Assignment 4

November 9, 2023 2:51 PM

SOLUTIONS BELOW QUESTIONS

Aaryan Nagpal
1007792596
nagpalaa

ESC384 Assignment 4

Due Wednesday, 22 November 2023, at 9:10am

The solution to the assignment must be uploaded to Quercus as a single PDF file at the specified time. For problems that require coding, please include a copy of the code in the aforementioned PDF file and also upload the source code as a single ZIP file to facilitate the grading process. In summary, there should be two separate files uploaded to Quercus: (i) a single PDF file with the entire solution; (ii) a single ZIP file with all the source code. Everything that you would like to have marked should be in the PDF file; TAs will nominally only look at the PDF file (and not the ZIP file). Finally, please adhere to the collaboration policy: the final write up must be prepared individually without consulting others. (See the syllabus for details.)

Problem 1. Heat equation: finite difference (50%)

Consider the heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} &= f \quad \text{in } (0,1)\times(0,T], \\ u(x=0,t) &= 0 \quad \text{on } \{x=0\}\times(0,T], \\ \frac{\partial u}{\partial x}(x=1,t) + u(x=1,t) &= 0 \quad \text{on } \{x=1\}\times(0,T], \\ u &= g \quad \text{on } (0,1)\times\{t=0\} \end{aligned}$$

for some initial condition function $g:(0,1)\to\mathbb{R}$ and source function $f:(0,1)\times(0,T]\to\mathbb{R}$. Note that a Robin boundary condition is imposed on the right boundary.

We will implement a finite difference heat equation solver in MATLAB. We will use $n+1$ grid points $0 \equiv x_0 < x_1 < \cdots < x_n \equiv 1$ to discretize the spatial domain, and $J+1$ time points $0 \equiv t^0 < t^1 \cdots < t^J = T$ to discretize the time interval. We assume both spatial and temporal points are equispaced. Our goal is to approximate the solution $u(x_i, t^j)$ for $i = 1,\dots,n$ and $j = 1,\dots,J$.

We first consider the semi-discrete form of the equation associated with the second-order accurate finite difference approximation in space.

(a) (10%) Find the semi-discrete equations for (i) the first unknown node $i=1$, (ii) the last unknown node $i=n$, and (iii) all other unknown nodes $i\in[2,n-1]$. Also identify the equation for (iv) the initial condition. Express the answer in terms of $\tilde{x}_i$, $\tilde{u}_i(t)$, $\tilde{f}_i(t) \equiv f(x_i,t)$, and $\tilde{g}_i \equiv g(x_i)$.

(b) (6%) The semi-discrete equations found in (a) can be expressed as

$$\frac{d\tilde{u}}{dt}(t) + \hat{A}\tilde{u}(t) = \hat{f}(t) \quad \text{in } \mathbb{R}^n,$$
$$\tilde{u}(t=0) = \hat{g} \quad \text{in } \mathbb{R}^n,$$

where $\tilde{u}(t)\in\mathbb{R}^n$, $\hat{A}\in\mathbb{R}^{n\times n}$, $\hat{f}(t)\in\mathbb{R}^n$, and $\hat{g}\in\mathbb{R}^n$. Find the expressions for the matrix $\hat{A}$ and vectors $\hat{f}(t)$ and $\hat{g}$.

1

We now consider the fully discrete form of the equation associated with the Crank-Nicolson approximation in time.

(c) (6%) The fully discrete equation associated with the semi-discrete equation found in (b) can be expressed as

$$\hat{C}\tilde{u}^j = \hat{D}\tilde{u}^{j-1} + \hat{F}(t^j, t^{j-1}), \quad j = 1,\dots,J,$$

where $\hat{C}\in\mathbb{R}^{n\times n}$, $\hat{D}\in\mathbb{R}^{n\times n}$, and $\hat{F}(t^j,t^{j-1})\in\mathbb{R}^n$. Find the expressions for the matrices and vectors.

We now implement the finite difference solver in MATLAB.

(d) (15%) Starting with the template `heat_fd_temp.m`, implement the finite difference solver.

Note. Please include (i) a copy of the code in the PDF file and (ii) the source code in the zip file to facilitate the grading process.

Note 2. You do *not* have to use the template if you would rather code everything from scratch.

(e) (5%) Let

$$f(x,t) = \exp(-3x)\exp(t),$$
$$g(x) = \frac{1}{2}x(2-x) + \frac{1}{6\pi}\sin(3\pi x).$$

Invoke the solver for $T=1$, $n=16$, $J=16$. Plot, in a single figure, the solution at the final time $t=0$, $t=1/16$, $t=1/8$, $t=1/4$, $1/2$, and $t=1$.

(f) (8%) We wish to verify the convergence of the solver. To this end, for the functions f and g given in (e), compute the solution for $(n,J) = (8,8), (16,16), (32,32)$, and $(64,64)$, and then evaluate (the approximation of) the output $s \equiv u(x=1,t=1)$ for each of the four discretizations. Also evaluate the reference output s_{ref} associated with $(n,J) = (512,512)$. Report, in a table, the error associated with the four different levels of discretizations. Report also the value of s_{ref} to at least six significant digits. Does the observed error behavior match your expectation?

Note. The table should have three columns with headings n, J, and $|s_{ref} - s|$. Please provide both the table and the value of s_{ref} in the hard copy of the assignment to facilitate the grading process.

Problem 2. Laplace's equation in an annular domain (30%)

Let Ω be an two-dimensional annular domain of the inner radius r_1 and the outer radius of r_2: i.e.,

$$\Omega \equiv \{x\in\mathbb{R}^2 \mid r_1 < \|x\| < r_2\}.$$

We consider Laplace's equation

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega, \\ u(r=r_1,\theta) &= 0, \quad \theta\in[0,2\pi), \\ u(r=r_2,\theta) &= g(\theta), \quad \theta\in[0,2\pi). \end{aligned}$$

In words, the homogeneous Dirichlet boundary condition is imposed on the inner boundary, and a nonhomogeneous Dirichlet boundary condition is imposed on the outer boundary. Answer the following questions:

2

(a) (14%) Find a family of functions of the form $u_m(r,\theta) = r_m(r)\Theta_m(\theta)$ that satisfies (i) the PDE and (ii) the inner boundary condition (but not necessarily the outer boundary condition). Express the answer in terms of r, θ, r_1, r_2, and some unknown coefficients.

(b) (8%) Let $g(\theta) = \cos(m\theta)$, where m is a positive integer. Find the associated solution u as a function of m.

Note. The integer m is positive; i.e., it does not take on 0.

(c) (8%) Let $r_1 = 1$ and $r_2 = 2$. Using MATLAB (or any other software), plot, in a single figure, $u(r,\theta=0)$ over $r\in[r_1,r_2]$ for $m=1$ and 10. Comment (i) if the relative decay in the two solutions away from the outer boundary is consistent with your expectation and (ii) if the maximum principle is satisfied.

Problem 3. Green's function (20%)

Consider Laplace's equation in the first quadrant $\Omega = \mathbb{R}_{>0}\times\mathbb{R}_{>0}$ with the bottom boundary $\Gamma_B \equiv \mathbb{R}_{>0}\times\{x_2=0\}$ and the left boundary $\Gamma_L \equiv \{x_1=0\}\times\mathbb{R}_{>0}$:

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega, \\ u &= g_B \quad \text{on } \Gamma_B, \\ u &= g_L \quad \text{on } \Gamma_L. \end{aligned}$$

Answer the following questions.

(a) (10%) Find the Green's function $G:\Omega\times\Omega\to\mathbb{R}$ such that, for any given $\xi\in\Omega$,

$$\begin{aligned} -\Delta G(x,\xi) &= \delta(x-\xi) \quad \forall x\in\Omega, \\ G(x,\xi) &= 0 \quad \forall x\in\Gamma_B\cup\Gamma_L. \end{aligned}$$

Identify an appropriate number of image points, and express them as $\xi^{(1)} = (\xi_1^{(1)},\xi_2^{(1)})$, $\xi^{(2)} = (\xi_1^{(2)},\xi_2^{(2)})$, ... in terms of $\xi = (\xi_1,\xi_2)$. Then express the solution in terms of the fundamental solution Φ of Laplace's equation and the image points.

(b) (10%) Let $g_L = 0$. Find an integral expression for the solution $u(x = (x_1,x_2))$ in terms of g_B.

Hint. Note that the outward directional derivative on Γ_B is $\nu_B\cdot\nabla_\xi G(x,\xi) = -\frac{\partial G(x,\xi)}{\partial\xi_2}$. Also note that $-\frac{\partial\Phi(x-\xi)}{\partial\xi_2} = \frac{\partial}{\partial\xi_2}\left(\frac{1}{2\pi}\log|\sqrt{(x_1-\xi_1)^2+(x_2-\xi_2)^2}|\right)$.

Hint 2. The final solution can be succinctly expressed in terms of Poisson's kernel for the upper half plane.

(c) (5%) Let $g_L = 0$, and

$$g_B(x_1) = \begin{cases} 1, & x_1\in(0,1), \\ 0, & \text{otherwise.} \end{cases}$$

Find an integral representation of the solution. The solution may contain only an integral over a finite interval.

3

Aaryan Nagpal
1007792596
nagpalaa

ESC384 PDE's Assignment 4 SOLUTIONS

Q1

a) PDE: $\frac{\partial u}{\partial t}(x,t) - \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t)$

i) Semi-discrete: $\frac{d\tilde{u}_i(t)}{dt} - \frac{\tilde{u}_{i+1}(t) - 2\tilde{u}_i(t) + \tilde{u}_{i-1}(t)}{\Delta x^2} = \hat{f}_i(t)$

Left Dirichlet B.C: • want $\tilde{u}_0(t) = u(x=0,t) = 0$

• substitute BC into the semi-discrete equation at $i=1$.

$$\frac{d\tilde{u}_1(t)}{dt} - \frac{1}{\Delta x^2}\left(\tilde{u}_2(t) - 2\tilde{u}_1(t) - \underbrace{\tilde{u}_0(t)}_{=0}\right) = \hat{f}_1(t)$$

semi-discrete equation for node $i=1$.

$$\Rightarrow \boxed{\frac{d\tilde{u}_1(t)}{dt} - \frac{1}{\Delta x^2}\left(\tilde{u}_2(t) - 2\tilde{u}_1(t)\right) = \hat{f}_1(t)}$$

this is the semi-discrete equation for node $i=1$.

ii) Right Robin B.C: • want $\tilde{u}_n(t) = \frac{\partial u}{\partial x}(x=1,t) + u(x=1,t) = 0$

• allows us to apply finite difference formula at $i=n$.

$$\Rightarrow \frac{\tilde{u}_{n+1} - \tilde{u}_{n-1}}{2\Delta x} + \tilde{u}_n = 0 \quad \Rightarrow \quad \tilde{u}_{n+1} = -\tilde{u}_n(2\Delta x) + \tilde{u}_{n-1}$$

Substitute into semi-discrete equation at $i=n$

$$\frac{d\tilde{u}_n}{dt} - \frac{1}{\Delta x^2}\left(\tilde{u}_{n+1} - 2\tilde{u}_n + \tilde{u}_{n-1}\right) = \hat{f}_n(t)$$

↳ found above

$$\frac{d\tilde{u}_n}{dt} - \frac{1}{\Delta x^2}\left((\tilde{u}_{n-1} - \tilde{u}_n(2\Delta x)) - 2\tilde{u}_n + \tilde{u}_{n-1}\right) = \hat{f}_n(t)$$

$$\boxed{\frac{d\tilde{u}_n}{dt} - \frac{1}{\Delta x^2}\left(2\tilde{u}_{n-1} - 2\tilde{u}_n\right) + \frac{1}{\Delta x}\left(2\tilde{u}_n\right) = \hat{f}_n(t)}$$

↳ Semi-discrete equation for node $i=n$.

this is the semi-discrete equation for node $i=n$.

iii) now we can apply finite difference formula at any $i\in[2,n-1]$

$$\Rightarrow \frac{\tilde{u}_{i+1} - \tilde{u}_{i-1}}{2\Delta x} + \tilde{u}_i = 0 \quad \Rightarrow \quad \tilde{u}_{i+1} = -\tilde{u}_i(2\Delta x) + \tilde{u}_{i-1}$$

Substitute into semi-discrete equation for $i\in[2,n-1]$ and we end with

$$\boxed{\frac{d\tilde{u}_i}{dt} - \frac{1}{\Delta x^2}\left(\tilde{u}_{i+1} - 2\tilde{u}_i + \tilde{u}_{i-1}\right) = \hat{f}_i(t)}$$

↳ Semi-discrete equation for unknown nodes $i\in[2,n-1]$

this is the semi-discrete equation for all other unknown nodes $i\in[2,n-1]$

iv) Lastly, we can express the equation for the initial condition as : $\hat{F}_i(t) = g(x_i)$, but we are given that $g(x_i) \equiv \hat{g}_i$

where $\hat{g}_i$ is the value of $u_i(t=0) = \hat{u}_i(0)$

So we finally have $\boxed{\hat{F}_i(t) = \hat{g}_i = \hat{u}_i(0) \quad \forall\, i \in [1,n]}$ $\hookrightarrow$ equation for I.C.

b) Semi-discrete equation can be expressed as

$$\frac{d\tilde{u}}{dt}(t) + \hat{A}\hat{u}(t) = \hat{F}(t) \quad \text{in } \mathbb{R}^n,$$

$$\tilde{u}(t=0) = \hat{g} \quad \text{in } \mathbb{R}^n$$

So matrix $\hat{A}$ can be written as,

$$\hat{A} = \frac{1}{\Delta x^2}\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -2 & 2+2\Delta x \end{bmatrix}$$

dependent on left B.C

$-1 \; 2 \; 1$ sequence shifts to right by 1 each time.

dependent on Right B.C

$\hookrightarrow$ From the $\frac{1}{\Delta x}(2\hat{u}_n)$ term.

now, $\hat{F}(t)$ and $\hat{g}$ vectors can be expressed as

$$\hat{F}(t) = \begin{bmatrix} \hat{F}_1(t) \\ \hat{F}_2(t) \\ \vdots \\ \hat{F}_n(t) \end{bmatrix}, \quad \hat{g} = \begin{bmatrix} \hat{g}_1 \\ \hat{g}_2 \\ \vdots \\ \hat{g}_n \end{bmatrix}$$

$\hookrightarrow$ expressions for matrix $\hat{A}$, vectors $\hat{F}(t)$, $\hat{g}$

c) Clark Nicolson approximation in time for fully discrete eqn

$\rightarrow C\hat{\tilde{u}}^j = D\hat{\tilde{u}}^{j-1} + \hat{F}(t^j, t^{j-1})$, $j = 0,1,2,3,\ldots,J$, where $\tilde{u}(t^j) = \hat{\tilde{u}}^j$

In the C-N method, we can express the semi-discrete equation as

$$\frac{\hat{\tilde{u}}^j - \hat{\tilde{u}}^{j-1}}{\Delta t} = \frac{1}{2}\left(F(\hat{\tilde{u}}^j, t^j) + F(\hat{\tilde{u}}^{j-1}, t^{j-1})\right) \quad \text{in } \mathbb{R}^n,$$

$\hookrightarrow$ avg of the derivatives

Now, we know that

$\Rightarrow F(\tilde{u}, t) = \frac{d^2\hat{\tilde{u}}}{dx^2}(t) + \hat{F}(t)$, and so we can use the $\hat{A}$ matrix to express this as done below

$$\Rightarrow \frac{\hat{\tilde{u}}^j - \hat{\tilde{u}}^{j-1}}{\Delta t} = \frac{1}{2}\left(-\hat{A}\hat{\tilde{u}}^j + \hat{F}(t)^j - \hat{A}\hat{\tilde{u}}^{j-1} + \hat{F}(t)^{j-1}\right)$$

Rearranging and using the identity matrix 'I' we can write this as

$$\Rightarrow \left(\frac{I}{\Delta t} + \frac{1}{2}\hat{A}\right)\hat{\tilde{u}}^j = \left(\frac{I}{\Delta t} - \frac{1}{2}\hat{A}\right)\hat{\tilde{u}}^{j-1} + \frac{1}{2}\left(\hat{F}(t)^j + \hat{F}(t)^{j-1}\right)$$

Now, if we compare this to the given fully discrete

$$\left(\frac{I}{\Delta t} + \frac{1}{2}\hat{A}\right)\hat{\tilde{u}}^j = \left(\frac{I}{\Delta t} - \frac{1}{2}\hat{A}\right)\hat{\tilde{u}}^{j-1} + \frac{1}{2}\left(\hat{F}(t)^j + \hat{F}(t)^{j-1}\right) \Rightarrow \hat{C}\hat{\tilde{u}}^j = D\hat{\tilde{u}}^{j-1} + F(t^j, t^{j-1})$$

we have the expressions for matrices C, D and vector F by comparison as

$$\Rightarrow \boxed{\hat{C} = \left(\frac{I}{\Delta t} + \frac{1}{2}\hat{A}\right), \quad \hat{D} = \left(\frac{I}{\Delta t} - \frac{1}{2}\hat{A}\right), \quad \text{and } \hat{F}(t^j, t^{j-1}) = \frac{1}{2}\left(\hat{F}(t)^j + \hat{F}(t)^{j-1}\right)}$$

$\rightarrow$ expressions for C, D matrices and F vector

d) Code over here for heat_fd_temp

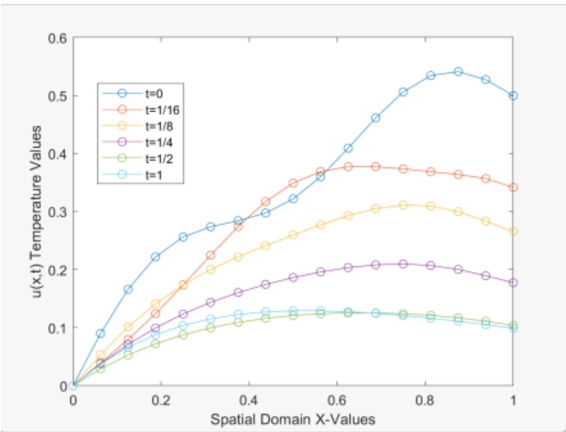
Question 1, part d heat_fd_temp code

```
function heat_fd_temp
% functions
gfun = @(x) 0.5*x.*(2-x) + 1/(6*pi)*sin(3*pi*x);
ffun = @(x,t) exp(-3*x).*exp(t);
% discretization parameters
n = 16;
J = 16;
T = 1; %over the time range (0,1) as given in the question
% setup spatial and temporal grids
xx = linspace(0,1,n+1)';
tt = linspace(0,T,J+1)';  %equally spaced points in the time domain
%discretization
dx = 1/n;
dt = T/J; %discretizing in the time domain over the given parameter
% assemble the A matrix
% (not the most effcient method, but it will work)
A = sparse([],[],[],n,n);
for i = 1:n
    if (i == 1)
        % left boundary
        A(i,i) = 2;  %obtained from first element in first row of A matrix
        A(i,i+1) = -1; %obtained from second element in first row of A matrix
    elseif (i == n)
        % right boundary
        A(i,i-1) = -2; %obtained from second last element in last row of A matrix
        A(i,i) = 2+2*dx; %obtained from last element in last row of A matrix
    else
        % all other points
        % all these points are obtained from the 3 shifting elements in
        %each row of the A matrix except for the first and last rows which
        %represent the boundary conditions
        A(i,i-1) = -1;
        A(i,i) = 2;
        A(i,i+1) = -1;
    end
end
%In my expression for the A matrix I also have a 1/dx^2 term in front of
%it, so I am adding that here for it to be consistent, once again using
%element wise division as A is a matrix!
A=A./dx^2;
% iteration
I = speye(n,n);
% assemble C and D matrices in terms of A, I, and dt
  %need to use elementwise division and multiplication as they are vector
  %representations of C and D
C = ((I./dt)+(1/2).*A);  %found in Q1c using C-N method
D = ((I./dt)-(1/2).*A);  %found in Q1c using C-N method
% initialize state
xxi = xx(2:end);  % x excluding the left node
U = gfun(xxi);
% record the initial state in UU; note the first node is left as 0
UU = zeros(n+1,J+1);
UU(2:end,1) = U;
for j = 1:J
    % evaluate F
    F = (1/2)*(ffun(xxi,tt(j))+ffun(xxi,tt(j+1))); %Using the expression we
    %found in Q1c but with indexing based on matlab formatting for the
    %steps in the function

    % solve linear system
    % expression involves U, C, D, and F. Do NOT use "inv".
    %using the expression for U from Q1c, and also what was explained in
    %lecture and using \ instead of inv
    U = C\(F + D*UU(2:end,j));

    % record solution in UU
    UU(2:end,j+1) = U;
end
% find time indices to plot
tplot = [0,1/16,1/8,1/4,1/2,1]';
[~,it] = intersect(tt,tplot);
% plot solution
figure(1), clf,
plot(xx,UU(:,it),'o-');
xlabel("Spatial Domain X-Values");
ylabel("u(x,t) Temperature Values");
legend("t=0", "t=1/16", "t=1/8", "t=1/4", "t=1/2", "t=1", 'Position', [0.2 0.6 0.1 0.2]);
% evaluate output
s = UU(end,end);
end
```

e) Solution representation for T=1, N=16, J=16 for various values of t using finite difference method



code over here for table

F) The observed error behaviour matches my expectation as it follows the second order scheme discussed in lecture for both spatial and time domains. It can be noted that the error reduces by a factor of ≈4 for a doubling in the n, J values each time. This is observed and shown in the table below.

See table for error convergence values and confirmation.

s_ref = 0.0997718155

n	J	|s_ref-s|
8	8	0.0047237
16	16	0.00078075
32	32	0.0001566
64	64	3.4994e-05

Question 1, part f heat_fd_solver and errorcalculator code

```
function s = heat_fd_solver(n,J)
% functions
gfun = @(x) 0.5*x.*(2-x) + 1/(6*pi)*sin(3*pi*x);
ffun = @(x,t) exp(-3*x).*exp(t);
% discretization parameters
T = 1; %over the time range (0,1) as given in the question
% setup spatial and temporal grids
xx = linspace(0,1,n+1)';
tt = linspace(0,T,J+1)';  %equally spaced points in the time domain
%discretization
dx = 1/n;
dt = T/J; %discretizing in the time domain over the given parameter
% assemble the A matrix
% (not the most effcient method, but it will work)
A = sparse([],[],[],n,n);
for i = 1:n
    if (i == 1)
        % left boundary
        A(i,i) = 2;  %obtained from first element in first row of A matrix
        A(i,i+1) = -1; %obtained from second element in first row of A matrix
    elseif (i == n)
        % right boundary
        A(i,i-1) = -2; %obtained from second last element in last row of A matrix
        A(i,i) = 2+2*dx; %obtained from last element in last row of A matrix
    else
        % all other points
        % all these points are obtained from the 3 shifting elements in
        %each row of the A matrix except for the first and last rows which
        %represent the boundary conditions
        A(i,i-1) = -1;
        A(i,i) = 2;
        A(i,i+1) = -1;
    end
end
%In my expression for the A matrix I also have a 1/dx^2 term in front of
%it, so I am adding that here for it to be consistent, once again using
%element wise division as A is a matrix!
A=A./dx^2;
% iteration
I = speye(n,n);
% assemble C and D matrices in terms of A, I, and dt
  %need to use elementwise division and multiplication as they are vector
  %representations of C and D
C = ((I./dt)+(1/2).*A);  %found in Q1c using C-N method
D = ((I./dt)-(1/2).*A);  %found in Q1c using C-N method
% initialize state
xxi = xx(2:end);  % x excluding the left node
U = gfun(xxi);
% record the initial state in UU; note the first node is left as 0
UU = zeros(n+1,J+1);
UU(2:end,1) = U;
for j = 1:J
    % evaluate F
    F = (1/2)*(ffun(xxi,tt(j))+ffun(xxi,tt(j+1))); %Using the expression we
    %found in Q1c but with indexing based on matlab formatting for the
    %steps in the function

    % solve linear system
    % expression involves U, C, D, and F. Do NOT use "inv".
    %using the expression for U from Q1c, and also what was explained in
    %lecture and using \ instead of inv
    U = C\(F + D*UU(2:end,j));

    % record solution in UU
    UU(2:end,j+1) = U;
end
% evaluate output
s = UU(end,end);
End


%Now writing the code needed to get the error for the different levels of
%discretization mentioned in Q1f
function errorcalculator
%forming vectors for the n, J values and reference levels of discretization
nJ_values = [8,16,32,64];
reference = [512,512];
%getting the value of s_ref for comparison by passing to heat_fd_temp
ref_n = reference(1);
ref_J = reference(2);
s_ref = heat_fd_solver(ref_n, ref_J);
%creating an error vector with zeros that holds the error values at each level of
%discretization we have above
error_values = zeros(length(nJ_values), size(nJ_values,1));
%obtaining the errors for each solution based on the parameters and storing
%them in the error_values vector we created earlier
for i = 1:length(nJ_values)
    s=heat_fd_solver(nJ_values(i),nJ_values(i));
    error_values(i)=abs(s_ref-s);
end
%Displaying the values in the form of a table to show the errors at each
%level of discretization
T=table(nJ_values',nJ_values',error_values, 'VariableNames', {'n','J','|s_ref-s|'});
disp(T)
end
```

Q2 (a) Given Laplace's Equation

$$-\Delta u = 0 \text{ in } \Omega$$
$$u(r=r_1, \theta) = 0 \quad , \theta \in [0, 2\pi]$$
$$u(r=r_2, \theta) = g(\theta) \;, \theta \in [0, 2\pi]$$

Laplacian in polar coordinates

$$\Delta u = \nabla \cdot \nabla u = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$$
$$= \frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2}$$

we can use
Seperation of variables

$$u_n(r,\theta) = R_n(r)\Theta_n(\theta) \rightarrow \text{Family of sols}$$

$$R_n''\Theta_n + \frac{1}{r}R_n'\Theta_n + \frac{1}{r^2}R_n\Theta_n'' = 0$$

multiply by $r^2 \frac{1}{R_n\Theta_n}$

$$\Rightarrow r^2\frac{R_n''}{R_n} + r\frac{R_n'}{R_n} + \frac{\Theta_n''}{\Theta_n} = 0$$

$$\Rightarrow \underbrace{r^2\frac{R_n''}{R_n} + r\frac{R_n'}{R_n}}_{\text{independent of } \theta} = \underbrace{-\frac{\Theta_n''}{\Theta_n}}_{\text{independent of } r} = \underbrace{\alpha_n}_{\text{constant}}$$

Now, we can solve for

Eigenproblem for Θ_n

$$-\Theta_n'' = \alpha_n\Theta_n \quad \text{in } (0, 2\pi)$$
$$\Theta_n(0) = \Theta_n(2\pi) \rightarrow \text{periodic B.C. since we want continuous solutions.}$$

General soln is of the form,

$$\Rightarrow \Theta_n(\theta) = A_n\cos(n\theta) + B_n\sin(n\theta) \quad , n = 0, 1, 2 \dots$$
$$\alpha_n = n^2$$

Now,
ODE for R_n

$$r^2R_n'' + rR_n' = \overset{\alpha_n}{n^2}R_n$$
$$r^2R_n'' + rR_n' - n^2R_n = 0 \qquad \text{(Cauchy-Euler equation of order 2)}$$

we can thus use the general sol for Cauchy-Euler equation of 2nd order

$$R_n(r) = \begin{cases} C_n + D_n\log(r) \;, & n = 0 \\ C_nr^n + D_nr^{-n} \;, & n = 1, 2, 3 \dots \end{cases}$$

Now, we impose the boundary conditions

B.C at $r = r_1$ and $n = 0$, we have

$$R_0(r_1) = C_0 + D_0\log(r_1) \overset{\text{want}}{=} 0 \,,$$
$$\Rightarrow C_0 = -D_0\log(r_1)$$

plugging back in the general soln, we have

$$R_0(r) = -D_0\log(r_1) + D_0\log(r)$$
$$R_0(r) = D_0\log(r/r_1)$$

B.C at $r = r_1$, and $n = 1, 2, \dots$, we have

$$R_n(r_1) = C_nr_1^n + D_nr_1^{-n} \overset{\text{want}}{=} 0$$
$$\Rightarrow C_nr_1^n = -D_nr_1^{-n}$$
$$\Rightarrow C_n = \frac{-D_nr_1^{-n}}{r_1^n} = \frac{-D_n}{r_1^{2n}}$$

plugging back in the general soln, we have

$$R_n(r) = -\frac{D_n}{r_1^{2n}}r^n + D_nr^{-n}$$
$$R_n(r) = D_n\left(-\frac{r^n}{r_1^{2n}} + r^{-n}\right) = D_n\left(-r_1^{-2n}\cdot r^n + r^{-n}\right)$$

Now using superposition and the general soln expression,
we have that $u_n(r,\theta) = R_n(r)\Theta_n(\theta)$

$$\Rightarrow u_n(r,\theta) = A_0R_0(r) + \sum_{n=1}^{\infty} R_n(r)(A_n\cos(n\theta) + B_n\sin(n\theta))$$

we have that $u_n(r,\theta) = R_n(r)\,\Theta_n(\theta)$

$\Rightarrow$ $u_n(r,\theta) = A_0R_0(r) + \sum_{n=1}^{\infty} R_n(r)(A_n\cos(n\theta) + B_n\sin(n\theta))$

$$\boxed{u_n(r,\theta) = \begin{cases} A_0(\log(r/r_1)) & , n=0 \\ D_n(-r_1^{-2n}\cdot r^n + r^{-n})(A_n\cos(n\theta) + B_n\sin(n\theta)) & , n=1,2,3\ldots \end{cases}}$$

satisfies PDE, inner B.C.

Family of functions satisfying PDE, inner B.C.

b) Now, to find the solution for the PDE with the other B.C.

$u(r=r_2,\theta) = g(\theta)$, where $g(\theta) \equiv \cos(m\theta)$, = +ve integer

we can use the general solution expression found in part (a) and impose the B.C at $r=r_2$ on it.

i.e, $u(r=r_2,\theta) = A_0\log(r_2/r_1) + \sum_{n=1}^{\infty}(r_1^{-2n}r_2^n - r_2^{-n})(A_n\cos(n\theta) + B_n\sin(n\theta)) = \cos(m\theta) \to g(\theta)$

The above expression in the RHS matches a fourier series form, so deducing the coefficients for it.

we know $g_0 = 2\cdot\frac{1}{L}\int_{-L}^{L} g(\theta)\,d\theta$

$\Rightarrow$ since the period in the PDE is $[0,2\pi)$, we can use $[-\pi,\pi)$, instead as $g(\theta) = \cos(m\theta)$, $m>0$ is even-periodic and the area under $[0,2\pi)$ is the same as the area under $[-\pi,\pi)$ and the integrals remain unchanged otherwise.

$\therefore$ we have, $g_0 = 2\cdot\frac{1}{\pi}\int_{-\pi}^{\pi}\cos(m\theta)\,d\theta \Rightarrow \frac{2}{\pi}[\overset{0}{\sin(m\pi)} - \overset{0}{\sin(-m\pi)}] = 0$

↳ odd-fn.

comparing this to $A_0\log(r_1/r_2)$

we have $A_0 = 0$.

now, we take $g_{n,\cos}$ and $g_{n,\sin}$

$$g_{n,\cos} = \frac{1}{L}\int_{-L}^{L}\cos(n\theta)g(\theta)\,d\theta \Rightarrow \frac{1}{\pi}\int_{-\pi}^{\pi}\cos(n\theta)\cos(m\theta)\,d\theta = \frac{2}{\pi}\int_{0}^{\pi}\cos(n\theta)\cos(m\theta)\,d\theta$$

using the integral list we have,

$$\int\cos(\alpha x)\cos(\beta x)\,dx = \frac{\sin((\alpha-\beta)x)}{2(\alpha-\beta)} + \frac{\sin((\alpha+\beta)x)}{2(\alpha+\beta)}, \quad \alpha\neq\beta$$

So we get

$$g_{n,\cos} = \frac{2}{\pi}\left[\frac{\sin((n-m)\theta)}{2(n-m)} + \frac{\sin((n+m)\theta)}{2(n+m)}\right]\Bigg|_0^{\pi}$$

$$g_{n,\cos} \Rightarrow \begin{cases} 1, & n=m \\ 0, & n\neq m \end{cases}$$

now, we take $g_{n,\sin}$

$$g_{n,\sin} = \frac{1}{L}\int_{-L}^{L}\sin(n\theta)g(\theta)\,d\theta \Rightarrow \frac{1}{\pi}\int_{-\pi}^{\pi}\sin(n\theta)\cos(m\theta)\,d\theta = \frac{2}{\pi}\int_{0}^{\pi}\sin(n\theta)\cos(m\theta)\,d\theta$$

using the integral list we have,

$$\int\sin(\alpha x)\cos(\beta x)\,dx = \frac{\cos((\alpha-\beta)x)}{2(\alpha-\beta)} + \frac{\cos((\alpha+\beta)x)}{2(\alpha+\beta)}, \quad \alpha\neq\beta$$

So we get

$$g_{n,\sin} = \frac{2}{\pi}\left[\frac{\cos((n-m)\theta)}{2(n-m)} + \frac{\cos((n+m)\theta)}{2(n+m)}\right]\Bigg|_0^{\pi}$$

$\Rightarrow g_{n,\sin} = 0$.

Now, if we compare coefficients with the expression of the soln we have

$$g_{n,\cos} = \begin{cases} 0, & n\neq m \\ 1, & n=m \end{cases} = (-r^{-2n}\cdot r_2^n - r_2^{-n})\,A_n\cos(n\theta)$$

$$\Rightarrow A_n\cos(n\theta) = \frac{1}{(-r^{-2n}\cdot r^n - r^{-n})} \quad -(1)$$

$\qquad\qquad\qquad \{ 1, n = m$

$$\Rightarrow A_n \cos(n\theta) = \frac{1}{(r_1^{-2n} \cdot r_2^{n} - r_2^{-n})} \quad -(1)$$

$$g_{n,\sin} = 0 \Rightarrow B_n \sin(n\theta) = 0 .$$

$\therefore$ plugging back into the original expression, we get

$$u(r,\theta) = \overset{\nearrow 0}{\cancel{A_0}} \log(r/r_1) + \sum_{n=1}^{\infty} (r_1^{-2n} r_2^{n} - r_2^{-n}) \Big(\underbrace{A_n \cos(n\theta)}_{\text{plug in } (1)} + \overset{\nearrow 0}{\cancel{B_n}} \sin(n\theta) \Big) = \overset{\nearrow g(\theta)}{\cos(m\theta)}$$

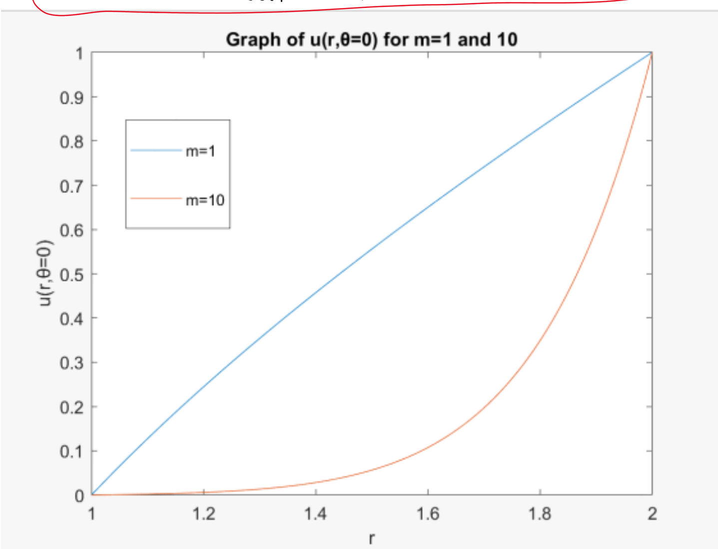
changing all 'n' subscripts to 'm' to match $\cos(m\theta)$, and shifting terms around we get

$$\boxed{u(r,\theta) = \frac{\cos(m\theta)\,(r_1^{-2m} r^{m} - r^{-m})}{(r_1^{-2m} r_2^{m} - r_2^{-m})}} \rightarrow \text{solution } u(r,\theta) \text{ for } g(\theta) = \cos(m\theta).$$

c) i) Yes, the graph obtained is consistent with my expectations as the solution with the higher mode ($m=10$) decays a lot more rapidly from the outer boundary as it moves to the inner one as can be seen in the graph.

ii) Yes, the maximum principle is also satisfied because the temperature value as seen in the graph between the two boundaries $[r_1, r_2]$ does not exceed its max value as seen on the r_2 boundary.

The graph for Laplaces solution at $m=1,10$ between the annular domain boundaries.



Code for (2c)

Question 2c, plot for m=1 and m=10 using the usolver function

```
%creating boundary values r_1, r_2 and also creating a vector with equally
%spaced points between them
r_1 = 1;
r_2 = 2;
r = linspace(r_1, r_2);
%creating a vector m for the two m values for which the solution will be
%graphed
m=[1,10];
%Plotting the solutions obtained by running the solver on the same plot
plot(r, usolver(r,m(1),r_1,r_2));
hold on
plot(r, usolver(r,m(2),r_1,r_2));
hold off
%labelling
xlabel('r');
ylabel('u(r,θ=0)');
title('Graph of u(r,θ=0) for m=1 and 10')
legend('m=1', 'm=10', 'Position', [0.2 0.6 0.1 0.2]);
function u = usolver(r,m,r_1,r_2)
%theta is 0 in the question
th = 0;
% Using the value of the solution we obtained in Q2b of the assignment
u = cos(m*th)*((r_1^(-2*m)*r.^m-(r.^(-m))))/(r_1^(-2*m)*r_2^m-(r_2^(-m)));
end
```

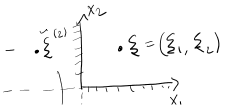
Q3 a) Laplace's Equation

$$-\Delta u = 0 \text{ on } \Omega \equiv \mathbb{R}_{>0} \times \mathbb{R}_{>0}$$
$$u = g_B \text{ on } \Gamma_B \equiv \mathbb{R}_{\geq 0} \times \{x_2 = 0\}$$
$$u = g_L \text{ on } \Gamma_L \equiv \{x_1 = 0\} \times \mathbb{R}_{>0}$$

To find green's function given $\xi \in \Omega$,

$$-\Delta G(x,\xi) = \delta(x-\xi) \;\; \forall x \in \Omega$$
$$G(x,\xi) = 0 \;\; \forall x\, \Gamma_B \cup \Gamma_L$$

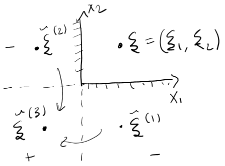
First, sketching the problem, using image points



Now we can express $\hat{\xi}^{(1)}, \hat{\xi}^{(2)}, \hat{\xi}^{(3)}$ in terms of ξ_1, ξ_2

$$\Rightarrow \hat{\xi}^{(1)} = (\xi_1, -\xi_2), \;\; \hat{\xi}^{(2)} = (-\xi_1, \xi_2), \;\; \hat{\xi}^{(3)} = (-\xi_1, -\xi_2)$$

... of the fundamental solution we have



Now we can express $\hat{\xi}$, $\hat{\xi}$, $\hat{\xi}$ in terms of ξ_1, ξ_2

$\Rightarrow \hat{\xi}^{(1)} = (\xi_1, -\xi_2)$, $\hat{\xi}^{(2)} = (-\xi_1, \xi_2)$, $\hat{\xi}^{(3)} = (-\xi_1, -\xi_2)$

writing out the solution in terms of the fundamental solution, we have

$G(x-\xi) = \Phi(x-\xi) + \psi(x,\xi)$ where $\psi(x,\xi) = -\Phi(x-\xi) \; \forall x \in \partial\Omega$

$\therefore$ here $\psi(x,\xi) = -\Phi(x-\hat{\xi}^{(1)}) - \Phi(x-\hat{\xi}^{(2)}) + \Phi(x-\hat{\xi}^{(3)})$,

and so $\boxed{G(x-\xi) = \Phi(x-\xi) - \Phi(x-\hat{\xi}^{(1)}) - \Phi(x-\hat{\xi}^{(2)}) + \Phi(x-\hat{\xi}^{(3)})}$, $\rightarrow$ greens Function representation for the laplace equation.

where $\|x-\xi\| = \|x-\hat{\xi}^{(1)}\|$ on Γ_B, $\|x-\xi\| = \|x-\hat{\xi}^{(2)}\|$ on Γ_L, and $\|x-\hat{\xi}^{(2)}\| = \|x-\hat{\xi}^{(3)}\|$ on Γ_B

b) $g_L = 0$, we need to find integral expression of solution $u(x = (x_1 x_2))$

we start by using the Green's Representation Formula

$$u(x) = -\int_{\partial\Omega} \nu \cdot \nabla_\xi G(x,\xi) g(\xi) d\xi$$

For $u(x_1, x_2)$ in our case this can be expended as Follows

$$u(x_1,x_2) = -\int_{\Gamma_B} \nu_B \cdot \nabla_\xi G(x,\xi) g_B(\xi) d\xi - \underbrace{\int_{\Gamma_L} \nu_L \cdot \nabla_\xi G(x,\xi) \underbrace{g_L(\xi)}_{=0 \text{ here as given}} d\xi}_{\text{thus the entire integral vanishes.}}$$

$\Rightarrow$ Now plugging in the $G(x,\xi)$ function we found in (a).

$$= -\int_{\Gamma_B} \nu_B \cdot \nabla_\xi \left(\Phi(x-\xi) - \Phi(x-\hat{\xi}^{(1)}) - \Phi(x-\hat{\xi}^{(2)}) + \Phi(x-\hat{\xi}^{(3)}) \right) g_B(\xi) d\xi \quad - (1)$$

Now, using the hint : we can compute $-\dfrac{\partial \Phi(x-\xi)}{\partial \xi_2} = \dfrac{\partial}{\partial \xi_2}\left(\dfrac{1}{2\pi} \log\left(\sqrt{(x_1-\xi_1)^2 + (x_2-\xi_2)^2} \right) \right)$

Plugging the expressions in, we get

$$= \frac{\partial}{\partial \xi_2}\left(\frac{1}{2\pi} \ln\left((x_1-\xi_1)^2 + (x_2-\xi_2)^2 \right)^{1/2} \right) - \frac{\partial}{\partial \xi_2}\left(\frac{1}{2\pi} \ln\left((x_1-\xi_1)^2 + (x_2+\xi_2)^2 \right)^{1/2} \right)$$

$$- \frac{\partial}{\partial \xi_2}\left(\frac{1}{2\pi} \ln\left((x_1+\xi_1)^2 + (x_2-\xi_2)^2 \right)^{1/2} \right) - \frac{\partial}{\partial \xi_2}\left(\frac{1}{2\pi} \ln\left((x_1+\xi_1)^2 + (x_2+\xi_2)^2 \right)^{1/2} \right)$$

Plugging this expression into (1) after solving the partial derivatives, we get

$$\Rightarrow -\int_{\Gamma_B} \frac{1}{2\pi} \left(\frac{\xi_2 - x_2}{(x_1-\xi_1)^2 + (x_2-\xi_2)^2} - \frac{\xi_2 + x_2}{(x_1-\xi_1)^2 + (x_2+\xi_2)^2} - \frac{\xi_2 - x_2}{(x_1+\xi_1)^2 + (x_2-\xi_2)^2} + \frac{\xi_2 + x_2}{(x_1+\xi_1)^2 + (x_2+\xi_2)^2} \right) g_B(\xi) d\xi$$

we also know that on Γ_B the ξ_2 term will be 0, so

we get $\Rightarrow$
$$-\int_{\Gamma_B} \frac{1}{2\pi} \left(\frac{-x_2}{(x_1-\xi_1)^2 + x_2^2} - \frac{x_2}{(x_1-\xi_1)^2 + x_2^2} + \frac{x_2}{(x_1+\xi_1)^2 + x_2^2} + \frac{x_2}{(x_1+\xi_1)^2 + x_2^2} \right) g_B(\xi) d\xi$$

$$\Rightarrow \int_{\Gamma_B} \frac{1}{2\pi} \left(\frac{2x_2}{(x_1-\xi_1)^2 + x_2^2} - \frac{2x_2}{(x_1+\xi_1)^2 + x_2^2} \right) g_B(\xi) d\xi$$

but Γ_b goes from 0 to ∞, so replacing limits and moving $\frac{1}{\pi}$ out after cancelling the Factor of 2

$$\boxed{u(x_1,x_2) = \frac{1}{\pi} \int_0^\infty \left(\frac{x_2}{(x_1-\xi_1)^2 + x_2^2} - \frac{x_2}{(x_1+\xi_1)^2 + x_2^2} \right) g(\xi) d\xi_1}$$

$\rightarrow$ integral representation of the solution with $g_L = 0$.

c) $g_L = 0$, $g_B(x_1) = \begin{cases} 1, & x_1 \in (0,1) \\ 0, & \text{o.w} \end{cases}$

To Find: integral representation of solution over finite limit

we can use the expression we found in (b) to Find the solution,

So all we need to do is plug in over the new limit 0 to 1, and evaluate $g(\xi)$

we know $g(\xi) = 1$ over the given limit of 0 to 1 based on the Function provided above, so we have

$$\boxed{u(x_1, x_2) = \frac{1}{\pi}\int_0^1 \left(\frac{x_2}{(x-\xi_1)^2 + x_2^2} - \frac{x_2}{(x_1+\xi_1)^2 + x_2^2} \right) d\xi_1}$$

↪ integral representation of solution given the $g_B(x_1)$ condition.