CS1112 Fall 2015 Project 4 Part A due Monday 10/26 at 11pm

You must work either on your own or with one partner. If you work with a partner you must first register as a group in CMS and then submit your work as a group. *Adhere to the Code of Academic Integrity.* For a group, "you" below refers to "your group." You may discuss background issues and general strategies with others and seek help from the course staff, but the work that you submit must be your own. In particular, you may discuss general ideas with others but you may not work out the detailed solutions with others. It is not OK for you to see or hear another student's code and it is certainly not OK to copy code from another person or from published/Internet sources. If you feel that you cannot complete the assignment on you own, seek help from the course staff.

Objectives

Completing this project will solidify your understanding of 2-dimensional and 3-dimensional arrays. In Part A, you will practice working with matrices and will use MATLAB as a (black box) tool to solve a system of linear equations. In Part B, you will practice working with 2-d and 3-d arrays through image processing applications.

1 The Ising Model

Complete parts **(a)**, **(b)**, and **(c)** of Problem **P7.3.6** (p.176) in *Insight.* By initializing `A` and then repeatedly applying the function `Sweep`, you now have a simple implementation of the Ising Model.

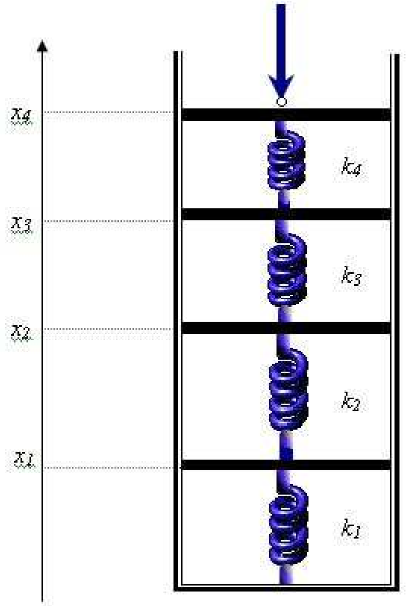
(d') Our model describes a system in which each cell tends to have the same state as its neighbors, subject to random fluctuations. And it turns out that these random fluctuations play a critical role in the long term behavior of the model. To explore this behavior, write a script `simIsing` to generate a visual representation of the simulation as it proceeds. Set `n` to be 50 and `p` to be 0.75; use built-in functions `pcolor`[1] and `pause` for the visualization. At the end of the script, provide a brief answer, in the form of program comments, to each of the following questions: (1) What is the longterm behavior of the model when `T`=0? When `T` is small? When `T` is large? (2) What values of `T` do you consider to be small/large? (3) In this model, `T` admits the physical interpretation of temperature. Are your observations consistent with such an interpretation? (Hint: How do the phases of matter relate to temperature?)

Submit your files `InitialIsing.m`, `Potential.m`, `Sweep.m`, and `simIsing.m` on CMS.

2 System of Linear Equations

Systems of linear equations often arise from the modeling of systems of interconnected elements. In your physics class you may have studied the displacement that results from applying a force to a set of blocks connected by springs. Such a model gives a coupled set of equations that must be solved simultaneously; the equations are coupled because the individual parts of the system are influenced by other parts.

[Our objective for this problem is to show how you can use MATLAB *as a solver of systems of linear equations. We do not expect you to know linear algebra nor are we trying to teach linear algebra. We will use the solver as a "black box," so your only real task is to turn a set of given equations into a matrix and a vector and then use the solver, which is just an operator. So there's no need to be afraid of the physics or math! This discussion and problem is adapted from Introduction to Computing for Engineers by Chapra and Canale.]*



---

[1] `pcolor(A)` gives a visualization of the values in matrix `A` by associating a color with an individual numeric value in `A`, see `Eg7_3` in *Insight.* For this problem, *do not* use the additional options `shading interp` and `caxis` as shown in `Eg7_3` since we do not want any interpolation—we have only two possible values in our matrix in this problem. If you want to learn more about `pcolor`, type `help pcolor` or `doc pcolor` at the MATLAB *Command Window.*

Consider a set of n linear algebraic equations of the general form

$$\begin{array}{ccccc} a_{11}x_1+ & a_{12}x_2+ & \ldots+ & a_{1n}x_n = & b_1 \\ a_{21}x_1+ & a_{22}x_2+ & \ldots+ & a_{2n}x_n = & b_2 \\ . & . & & . & . \\ . & . & & . & . \\ . & . & & . & . \\ a_{n1}x_1+ & a_{n2}x_2+ & \ldots+ & a_{nn}x_n = & b_n \end{array} \tag{1}$$

where the a's are known constant coefficients, the b's are known constants, and the n unknowns, x_1, x_2, …, x_n, are raised to the first power. This system of equations can be expressed in matrix notation as

$$\mathbf{Ax} = \mathbf{b}$$

or

$$\begin{bmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ . \\ . \\ . \\ b_n \end{bmatrix} \tag{2}$$

"Solving" this linear system of equations is to find the values x_1, x_2, …, x_n such that $\mathbf{Ax} = \mathbf{b}$. In MATLAB, the solution can be found using the backslash, called the *matrix left divide* operator:

```
x = A\b
```
(3)

where `A` is the n-by-n matrix of coefficients, `b` is the length n *column* vector of constants, and the result `x` is the length n column vector of values such that $\mathbf{Ax} = \mathbf{b}$.

Your job: Write a script `springSystem.m` to find the positions x_1, x_2, x_3, and x_4 of the spring-mass system shown above when a force F of 2000 lbs is applied. The spring constants k_1 through k_4 are 100, 50, 75, and 200 lb/in, respectively. The force-balance equations that define the relationships among the springs are

$$\begin{aligned} k_2(x_2 - x_1) &= k_1 x_1 \\ k_3(x_3 - x_2) &= k_2(x_2 - x_1) \\ k_4(x_4 - x_3) &= k_3(x_3 - x_2) \\ F &= k_4(x_4 - x_3) \end{aligned}$$

You need to

1. Rewrite these equations in the general form of (1) by collecting terms together, where the unknowns are x_1, x_2, x_3, and x_4.
2. Create the 4-by-4 matrix `A` and length 4 column vector `b` as shown in (2).
3. Apply the matrix left division operator as shown in (3) to solve for x_1, x_2, x_3, and x_4.
4. Finally display the values of x_1 through x_4 neatly.

Submit your file `springSystem.m` on CMS.

Part B of Project 4 will appear in a separate document. Both Parts A and B have the same due date.