Math 307: Problems for section 1.3

1. **Write down the vector approximating $f''(x)$ at interior points, the vector approximating $xf(x)$ at interior points, and the finite difference matrix equation for the finite difference approximation with $N = 4$ for the differential equation**

$$f''(x) + xf(x) = 0$$

for $1 \le x \le 3$ subject to

$$f(1) = 1, \quad f(3) = -1.$$

2. **Write down the matrix equation to solve in order to find the finite difference approximation with $N = 4$ for the same differential equation**

$$f''(x) + xf(x) = 0$$

for $1 \le x \le 3$ but now subject to

$$f'(1) = 1, \quad f(3) = -1$$

3. **Use MATLAB/Octave to solve the matrix equations you derived in the last two problems for the vector F that approximates the solution (i.e., with $N = 4$). Then redo the calculation with $N = 50$ and plot the resulting functions.**

   Questions 4–6 deal with the steady heat equation in a one-dimensional rod considered in the notes:

$$0 = kT''(x) - HT(x) + S(x),$$

   where k and H are constants, subject to the boundary conditions

$$T = T_l \text{ at } x = x_l \text{ and } T = T_r \text{ at } x = x_r.$$

   The MATLAB/Octave commands needed to find the finite difference approximation for $T(x)$ in the case $k = 1,\ H = 0,\ S(x) = 1,\ T_l = T_r = 1,\ x_l = 0$ and $x_r = 1$ are provided in `heat.m`.

4. **Modify the commands provided in `heat.m` to calculate the temperature profile in a rod cooled by the air in the case $k = 1$, $H = 1$, $S(x) = 1$, $T_l = 0$, $T_r = 2$, $x_l = -1$ and $x_r = 1$. Describe briefly the modifications made, and hand in a plot of the solution for $n = 50$.**

5. **For the case given in Q4, compute the finite difference approximation at $x = -0.5$ for $n = 4$, 40 and 400. The true solution at this point is $1 - \sinh 0.5/\sinh 1$. Make a log-log plot of the magnitude of the error in the finite difference approximation against Δx. What is the approximate slope of this curve?**

6. **The boundary condition $T'(x) = 0$ at $x = x_l$ or $x = x_r$ describes an insulating end to the rod. Write down an approximation for $T'(x_l)$ using T_0 and T_1. Also write down an approximation for $T'(x_r)$ using T_{n-1} and T_n. Find the modification needed to the matrix equation if insulating boundary conditions are placed at $x = x_l$ and $x = x_r$ (you should find that two rows of the matrix change and two entries of the vector on the right-hand-side change). Modify the commands provided in `heat.m` to calculate the temperature profile in a heated rod in the case $k = 1$, $H = 0$, $S(x) = 1$, with insulating boundary conditions at $x = 0$ and $x = 1$ (representing a continuously heated rod from which no heat escapes). Try to find the solution for $n = 10$. Is the solution reasonable?**

7. **In this problem we will use finite differences to solve Laplace's equation on a square. We want to approximate the solution $f(x, y)$ to the partial differential equation (Laplace's equation)**

$$f_{xx}(x, y) + f_{yy}(x, y) = 0 \quad 0 \le x \le 1, \quad 0 \le y \le 1$$

subject to the boundary conditions

$$\begin{aligned} f(x, 0) &= a_1(x) \quad 0 \le x \le 1 \\ f(0, y) &= a_2(y) \quad 0 \le y \le 1 \\ f(x, 1) &= a_3(x) \quad 0 \le x \le 1 \\ f(1, y) &= a_4(y) \quad 0 \le y \le 1 \end{aligned}$$

You can think of $f(x, y)$ as the shape (i.e., the height) of a streched rubber membrane attached along the edges of a square to a wire described by the four known functions $a_1, \ldots, a_4$.

Pick N equally spaced points $x_k = k/N$ and $y_k = k/N$, $k = 0, \ldots N$ along the x and y axes with spacing $\Delta x = \Delta y = (1/N)$. Then, in our equations the unknown function $f(x, y)$ will be replaced by a grid of unknown values $f_{i,j}$ with $i = 0, \ldots N$ and $j = 0, \ldots N$ with the idea that $f(x_i, y_j) \sim f_{i,j}$.

For interior points (x_i, y_j) (i.e., $1 \le i \le N - 1$ and $1 \le j \le N - 1$) we can write down approximations to the second partial derivatives using the formula we derived for single variable B.V.P.:

$$\begin{aligned} f_{x,x}(x_i, y_j) &\sim (\Delta x)^{-2} (f_{i+1,j} - 2f_{i,j} + f_{i-1,j}) \\ f_{y,y}(x_i, y_j) &\sim (\Delta y)^{-2} (f_{i,j+1} - 2f_{i,j} + f_{i,j-1}) \end{aligned}$$

Adding these together and setting the result to zero is the discrete analogue of Laplace's equation. This gives a linear equation in the unknowns $f_{i,j}$ for each interior point:

$$f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4f_{i,j} = 0. \tag{1}$$

Here we have cancelled a factor of $N^2 = (\Delta x)^{-2} = (\Delta y)^{-2}$ from each side of the equation.

Next we set the boundary values. This is done with the equations

$$\begin{aligned} f_{i,0} &= a_1(x_i), \quad i = 0, \ldots N && (2) \\ f_{0,j} &= a_2(y_j), \quad j = 0, \ldots N && (3) \\ f_{i,N} &= a_3(x_i), \quad i = 0, \ldots N && (4) \\ f_{N,j} &= a_4(y_j), \quad j = 0, \ldots N && (5) \\ & && (6) \end{aligned}$$

In total this gives $(N+1)^2$ equations in $(N+1)^2$ unknowns $f_{i,j}$.

The only real difficulty in writing this system down as a matrix equation is that the unknowns $f_{i,j}$ are indexed by a double index. To write the matrix equation we need to number the unknowns by a single index. In other words we want a vector $F = [F_1, F_2, \ldots, F_{(N+1)^2}]^T$ that contains the $f_{i,j}$'s in some order. Then, we can write the equation in the form $LF = Y$ for some matrix L and vector Y. If we order the $f_{i,j}$'s by starting on the bottom row and working our way up, i.e.,

$$\begin{bmatrix} f_{0,N} & f_{1,N} & f_{2,N} & \cdots & f_{N,N} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ f_{0,2} & f_{1,2} & f_{2,2} & \cdots & f_{N,2} \\ f_{0,1} & f_{1,1} & f_{2,1} & \cdots & f_{N,1} \\ f_{0,0} & f_{1,0} & f_{2,0} & \cdots & f_{N,0} \end{bmatrix} = \begin{bmatrix} F_{N(N+1)+1} & F_{N(N+1)+2} & F_{N(N+1)+3} & \cdots & F_{(N+1)^2} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ F_{2(N+1)+1} & F_{2(N+1)+2} & F_{2(N+1)+3} & \cdots & F_{3(N+1)} \\ F_{(N+1)+1} & F_{(N+1)+2} & F_{(N+1)+3} & \cdots & F_{2(N+1)} \\ F_1 & F_2 & F_3 & \cdots & F_{N+1} \end{bmatrix}$$

then we see that

$$f_{i,j} = F_{\kappa(i,j)}$$

where $\kappa(i,j) = j(N+1) + i + 1$ is a re-indexing function. Now the equations 1 for each interior point (i,j) correspond to the $\kappa(i,j)$th row in our equation $LF = Y$ and can be written

$$F_{\kappa(i+1,j)} + F_{\kappa(i-1,j)} + F_{\kappa(i,j+1)} + F_{\kappa(i,j-1)} - 4F_{\kappa(i,j)} = 0$$

In other words, the $\kappa(i,j)$th row in the matrix L has a 1 in the $\kappa(i+1,j)$, $\kappa(i-1,j)$, $\kappa(i,j+1)$ and $\kappa(i,j-1)$ spots, and a -4 in the $\kappa(i,j)$ spot. The vector Y has a 0 in the $\kappa(i,j)$ spot.

Similarly, the equations 2 for each boundary point also correspond to a row in our equation $LF = b$. These equations can be written

$$\begin{aligned} F_{\kappa(i,0)} &= a_1(x_i), \quad i = 0, \ldots N \\ F_{\kappa(0,j)} &= a_2(y_j), \quad j = 0, \ldots N \\ F_{\kappa(i,N)} &= a_3(x_i), \quad i = 0, \ldots N \\ F_{\kappa(N,j)} &= a_4(y_j), \quad j = 0, \ldots N \end{aligned}$$

from which the entries in L and Y can be deduced.

Here are MATLAB/Octave commands to implement this procedure when

$$\begin{aligned} a_1(x) &= \sin(\pi x) \\ a_2(y) &= 0 \\ a_3(x) &= 0 \\ a_4(y) &= 0. \end{aligned}$$

These commands can be found in the file `laplaceeqn.m`

First we choose `N`**, initialize the matrix** `L` **and the vector** `Y`**, and set** `X` **to the vector of $N+1$ equally spaced points between 0 and 1.**

```
N=30
L=zeros((N+1)^2,(N+1)^2);
Y=zeros((N+1)^2,1);
X=linspace(0,1,(N+1));
```

Next we define the re-indexing function that converts the double index (i, j) into the single index $\kappa(i, j) = i(N + 1) + j + 1$. In Octave, as long as a function is not the first thing in a .m file, the filename does not have to match the function name.

In MATLAB you have to take the following three lines and put them in a separate file called `k.m`

```
function k=k(i,j,N)
    k=j*(N+1)+i+1;
end
```

Now we define the parts of the matrix L and vector Y that correspond to the boundary conditions along the sides of the square.

```
for n=0:N
    L(k(0,n,N),k(0,n,N))=1;
    Y(k(0,n,N))=0;
    L(k(N,n,N),k(N,n,N))=1;
    Y(k(N,n,N))=0;
end
```

Next we define the parts of the matrix L and vector Y that correspond to the boundary conditions along the bottom and top of the square.

```
for n=1:N-1
    L(k(n,0,N),k(n,0,N))=1;
    Y(k(n,0,N))=sin(pi*X(n+1));
    L(k(n,N,N),k(n,N,N))=1;
    Y(k(n,N,N))=0;
end;
```

Finally we define the parts of the matrix L and vector Y that correspond to the equations for the interior points.

```
for i=1:N-1
    for j=1:N-1
        L(k(i,j,N),k(i,j,N))=-4;
        L(k(i,j,N),k(i+1,j,N))=1;
        L(k(i,j,N),k(i-1,j,N))=1;
        L(k(i,j,N),k(i,j+1,N))=1;
        L(k(i,j,N),k(i,j-1,N))=1;
    end
end
```

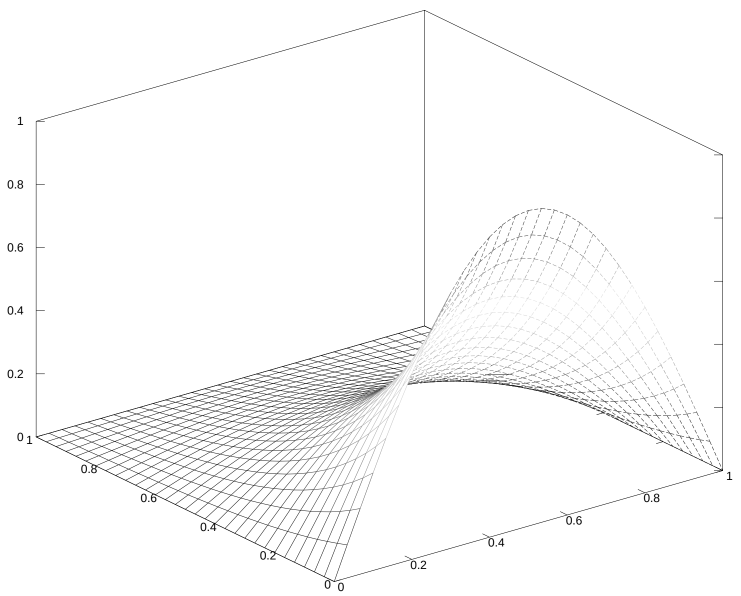
Now we can solve the equation for F and plot the result. To do this we have to put the `F` **values in a two dimensional grid** `FF` **and use the** `mesh` **command to do the 3-d plot. If** `X` **and** `Y` **are vectors of length n and** `Z` **is an nxn matrix then** `mesh(X,Y,Z)` **plots the points** `[X(j),Y(i),Z(i,j)]`**.**

```
F=L\Y;
```

```
FF=zeros(N+1,N+1);
for i=0:N
    for j=0:N
        FF(j+1,i+1)=F(k(i,j,N));
    end
end

mesh(X,X,FF);
```

We can print out the resulting graph using `print laplace1.pdf` **(or** `print laplace1.jpg` **or** `print laplace1.eps`**). This will produce a pdf file (or jpg or eps file) containing the graph. Here is the result:**



Run the file `laplaceeqn.m` **to produce this picture. Then modify the code to solve Laplace's equation with boundary conditions:**

$$f(x,0) = \sin(\pi x) \quad 0 \le x \le 1$$
$$f(0,y) = 0 \quad 0 \le y \le 1$$
$$f(x,1) = \sin(\pi x) \quad 0 \le x \le 1$$
$$f(1,y) = 0 \quad 0 \le y \le 1$$

Say what code you modified, and hand in the resulting picture. Finally modify the code

to solve Laplace's equation with boundary conditions:

$$\begin{aligned} f(x,0) &= \sin(\pi x) \quad 0 \le x \le 1 \\ f(0,y) &= 0 \quad 0 \le y \le 1 \\ f_y(x,1) &= 0 \quad 0 \le x \le 1 \\ f(1,y) &= 0 \quad 0 \le y \le 1 \end{aligned}$$

The third boundary condition is called a Neumann boundary condition, and corresponds to detaching the rubber membrane from the wire along the top boundary. Again, say what code you modified, and hand in the resulting picture.