ST/ NUMERICAL MODELING
CS559 Fall 2020
Oregon State University

Homework 3

Tianle Yuan (Mark) 933-946-576

November, 15, 2020

This homework answers the problem set sequentially:

3.1 **Implement the algorithm for solving numerically the (discrete) heat equation, assuming fixed temperature boundary conditions:**

$$u(0,t) = 0, \qquad u(L,t) = 10, \qquad L = 1,$$

and initial value

$$u_j(0) = u(x_j, 0) = 10x_j + 5sin^2 2\pi x_j, \qquad 1 \le j \le N.$$

Use K= 0.1 as diffusion coefficient. Then:

1. With different numbers of discretization points (e.g., n = 10, n = 50, n = 80, n = 100, n = 200,...), investigate the spread of time constants of the system. Use the command *eig* in MATLAB, and plot the time constants in a logarithmic scale (see the command *semilogy*.) What can you say about the stiffness of the system?

Solution:

For the time constant, here we use the Condition Number to express it:

$$k(A) = \frac{|\lambda_{max}(A)|}{|\lambda_{min}(A)|} \tag{1}$$

where $\lambda_{max}(A)$ and $\lambda_{min}(A)$ are maximal and minimal eigenvalues of A respectively.

The time constant is a constant try to describe how inaccurate the solution of the equation will be after approximation. High time constant means the final solution exists big error. Low time constant means it is not hard for the system to find a numerical solution which is close to the exact solution.

Here, we get points with numbers of discretization points:10, 50, 80, 100, 200. Our matrix A is $K\widetilde{L}$:

$$A = K\widetilde{L} = \frac{K}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & & ... & & \\ & & 1 & -2 & 1 \\ & & & 1 & -2 \end{pmatrix} \in \mathbb{R}^{(n-2)\times(n-2)} \tag{2}$$

The Condition Number can also be seem as the Stiffness Ratio, which is used to measure the stiffness of differential equations. The ratio is larger, the stiffer the system becomes. The final result can be shown below:

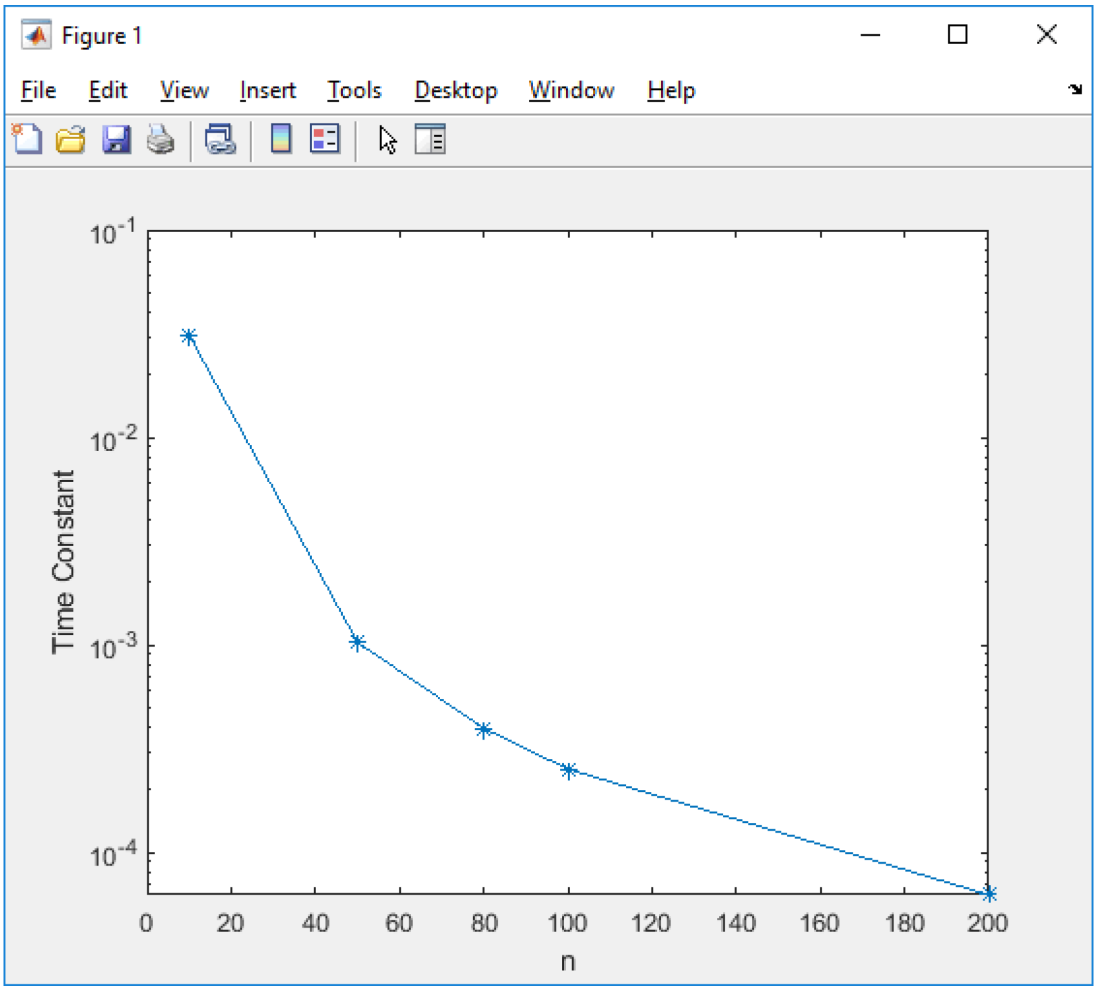

Figure 1: The change of time constant under different n

We can get a conclusion that as the numbers of discretization points increase, the system will be come less stiff, which means the size of the time interval will reduce the influence on the stability of the solution.

2. Propagate the solution from t = 0 to t = 1. Try both *ode*45 **and** *ode*23*s* **with different numbers of discretization points. What is the difference between the solutions? Clock the solvers using the command** *tic toc* **in MATLAB. (Hint: run first with** *ode*23*s*, **otherwise you will probably run out of patience (and memory) as** n **grows.)**

Visualize the results using the *mesh* **command in MATLAB. Remember: The time steps are not uniform, so in order to have a meaningful figure, you need to define the** $x-$ **and** $t-$**variables for the** *mesh*−**command. In addition, make sure that your solution** $u(x,t)$ **also contains the constant boundary values, not just values in the interior points** x_j, $2 \le j \le n$.

Solution:

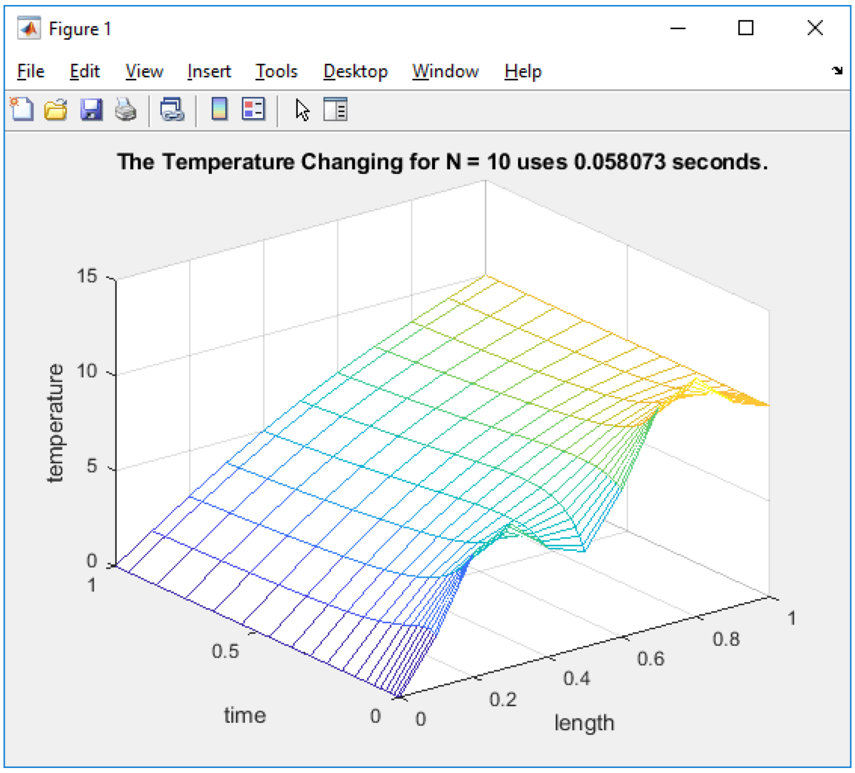

Figure 2: The temperature changing when using ode23s with n =10

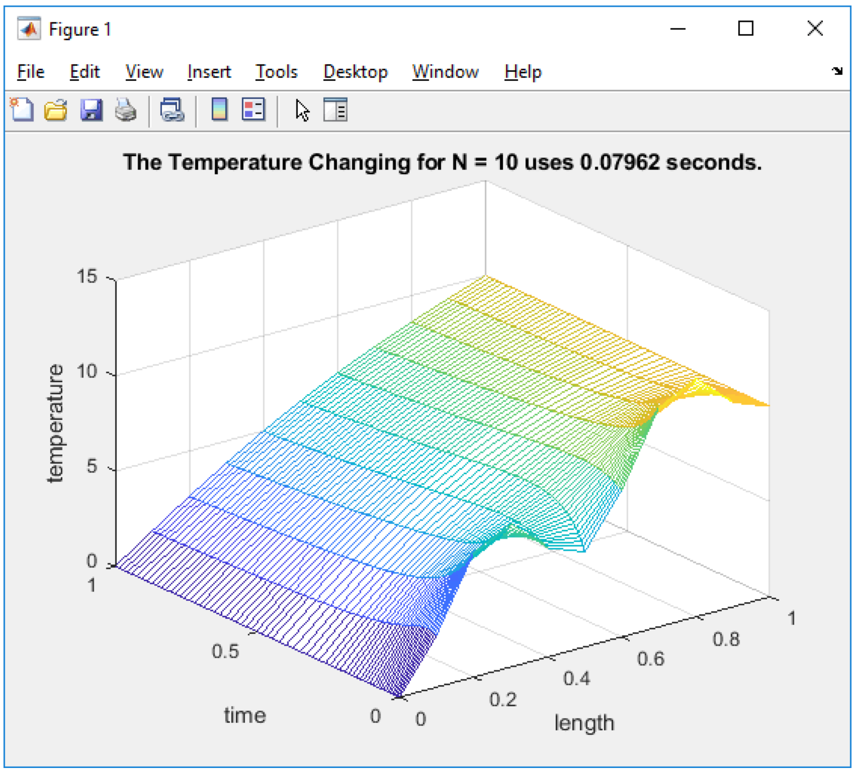

Figure 3: The temperature changing when using ode45 with n =10

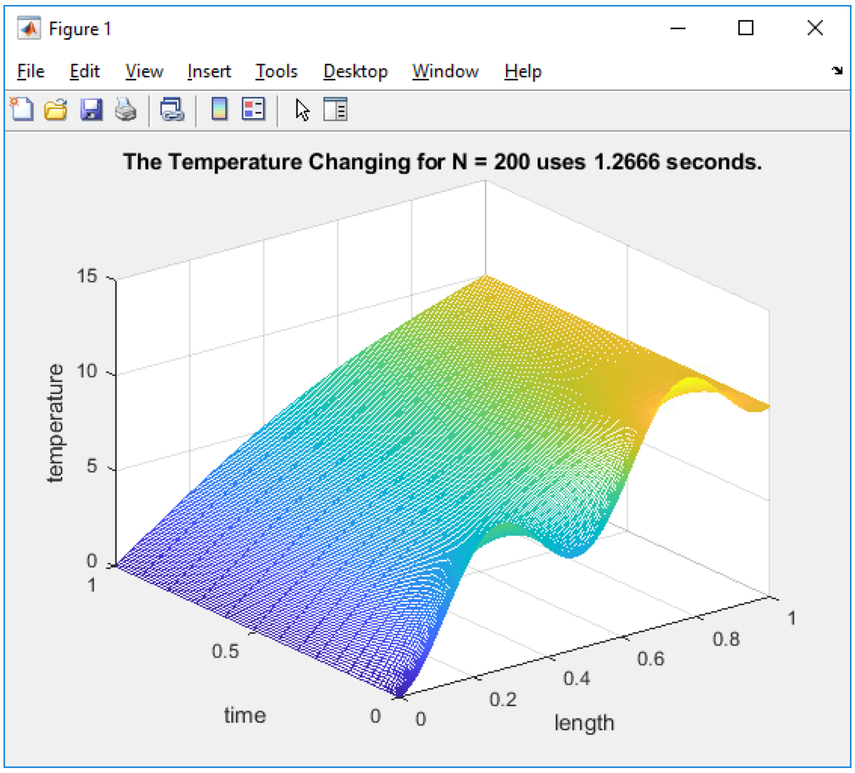

Figure 4: The temperature changing when using ode23s with n =200

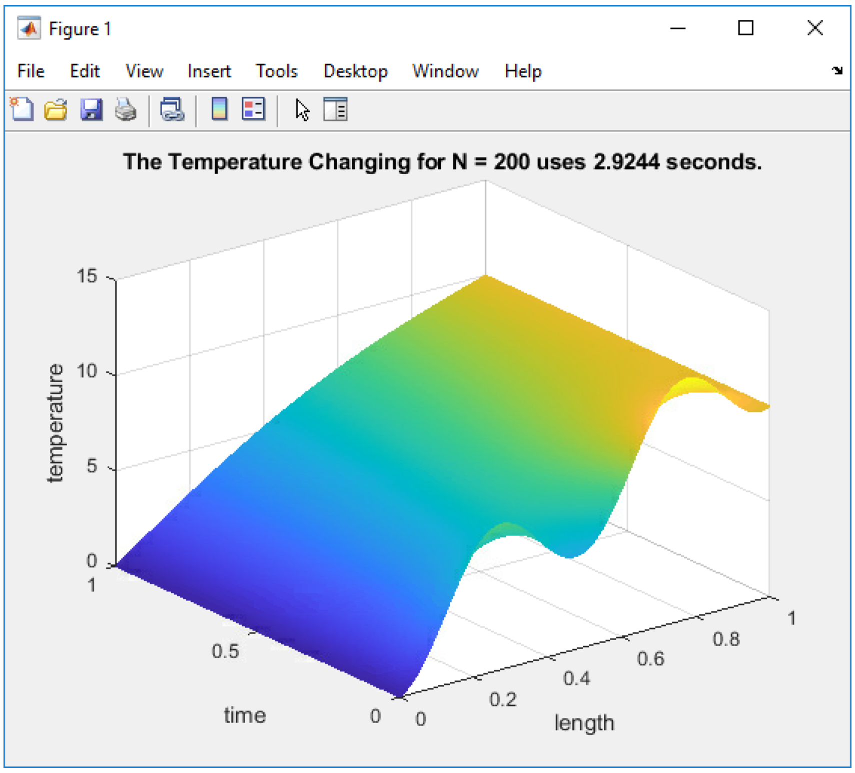

Figure 5: The temperature changing when using ode45 with n =200

From pictures shown above, we can get a conclusion that in the normal case the solution of ode45 is more precise than ode23s', but need much more time. Another conclusion is that as the numbers of discretization points increase, the system get less stiff, thus ode45 can still be a good method to get precise solution for this equation system.

Matlab-Code

Problem 1:

```
n1 = 10;
n2 = 50;
n3 = 80;
n4 = 100;
n5 = 200;
x = [n1,n2,n3,n4,n5];
y = [constant(n1),constant(n2),constant(n3),constant(n4),constant(n5)];
semilogy(x,y,'*-');
xlabel ('n')
ylabel('Time Constant');

function stiffness = constant(n)
L = 1;
h = L/n;
K = 0.1;
b = diag(repmat([-2],1,(n-2)))+diag(repmat([1],1,(n-3)),1)+diag(repmat([1],1,
(n-3)),-1);
A = K/h^2.*b;
Emax = abs(max(eig(A)));
Emin = abs(min(eig(A)));
stiffness = Emax/Emin;
end
```

Problem 2:

```
tspan=[0 1];
n = 10;
L = 1;
h = L/n;
x = 0:h:L;
u0 = 10.*x+5.*(sin(2*pi.*x)).^2;

tic;
[t,u]=ode23s(@odefun,tspan,u0);
time = toc;

mesh(x,t,u);
```

```
title('The behavior of the fish population');
xlabel('length');ylabel('time'); zlabel('temperature');
title(['The Temperature Changing for N = ' num2str(n) ' uses ' num2str(time) '
seconds.']);


function dudt = odefun(t,u)
n = 11;
L = 1;
h = L/n;
K = 0.1;
dudt = zeros(n,size(u,2)); %preallocate space

i = 2;
dudt(i,:)= K*(-2*u(i,:)+u(i+1,:))/(h^2);
i = 2:1:n-2;
dudt(i,:)= K*(u(i+1,:)-2*u(i,:)+u(i-1,:))/(h^2);
i = n-1;
dudt(i,:)= K*(-2*u(i,:)+u(i-1,:))/(h^2);

b = zeros(n,1);
b(2) = 0;
b(n-1) = 10;

i = 1:n;
dudt(i,:) = dudt(i,:)+ (K/h^2*b(i,:));
end
```