ME 254 Computational Fluid Dynamics
Homework 3

Due on 02/19/19 at 11:59 pm (through Catcourses)
Maximum points: 100

1. (**55 points**) Write a code to solve the 1-D heat equation (given below) using an explicit forward-time centered-space finite difference scheme.

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{1}$$

The computational domain extends from $x = 0$ to $x = 1$. The boundary conditions are given by $T(0,t) = 700\ K$ and $T(1,t) = 300\ K$. The initial condition can be assumed to be $T(x,0) = 500\ K$. Assume $\alpha = 1$ and $\Delta x = 0.05$

   (a) Use $\alpha \Delta t/(\Delta x)^2 = 0.5$ and plot the temperature profiles (on the same plot to show evolution) at $t = \Delta t$, $2\Delta t$, $5\Delta t$, $10\Delta t$, $20\Delta t$, $50\Delta t$, $100\Delta t$, and $150\Delta t$.

   (b) Does the temperature profile reach a steady state.

   (c) Now increase $\alpha \Delta t/(\Delta x)^2$ to 1.5 and re-run your code. Compare your results with those obtained above.

2. (**15 points**) The DuFort-Frankel method for solving the heat equation requires solution of the difference equation

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = \frac{\alpha}{(\Delta x)^2}(u_{j+1}^n - u_j^{n+1} - u_j^{n-1} + u_{j-1}^n) \tag{2}$$

Determine the stability restrictions for this scheme.

3. (**15 points**) An implicit scheme for solving the heat equation is given by

$$u_j^{n+1} = u_j^n + \frac{\alpha \Delta t}{(\Delta x)^2}\left[\frac{1}{3}(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}) + \frac{2}{3}(u_{j+1}^n - 2u_j^n + u_{j-1}^n)\right] \tag{3}$$

Determine the stability restrictions for this scheme.

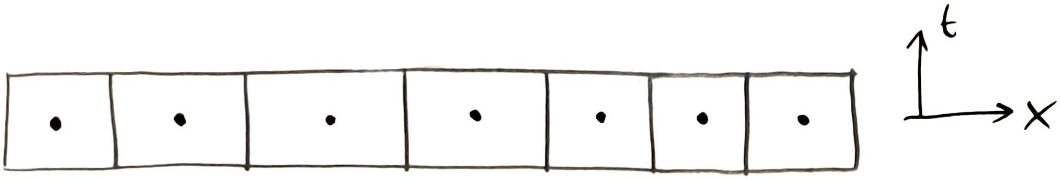
4. (**15 points**) The leap frog method for solving the 1-D wave equation is given by

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + c\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} = 0 \tag{4}$$

Determine the stability restrictions for this scheme.

Write a code to solve 1-D heat equation (given below) using an explicit/foward-time/centered space/finite difference scheme

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}$$



foward - time

$$\left.\frac{\partial T}{\partial t}\right|_{i,j} = \frac{T_i^{N+1} - T_i^N}{\Delta t}$$

centered scheme

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{i,j} = \frac{T_{i+1}^N - 2T_i^N + T_{i-1}^N}{(\Delta x)^2}$$

total equation

$$\frac{T_i^{N+1} - T_i^N}{\Delta t} = \frac{\alpha \left[T_{i+1}^N - 2T_i^N + T_{i-1}^N \right]}{(\Delta x)^2}$$

B.C's

$x = 0 \rightarrow x = 1$ $\qquad$ $\alpha = 0.1$

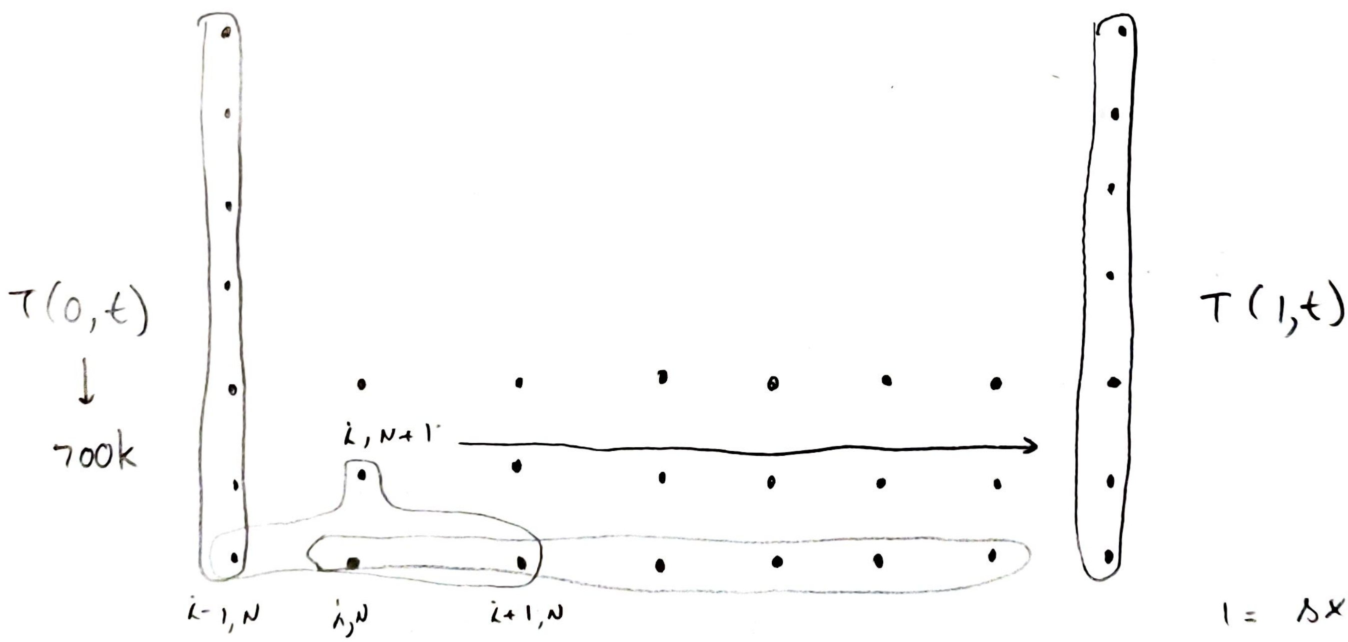
$T(0,t) = 700k$ $\qquad$ $\Delta x = 0.05$

$T(1,t) = 300k$

$T(x,0) = 500k$

$$T_i^{N+1} - T_i^N = \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right)\left[T_{i+1}^N - 2T_i^N + T_{i-1}^N\right]$$

$$T_i^{N+1} = \left(\frac{\alpha \Delta t}{(\Delta x)^2}\right)\left[T_{i+1}^N - 2T_i^N + T_{i-1}^N\right] + T_i^N$$



$1 = \Delta x(n)$

$1 = 0.5\,(n)$

$T(x,0) = 500k$

$$\underset{1}{u_j^{N+1}} - \underset{2}{u_j^{N-1}} = 2\gamma\left(\underset{3}{u_{j+1}^{N}} - \underset{4}{u_j^{N+1}} - \underset{5}{u_j^{N-1}} + u_{j-1}^{N}\right)$$

$$\gamma = \frac{\alpha \Delta t}{(\Delta x)^2}$$

1 $$\frac{b_m(t+\Delta t)\, e^{ik_m x}}{b_m(t)\, e^{ik_m x}} = G$$

2 $$\frac{b_m(t-\Delta t)\, e^{ik_m x}}{b_m(t)\, e^{ik_m x}} = \frac{1}{G}$$

$$\frac{e^{at-a\Delta t}}{e^{at}} = e^{-a\Delta t} = \frac{1}{e^{a\Delta t}} = \frac{1}{G}$$

3 $$\frac{b_m(t)\, e^{ik_m(x+\Delta x)}}{b_m(t)\, e^{ik_m x}} = e^{ik_m \Delta x}$$

4 $$\frac{b_m(t+\Delta t)\, \cancel{e^{ik_m x}}}{b_m(t)\, \cancel{e^{ik_m x}}} = G$$

5 $$\frac{1}{G}$$

6 $$\frac{b_m(t)\, e^{ik_m(x-\Delta x)}}{b_m(t)\, e^{ik_m x}} = e^{-ik_m \Delta x}$$

$$G - \frac{1}{G} = 2\gamma\left(e^{i k_m \Delta x} - G - \frac{1}{G} + e^{-i k_m \Delta x} \right)$$

$$e^{i k_m \Delta x} + e^{-i k_m \Delta x} = e^{i\beta} + e^{-i\beta} = \cos\beta + \cancel{i\sin\beta} + \cos\beta - \cancel{i\sin\beta} = 2\cos\beta$$

$$G - \frac{1}{G} = 2\gamma\left(2\cos\beta - G - \frac{1}{G} \right)$$

Solving for G this gives...

$$G = \frac{2\gamma\cos\beta \pm \sqrt{1 - 2\gamma^2 + 2\gamma^2\cos[2\beta]}}{1 + 2\gamma}$$

$$\rightarrow \quad 1 - 2\gamma^2\left(1 - \cos^2\beta + \sin^2\beta\right)$$

$$1 - 2\gamma^2\left(2\sin^2\beta\right) \rightarrow 1 - 4\gamma^2\sin^2\beta$$

$$G = \frac{2\gamma\cos\beta \pm \sqrt{1 - 4\gamma^2\sin^2\beta}}{1 + 2\gamma}$$

$$\left| \frac{2\gamma \cos\beta + \sqrt{1 - 4\gamma^2 \sin^2\beta}}{1 + 2\gamma} \right| \geq 1$$

$$\left(\left(\frac{2\gamma \cos\beta + \sqrt{1 - 4\gamma^2 \sin^2\beta}}{1 + 2\gamma} \right)^2 \right)^{1/2} \geq 1$$

$$\frac{2\gamma \cos\beta + \sqrt{1 - 4\gamma \sin^2\beta}}{1 + 2\gamma} \geq 1$$

$$2\gamma \cos\beta + \sqrt{1 - 4\gamma \sin^2\beta} \geq 1 + 2\gamma$$

$$\gamma \leq \frac{-1 + \cos\beta - \sin\beta}{1 - 2\cos\beta + \cos^2\beta}$$

$[1, -1]$

↳ at $\cos(x) = 1$

$$\gamma \leq \frac{-1 + \cos\beta - \sin\beta}{(-1 + \cos\beta)^2}$$

$$\gamma \leq \frac{(-1 + 1 - 0)}{(-1 + 1)}$$ → Nope

at $\cos(x) \to -1$

$$\gamma \leq \frac{-1-1-0}{-1-1}$$

$$\left[\gamma \leq \frac{-2}{-2} = 1 \right]$$

unconditionally stable

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{\alpha}{(\Delta x)^2}\left[\frac{1}{3}\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right) + \frac{2}{3}\left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)\right]$$

LHS

$$\frac{b_m(t+\Delta t)e^{ik_m x} - b_m(t)e^{ik_m x}}{\Delta t}$$

$$\left[\frac{b_m(t+\Delta t)e^{ik_m x}}{b_m(t)e^{ik_m x}} - \frac{b_m(t)e^{ik_m x}}{b_m(t)e^{ik_m x}}\right]\frac{1}{\Delta t}$$

$$\frac{1}{\Delta t}[G - 1]$$

RHS

A

$$\frac{\alpha}{(\Delta x)^2}\left[\frac{1}{3}\left(b_m(t+\Delta t)e^{ik_m(x+\Delta x)} - 2b_m(t+\Delta t)e^{ik_m x} + b_m(t+\Delta t)e^{ik_m(x-\Delta x)}\right)\right.$$

$$\left. + \frac{2}{3}\left(b_m(t)e^{ik_m(x+\Delta x)} - 2b_m(t)e^{ik_m x} + b_m(t)e^{ik_m(x-\Delta x)}\right)\right]$$

B

(A)

$$\frac{\alpha}{(\Delta x)^2}\left[\left(\frac{1}{3}\right)\left(\frac{G_m(t+\Delta t)\, e^{ik_m(x+\Delta x)}}{G_m(t)\, e^{ik_m x}} - \frac{2G_m(t)\, e^{ik_m x}}{\cancel{G_m(t) e^{ik_m x}}} + \frac{G_m(t+\Delta t)\, e^{ik_m(x-\Delta x)}}{G_m(t)\, e^{ik_m x}}\right)\right]$$

$$\frac{\alpha}{(\Delta x)^2}\left[\left(\frac{1}{3}\right)\left(G e^{ik_m \Delta x} - 2G + G e^{-ik_m \Delta x}\right)\right]$$

(B)

$$\frac{\alpha}{(\Delta x)^2}\left[\left(\frac{2}{3}\right)\left(\frac{G_m(t) e^{ik_m(x+\Delta x)}}{G_m(t)\, e^{ik_m x}} - \frac{2G_m(t)\, e^{ik_m x}}{\cancel{G_m(t) e^{ik_m x}}} + \frac{G_m(t)\, e^{ik_m(x-\Delta x)}}{G_m(t)\, e^{ik_m(x)}}\right)\right]$$

$$\frac{\alpha}{(\Delta x)^2}\left[\left(\frac{2}{3}\right)\left(e^{ik_m \Delta x} - 2 + e^{-ik_m \Delta x}\right)\right]$$

AA BB

$$G-1 = \frac{\alpha \Delta t}{(\Delta x)^2}\left[\left(\frac{1}{3}\right)\left(G e^{ik_m\Delta x} - 2G + G e^{-ik_m \Delta x}\right) + \left(\frac{2}{3}\right)\left(e^{ik_m \Delta x} - 2 + e^{-ik_m\Delta x}\right)\right]$$

(AA)

$$G\left(e^{i\beta} + e^{-i\beta}\right) - 2G = G\left(\cos\beta + \cancel{i\sin\beta} + \cos\beta - \cancel{i\sin\beta}\right) - 2G$$

$$= G(2\cos\beta) - 2G$$

(BB)

$(2\cos\beta) - 2$

$$\gamma = \frac{\alpha \Delta t}{(\Delta x)^2}$$

$$G-1 = \gamma\left[\left(\tfrac{1}{3}\right)\left(G(2\cos\beta)-2G\right) + \left(\tfrac{2}{3}\right)\left(2\cos\beta - 2\right)\right]$$

Solving for G gives...

$$G \rightarrow \frac{3-4\gamma+4\gamma\cos\beta}{3+2\gamma-2\gamma\cos\beta}$$

$$\left|\frac{3-4\gamma+4\gamma\cos\beta}{3+2\gamma-2\gamma\cos\beta}\right| \leq 1$$

$$\cancel{3}-4\gamma+4\gamma\cos\beta \leq \cancel{3}+2\gamma-2\gamma\cos\beta$$

$$-6\gamma \leq -6\gamma\cos\beta$$

$$1 \geq \cos\beta$$

$$\frac{3-4\gamma+4\gamma\cos\beta}{3+2\gamma-2\gamma\cos\beta} \geq -1$$

$$\frac{3-4\gamma+4\gamma\cos\beta}{3+2\gamma-2\gamma\cos\beta} \geq -1$$

$$3-4\gamma+4\gamma\cos\beta \geq -3-2\gamma+2\gamma\cos\beta$$

$$-6 \geq -2\gamma+2\gamma\cos\beta+4\gamma-4\gamma\cos\beta$$

$$-6 \geq 2\gamma-2\gamma\cos\beta$$

$$-6 \geq 2\gamma(1-\cos\beta)$$

$$\frac{-6}{2(1-\cos\beta)} \geq \gamma \qquad \cos\beta \quad [-1,1]$$

$$\frac{-6}{2(1-0.5)}$$

$$\frac{-6}{2(1+1)} \geq \gamma \qquad \frac{-6}{2} \to -3 \qquad \frac{-6}{2(0.5)} = -6$$

$$\left[\frac{-6}{4} \geq \gamma\right]$$

$$\left[-\frac{3}{2} \geq \gamma\right]$$

[conditionally stable]

$$\frac{u_j^{N+1} - u^{N-1}}{2\delta t} + \frac{c\left(u_{j+1}^N - u_{j-1}^N\right)}{2\delta x} = 0$$

$$\frac{u_j^{N+1} - u^{N-1}}{2\delta t} = \left(\frac{c}{2\delta x}\right)\left(u_{j-1}^N - u_{j+1}^N\right)$$

LHS

$$\left(\frac{b_m(t+\delta t)e^{ik_m x} - b_m(t-\delta t)e^{ik_m x}}{2\delta t}\right)\left(\frac{1}{b_m(t)e^{ik_m x}}\right)$$

$$\left(\frac{1}{2\delta t}\right)\left(b_m(t+\delta t)\cancel{e^{ik_m x}}/b_m(t)\cancel{e^{ik_m x}} - b_m(t-\delta t)\cancel{e^{ik_m x}}/b_m(t)\cancel{e^{ik_m x}}\right)$$

$$\left(\frac{1}{2\delta t}\right)\left(G - \frac{1}{G}\right)$$

RHS

$$\left(\frac{c}{2\delta x}\right)\left(\frac{b_m(t)\,e^{ik_m(x+\delta x)}}{b_m(t)\,e^{ik_m(x)}} - \frac{b_m(t)\,e^{ik_m(x-\delta x)}}{b_m(t)\,e^{ik_m(x)}}\right)$$

$$\left(\frac{c}{2\Delta x}\right)\left(e^{ik_m \Delta x} - e^{-ik_m \Delta x}\right)$$

$$\left(\frac{c}{2\Delta x}\right)\left(e^{i\beta} - e^{-i\beta}\right)$$

ok... the equation is

$$\left(\frac{1}{2\Delta t}\right)\left(G - \frac{1}{G}\right) = \frac{c}{2\Delta x}\left(e^{i\beta} - e^{-i\beta}\right)$$

$$G - \frac{1}{G} = \frac{c\,\Delta t}{\Delta x}\left(e^{i\beta} - e^{-i\beta}\right)$$

A

$$G - \frac{1}{G} = \gamma\left(e^{i\beta} - e^{-i\beta}\right)$$

A $\cancel{\cos\beta} + i\sin\beta - (\cancel{\cos\beta} - i\sin\beta)$

$= 2i\sin\beta$

$$G - \frac{1}{G} = \gamma\left(2i\sin\beta\right)$$

$$G = \frac{1}{2}\left[2i\gamma\sin\beta \pm \sqrt{4 - 4\gamma^2\sin^2\beta}\right]$$

$$G = \gamma \sin\beta\, i \pm \sqrt{1-\gamma^2 \sin^2\beta}$$ (now, lets do G)

$$\left| \gamma \sin\beta\, i + \sqrt{1-\gamma^2\sin^2\beta} \right| \le 1 \quad \text{A}$$

$$\left| \gamma \sin\beta\, i - \sqrt{1-\gamma^2\sin^2\beta} \right| \le 1 \quad \text{B}$$

(for A)

$$\left[\, |a+bi| = \sqrt{a^2+b^2} \,\right]$$

$$\sqrt{(\gamma \sin\beta)^2 + \left(\left(1-\gamma^2\sin^2\beta\right)^{1/2}\right)^2} \le 1$$

$$\left(\cancel{\gamma^2\sin^2\beta} + \left(1-\cancel{\gamma^2\sin^2\beta}\right)\right)^{1/2} \le 1$$

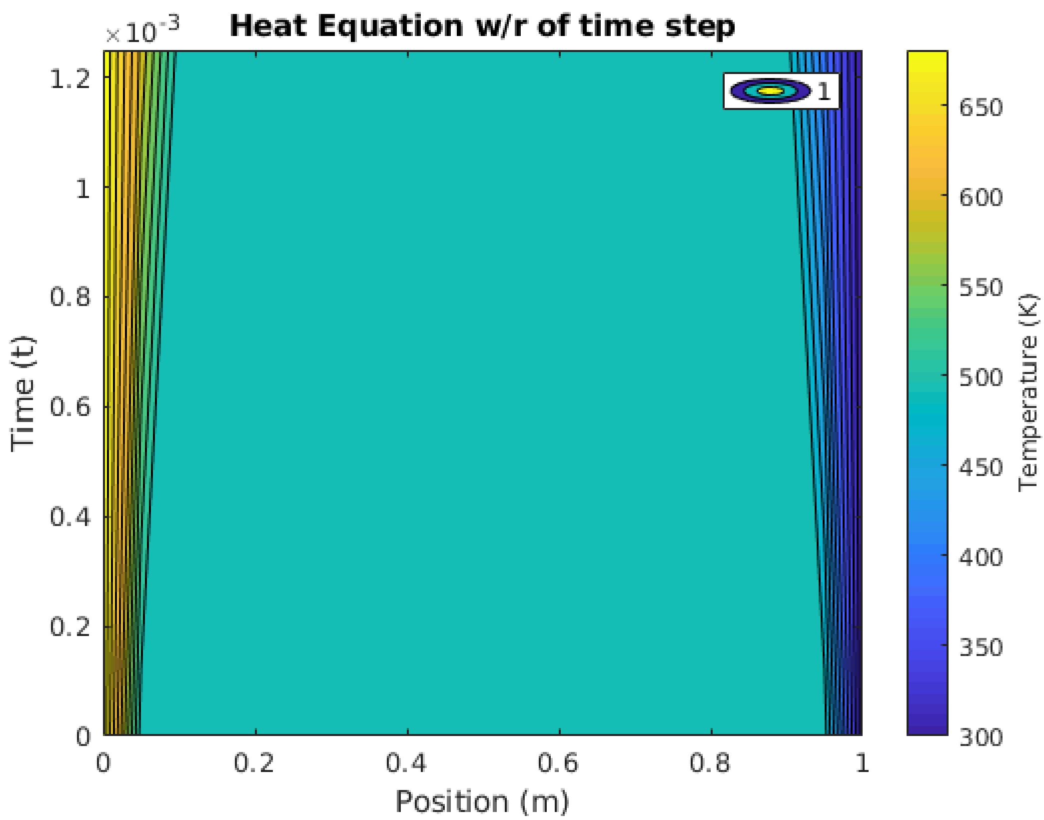
$$(1)^{1/2} \le 1$$

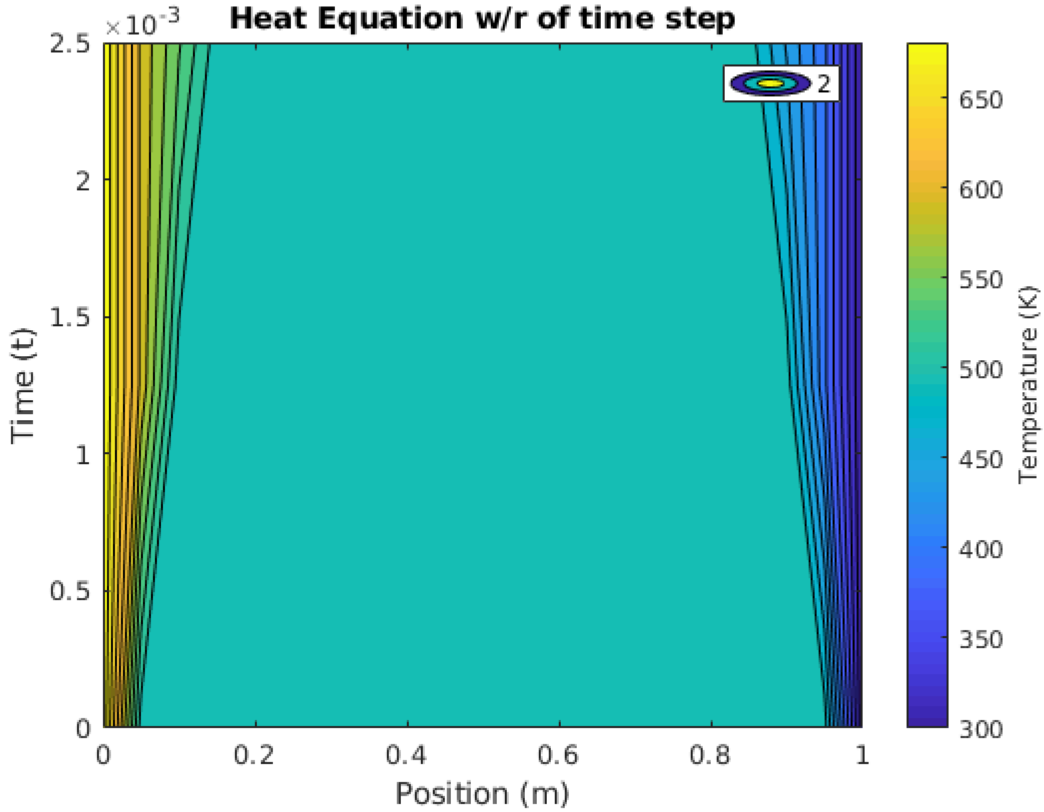
$$[1 \le 1]$$ Unconditionally stable

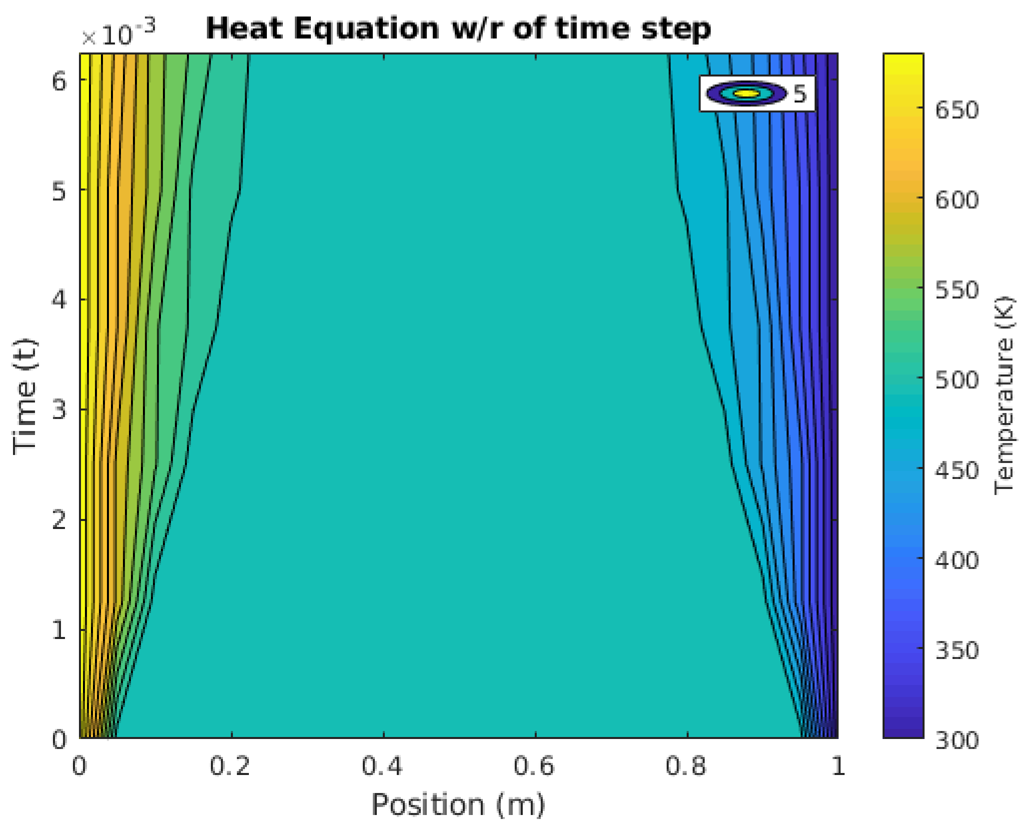
→ is going to be the same for B...

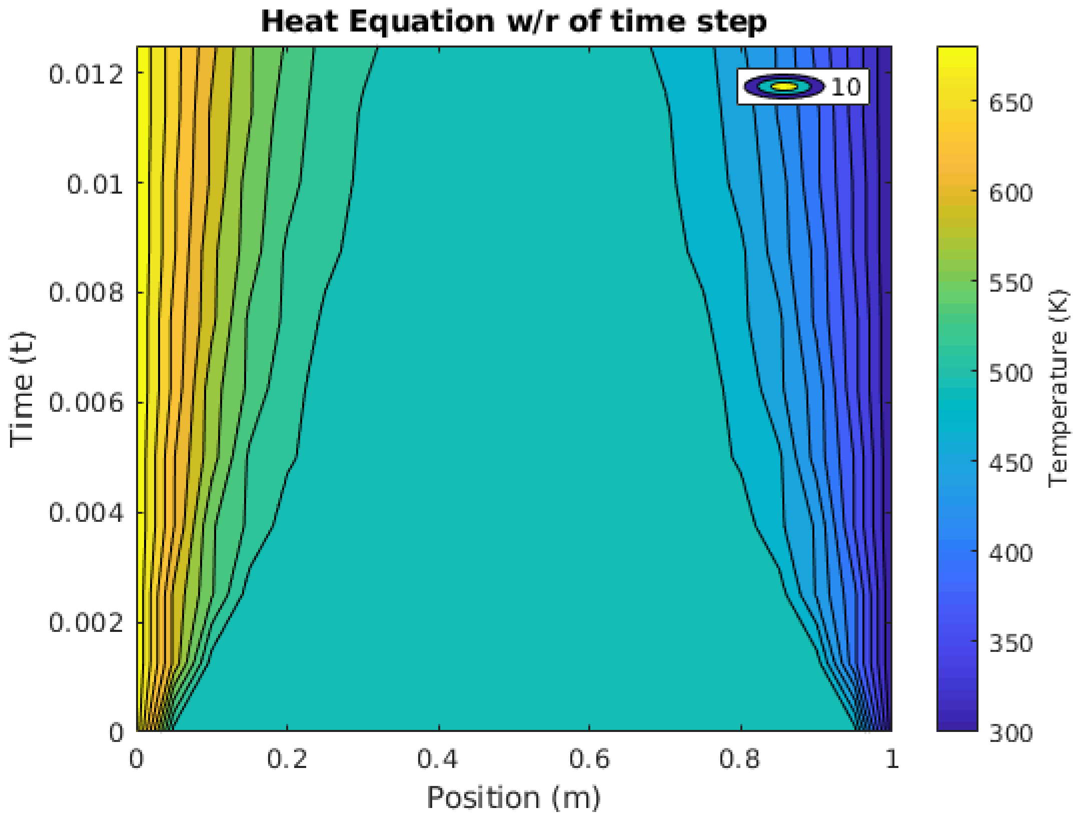
```
% The following script solves the 1-D heat equation using a
% explicit foward-time centered space finite difference scheme
% For this case I use a Y of 0.5

A = [1,2,5,10,20,50,100,150];
heatTransferPlots(0.5,A);
%temperaturePosition(0.5,A,length(A));
%heatTransferPlots(1.5,A);
```

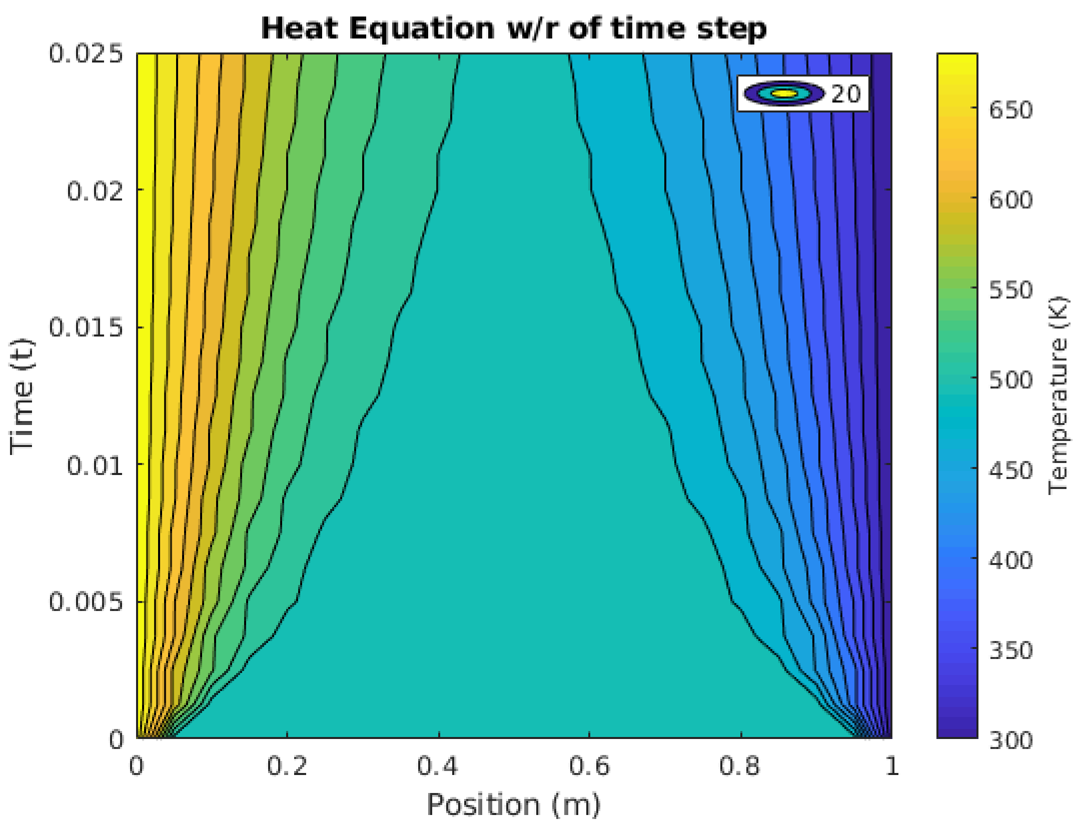
```
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
```

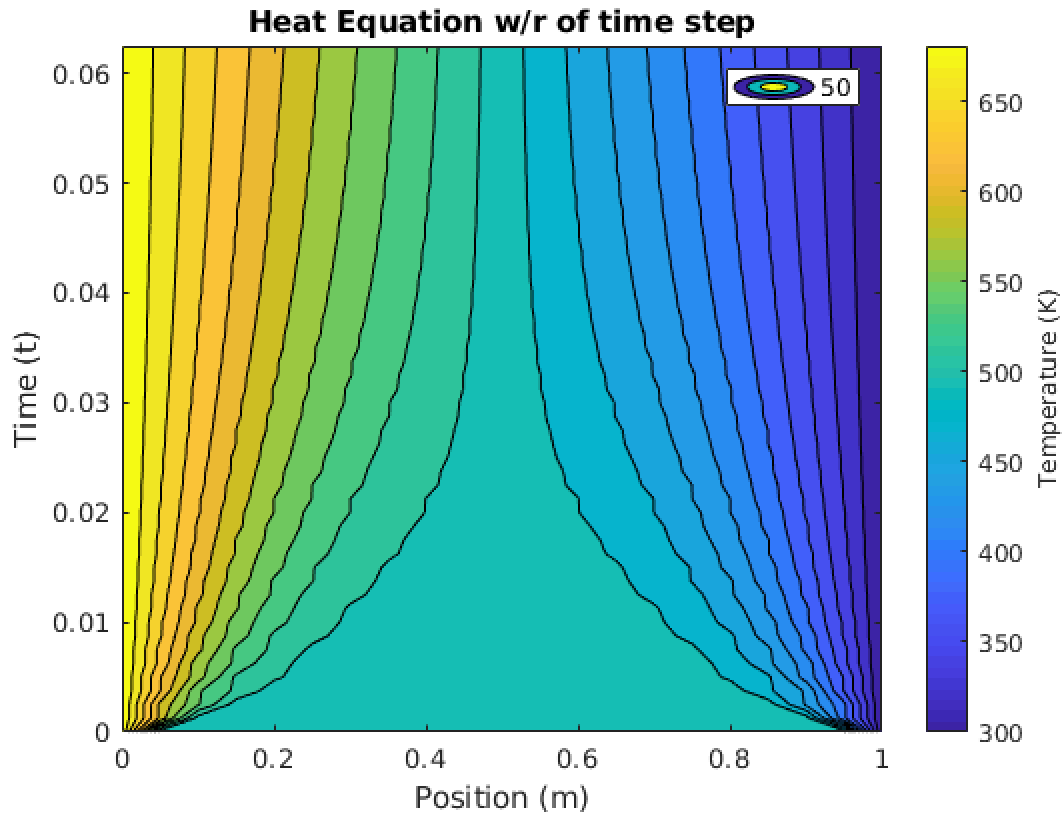


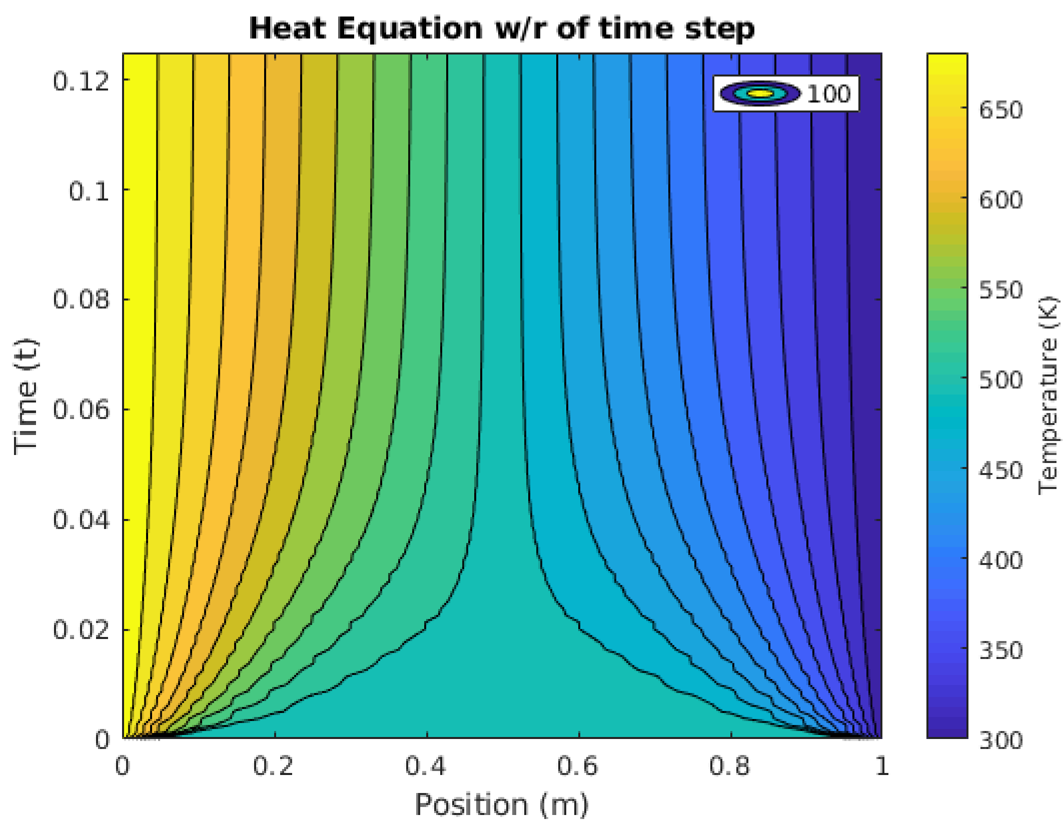

Heat Equation w/r of time step
×10^-3
2.5
2
1.5
1
0.5
0
Time (t)
0
0.2
0.4
0.6
0.8
1
Position (m)
2
650
600
550
500
450
400
350
300
Temperature (K)

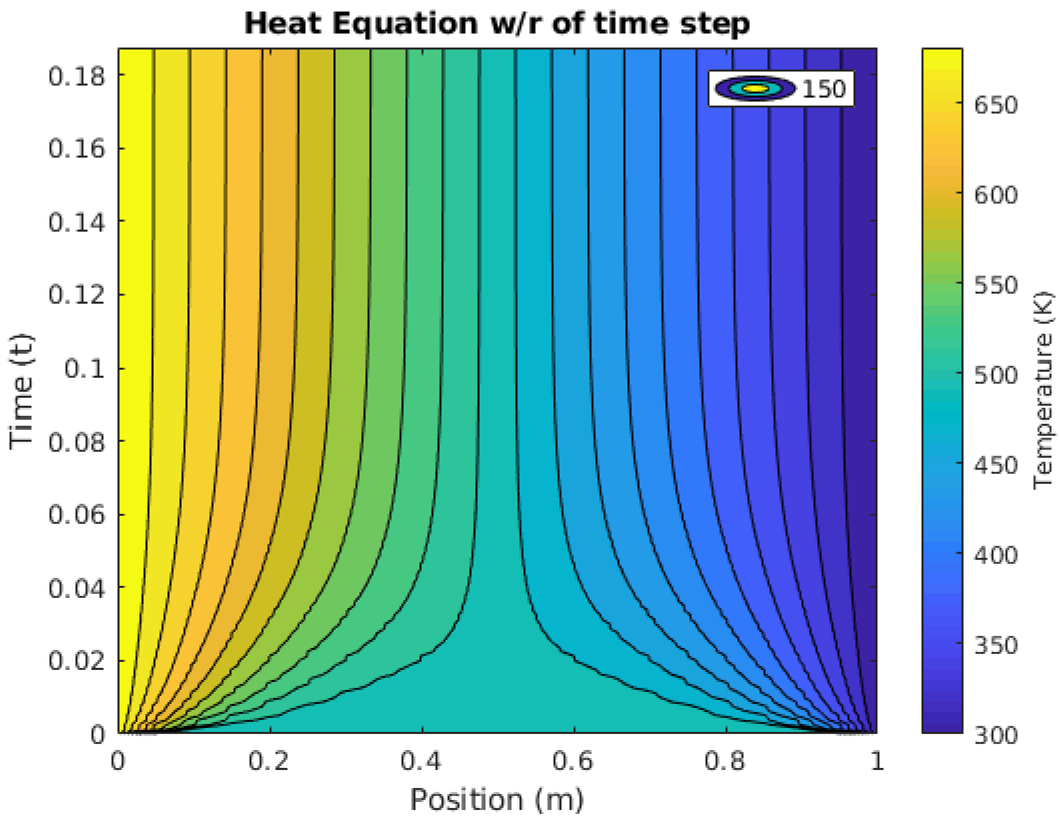

Heat Equation w/r of time step
×10^-3
6
5
4
3
2
1
0
Time (t)
0
0.2
0.4
0.6
0.8
1
Position (m)
5
650
600
550
500
450
400
350
300
Temperature (K)


Heat Equation w/r of time step
10
Time (t)
Position (m)
Temperature (K)
0.012
0.01
0.008
0.006
0.004
0.002
0
0.2
0.4
0.6
0.8
1
650
600
550
500
450
400
350
300


Heat Equation w/r of time step
20
Time (t)
Position (m)
Temperature (K)
0.025
0.02
0.015
0.01
0.005
0
0.2
0.4
0.6
0.8
1
650
600
550
500
450
400
350
300

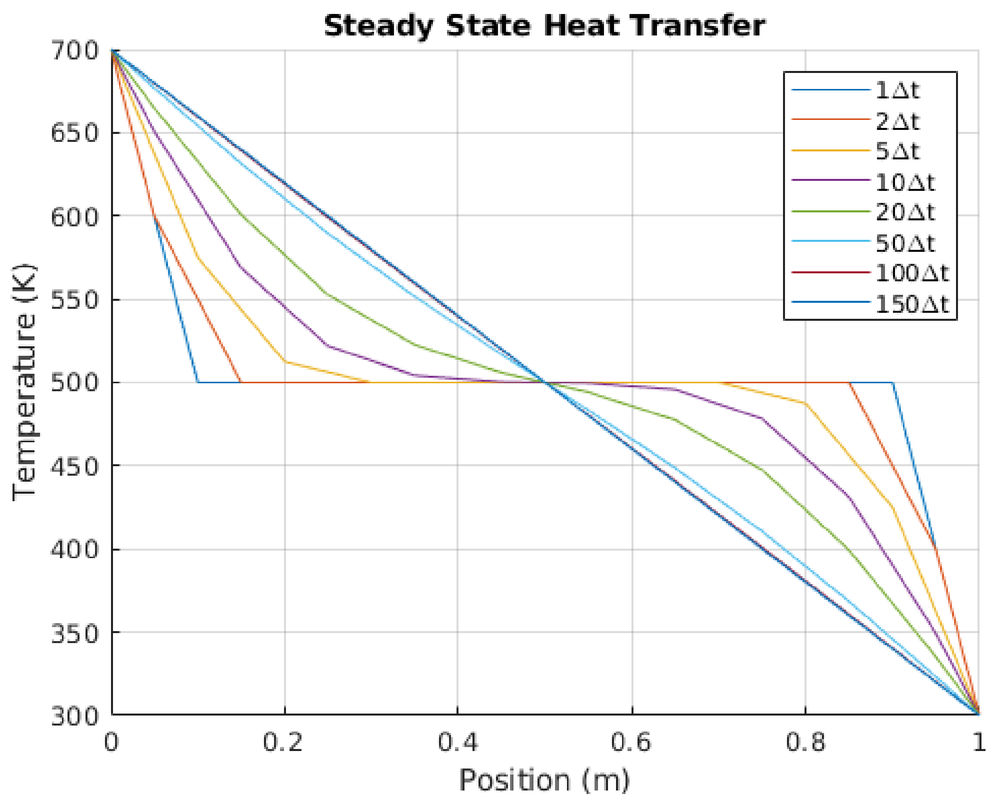

Heat Equation w/r of time step
Time (t)
Position (m)
50
Temperature (K)
0.06
0.05
0.04
0.03
0.02
0.01
0
0.2
0.4
0.6
0.8
1
650
600
550
500
450
400
350
300


Heat Equation w/r of time step
Time (t)
Position (m)
100
Temperature (K)
0.12
0.1
0.08
0.06
0.04
0.02
0
0.2
0.4
0.6
0.8
1
650
600
550
500
450
400
350
300



Published with MATLAB® R2018a

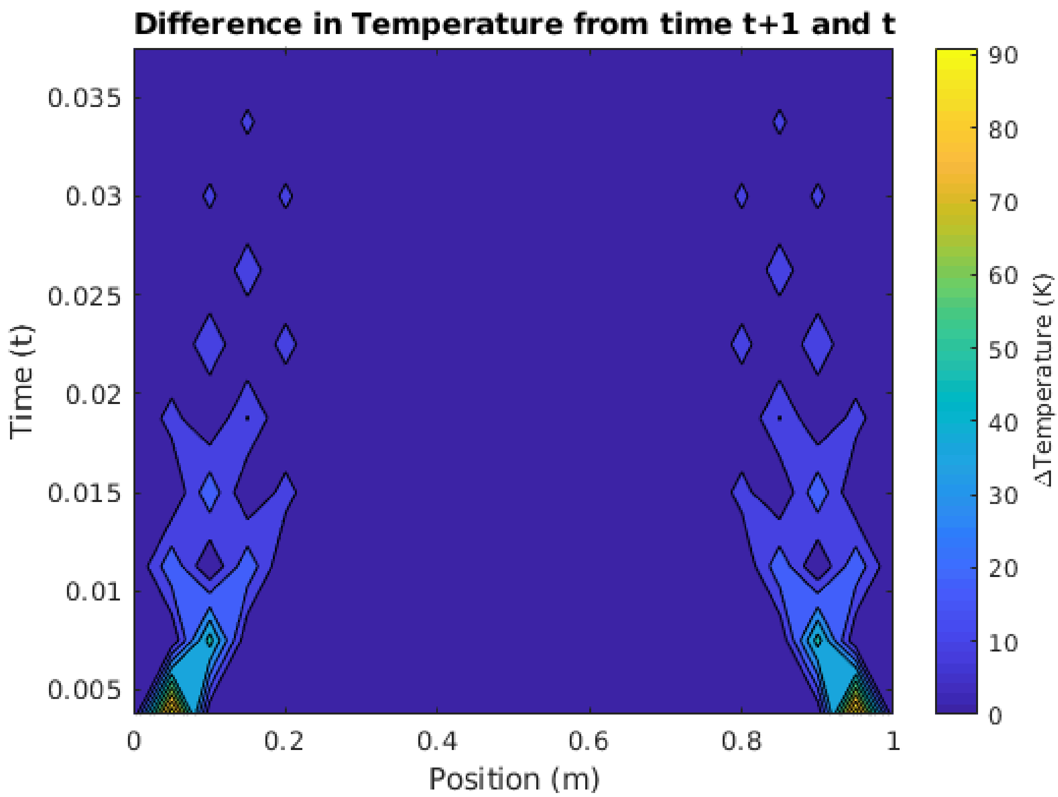
```
% The following script solves the 1-D heat equation using a
% explicit foward-time centered space finite difference scheme
% For this case I use a Y of 0.5
% This time I only plot position and temperature
clc
clear all
close all

A = [1,2,5,10,20,50,100,150];
%heatTransferPlots(0.5,A);
temperaturePosition(0.5,A,length(A));
%heatTransferPlots(1.5,A);
```



Published with MATLAB® R2018a

```
% The following script compare the value at a initial iteration
% with the following iteration in order to see whenever there is a
% convergence. In other words, if I see in the countour plot a value
% other than zero that means that more iterations are needed in order
% to reach convergence. Tested until 10 time steps in order to see
% results

clc;
clear all;

dx = 0.05;
Y = 1.5;
dt = Y*(dx)^2;

n_time = 10;

% this define columns needed
n_x = (1/dx)+1;
% this define rows needed
n_t = n_time+1;

T = zeros(n_t,n_x);
%this gives us our boundary conditions
boundaryx0 = 700*ones(n_t,1);
boundaryx1 = 300*ones(n_t,1);
boundaryt0 = 500*ones(1,n_x-2);

T(:,1) = boundaryx0;
T(:,n_x) = boundaryx1;
T(n_t,2:n_x-1) = boundaryt0;

if n_t>1
    for i=n_t-1:-1:1
        for w=2:1:n_x-1
            T(i,w) = (0.5)*(T(i+1,w+1)-2*T(i+1,w)+T(i+1,w-1))+T(i
+1,w);
        end
    end
end

% for x values the following is done
valuedx = 0:dx:1;

% for time values the following is done
valuedt = n_time*dt:-dt:0;

matrix_x_direction = valuedx.*ones(n_t,n_x);
matrix_t_direction = (valuedt)'.*ones(n_t,n_x);

change_t_direction_d = matrix_t_direction(1:n_t-1,:);
matrix_x_direction_d = matrix_x_direction(1:n_t-1,:);
change_T_direction1 = zeros((n_t-1),n_x);
```

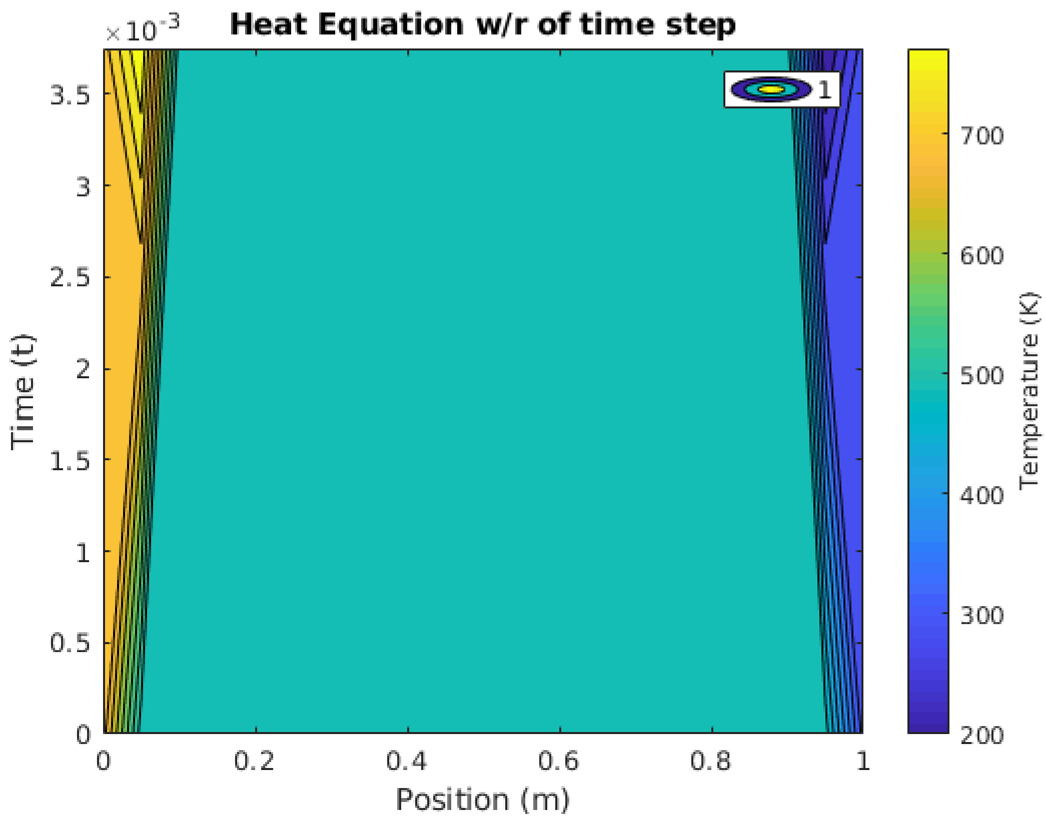
```
for i=1:1:n_t-1
    change_T_direction1(i,:) = abs(T(i,:)-T(i+1,:));
end


figure(1)
contourf(matrix_x_direction_d,change_t_direction_d,change_T_direction1,n_time)
c = contourcbar;
c.Label.String = '\DeltaTemperature (K)';
title('Difference in Temperature from time t+1 and t');
xlabel('Position (m)');
ylabel('Time (t)');
```

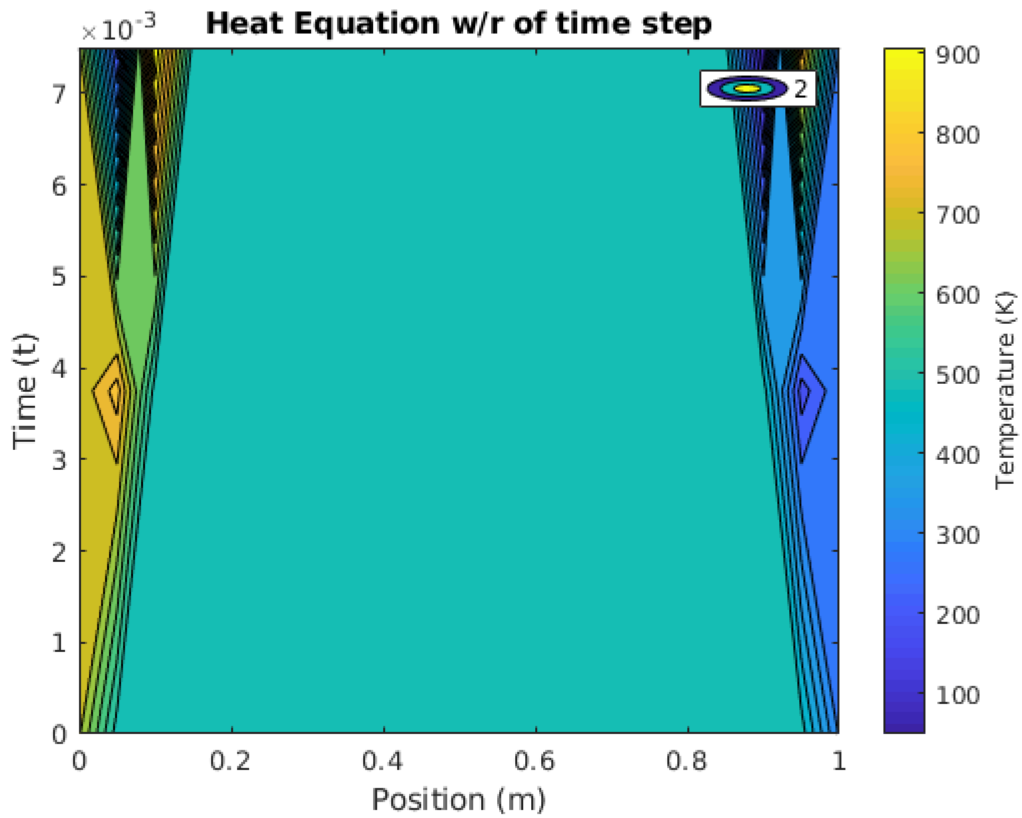


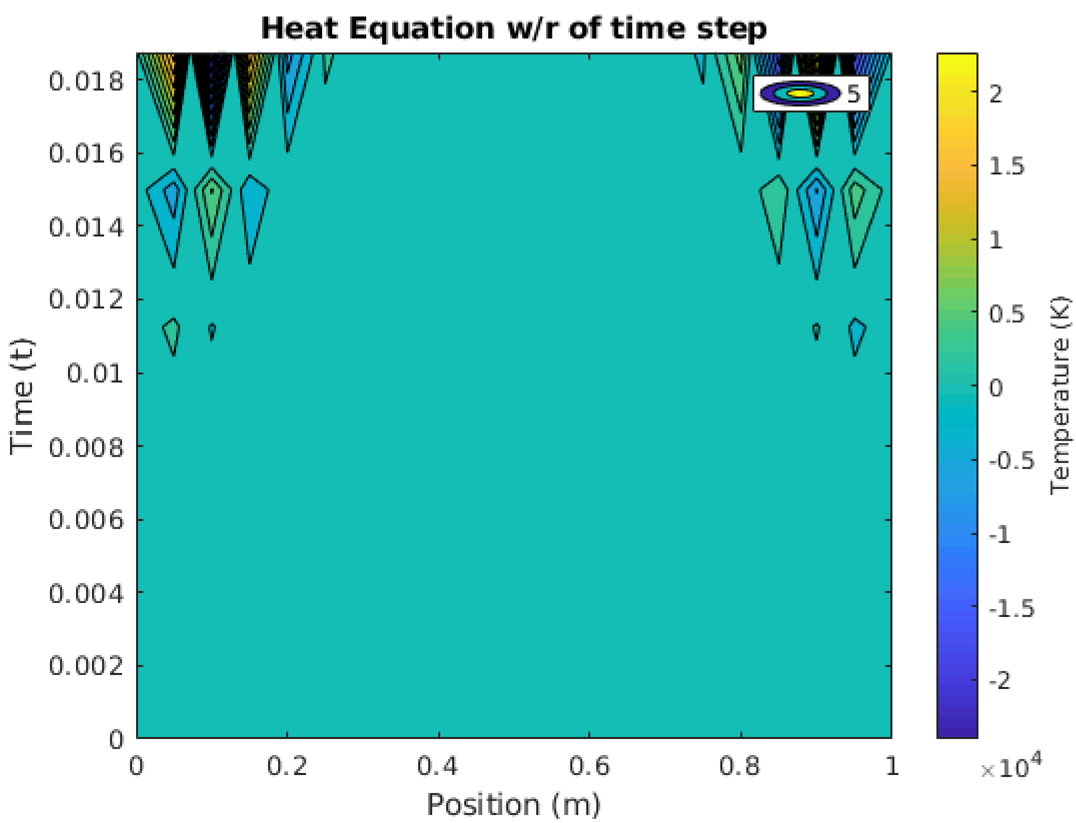
Published with MATLAB® R2018a

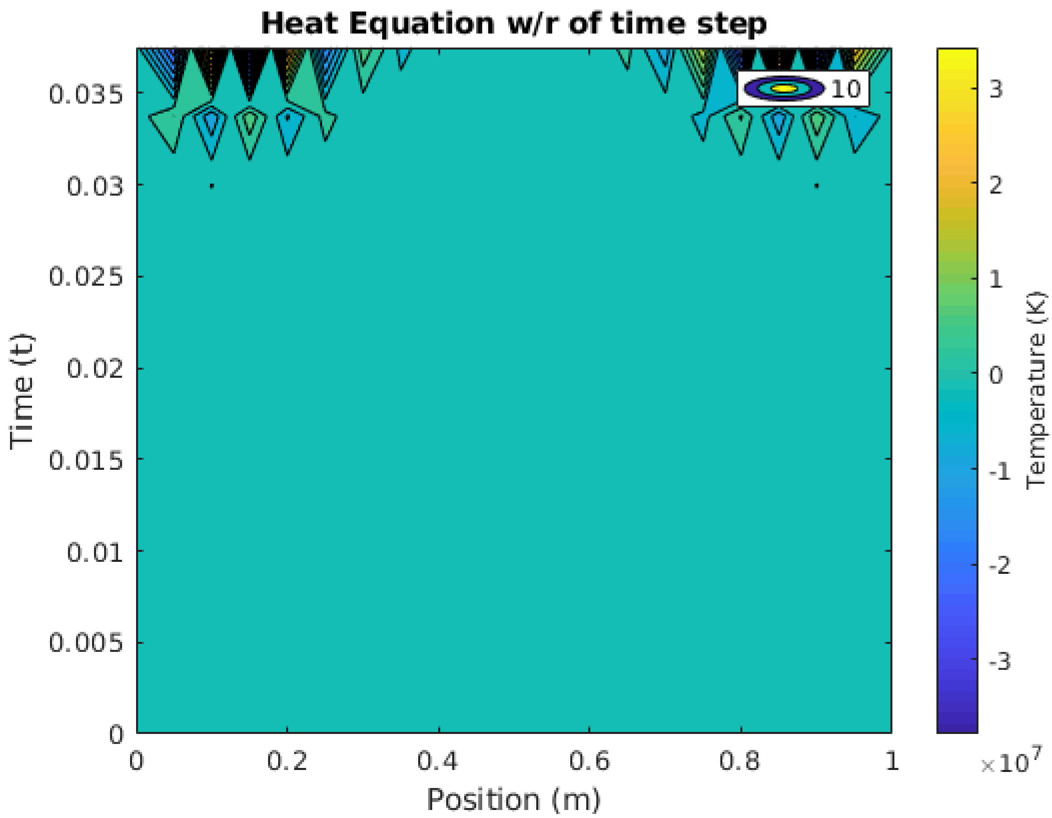
```
% The following script solves the 1-D heat equation using a
% explicit foward-time centered space finite difference scheme
% For this case I use a Y of 1.5
% This time I will plot the contour plot
clc
clear all
close all

A = [1,2,5,10,20,50,100,150];
%heatTransferPlots(0.5,A);
%temperaturePosition(0.5,A,length(A));
heatTransferPlots(1.5,A);
```

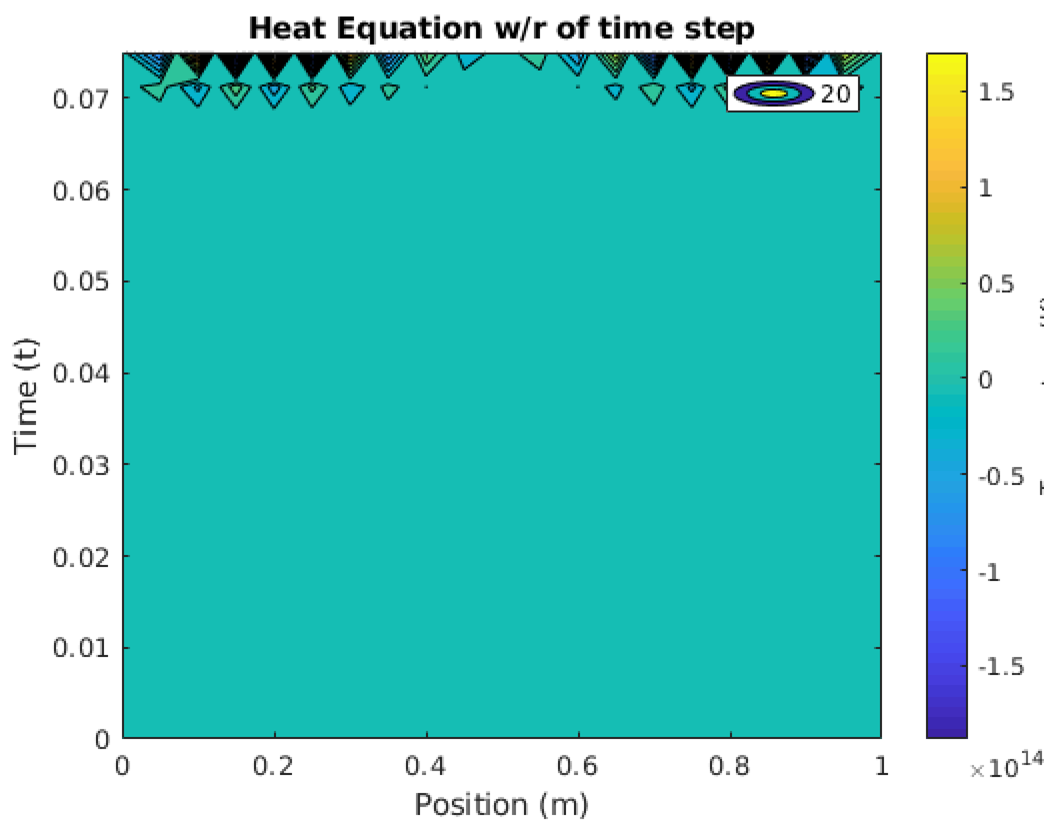
```
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
Warning: Ignoring extra legend entries.
```

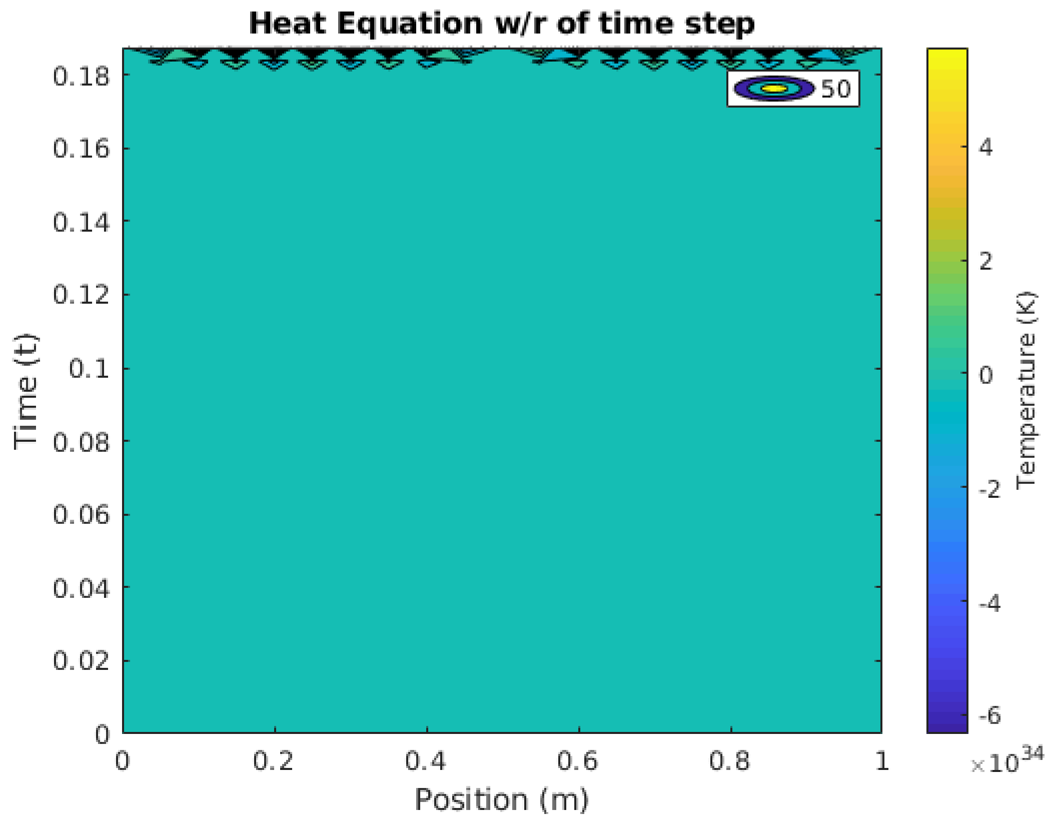


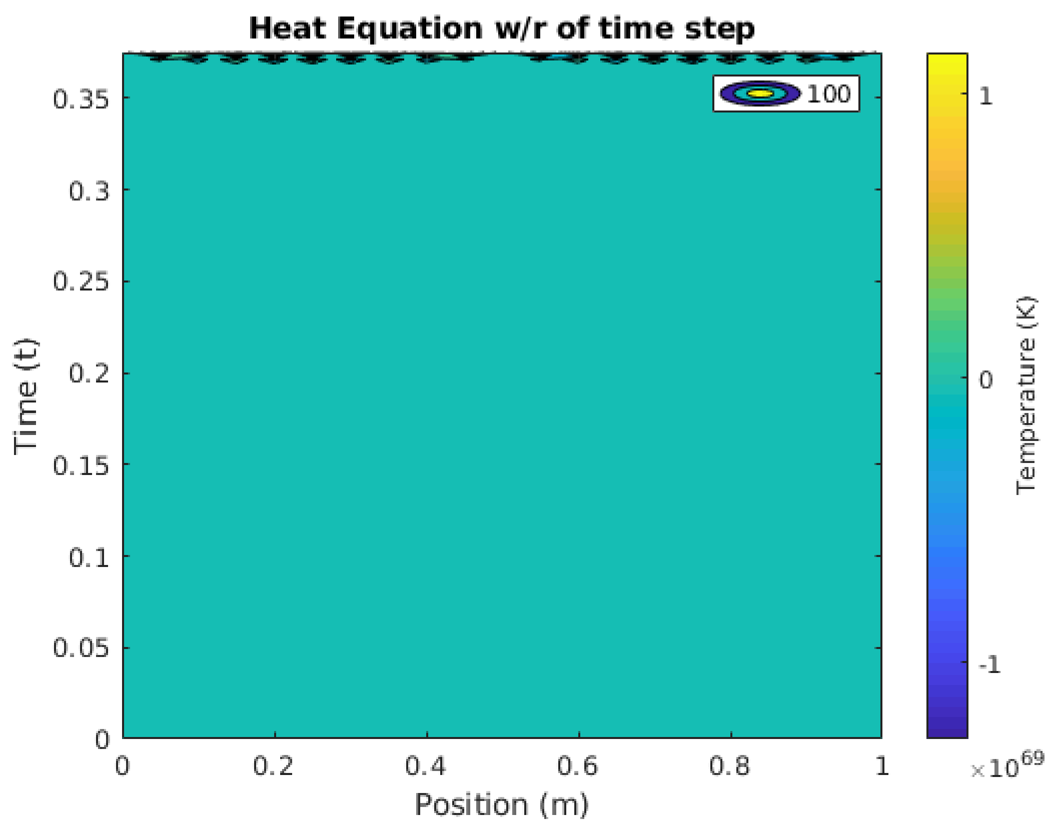

Heat Equation w/r of time step
$\times 10^{-3}$
Time (t)
Position (m)
Temperature (K)
2
0
1
2
3
4
5
6
7
0
0.2
0.4
0.6
0.8
1
100
200
300
400
500
600
700
800
900

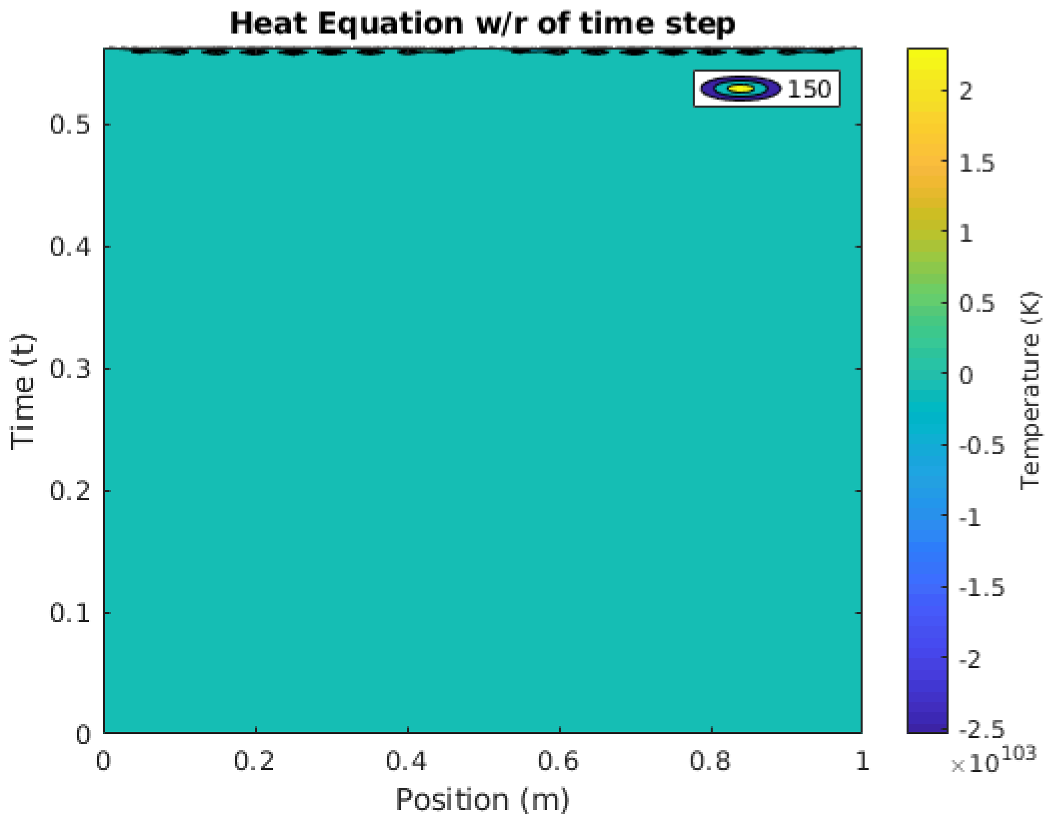

Heat Equation w/r of time step
Time (t)
Position (m)
Temperature (K)
$\times 10^{4}$
5
0
0.002
0.004
0.006
0.008
0.01
0.012
0.014
0.016
0.018
0
0.2
0.4
0.6
0.8
1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Heat Equation w/r of time step
Time (t)
Position (m)
Temperature (K)
10
×10^7


Heat Equation w/r of time step
Time (t)
Position (m)
Temperature (K)
20
×10^14

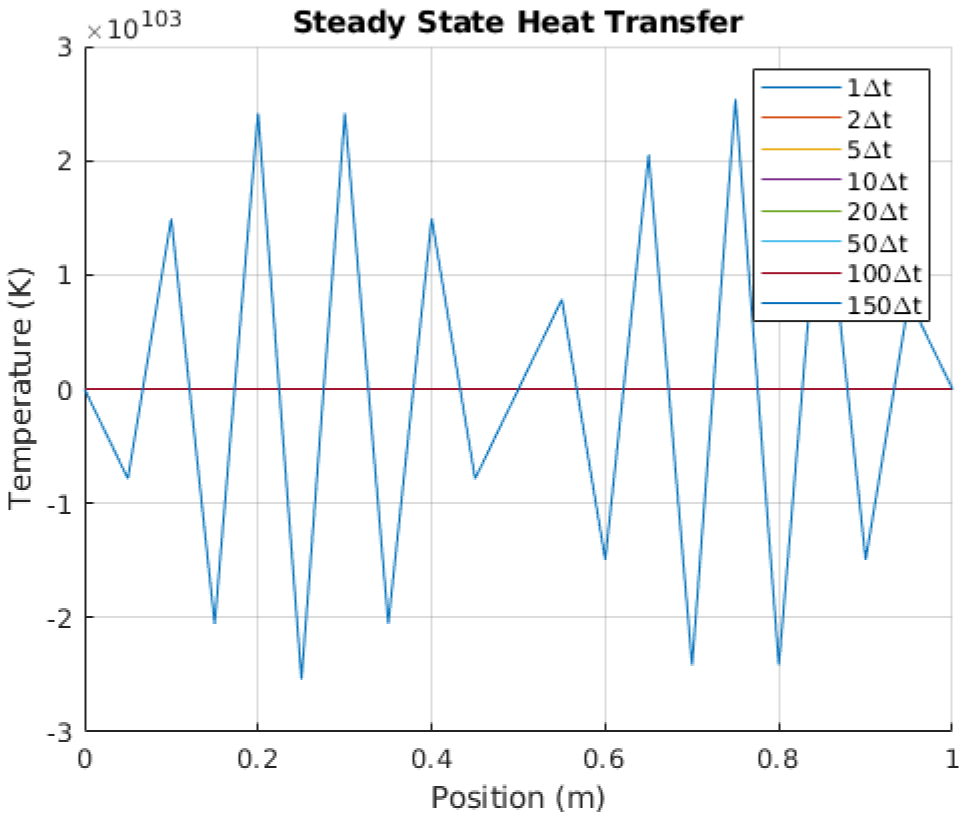

Heat Equation w/r of time step
50
Time (t)
Position (m)
Temperature (K)
$\times10^{34}$


Heat Equation w/r of time step
100
Time (t)
Position (m)
Temperature (K)
$\times10^{69}$



Published with MATLAB® R2018a

```
% The following script solves the 1-D heat equation using a
% explicit foward-time centered space finite difference scheme
% For this case I use a Y of 1.5
% This time I will plot temperature with respect of position
% clearly this is diverging and the answer makes no sence

clc
clear all
close all

A = [1,2,5,10,20,50,100,150];
%heatTransferPlots(0.5,A);
%temperaturePosition(0.5,A,length(A));
%heatTransferPlots(1.5,A);
temperaturePosition(1.5,A,length(A));
```



Published with MATLAB® R2018a

```
function temperaturePosition(Y,n_time,figure_number)

for x=1:1:length(n_time)

dx = 0.05;
% this define columns needed
n_x = (1/dx)+1;
% this define rows needed
n_t = n_time(x)+1;

T = zeros(n_t,n_x);
%this gives us our boundary conditions
boundaryx0 = 700*ones(n_t,1);
boundaryx1 = 300*ones(n_t,1);
boundaryt0 = 500*ones(1,n_x-2);

T(:,1) = boundaryx0;
T(:,n_x) = boundaryx1;
T(n_t,2:n_x-1) = boundaryt0;

if n_t>1
    for i=n_t-1:-1:1
        for w=2:1:n_x-1
            T(i,w) = (Y)*(T(i+1,w+1)-2*T(i+1,w)+T(i+1,w-1))+T(i+1,w);
        end
    end
end

% for x values the following is done
valuedx = 0:dx:1;

figure(figure_number)
hold on
plot(valuedx,T(1,:),'DisplayName',[num2str(n_time(x)) '\Deltat'])
title('Steady State Heat Transfer');
xlabel('Position (m)');
ylabel('Temperature (K)');
legend(gca,'show')
%legend('\Delta',num2str(x),'t')
%legend(num2str(x))
grid on
end

end
```

```
Not enough input arguments.

Error in temperaturePosition (line 4)
for x=1:1:length(n_time)
```

Published with MATLAB® R2018a

```
function heatTransferPlots(Y,n_time)

for x=1:1:length(n_time)
dx = 0.05;
dt = Y*(dx)^2;
% this define columns needed
n_x = (1/dx)+1;
% this define rows needed
n_t = n_time(x)+1;

T = zeros(n_t,n_x);
%this gives us our boundary conditions
boundaryx0 = 700*ones(n_t,1);
boundaryx1 = 300*ones(n_t,1);
boundaryt0 = 500*ones(1,n_x-2);

T(:,1) = boundaryx0;
T(:,n_x) = boundaryx1;
T(n_t,2:n_x-1) = boundaryt0;

if n_t>1
    for i=n_t-1:-1:1
        for w=2:1:n_x-1
            T(i,w) = (Y)*(T(i+1,w+1)-2*T(i+1,w)+T(i+1,w-1))+T(i+1,w);
        end
    end
end

% for x values the following is done
valuedx = 0:dx:1;

% for time values the following is done
valuedt = n_time(x)*dt:-dt:0;

matrix_x_direction = valuedx.*ones(n_t,n_x);
matrix_t_direction = (valuedt)'.*ones(n_t,n_x);

figure(x)
contourf(matrix_x_direction,matrix_t_direction,T,20)
c = contourcbar;
c.Label.String = 'Temperature (K)';
title('Heat Equation w/r of time step');
xlabel('Position (m)');
ylabel('Time (t)');
legend(num2str(n_time(x)),'/Deltat')
end
end
```

Not enough input arguments.

Error in heatTransferPlots (line 3)
for x=1:1:length(n_time)