Math 4540/MSSC 5540 - Homework #4

Henri Medeiros Dos Reis

December 4, 2023

1. Write matlab code to solve our heat equation wit Neumann BCs using finite differences method

```
%% Finite Difference for BVPs
clear all; close all
%Solve the 1D Heat equation with Neumann BCs.
L = 10; Tair = 200; T0 = 40; TL = 400; w = 0.05;q0 = 0;
h = 2;
a1 = 2+w*h*h;
a2 = w*h*h*Tair;

A = [a1 -2 0 0 0;
    -1 a1 -1 0 0;
    0 -1 a1 -1 0;
    0 0 -1 a1 -1;
    0 0 0 -1 a1];
b = [w*h^2*Tair-2*h*q0 a2 a2 a2 TL+a2]';

uin = A\b;
u = [uin' 400];
x = 0:h:L;
plot(x,u)
```

2. An insulated heated rod with a uniform heat source can be modeled by Poisson's equation:

$$\frac{d^2u}{dx^2} = -f(x)$$

Given a heat source $f(x) = 25$ and boundary conditions $u(0) = 40$ and $u(10) = 200$

i) Write the finite differences system of equations with h=2

$$i = 1, \frac{u_0 - 2u_1 + u_2}{h^2} = -25 \Rightarrow 2u_1 - u_2 = 25h^2 + u_0$$

$$i = 2, -u_1 + 2u_2 - u_3 = 25h^2$$
$$i = 3, -u_2 + 2u_3 - u_4 = 25h^2$$
$$i = 4, -u_3 + 2u_4 = 25h^2 + u_f$$

Then
$$\begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 25h^2 + u_0 \\ 25h^2 \\ 25h^2 \\ 25h^2 + u_f \end{bmatrix}$$

ii) Solve using matlab

```
%% Finite Difference for BVPs
clear all; close all
L=10;
u0 = 40;
uf = 200;
h = 2;
a1 = 2;
a2 = 25*h^2;

A = [a1 -1 0 0;
     -1 a1 -1 0;
     0 -1 a1 -1;
     0 0 -1 a1];
b = [a2+u0 a2 a2 a2+uf]';

uin = A\b;
u = [u0 uin' uf];
x = 0:h:L;
plot(x,u)
```

iii) Solve in matlab using h=0.2 (or larger if required by matlab)

```
%% iii
% solve the same with different h
clear all; close all
L=10;
u0 = 40;
uf = 200;
h = 0.02;
a1 = 2;
a2 = 25*h^2;
```

```
m_size = L/h-1;
A = diag(a1*ones(1,m_size)) + diag( ...
    -1*ones(1,m_size-1),1) + diag( ...
    -1*ones(1,m_size-1),-1);

b = a2+zeros(m_size,1);
b(1) = b(1)+u0;
b(end) = b(end)+uf;

uin = A\b;
u = [u0 uin' uf];
x = 0:h:L;
plot(x,u)
```

3. The following is a simple reaction-diffusion equation describing the stead-state concentration, c, of a substance that reacts in a long reactor and disperses axially:

$$D\frac{d^2c}{dx^2} - kc = 0$$

Where $D = 1.5$ the dispersion coefficient, $k - 5$ the reaction time, and $L = 100$

Boundary conditions are given by c(0)=0.1 and c(L) = 1.

(a) Write the system of equations

$$i = 1, D(\frac{u_0 - 2u_1 + u_2}{h^2} - Ku_1 = 0 \Rightarrow u_1\left(\frac{2D}{h^2} + k\right) - \frac{Du_2}{h^2} = \frac{Du_0}{h^2}$$

$$i = 2, \frac{-Du_1}{h^2} + u_2\left(\frac{2D}{h^2} + k\right) - \frac{Du_3}{h^2} = 0$$

$$i = 3, \frac{-Du_2}{h^2} + u_3\left(\frac{2D}{h^2} + k\right) - \frac{Du_4}{h^2} = 0$$

$$i = 3, \frac{-Du_3}{h^2} + u_4\left(\frac{2D}{h^2} + k\right) = \frac{Du_f}{h^2}$$

$$\begin{bmatrix} \frac{2D}{h^2} + k & -\frac{D}{h^2} & 0 & 0 \\ -\frac{D}{h^2} & \frac{2D}{h^2} + k & -\frac{D}{h^2} & 0 \\ 0 & -\frac{D}{h^2} & \frac{2D}{h^2} + k & -\frac{D}{h^2} \\ 0 & 0 & -\frac{D}{h^2} & \frac{2D}{h^2} + k \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \begin{bmatrix} u0\frac{D}{h^2} \\ 0 \\ 0 \\ uf\frac{D}{h^2} \end{bmatrix}$$

(b) Solve in matlab using finite differences with h=20

```matlab
%% Finite Difference for BVPs
clear all; close all
L=100;
k=5;
D=1.5;
u0 = 0.1;
uf = 1;
h = 20;

a1 = 2*D/h^2+k;
a2 = 0;
c = D/h^2;
A = [a1 -c 0 0;
     -c a1 -c 0;
     0 -c a1 -c;
     0 0 -c a1];
b = [a2+u0*D/h^2 a2 a2 a2+uf*D/h^2]';

uin = A\b;
u = [u0 uin' uf];
x = 0:h:L;
plot(x,u)
```

(c) Repeat with h=2 (or as small as matlab will permit)

```matlab
%% iii
% solve the same with different h
L=100;
k=5;
D=1.5;
u0 = 0.1;
uf = 1;
h = 2;

a1 = 2*D/h^2+k;
a2 = 0;
c = D/h^2;

m_size = L/h-1;
A = diag(a1*ones(1,m_size)) + diag( ...
    -c*ones(1,m_size-1),1) + diag( ...
    -c*ones(1,m_size-1),-1);
```

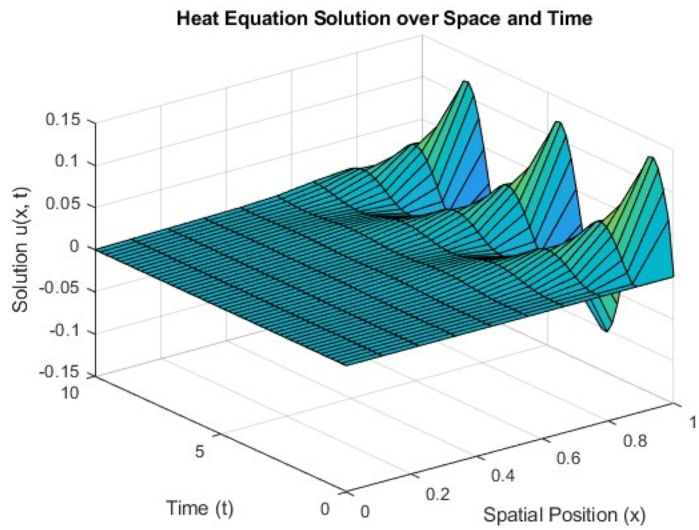
```
b = a2+zeros(m_size,1);
b(1) = b(1)+u0*D/h^2;
b(end) = b(end)+uf*D/h^2;

uin = A\b;
u = [u0 uin' uf];
x = 0:h:L;
plot(x,u)
```

Partial Differential Equations:

4. Problem from class on 11/27 with heat flow proportional to the temperature difference at x = 1. Submit code and a graph of the solution with labeled axes. Use forward difference, finite difference method.

```
clear all; close all;
m = 50;
n=10;
T = 10;
alpha = 0.1;
gi = @(w) sin(pi*w);

dt = T/(m-1);
dx = 1/(n-1);
R = alpha*alpha*dt/(dx^2);
u = zeros(m,n);

u(1, 1:n) = 0;
u(1:m,1) = 0;

for i=1:m-1
    for j=2:n-1
        u(i+1,j)=u(i,j)+R*(u(i,j+1)-2*u(i,j)+u(i,j-1));
    end
    u(i+1,n) = (u(i+1,n-1)+dx*gi((i+1)*dx))/(1+dx);
end

x = (0:n-1)*dx;
t=(0:m-1)*dt;
surf(x,t,u)


xlabel('Spatial Position (x)')
```

```
ylabel('Time (t)')
zlabel('Solution u(x, t)')
title('Heat Equation Solution over Space and Time')
```



5. Consider the equation $\frac{\partial u}{\partial t} = \frac{1}{\pi}\frac{\partial^2 u}{dx^2},\ t > 0,\ 0 < x < 1,$ with $u\,(0, x) = sin\pi x,\ \ u\,(t, 0) = u\,(t, 1) = 0.$

   i) Show that $u\,(t, x) = \ e^{-\pi t}sin\,(\pi x)$is the solution of the problem, i.e. satisfies the PDE and the side conditions.

$$\frac{\partial u}{\partial t} = (-e^{-\pi t})(-\pi)sin(\pi x) = \pi e^{\pi t}sin(\pi x)$$

   and

$$\frac{\partial u}{\partial x} = -e^{\pi t}cos(\pi x)\pi \Rightarrow \frac{1}{pi}\frac{\partial u^2}{\partial x^2} = \frac{1}{\pi}e^{\pi t}sin(\pi x)\pi^2$$

   So the solution holds, now for the initial conditions

$$u(0, x) = e^0 sin(\pi x) = sin(\pi x)$$
$$u(t, 0) = e^{-\pi t}sin(0) = 0$$
$$u(t, 1) = e^{-\pi t}sin(\pi) = 0$$

   ii) Show the modification of the forward difference, finite difference method to the discretization for this problem.

$$\frac{\partial u}{\partial t} \approx \frac{u(t + \Delta t) - u(t, x)}{\Delta t} = \frac{1}{\Delta t}(u_{t+1,j} - u_{ij})$$

$$\frac{\partial u^2}{\partial x^2} \approx \frac{u(t, x - \Delta x - 2u(t, x) + u(t, x + \Delta x}{\Delta x^2} = \frac{1}{\Delta x^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})$$

Then plug to PDE and solve for $u_{i+1,j}$

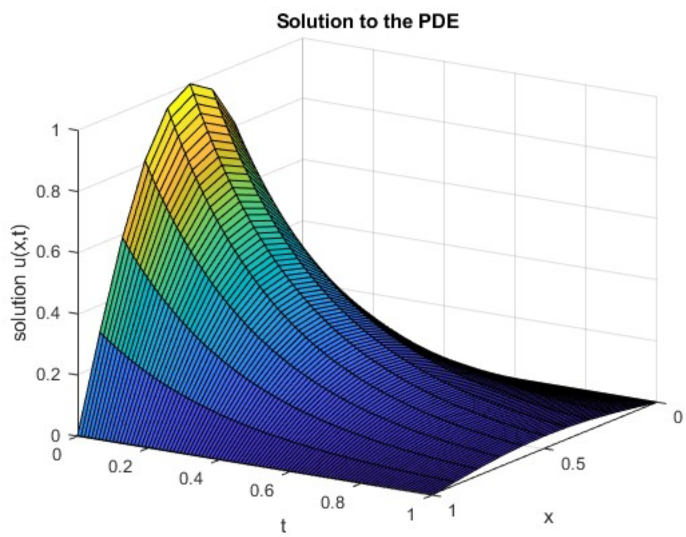
$$\frac{1}{\Delta t}(u_{t+1,j} - u_{ij}) = \frac{1}{\pi}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})$$

$$u_{i+1,j} = \frac{\Delta t}{\pi \Delta x^2}(ui, j-1) - 2u_{ij} + u_{i,j+1}) + u_{ij}$$

iii) Solve the problem using forward difference, finite difference method in matlab with dx = 0.1 and dt = 0.01 Submit your matlab code and a graph of the solution.

```
clear all; close all;
dx = 0.1;
dt = 0.01;
xl = 0; xr = 1;
yb = 0; yt = 1;

M = (xr - xl) / dx; % Number of space steps
N = (yt - yb) / dt; % Number of time steps

f = @(x) sin(pi * x);

D = 1 / pi; % Diffusion coefficient


m = M - 1;
n = N;

sigma = D * dt / (dx^2);

% Create the tridiagonal matrix 'a'
a = diag(1 - 2 * sigma * ones(m, 1)) + diag(sigma * ones
   (m - 1, 1), 1);
a = a + diag(sigma * ones(m - 1, 1), -1);

lside =(0:n) * 0;
rside = (0:n) * 0;

w(:, 1) = f(xl + (1:m) * dx)'; % Initial conditions

for j = 1:n
```

```
    w(:, j + 1) = a * w(:, j) + sigma * [0; zeros(m - 2,
        1); 0];
end

w = [lside; w; rside]; % Attach boundary conditions

x = (0:m + 1) * dx;
t = (0:n) * dt;

% Plot the solution
surf(x, t, w');
title('Solution to the PDE');
xlabel('x');
ylabel('t');
zlabel('solution u(x,t)');
```

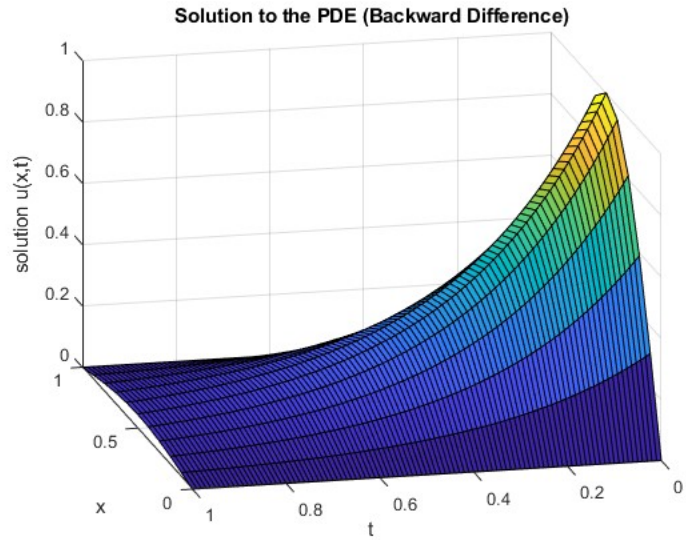


iv) For what step sizes dt, is the forward difference method stable, given dx = 0.1? Check this out in your matlab code by changing dt and seeing what happens. Write down what you observe.

The solution seems to be stable until dt = 0.4 or larger. And it is not completely unstable as we saw in other problems before, it goes just a little up and down, then stabilizes again.

6. Now consider the problem in 5. above again.

i) Use the backward difference method to the solve this problem. You may use the code posted and discussed in class on 11/29. Submit your graph only.



ii) Using the backward difference method, make a table of the exact value, the approximate value, and the error at x = 0.3, t = 1 for step sizes dx = 0.1 and dt = 0.02, 0.01, 0.005.

```
    t      | x_approximate |     x_exact      |      Error
--------------------------------------------------------
   0.020 |   0.00010605    |   0.03496079    |   0.03485475
   0.010 |   0.00007115    |   0.03496079    |   0.03488964
   0.005 |   0.00005722    |   0.03496079    |   0.03490358
--------------------------------------------------------
```

7. Consider the equation $\frac{\partial u}{\partial t} = \frac{1}{\pi}\frac{\partial^2 u}{dx^2}$, $t > 0,\ 0 < x < 1$, with $u\left(0, x\right) = sin\pi x$, $\frac{\partial u}{\partial x}\left(t, 0\right) = \frac{\partial u}{\partial x}\left(t, 1\right) = 0$, insulated on both ends.

i) Show the modification to the discretization for this problem.

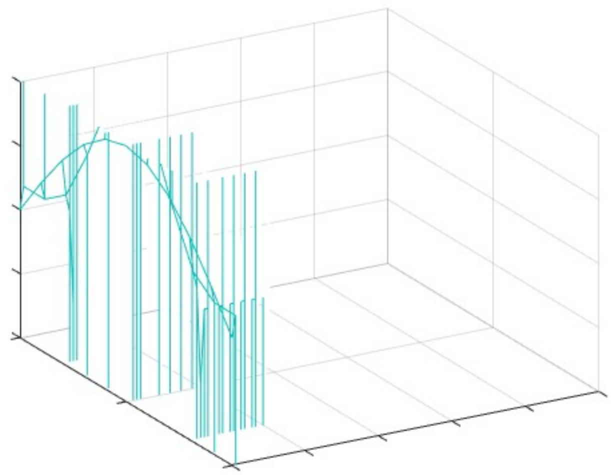
All we need to change are the boundary conditions, which now are functions different than just 0.

$$\frac{\partial u}{\partial t}(t, 0) = \frac{1}{\Delta t}(u_{i,0} - u_{i+1,0})$$
$$\frac{\partial u}{\partial t}(t, 1) = \frac{1}{\Delta t}(u_{t-1,1} - u_{t,1})$$

ii) Solve the problem using forward differences in matlab with dx = 0.1 and dt = 0.01 Submit your matlab code and a graph of the solution.

```
 clear all;close all;

w = heatfdn(0, 1, 0, 1, 0.1, 0.01);
```

```
function w = heatfdn(xl, xr, yb, yt, dx, dt)
    f = @(x) sin(pi * x);
    D = 1; % diffusion coefficient
    M = (xr - xl) / dx; N = yt / dt;
    h = dx; k = dt;
    m = M + 1; n = N;

    sigma = D * k / (h^2);

    a = diag(1 - 2 * sigma * ones(m, 1)) + diag(sigma *
       ones(m - 1, 1), 1);
    a = a + diag(sigma * ones(m - 1, 1), -1); % define
       matrix a
    a(1, :) = [1 -1 zeros(1, m - 2)]; % Neumann
       conditions
    a(m, :) = [zeros(1, m - 2) -1 1];

    w(:, 1) = f(xl + (0:M) * h)'; % initial conditions
    for j = 1:n
        b = w(:, j); b(1) = 0; b(m) = 0;
        w(:, j + 1) = a \ b;
    end

    x = (0:M) * h; t = (0:n) * k;
    mesh(x, t, w') % 3-D plot of solution w
    view(60, 30); axis([xl xr yb yt -1 1])
end
```

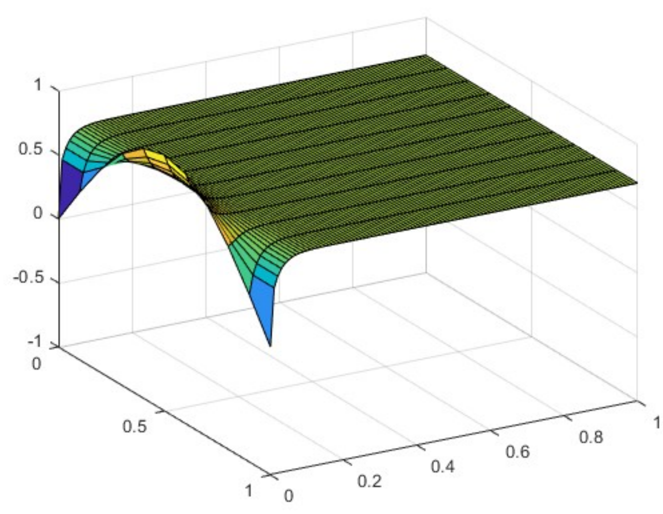


iii) For what step sizes dt, is the forward difference method stable, given dx = 0.1? Check this out in your matlab code by changing dt and seeing what happens. Write down what you observe.

The solution seems to stabilize once we get close to dx, dt = 0.09 and larger values seems to make it stable.

8. Now consider the problem in 7. above again.

i) Use the backward difference method to the solve this problem. You may use the code posted and discussed in class on 11/29. Submit your graph only.



ii) Using the backward difference method, make a table of the exact value, the approximate value, and the error at x = 0.3, t = 1 for step sizes dx = 0.1 and dt = 0.02, 0.01, 0.005.

```
-----------------------------------------------------------
   t    |  x_approximate  |     x_exact      |     Error
-----------------------------------------------------------
 0.020 |    0.70152795   |   0.63661977   |   0.06490817
 0.010 |    0.70152795   |   0.63661977   |   0.06490817
 0.005 |    0.70152795   |   0.63661977   |   0.06490817
-----------------------------------------------------------
```