MA402 Project I: Numerical Analysis and Simulations for Some Partial Differential Equations

Zhilin Li, ID: 000-00-0000

Abstract

In this project, we derive a simple mathematical model for $\cdots$. Some finite difference methods have been developed to solve the model problem numerically. Some analysis including the accuracy, the consistency, the stability are given for the finite difference methods. The numerical results presented here show physical reasonable results. Through the numerical tests, some conclusions are observed from our numerical tests including the effect of the parameters on the solutions.

1 A model problem.

Choose an application problem that can be modeled as ***one*** of the following partial differential equations.

- The heat diffusion:

$$u_t = \beta u_{xx} + a u_x - cu + f(x,t), \tag{1}$$

  where $u_t = \frac{\partial u}{\partial t}$, $u_{xx} = \frac{\partial^2 u}{\partial x^2}$ and so forth.

- The one-way wave equations:

$$u_t = \beta u_x - cu + f(x,t) \tag{2}$$

The problem may be similar to one in the text book, or it may be related for another course or your job. If you can find an similar problem outside the text book, you will get *extra 5 points*. If you can not find any problem outside of the text book, then you can use one from the text book.

<u>You need to:</u>

- Identify the physical laws, if any, in deriving the model.
- Identify all the dependent and dependent variables and their physical meaning.
- Explain the meanings of each term in the model such as the rate of change with time, the source/sink term, etc.
- Explain each parameter in the model.

2 Possible theoretical analysis

- Are there any analytic solutions? You can use the knowledge from your ODE/PDE courses.
- Is there a steady state solution? If your answer is yes, what is the differential equation for the steady state solution? Does the state solution depend on the initial condition?

3 The finite difference methods for the model.

3.1 Set-up a grid.

Assume that $0 \le x \le L$, and $0 < t < t_{final}$, generate a finite difference grid that can be used to solve the model problem numerically, that is:

$$\begin{aligned} x_i &= ?, \qquad i = ? \cdots ?, \quad h = \frac{L}{?} \\ \text{Given } \Delta t, \quad t^k &= ?, \qquad k = ? \cdots ?, \quad k_{final} = \frac{?}{?}. \end{aligned}$$

3.2 The finite difference schemes.

Write down three different finite difference schemes using:

1. The forward discretization for u_t at (x_i, t^k), and the central formula for the space discretization (i.e. the right hand side of the PDE) at (x_i, t^k).
2. The Backward discretization for u_t at (x_i, t^k), and the central formula for the space discretization at (x_i, t^k).
3. The central discretization for u_t at (x_i, t^k), and the central formula for the space discretization at (x_i, t^k).

Analyze the three finite difference schemes by discussing the following questions:

- Are the finite difference schemes consistent? If so, what are the oder of the truncation errors $((\Delta t)^? + h^?)$?
- Discuss the feasibility of the three different schemes and possible difficulties in implementations.

4 Numerical Experiments and Analysis

Referring to the sample codes *heat_im.m* and *wave_im.m* from the class web-page, implement the finite difference scheme for one of the following two PDEs.

- The one dimensional diffusion equation

$$u_t = \beta u_{xx} - cu + f(x,t). \tag{3}$$

Use the Crank-Nicholson scheme (see heat_im.m from the sample file):

$$\begin{aligned} \frac{u_i^{k+1} - u_i^k}{\Delta t} &= \frac{\beta}{2}\left(\frac{u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}}{h^2} + \frac{u_{i-1}^{k} - 2u_i^{k} + u_{i+1}^{k}}{h^2}\right) & (4) \\ &+ \frac{c}{2}\left(u_i^{k+1} + u_i^k\right) + f_i^{k+\frac{1}{2}}, \qquad\qquad \beta > 0, & (5) \end{aligned}$$

to solve the problem, where $f_i^{k+\frac{1}{2}} = f(x_i, t^k + h/2)$. Use the following exact solution to check your code:

$$u_e(x,t) = \frac{\sin(\pi x)}{1+t^2}, \quad \text{and} \quad f(x,t) = \frac{\partial u_e}{\partial t} - \beta\frac{\partial^2 u_e}{\partial x^2} + cu_e \tag{6}$$

to show that the Crank-Nicholson method is second order accurate $O((\Delta t)^2 + h^2)$ and it is unconditionally stable (works for any Δt) by testing.

1. Write down the matrix-vector form of the finite difference scheme

$$A\mathbf{u}^{k+1} = B\mathbf{u}^k + \mathbf{b}^k, \tag{7}$$

where A and B are two matrices, and **b** is a vector.

2. Fill the similar table in HW#3 with the infinity norm with $\Delta t = h$, with $t_{final} = 5$.
3. Try your code with $\Delta t = 10\,h,\ t_{final} = 5, \quad \Delta t = 100\,h,\ t_{final} = 50$,
4. Now set $f(x,t) = 0,\ u(x,0) = \sin(\pi x),\ u(0,t) = 0$, and $u(1,t) = 0$. Try different β to see how the change of β would effect the solution. Does your observation agree with physics?

- The one dimensional one-way wave equation

$$u_t = \beta u_x - cu + f(x,t), \qquad\qquad \beta > 0. \tag{8}$$

Use the backward Euler scheme

$$\frac{u_i^{k+1} - u_i^k}{\Delta t} = \beta\left(\frac{u_{i+1}^{k+1} - u_i^{k+1}}{h}\right) - cu_i^{k+1} + f_i^{k+1}, \tag{9}$$

to solve the problem, where $f_i^{k+1} = f(x_i, t^k + h)$. Use the following exact solution to check your code:

$$u_e(x,t) = \frac{\sin(\pi x)}{1+t^2}, \quad \text{and} \quad f(x,t) = \frac{\partial u_e}{\partial t} - \beta\frac{\partial u_e}{\partial x} + cu_e \tag{10}$$

and show that the backward Euler method is first order accurate $O(\Delta t + h)$ and it is unconditionally stable (works for any Δt) by testing.

1. Fill the similar table in HW#3 with the infinity norm with $\Delta t = h$, with $t_{final} = 5$.
2. Try your code with $\Delta t = 10\,h,\ t_{final} = 5$, $\Delta t = 100\,h,\ t_{final} = 50$,
3. Do you think the backward finite difference scheme is superior to the forward finite difference scheme? (**Hint:** Consider accuracy, the time step constraint, and implementation).
4. Where should we have the boundary condition?
5. Now set $f(x,t) = 0$, $u(x,0) = \sin(\pi x)$, and $u(1,t) = 0$. Try different β to see how the change in β would effect the solution. Does your observation agree with physics?

You should itemize your experiments and your discussions, for example,

4.1 Test1:

In this test, the parameters are $\cdots$. Table 1 shows the grid refinement analysis. We can see the error decreases by fact 4 as we refine the mesh by double n. ... The attached Figure 1 is the computed solution at $t_{fina} = 5$, ...

4.2 Test2:

Now we test the effect of the parameter β on the solution, we can see that ...

4.3 Extra credits

If you find a problem outside the class, you should also try to do some computations for your problems. You will get extra credits in doing this.

5 Conclusions

In this projects, we have developed a mathematical model for one dimensional heat diffusion in a thin rod. We have also developed several finite difference schemes to solve the model problem numerically. Through the numerical experiments, we believer the Crank-Nicholson method is the best for our model problem because it is second order accurate and unconditionally stable. We have also investigate the effect of the parameters on the solution of the differential equations.

6 Acknowledgment

The authors would like to thank the instructor of MA402 for the sample codes posted on the course web-page [1].

References

[1] Zhilin Li. MA402 web-page, Fall 2001.
http://www4.ncsu.edu/~zhilin/TEACHING/MA402/index.html

7 Appendix I: Computer Code(s):

```
%  Heat Diffusion in a Thin Insulated Wire

clear all; close all;
global beta c

beta = 2;   c = 0.5;                %  material parameters
L = 5.0;                            %  Length of the Wire
tfinal = 1;                         %  Final Time

n = 40.; h = L/n;                  %Number of Space Steps

dt = h;
maxk = fix(tfinal/dt),               %  Number of Time Steps

%  Initial Temperature
for i = 1:n+1
        x(i) = (i-1)*h;
        u0(i) = uexact(x(i),0);
end

plot(x,u0); hold

t = 0;

A = sparse(n+1,n+1); b=zeros(n+1,1);
A(1,1) = 1;
for i=2:n
  A(i,i) = 1/dt + beta/(h*h) + c/2;
  A(i,i-1) = -beta/(2*h*h);       A(i,i+1) = -beta/(2*h*h);
end
A(n+1,n+1) = 1;


for k=1:maxk                % Time Loop

   b(1) = 0;   b(n+1) = 0;      % The boundary condition
   for i=2:n;               % Space Loop
      b(i) = f(x(i),t+0.5*dt) + u0(i)/dt+ 0.5*beta*( u0(i-1) -2*u0(i) + ...
             u0(i+1) )/(h*h) -c*u0(i)/2;
   end
```

```
   u1 = A\b;           % Solve the system of equations.

   u0 = u1;   plot(x,u1)
   t = t + dt;
end

for i=1:n+1
  err(i) = u1(i)- uexact(x(i),t);
end

  figure(2); plot(x,err)
  e2 = [norm(err,1),norm(err,2),norm(err,inf)],
```

7.1 Matlab function of the source term *f.m* and global variables

```
function y = f(x,t)

global beta c

 y= exp(-t)* ( -sin(pi*x) + beta*pi*pi*sin(pi*x) +c*sin(pi*x) );
```

8 Appendix II: Some plots.

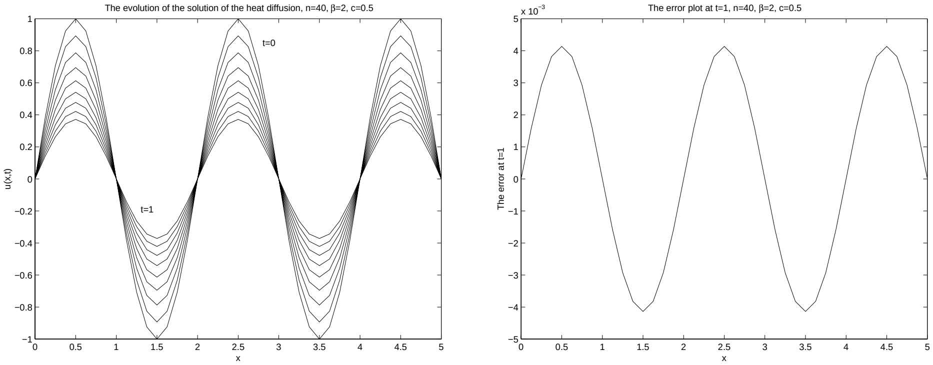

Figure 1: (a) The evolution of the computed solution. We can see that as t increases, the magnitude of the solution decreases. The parameters are $\beta = 2$, $c = 0.5$, $L = 5$, $t_{final} = 1$, and $\Delta t = h$. (b) The error plot. The exact solution is $u(x,t) = e^{-t}\sin(\pi x)$.