial x_{(i,j)}}[(r^{(i,j)}_{(i+1,j)} - l_b)^2 + (r^{(i,j)}_{(i-1,j)} - l_b)^2 + (r^{(i,j)}_{(i,j+1)} - l_b)^2 + (r^{(i,j)}_{(i,j-1)} - l_b)^2]$$

$$\begin{aligned} \frac{\partial U}{\partial x_{(i,j)}} = \frac{K}{2}\Bigg[& 2(r^{(i,j)}_{(i+1,j)} - l_b)\frac{\partial r^{(i,j)}_{(i+1,j)}}{\partial x_{(i,j)}} + 2(r^{(i,j)}_{(i-1,j)} - l_b)\frac{\partial r^{(i,j)}_{(i-1,j)}}{\partial x_{(i,j)}} \\ & + 2(r^{(i,j)}_{(i,j+1)} - l_b)\frac{\partial r^{(i,j)}_{(i,j+1)}}{\partial x_{(i,j)}} + 2(r^{(i,j)}_{(i,j-1)} - l_b)\frac{\partial r^{(i,j)}_{(i,j-1)}}{\partial x_{(i,j)}}\Bigg] \end{aligned} \tag{EQ 8}$$

$$\begin{aligned} \frac{\partial U}{\partial x_{(i,j)}} = K\Bigg[& (r^{(i,j)}_{(i+1,j)} - l_b)\frac{\Delta x^{(i,j)}_{(i+1,j)}}{r^{(i,j)}_{(i+1,j)}} + (r^{(i,j)}_{(i-1,j)} - l_b)\frac{\Delta x^{(i,j)}_{(i-1,j)}}{r^{(i,j)}_{(i-1,j)}} \\ & + (r^{(i,j)}_{(i,j+1)} - l_b)\frac{\Delta x^{(i,j)}_{(i,j+1)}}{r^{(i,j)}_{(i,j+1)}} + (r^{(i,j)}_{(i,j-1)} - l_b)\frac{\Delta x^{(i,j)}_{(i,j-1)}}{r^{(i,j)}_{(i,j-1)}}\Bigg] \end{aligned}$$

$$\frac{\partial U}{\partial y_{(i,j)}} = K\left[(r^{(i,j)}_{(i+1,j)} - l_b)\frac{\Delta y^{(i,j)}_{(i+1,j)}}{r^{(i,j)}_{(i+1,j)}} + (r^{(i,j)}_{(i-1,j)} - l_b)\frac{\Delta y^{(i,j)}_{(i-1,j)}}{r^{(i,j)}_{(i-1,j)}} + (r^{(i,j)}_{(i,j+1)} - l_b)\frac{\Delta y^{(i,j)}_{(i,j+1)}}{r^{(i,j)}_{(i,j+1)}} + (r^{(i,j)}_{(i,j-1)} - l_b)\frac{\Delta y^{(i,j)}_{(i,j-1)}}{r^{(i,j)}_{(i,j-1)}}\right] \tag{EQ 9}$$

We define also a matrix of size $F \times F$ of the second derivatives of the energy with respect to each degree of freedom, the so-called Hessian matrix H, with the elements

$$H_{mn} = \frac{\partial^2 U}{\partial q_m \partial q_n} = \frac{\partial^2 U}{\partial q_n \partial q_m} = H_{nm} \tag{EQ 10}$$

Here we see that because the order of differentiation does not matter, the Hessian matrix is necessarily real and symmetric, so that all eigenvalues of H are real and all eigenvectors of H are mutually orthogonal.

$$H = \begin{bmatrix} \frac{\partial^2 U}{\partial q_1 \partial q_1} & \frac{\partial^2 U}{\partial q_1 \partial q_2} & \cdots & \frac{\partial^2 U}{\partial q_1 \partial q_F} \\ \frac{\partial^2 U}{\partial q_2 \partial q_1} & \frac{\partial^2 U}{\partial q_2 \partial q_2} & \cdots & \frac{\partial^2 U}{\partial q_2 \partial q_F} \\ : & : & & : \\ \frac{\partial^2 U}{\partial q_F \partial q_1} & \frac{\partial^2 U}{\partial q_F \partial q_2} & \cdots & \frac{\partial^2 U}{\partial q_F \partial q_F} \end{bmatrix} \tag{EQ 11}$$

Rather than derive the Hessian analytically, we will compute the value of the Hessian - evaluated at $\underline{q} = \hat{\underline{q}}$ using finite difference approximations,

$$\left.\frac{\partial^2 U}{\partial q_i \partial q_j}\right|_{\hat{\underline{q}}} \approx \frac{(\partial U/\partial q_i)\big|_{\hat{\underline{q}} + \varepsilon \underline{e}^{[j]}} - (\partial U/\partial q_i)\big|_{\hat{\underline{q}}}}{\varepsilon} \tag{EQ 12}$$

For each of the degrees of freedom $j = 1, 2, \ldots, F$ in the system, we change the value of q_j by some small value ε and evaluate again the vector of first derivatives. Application of the formula above provides a numerical estimate of the values in the j^{th} column of the Hessian matrix.

In general, we are given the potential energy function, and must find the state of minimum energy numerically. Below, we show in an optional discussion section how one may modify Newton's method to find the minimum. For this system, we can identify *a priori* the values of the generalized coordinates at the minimum energy state, $\underline{q} = \hat{\underline{q}}$.

$$\hat{\underline{q}} = \begin{bmatrix} \hat{q}_1 \\ \hat{q}_2 \\ \hat{q}_3 \\ \hat{q}_4 \\ : \\ \hat{q}_{F-1} \\ \hat{q}_F \end{bmatrix} = \begin{bmatrix} \hat{x}_{(1,1)} \\ \hat{y}_{(1,1)} \\ \hat{x}_{(2,1)} \\ \hat{y}_{(2,2)} \\ : \\ \hat{x}_{(N,N)} \\ \hat{y}_{(N,N)} \end{bmatrix} \qquad \text{(EQ 13)}$$

We now wish to consider the nature of the vibrations of the system around this minimum energy state. To address this question, we write the time-dependent values of the system coordinates as the sum of the minimum energy state plus a small time-varying “departure” term,

$$\underline{q}(t) = \hat{\underline{q}} + \underline{\delta}(t) \qquad \text{(EQ 14)}$$

The equations of motion for each degree of freedom are of the form,

$$m_j \frac{d^2 q_j}{dt^2} = m_j \frac{d^2 \delta_j}{dt^2} = f_j = -\frac{\partial}{\partial \delta_j} U(\hat{q}_1 + \delta_1, \ldots, \hat{q}_F + \delta_F) \qquad \text{(EQ 15)}$$

For small $\underline{\delta}(t)$,we can expand the potential energy function as a truncated Taylor series around the minimum energy state,

$$U(\hat{q}_1 + \delta_1, \ldots, \hat{q}_F + \delta_F) \approx U(\hat{q}_1, \ldots, \hat{q}_F) + \sum_{m=1}^{F} \left(\left. \frac{\partial U}{\partial q_m} \right|_{\hat{q}} \right) \delta_m + \frac{1}{2} \sum_{m=1}^{F} \sum_{n=1}^{F} \delta_m \left(\left. \frac{\partial^2 U}{\partial q_m \partial q_n} \right|_{\hat{q}} \right) \delta_n \qquad \text{(EQ 16)}$$

Since at the minimum energy state, all forces are zero, the first order derivatives in this expansion are zero, and the second order derivatives are just the elements of the Hessian matrix H evaluated at the minimum energy state. Therefore, for small departures from the minimum energy state, we can approximate the potential energy as

$$U(\hat{q}_1 + \delta_1, \ldots, \hat{q}_F + \delta_F) \approx U(\hat{q}_1, \ldots, \hat{q}_F) + \frac{1}{2} \sum_{m=1}^{F} \sum_{n=1}^{F} \delta_m H_{mn} \delta_n \qquad \text{(EQ 17)}$$

In general, this quadratic expression for the potential energy in the vicinity of a minimum is an approximation, but for our system it is exact, since our potential energy is simply the sum of a number of harmonic springs.

Substituting this equation into the equation of motion for $\delta_j(t)$ yields

$$\begin{aligned}
m_j \frac{d^2\delta_j}{dt^2} &= -\frac{\partial}{\partial\delta_j}\left[U(\hat{q}_1, \ldots, \hat{q}_F) + \frac{1}{2}\sum_{m=1}^{F}\sum_{n=1}^{F}\delta_m H_{mn}\delta_n\right] \\
m_j \frac{d^2\delta_j}{dt^2} &= -\frac{1}{2}\sum_{n=1}^{F} H_{jn}\delta_n - \frac{1}{2}\sum_{m=1}^{F}\delta_m H_{mj} \qquad \text{(EQ 18)} \\
m_j \frac{d^2\delta_j}{dt^2} &= -\sum_{n=1}^{F} H_{jn}\delta_n = -(H\underline{\delta})_j
\end{aligned}$$

where in the last line we have used the symmetry of the Hessian matrix. If we assume an equal mass m for each lattice site, we define the mass matrix

$$M = \begin{bmatrix} m_1 & & & \\ & m_2 & & \\ & & \cdots & \\ & & & m_F \end{bmatrix} = mI \tag{EQ 19}$$

Defining the vector of the second time derivatives of the coordinates as,

$$\ddot{\underline{\delta}} = \begin{bmatrix} \frac{d^2\delta_1}{dt^2} \\ \frac{d^2\delta_2}{dt^2} \\ \vdots \\ \frac{d^2\delta_F}{dt^2} \end{bmatrix} \tag{EQ 20}$$

the system of differential equations for the vibrational motion around the minimum energy state is

$$M\ddot{\underline{\delta}} = -H\underline{\delta} \tag{EQ 21}$$

This appears very similar to a matrix eigenvalue problem except for the appearance of the mass matrix on the left hand side. This is known as a generalized eigenvalue problem (the MATLAB commands **eig()** and **eigs()** can solve these types of problems as well). For our particular case where $M = mI$, it is easy to convert the problem into a simple eigenvalue problem of the sort that we have been considering,

$$m\ddot{\underline{\delta}} = -H\underline{\delta} \tag{EQ 22}$$

We now define the eigenvalues and eigenvectors of the Hessian matrix as

$$H\underline{w}^{[j]} = \lambda_j\underline{w}^{[j]} \qquad j = 1, 2, \ldots, F \tag{EQ 23}$$

Since the Hessian matrix is real and symmetric, we know that the eigenvalues of H are real and the eigenvectors are mutually orthogonal, and so form a convenient basis set for representing any vector. We therefore write a trial form of $\underline{\delta}(t)$ as the linear combination

$$\begin{gathered}\underline{\delta}(t) = c_1(t)\underline{w}^{[1]} + \ldots + c_F(t)\underline{w}^{[F]} \\ \ddot{\underline{\delta}}(t) = \ddot{c}_1\underline{w}^{[1]} + \ldots + \ddot{c}_F\underline{w}^{[F]}\end{gathered} \tag{EQ 24}$$

So that the system of equations becomes,

$$\begin{gathered}m[\ddot{c}_1\underline{w}^{[1]} + \ldots + \ddot{c}_F\underline{w}^{[F]}] = -H[c_1\underline{w}^{[1]} + \ldots + c_F\underline{w}^{[F]}] \\ m\ddot{c}_1\underline{w}^{[1]} + \ldots + m\ddot{c}_F\underline{w}^{[F]} = -[c_1H\underline{w}^{[1]} + \ldots + c_FH\underline{w}^{[F]}] \\ m\ddot{c}_1\underline{w}^{[1]} + \ldots + m\ddot{c}_F\underline{w}^{[F]} = -[c_1\lambda_1\underline{w}^{[1]} + \ldots + c_F\lambda_F\underline{w}^{[F]}]\end{gathered} \tag{EQ 25}$$

Equating the left and right sides separately in each eigenvector direction yields the following uncoupled set of equations for each normal coordinate $c_j(t)$,

$$m\frac{d^2c_j}{dt^2} = -\lambda_jc_j \tag{EQ 26}$$

We write a trial form of the solution as

$$c_j(t) = a_j\sin(\omega_jt) \tag{EQ 27}$$

and substitute it into the differential equation,

$$\frac{d^2c_j}{dt^2} = a_j\omega_j^2\sin(\omega_jt) = -m^{-1}\lambda_ja_j\sin(\omega_jt) = -m^{-1}\lambda_jc_j \tag{EQ 28}$$

We see that the trial form of the solution is valid, and identify the angular frequency of the normal mode as

$$\omega_j = \sqrt{\frac{\lambda_j}{m}} \tag{EQ 29}$$

For this frequency to be a real number, we need each eigenvalue of H to be positive, or at-least non-negative $\lambda_j \geq 0$. To show that this is the case, we write the expansion of the potential energy around the minimum energy state as

$$U(\underline{q}) - U(\hat{\underline{q}}) = \frac{1}{2}\sum_{m=1}^{F}\sum_{n=1}^{F}\delta_m H_{mn}\delta_n = \frac{1}{2}\underline{\delta}^T H\underline{\delta} \tag{EQ 30}$$

Expanding the departure vector in terms of the eigenvectors of H,

$$\underline{\delta} = c_1\underline{w}^{[1]} + \ldots + c_F\underline{w}^{[F]} \tag{EQ 31}$$

we obtain

$$\begin{gathered}
U(\underline{q}) - U(\hat{\underline{q}}) = \frac{1}{2}\underline{\delta}^T H[c_1\underline{w}^{[1]} + \ldots + c_F\underline{w}^{[F]}] \\
U(\underline{q}) - U(\hat{\underline{q}}) = \frac{1}{2}\underline{\delta}^T[c_1 H\underline{w}^{[1]} + \ldots + c_F H\underline{w}^{[F]}] \\
U(\underline{q}) - U(\hat{\underline{q}}) = \frac{1}{2}[c_1\underline{w}^{[1]} + \ldots + c_F\underline{w}^{[F]}]^T[c_1\lambda_1\underline{w}^{[1]} + \ldots + c_F\lambda_F\underline{w}^{[F]}] \\
U(\underline{q}) - U(\hat{\underline{q}}) = \frac{1}{2}[c_1^2\lambda_1 + \ldots + c_F^2\lambda_F]
\end{gathered} \tag{EQ 32}$$

where in the last line we have used the orthonormality of the eigenvectors of the real, symmetric matrix H. If $\hat{\underline{q}}$ is a true minimum of the potential energy, then $U(\underline{q}) - U(\hat{\underline{q}}) > 0$, requiring that all eigenvalues of H be positive (i.e. H is said to be *positive-definite*).

For our 2-D lattice, we use a potential energy function in which the energy depends only on the relative separation distance between bonded lattice sites. We can therefore translate the entire system uniformly in any direction without affecting the total energy. Since the potential energy is invariant with respect to the two degrees of freedom of the system involving uniform translation, there will be two eigenvectors of H with eigenvalues of zero. These two translational eigenvectors will be linear combinations of the vectors,

$$\underline{w}^{[\text{trans,x}]} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ : \\ x_S \\ y_S \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ : \\ 1 \\ 0 \end{bmatrix} \qquad \underline{w}^{[\text{trans,y}]} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ : \\ x_S \\ y_S \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ : \\ 0 \\ 1 \end{bmatrix} \qquad \text{(EQ 33)}$$

Similarly, we can perform any rigid body rotation on the lattice without changing the potential energy, so that there will be an additional two invariant rotational degrees of freedom that introduce another two eigenvectors corresponding to zero eigenvalues.

In general, these would be the only energy-invariant eigenvectors, but for our 2-D lattice, there are a number of other zero-energy eigenvectors corresponding to shear deformation modes. From inspection of the regular 2-D lattice, we see that we can take any row or column and displace it uniformly by a small amount without changing, to first order, the distance between any bonds. For this type of motion, we are only rotating the bonds, not stretching them. Therefore, this regular 2-D lattice will have the unusual, and unphysical, property of a zero shear modulus (no resistance to infinitesimal shear deformations). Due to the presence of these zero-frequency modes, for this potential energy function, $U(\underline{q}) - U(\hat{\underline{q}}) \geq 0$, and all eigenvalues of H are greater than or equal to zero (i.e. H is *positive semi-definite*).

Once we have computed the eigenvectors and eigenvalues of H, and the corresponding natural frequencies ω_j of each vibrational mode, we can "animate" each mode separately by plotting the positions of the lattice sites as the coordinate vector $\underline{q}$ changes in time according to the equation

$$\underline{q}(t) = \hat{\underline{q}} + a_j \sin(\omega_j t)\underline{w}^{[j]} \qquad \text{(EQ 34)}$$

a_j is an arbitrary amplitude of the vibration. To make the nature of the vibration clearly visible on a plot of the lattice positions as functions of time, we could choose $a_j = l_b/2$.

The MATLAB program **lattice_2D_vib.m** (posted on the 10.34 assignments web site) calculates for the potential energy function given above the normal modes from eigenvalue analysis of the Hessian matrix. This program stores the computed normal mode data in a .mat file. A second program **animate_2D_vib.m** reads this .mat file and "animates" a selected vibrational mode.

In lattice_2D_vib.m, you are asked to input the number N of sites per row or column in the lattice and whether to use add a random small number of each spring constant. This "randomization" is done to remove the degeneracy (many linearly independent modes with the same frequency) inherent to a regular 2-D lattice. Let us say that we have 10 different modes that all have the same frequency. We could take any linear combination of these modes to produce a new “mixture” of modes that is also an eigenvector with the same frequency. If we add a small random number individually to each spring constant, we remove this “mixing” between degenerate modes, and can animate more localized eigenvectors that are easier to interpret visually.

In addition, the user is asked whether to compute all eigenvalues/eigenvectors using the QR algorithm of **eig()**, or to compute a few largest or smallest eigenvalues using **eigs()**. Since the former method requires full matrix storage of the Hessian and $\sim N^3$ operations, this option is recommended for only systems with a relative small number of rows/columns of the lattice (say under 20).

<u>Your assignment is the following:</u>

1. First, use this program to compute the normal modes of a 3x3 lattice. Visualize the modes both with and without the randomization of the spring constants. With a small random number added to each spring constant, you should observe more localized versions of the individual modes. Visualize each zero-energy mode. Can you explain why they are not penalized by the energy model in terms of the stretching of the springs?

2. Compute the normal modes of a 10x10 lattice. First, compute the individual frequencies without randomization, and use the MATLAB **hist()** command to make a histogram of the frequencies (you will have to read in the .mat output file). Once you have identified the distinct frequencies, recompute the modes using randomization to “localize” the modes and visualize a mode of each distinct frequency. Print out the graph that shows the maximum amplitude offset of the lattice points and compare the natures of the low and high frequency modes.

You should be able to see that the lowest (non-zero) frequency modes involve long-ranged collective motion of many sites. Two sites that are close together are displaced at any one time by similar distances. By contrast, higher-frequency modes - those penalized more by the potential energy function - involve very different displacements among neighboring sites. This is a general phenomenon - the fastest-modes are of the shortest length scales and involve the local motion of atoms and the slowest-modes have the longest length scales and involve the collective motion of many atoms.

3. Plot as a function of the number of row(columns) in the system the values of the lowest and highest non-zero frequencies for lattices of size $N = 6, 7, 8, 9, 10$. Do not use randomization.

You merely have to perform these calculations, print out the graph, and discuss briefly your observations.

Optional discussion - finding the minimum energy state numerically

For the regular 2-D lattice, we know *a priori* the positions of each site in the minimum energy state. In general, this is not the case. We have some model that tells us the total energy and the gradient (the vector of first derivatives with respect to each degree of freedom). Perhaps also, we have some guess of what the minimum energy state should look like, but we must compute the positions of the atoms in the minimum energy state from our energy model. Only once we have identified the minimum energy state can we apply the normal mode analysis outlined above.

This is an example of an optimization problem. We have some vector of state variables,

$$\underline{x} = [x_1, x_2, \ldots, x_N]^T \tag{EQ 35}$$

and we want to find the values of these variables, $\hat{\underline{x}}$, that minimizes some specified cost function $F(\underline{x})$. In this case the state variables are the positions of each lattice site and the cost function is the total potential energy of the system. How do we find the minimum numerically?

Since at the minimum, we know that the gradient function, defined as

$$\underline{g}(\underline{x}) \equiv \underline{\nabla}F \tag{EQ 36}$$

is zero, we expect that we might be able to find the minimum by applying Newton's method to the system of equations

$$\underline{g}(\hat{\underline{x}}) = \underline{0} \tag{EQ 37}$$

We can in fact find the minimum using this approach, but we need to modify the algorithm slightly to make sure that we find a minimum of $F(\underline{x})$ as opposed to a maximum or a saddle point (see figure below).

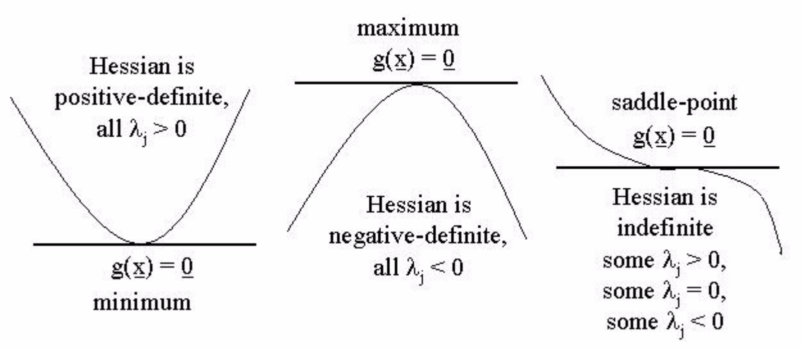

FIGURE 4. Characterization of extrema based on eigenvalues of Hessian

We assume that we have an model that allows us to calculate the value of the cost function and the gradient vector. Additionally, it would be nice to be able to calculate the Hessian analytically, but we can use finite differences to estimate it if this is not the case. If we make an initial guess of the minimum, $\underline{x}^{[0]}$, how do we generate a sequence of points $\underline{x}^{[1]}, \underline{x}^{[2]}, \underline{x}^{[3]}, \ldots$ that approach the minimum at $\underline{x} = \hat{\underline{x}}$?

We apply Newton's method to the system $\underline{g}(\hat{\underline{x}}) = \underline{0}$ to obtain the update rule,

$$\begin{aligned} H(\underline{x}^{[k]})\Delta\underline{x}^{[k]} &= -\underline{g}(\underline{x}^{[k]}) \\ \underline{x}^{[k+1]} &= \underline{x}^{[k]} + \alpha_k \Delta\underline{x}^{[k]} \end{aligned} \tag{EQ 38}$$

where we solve the linear system for $\Delta\underline{x}^{[k]}$, e.g. using Gaussian elimination, and then only take some fraction $\alpha_k > 0$ of this step such that the cost function is reduced (for example with the weak line search algorithm that we introduced for the reduced-step Newton algorithm). If we can ensure that the cost function is reduced at each step, we will end up at a minimum rather than a maximum, but how can we be sure that an appropriate $\alpha_k > 0$ exists such that $F(\underline{x}^{[k]} + \alpha_k \Delta\underline{x}^{[k]}) < F(\underline{x}^{[k]})$?

For very small values of α_k, we can use a Taylor series expansion to write

$$F(\underline{x}^{[k]} + \alpha_k \Delta\underline{x}^{[k]}) - F(\underline{x}^{[k]}) \approx \alpha_k \Delta\underline{x}^{[k]} \bullet \underline{g}(\underline{x}^{[k]}) \tag{EQ 39}$$

We use the Newton update equation to substitute for the gradient to obtain

$$F(\underline{x}^{[k]} + \alpha_k \Delta\underline{x}^{[k]}) - F(\underline{x}^{[k]}) \approx -\alpha_k \Delta\underline{x}^{[k]} \bullet H(\underline{x}^{[k]})\Delta\underline{x}^{[k]} \tag{EQ 40}$$

Therefore, as long as the Hessian is positive-definite, Newton's method will point in a "downhill" direction towards a minimum.

It is certainly possible that when our initial guess is far away from the local minimum that the Hessian is not positive-definite. When this happens we can modify the update equation by adding to the Hessian a scalar $\tau_k \geq 0$ times the identity matrix.

$$[H(\underline{x}^{[k]}) + \tau_k I]\Delta\underline{x}^{[k]} = -\underline{g}(\underline{x}^{[k]}) \tag{EQ 41}$$

We choose the value of τ_k to be large enough that we can ensure that the matrix $H(\underline{x}^{[k]}) + \tau_k I$ is positive-definite, for example by applying Gershogin's theorem.

Later we will use this technique of optimization to determine the value of parameters in nonlinear models that fit with experimental data.