ECE4390 Lab 1

1. The 1D wave equation, $\frac{\partial^2 \underline{u}(x,t)}{\partial x^2} - \frac{1}{c^2}\frac{\partial^2 \underline{u}(x,t)}{\partial t^2} = 0$, describes the motion of a vibrating string. It may be solved with initial conditions by discretizing space and iterating the following equation over time:

$$\underline{u}^{n+1} = 2\underline{u}^n - \underline{u}^{n-1} + \frac{\Delta t^2 c^2}{\Delta x^2} A \underline{u}^n.$$

Here, A describes the spatial interactions of the system and is a tridiagonal matrix who's diagonal entries are $[1, -2, 1]$. The time step discretizations are denoted by the subscripts on $\underline{u}$. For this question we will consider a string of length $l = 10$ (ie. domain is $L = [0, 10] \subset \mathcal{R}$), a time period of 20 seconds and a constant $c = 1$. The initial conditions are as follows: $\underline{u}^{-1}(x) = \underline{u}^0(x) = h\sin(\frac{x\pi}{l})$, $x \in L$, $h = 2.5$, this is to say that the system starts with the string already pulled-up a height h.

a) For $\Delta x = 0.05$ and $\Delta t = 0.05$ solve for $\underline{u}(x)$ ($x \in L$) at each time step. Plot the results with respect to x at each time step. One way of doing so is to create a MATLAB `movie()`. Another is to plot each time step to the same MATLAB figure with a time delay (using `pause()`) between each plot. Try solving the system with different values for Δx and Δt.

b) Implement A as a sparse matrix using the `sparse()` or `spdiags()` functions. Compare program memory usage between the dense and sparse cases using the `whos` command at the MATLAB console.

2. Write a function that returns data structures from an input text file containing circuit information (suggested MATLAB I/O functions: `fopen(), fclose(), fgetl(), sscanf()`). For the moment, only five circuit elements will be considered. They are the resistor (R), inductor (L), capacitor (C), voltage source (V) and current source (I). Their description in the file will have space delimited data in the following format:

Type Name Node1 Node2 Value

For example, a text file for a simple 3-node RC-circuit with a 5V source, 100Ω resistor and a 1nF capacitor would look like the following:

V Source1 10 20 5
R Resistor1 20 30 100
C Capacitor1 30 10 1e-9

Try drawing this example circuit.

3. Gmsh is a finite element mesh generator software application that will be used in this course. The program is available under the GNU General Public License at http://geuz.org/gmsh/. Documentation is also provided at the website.

a) A Gmsh geometry file and associated mesh file have been provided. They are named “mesh.geo” and “mesh.msh” respectively. The geometry file contains point, line and physics number directives that define a geometry in the Gmsh context. Try experimenting with the geometry by modifying the directives. Try using Gmsh to create a mesh over this geometry and save the result to a file. The mesh file contains node, line and element information pertaining to the mesh. Open the mesh file in a text editor and gain an understanding of the formatting of the data.

b) A MATLAB function, “ReadGmsh.m” that returns data structures from a Gmsh mesh ASCII input file (version 2.0) in a single MATLAB “struct” has been provided. Relevant data structures are: 1. A table of xy-coordinates for each node; 2. A table of node indices for each element; 3. A list of physics numbers for each element. Assume that the mesh is two dimensional. Read the mesh file with the given program and examine the data that is returned.

c) Plot the mesh in MATLAB using the `patch()` and `imagesc()` commands. Have the plots differentiate between the regions of different physics numbers.