Caleb Logemann
AER E 546 Fluid Mechanics and Heat Transfer I
Homework 2

1. The heat fin equation is the linear o.d.e.

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = MT$$

where M is a sort of thermal mass. First write the finite difference equation in terms of a tridiagonal matrix. Solve that equation using the Thomas algorithm (Gaussian elimination) for:

(a) Compute a solution with the boundary conditions $T(0) = 1$ and $T(1) = 0$. This corresponds to a fin that is between a hot and a cold reservoir. In non-dimensional terms, the heat flux into the cold reservoir is $-\frac{\mathrm{d}T}{\mathrm{d}x}$ at $x = 1$. Obtain the heat flux as $x = 1$ for $M = 1, 5, 9$. Use enough grid points to obtain 1% accuracy. Provide your three numerical values of the heat flux. Provide a single graph with curves of $T(x)$ for the 3 values of M.

First I will establish a some notation. I will discretize the fin into $N + 1$ points. Let $x_i = \frac{i}{N}$, then $x_0 = 0$ and $x_N = 1$. Let the approximate solution at x_i be represented by T_i. Then a numerical solution consists of a set of values T_i for $i \in \mathbb{N}$, $0 \le i \le N$.

Next I will discretize the partial differential equation into a discrete equation. The second order central finite difference for the second derivative is

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2}$$

Plugging this into the partial differential equation gives the following difference equation

$$\frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} = MT_i$$

Simplifying this gives

$$T_{i-1} + \left(-2 - M\Delta x^2\right)T_i + T_{i+1} = 0$$

For this problem the boundary conditions are $T(0) = 1$ and $T(1) = 0$. This can be encoded into the numerical solution at $T_0 = 1$ and $T_N = 0$. Now only the values for T_i for $1 \le i \le N-1$ need to be found. These can be found by solving the equations

$$T_{i-1} + \left(-2 - M\Delta x^2\right)T_i + T_{i+1} = 0$$

for $i = 1$, this become

$$\left(-2 - M\Delta x^2\right)T_1 + T_2 = -1$$

and for $i = N - 1$ the equation is

$$T_{N-2} + \left(-2 - M\Delta x^2\right)T_{N-1} = 0$$

These equation can be written in matrix form as

$$\begin{bmatrix} -2-\Delta x^2 & 1 & & & 0 \\ 1 & -2-\Delta x^2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2-\Delta x^2 & 1 \\ 0 & & & 1 & -2-\Delta x^2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N-2} \\ T_{N-1} \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

This is a tridiagonal system which can be solved easily with the Thomas algorithm.

The following function evaluates the Thomas algorithm on a tridiagonal system.

```
function [y] = tridiag(n, a, b, c, v);
    % solve a tridiagonal system with Gaussian elimination
    % Also known as Thomas Algorithm
    % (a_1 c_1   0   0                   0  )(   y_1   )  (   v_1   )
    % (b_1 a_2 c_2   0                   0  )(   y_2   )  (   v_2   )
    % (  0 b_2 a_3 c_3                   0  )(   y_3   )= (   v_3   )
    % (  0                               0  )(         )= (         )
    % (  .                a_{n-1} c_{n-1})( y_{n-1} )  ( v_{n-1} )
    % (  0                b_{n-1}     a_n )(   y_n   )  (   v_n   )

    % create array zero to store solutions
    y = zeros(size(v));

    % eliminate b_i's
    for(i=1:n-1)
        a(i+1) = a(i+1) + c(i)*(-b(i)/a(i));
        v(i+1) = v(i+1) + v(i)*(-b(i)/a(i));
    end

    % solve for y_n
    y(n) = v(n)/a(n);

    for(i=(n-1):-1:1)
        y(i) = (v(i) - c(i)*y(i+1))/a(i);
    end

end
```

The following script uses the previous function to run the Thomas algorithm on the tridiagonal system that was found earlier.

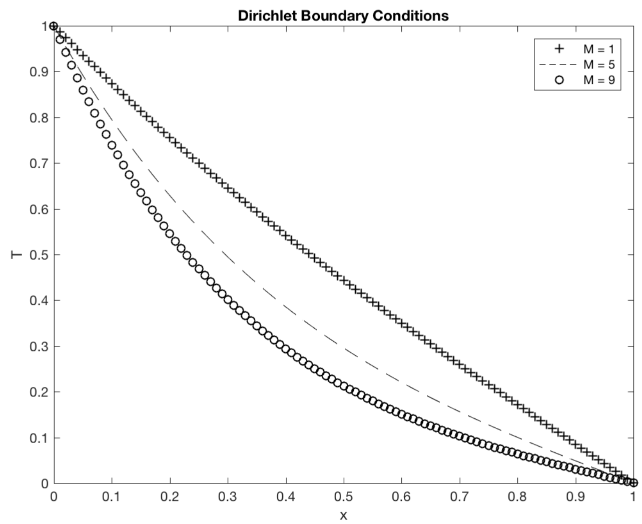
```
%% Problem 2a
n = 100;
deltaX = 1/n;
nM = 3;
sol = zeros(nM, n+1);
sol(:,1) = ones(nM,1)';
for iter = [1, 5, 9; 1:nM]
    M = iter(1);
    i = iter(2);
    mainDiagonal  = (-M*deltaX^2 - 2)*ones(n-1,1);
    lowerDiagonal = ones(n-2,1);
    upperDiagonal = ones(n-2,1);
    RHS = zeros(n-1,1);
    % boundary conditions
    RHS(1) = -1;
    sol(i,2:end-1) = tridiag(n-1, mainDiagonal, lowerDiagonal, upperDiagonal, RHS);
end

x = linspace(0, 1, n+1);
plot(x, sol(1,:), 'k+', x, sol(2,:), 'k--', x, sol(3,:), 'ko');
legend('M = 1', 'M = 5', 'M = 9');
title('Dirichlet Boundary Conditions');
xlabel('x');
ylabel('T');
saveas(gcf, 'Figures/02_01.png', 'png');

% -dT/dx at x = 1 can be approximated by -dT/dx = (T_(n-1) - T_n)/deltaX
% T_n = sol(i, end) = sol(i, n+1)
heatFlux = zeros(nM, 1);
```

```
for i = 1:nM
    heatFlux(i) = (sol(i, end-1) - sol(i,end))/deltaX;
end
disp(heatFlux);
```

This script outputs the following image, which shows the three plots for $M = 1, 5, 9$.



The values of the heat flux at $x = 1$ are given to be

$$M = 1 \qquad \left(-\frac{\mathrm{d}T}{\mathrm{d}x}\right)\bigg|_{x=1} = 0.850933420029442$$
$$M = 5 \qquad \left(-\frac{\mathrm{d}T}{\mathrm{d}x}\right)\bigg|_{x=1} = 0.483548793162072$$
$$M = 9 \qquad \left(-\frac{\mathrm{d}T}{\mathrm{d}x}\right)\bigg|_{x=1} = 0.299532256416841$$

Note that these values are found by evaluating

$$\left(-\frac{\mathrm{d}T}{\mathrm{d}x}\right)\bigg|_{x=1} \approx \frac{T_{n-1} - T_n}{\Delta x}$$

(b) Compute a solution with the boundary conditions $T(0) = 1$, $\frac{\mathrm{d}T(1)}{\mathrm{d}x} = 0$. This corresponds to a fin that is insulated at one end. Solve for the temperature, $T(1)$, at the insulated end for $M = 1, 5, 9$. Provide your three numerical values of $T(1)$. Also plot $T(x)$ for $M = 9$ with each pair of boundary conditions and compare to the exact solution.

For this problem we begin with the same difference equation as in part (a).

$$T_{i-1} + \left(-2 - M\Delta x^2\right)T_i + T_{i+1} = 0$$

Again we have the boundary condition that $T(0) = 1$, in results in the same modified equation for $i = 1$.

$$\left(-2 - M\Delta x^2\right)T_1 + T_2 = -1$$

However we have a different boundary condition at $x = 1$. In this case we wish to enforce $\frac{\mathrm{d}T(1)}{\mathrm{d}x} = 0$. In order to enforce this consdition we will consider an imaginary point on the fin T_{N+1}, this point doesn't actually exist on the fin, but if it did exist and the first derivative was zero then

$$\frac{T_{N+1} - T_N}{\Delta x} = 0.$$

In other words the finite difference for the first derivative should be zero, this simplifies to $T_N = T_{N+1}$. This should make intuitive sense, as if the derivative is zero the value shouldn't change past then endpoint. Using this condition in the difference equation for $i = N$ gives

$$\begin{aligned} T_{N-1} + \left(-2 - M\Delta x^2\right)T_N + T_{N+1} &= 0 \\ T_{N-1} + \left(-2 - M\Delta x^2\right)T_N + T_N &= 0 \\ T_{N-1} + \left(-1 - M\Delta x^2\right)T_N &= 0 \end{aligned}$$

We now have N equations for N unknowns. This forms the following matrix equation.

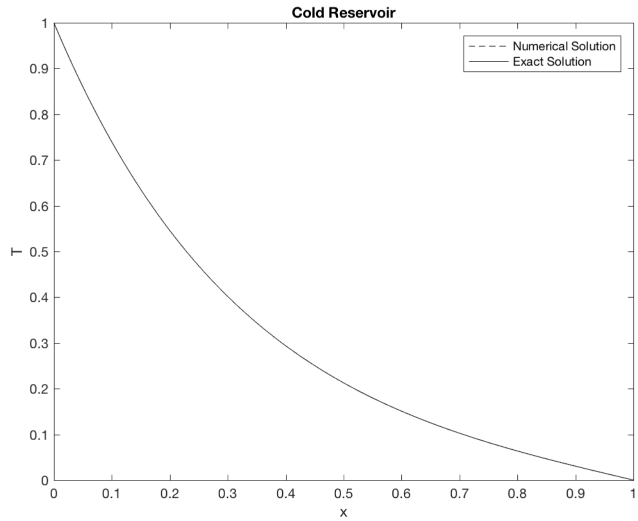
$$\begin{bmatrix} -2-\Delta x^2 & 1 & & & 0 \\ 1 & -2-\Delta x^2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2-\Delta x^2 & 1 \\ 0 & & & 1 & -1-\Delta x^2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N-1} \\ T_N \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

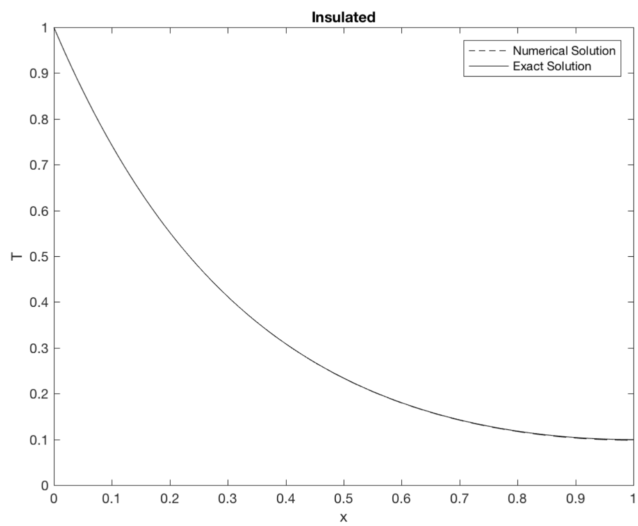
Note that this system is one equation larger than in part (a) as the value of T_N must be found. Still this is a tridiagonal system that can be solved using the Thomas algorithm, which was shown in part (a). The following script using the Thomas algorithm to solve this system for $M = 1, 5, 9$.

```
%% Problem 2b
n = 100;
deltaX = 1/n;
nM = 3;
sol2 = zeros(nM, n+1);
sol2(:,1) = ones(nM,1)';
for iter = [1, 5, 9; 1:nM]
    M = iter(1);
    i = iter(2);
    mainDiagonal  = (-M*deltaX^2 - 2)*ones(n,1);
    mainDiagonal(end) = (-M*deltaX^2 - 1);
    lowerDiagonal = ones(n-1,1);
    upperDiagonal = ones(n-1,1);
    RHS = zeros(n,1);
    % boundary conditions
    RHS(1) = -1;
    sol2(i,2:end) = tridiag(n, mainDiagonal, lowerDiagonal, upperDiagonal, RHS);
end

exactSol1 = @(x) (exp(-3)/(exp(-3) - exp(3))) * exp(3*x) + (-exp(3)/(exp(-3) - exp
    ↪ (3))) * exp(-3*x);
exactSol2 = @(x) (exp(-3)/(exp(-3) + exp(3))) * exp(3*x) + (exp(3)/(exp(-3) + exp
    ↪ (3))) * exp(-3*x);
x = linspace(0, 1, n+1);
plot(x, sol(3,:), 'k--', x, exactSol1(x), 'k-');
legend('Numerical Solution', 'Exact Solution ');
title('Cold Reservoir');
xlabel('x');
ylabel('T');
saveas(gcf, 'Figures/02_02.png', 'png');
```

The script output the following images. Note that the numerical solutions and the exact solutions are almost indistinguishable.





The following are the numerical values found for $T(1)$ for $M = 1, 5, 9$.

$$\begin{aligned} M &= 1 \qquad T(1) = 0.645597948372887 \\ M &= 5 \qquad T(1) = 0.209066847236553 \\ M &= 9 \qquad T(1) = 0.097878298010251 \end{aligned}$$

(c) Add a distributed heat source: Compute and plot a solution of the non-homogeneous equation

$$\frac{\mathrm{d}^2 T}{\mathrm{d}x^2} = MT - 100x^2(1-x)^2$$

with $M = 9$, $T(0) = 1$ and $\frac{\mathrm{d}T(1)}{\mathrm{d}x} = 0$.
In this problem we can start with the tridiagonal system given in (b) however the heat source requires changing the RHS.
Now the equation for each point i becomes

$$T_{i-1} + \left(-2 - M\Delta x^2\right)T_i + T_{i+1} = -100x_i^2(1-x_i)^2\Delta x^2.$$

The equation for $i = 1$ is

$$\left(-2 - M\Delta x^2\right)T_1 + T_2 = -100x_1^2(1-x_1)^2\Delta x^2 - 1$$

and the equation for $i = N$ is

$$T_{N-1} + \left(-1 - M\Delta x^2\right)T_N = -100x_N^2(1-x_N)^2\Delta x^2.$$

This create the following tridiagonal system.

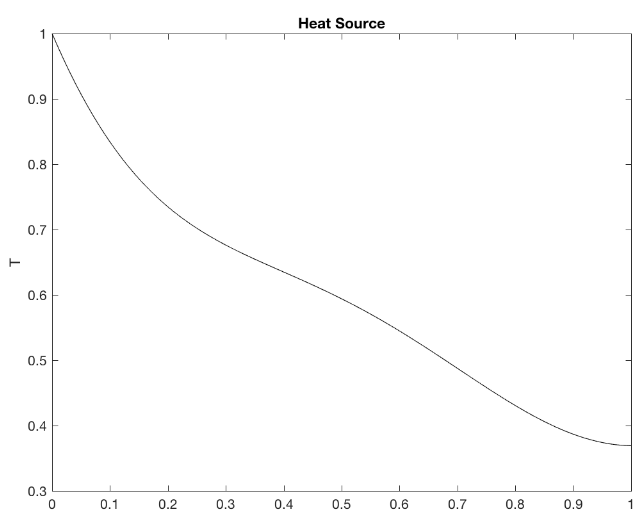
$$\begin{bmatrix} -2-\Delta x^2 & 1 & & & 0 \\ 1 & -2-\Delta x^2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2-\Delta x^2 & 1 \\ 0 & & & 1 & -1-\Delta x^2 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ \vdots \\ T_{N-1} \\ T_N \end{bmatrix} = \begin{bmatrix} -1-100x_1^2(1-x_1)^2\Delta x^2 \\ -100x_2^2(1-x_2)^2\Delta x^2 \\ \vdots \\ -100x_{N-1}^2(1-x_{N-1})^2\Delta x^2 \\ -100x_N^2(1-x_N)^2\Delta x^2 \end{bmatrix}$$

The following script solves the previous tridiagonal system using the Thomas algorithm.

```
%% Problem 2c
n = 100;
deltaX = 1/n;
sol3 = zeros(1, n+1);
sol3(1,1) = 1;
M = 9;
mainDiagonal  = (-M*deltaX^2 - 2)*ones(n,1);
mainDiagonal(end) = (-M*deltaX^2 - 1);
lowerDiagonal = ones(n-1,1);
upperDiagonal = ones(n-1,1);
x = linspace(deltaX, 1, n);
RHS = (-100*deltaX^2)*((x.^2).*((1-x).^2));
% boundary conditions
RHS(1) = RHS(1) - 1;
sol3(1,2:end) = tridiag(n, mainDiagonal, lowerDiagonal, upperDiagonal, RHS);

x = linspace(0,1,n+1);
plot(x, sol3(1,:), 'k-');
title('Heat Source');
xlabel('x');
ylabel('T');
saveas(gcf, 'Figures/02_04.png', 'png');
```

The script outputs the following image.



2. (i) What type of p.d.e. is

$$\frac{\partial^2 \phi}{\partial x \partial y} + \phi = 25?$$

This is a hyperbolic p.d.e. because the descriminant is greater than zero.

$$b^2 - 4ac = 1^2 - 4 \times 0 \times 0 = 1 > 0$$

(ii) What type of p.d.e. does the velocity potential, ϕ, satisfy if

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0$$

with

$$u = \frac{\partial \phi}{\partial x} \qquad v = \frac{\partial \phi}{\partial y}?$$

This p.d.e. can be rewritten as

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = 0$$

This is an elliptic p.d.e. because the descriminant is less than zero as shown below.

$$b^2 - 4ac = 0 - 4 \times 1 \times 1 = -4 < 0$$

(iii) The boundary layer momentum equation is

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = \frac{1}{Re}\frac{\partial^2 u}{\partial y^2}$$

where Re is the Reynolds number. What type is this equation?
This is a parabolic p.d.e because the descriminant is zero as shown below.

$$b^2 - 4ac = 0^2 - 4 \times 0 \times \frac{1}{Re} = 0$$