MATH 612

Numerical Methods for Partial Differential Equations

Spring Term 2004

Instructor: Georg Dolzmann

Solutions to Homework Set 6

Problem 1: The solution of the heat equation $u_t = u_{xx}$ on the interval $[-1, 1]$ with periodic boundary conditions and initial data

$$u^0(x) = \begin{cases} 1 & \text{if } |x| < \frac{1}{2}, \\ \frac{1}{2} & \text{if } |x| = \frac{1}{2}, \\ 0 & \text{otherwise,} \end{cases}$$

is given by

$$u^0(x,t) = \frac{1}{2} + \frac{2}{\pi}\sum_{k=0}^{\infty} e^{-t(2k+1)^2\pi^2}\frac{(-1)^k}{2k+1}\cos\big((2k+1)\pi x\big). \tag{1}$$

We now solve the heat equation with Dirichlet conditions where we take the boundary values from the exact solution given by (1). The solution is again given by (1).

(a) Use the Crank–Nicholson scheme with $h = 1/10$, $1/20$, $1/40$, and $1/80$, and with $\Delta x = \Delta t$. Compare the convergence in the L^2 norm and in the L^∞ norm,

$$e^n_\infty = \max_{j=1,\dots,J-1} |U^n_j - u(x_j, t_n)|,$$

and determine the experimental rate of convergence. Plot the solution and the amplification factors for the Crank–Nicholson scheme. You may use commands of the form

```
%
% plot the function with h=1/10
%
  subplot(3,2,1);
  plot(x,u);
%
% plot the amplification factors
%
  subplot(3,2,2);
  plot(k,lambda);
%
% plot the function with h=1/20
%
  subplot(3,2,3);
  ...
```

This gives you a nice comparison of the solution and the amplification factors.

If you want to implement the exact solution as an mfile, say `u_ex`, and pass the exact solution as an argument to your solver for the heat equation, then you need to use the `@` command:

```
%
% the m-file for the solver with Dirichlet conditions dc
%
  function u=cn(dc,...)

  u(1)=feval(dc,0,t);
%
% in the main routine
%
  u=cn(@u_ex,...)
```

You also need to use the `feval` command in the subroutine `cn` to evaluate the Dirichlet conditions.

(b) Repeat (a) with the fully implicit scheme, $\theta = 1$. Compare the results in (a) and (b) and explain your observations.

Solution: (a) We need to modify the implementation of the θ scheme in order to allow non-zero boundary conditions. Here is a possible implementation:

```
%
% function u=h1ddth(u0,dc,J,dt,nt,theta)
%
% solves the heat equation u_t=u_{xx} on (0,1) with
% the theta scheme
%
% the initial conditions are given as a function u0
% and the Dirichlet boundary conditions as a function dc
%
% J+1 nodes x_0,...,x_J in the x-variable, dx=1/J
%
% final time given by dt*nt
%
%
  function u=h1ddcn(u0,dc,J,dt,nt,theta)

  dx=2/J;
  x=[-1:dx:-.5-dx,-.5:dx:.5-dx,.5:dx:1];
  un=u0(x)';
  nu=dt/(dx^2);
%
% generate a warning if the stability condition is violated
%
  if nu*(1-2*theta)>1/2
    disp('WARNING: the stability condition is violated')
  end;
%
```

```
% un = solution at time n
%
% note that un has J+1 components labeled 1,...,J+1
%
%
  un=u0(x)';
%
% build the system matrix
%
% start with eye and add only 2*theta*nu to the diagonal
%
  a=eye(length(x));
  a=a+diag([0; 2*nu*theta*ones(J-1,1); 0]);
  a=a+diag([-theta*nu*ones(J-1,1);0],-1);
  a=a+diag([0;-theta*nu*ones(J-1,1)],1);

  for i=1:nt,
    f=[feval(dc,-1,i*dt,5); ...
       un(2:J)+(1-theta)*nu*(un(1:J-1)-2*un(2:J)+un(3:J+1)); ...
       feval(dc,1,i*dt,5)];
    un=a\f;
  end

  u=un';
```

We use this subroutine to solve the heat equation numerically. The following mfile contains most of the commands:

```
%
% solve the heat equation witht the theta scheme and
% plot the solution and the amplification factors
%
  function [e2,einf]=solve_n_plot(theta,J,kmax,maxplotno,plotno)
%
% plot the amplification factors
%
  k=1:kmax;

  dx=2/J; dt=dx; nt=1/dt;
  u=h1ddth(u0,@ux,J,dt,nt,1/2);
  x=-1:2/J:1;
  d=u-ux(x,nt*dt,4);
  e2=sqrt(dx*sum(d(1:J).*d(1:J)))
  einf=max(abs(d(1:J-1)))
  subplot(3,2,1);
  plot(x,u);
  nu=dt/dx^2;
  subplot(3,2,2);
  plot(k,(1-2*nu*(sin(k*dx/2).^2))./(1+2*nu*(sin(k*dx/2).^2)));
```

We now generate the plots in Figure 1 with the following commands:

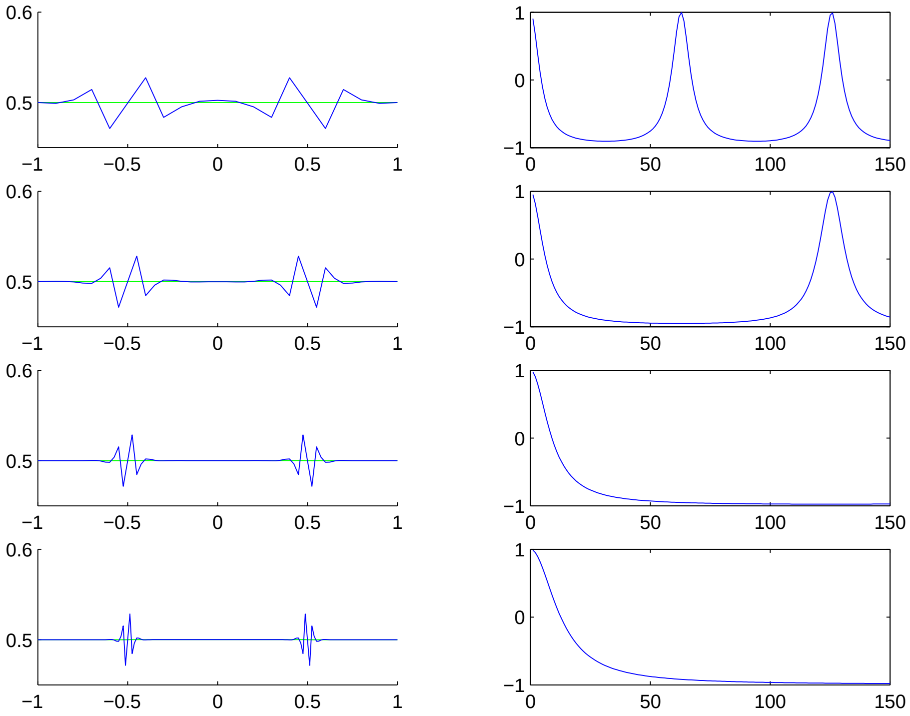

FIGURE 1. The solution of the heat equation with discontinuous initial data and the Cranck–Nicholson scheme. The correct solution is the green line $u \sim .5$

```
  close all; clear all
  u0=inline('(x>-0.5).*(x<.5)+.5*(x==.5)+.5*(x==-.5)');
%
% the calculations with the Cranck-Nicholson scheme
%
  theta=0.5;

  error=zeros(4,2);
%
% h=1/10 corresponds to J=20;
%
  [error(1,1),error(1,2)]=solve_n_plot(theta,20,u0,@ux,150,4,1)
%
% h=1/20 corresponds to J=40;
%
  [error(2,1),error(2,2)]=solve_n_plot(theta,40,u0,@ux,150,4,2)
%
% h=1/40 corresponds to J=80;
%
  [error(3,1),error(3,2)]=solve_n_plot(theta,80,u0,@ux,150,4,3)
```

J	L^2-error	EOC	L^∞-error	EOC
20	0.0205		0.0288	
40	0.0146	0.4942	0.0284	0.0192
90	0.0103	0.4988	0.0284	0.0011
160	0.0073	0.4997	0.0284	0.0007

TABLE 1. The L^2 error and the L^∞ error in the calculation of the solution heat equation with Dirichlet boundary conditions and the Crank–Nicholson scheme.

```
%
% h=1/80 corresponds to J=160;
%
  [error(4,1),error(4,2)]=solve_n_plot(theta,160,u0,@ux,150,4,4)

  for j=1:3,
   e2(j)  = -log(error(j+1,1)/error(j,1))/log(2)
   einf(j)= -log(error(j+1,2)/error(j,2))/log(2)
  end
```

The results are summarized in the following table:

(b) We now use the fully implicit method and generate the required output with the following commands:

```
  close all; clear all
  u0=inline('(x>-0.5).*(x<.5)+.5*(x==.5)+.5*(x==-.5)');
%
% the calculations with the implict scheme
%
  theta=1;

  error=zeros(4,2);
%
% h=1/10 corresponds to J=20;
%
  [error(1,1),error(1,2)]=solve_n_plot(theta,20,u0,@ux,150,4,1)
%
% h=1/20 corresponds to J=40;
%
  [error(2,1),error(2,2)]=solve_n_plot(theta,40,u0,@ux,150,4,2)
%
% h=1/40 corresponds to J=80;
%
  [error(3,1),error(3,2)]=solve_n_plot(theta,80,u0,@ux,150,4,3)
%
% h=1/80 corresponds to J=160;
%
  [error(4,1),error(4,2)]=solve_n_plot(theta,160,u0,@ux,150,4,4)
```

J	L^2-error	EOC	L^∞-error	EOC
20	0.0138		0.0138	
40	0.0068	1.0133	0.0068	1.0146
80	0.0033	1.0331	0.0033	1.0334
160	0.0017	1.0135	0.0017	1.0136

TABLE 2. The L^2 error and the L^∞ error in the calculation of the solution heat equation with Dirichlet boundary conditions and the implicit scheme.

```
for j=1:3,
 e2(j)  = -log(error(j+1,1)/error(j,1))/log(2)
 einf(j)= -log(error(j+1,2)/error(j,2))/log(2)
end
```

The results are summarized in the Figure 2 and Table 2: A comparison of the results in Part (a) and (b) shows that the solution with the Cranck–Nicholson scheme does not converge in L^∞ and only of order .5 in L^2 even though the scheme is formally of quadratic order for *smooth* solutions. The reason is that most of the amplification factors are close to -1 and the high frequencies that are needed to approximate the jump discontinuity in the initial data are not damped. They are rapidly damped by the fully implicit scheme for which most of the amplification

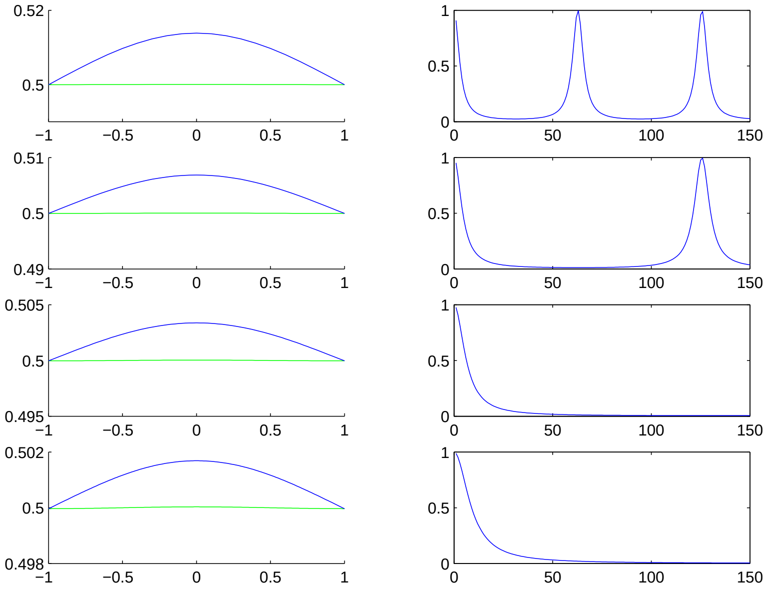

FIGURE 2. The solution of the heat equation with discontinuous initial data and the implicit scheme. The green line $u \sim .5$ is the correct solution.

factors are close to zero. Thus the implicit scheme can be advantageous despite is inferior rate of convergence.

Problem 2: Consider the hyperbolic system

$$\begin{pmatrix} u^1 \\ u^2 \end{pmatrix}_t + \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \end{pmatrix}_x = 0$$

for $x \in (0,1)$ and $t \in (0,1)$. Suppose that the boundary conditions and initial data are given by $u^1(0,t) = 0$, $u^1(1,t) = 0$ for $t \in (0,1)$ and $u^1(x,0) = x$, $u^2(x,0) = 1$ for $x \in (0,1)$. Show that the solution of this system is given by

$$\begin{pmatrix} u^1(x,t) \\ u^2(x,t) \end{pmatrix} = \begin{cases} \begin{pmatrix} x \\ 1-t \end{pmatrix} & \text{if } 0 \le x < 1-t, \\ \begin{pmatrix} x-1 \\ 2-t \end{pmatrix} & \text{if } t-1 \le x < 1. \end{cases}$$

Hint: You need to solve the system with the method of characteristics in order to get credit for this problem. It is not sufficient to check that the function given in the problem is really a solution of the system with the given boundary and initial conditions.

Solution: We diagonalize the system. The characteristic speeds are $\lambda_1 = 1$ and $\lambda_2 = -1$ and the corresponding eigenvectors are $v_1 = (1,1)$ and $v_2 = (-1,1)$. The transformation matrix P is then given by

$$P^{-1} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}, \quad P = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}.$$

The function $w = Pu$ then satisfies

$$w_t + \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} w_x = 0$$

or

$$w_t^1 + w_x^1 = 0, \quad w_t^2 - w_x^2 = 0.$$

The initial conditions are given by

$$w(x,0) = Pu(x,0) = \frac{1}{2} \begin{pmatrix} u^1(x,0) + u^2(x,0) \\ -u^1(x,0) + u^2(x,0) \end{pmatrix} = \begin{pmatrix} x+1 \\ -x+1 \end{pmatrix}.$$

Note that not all the characteristic curves through points in the square pass though the unit interval on the $t = 0$ axis and we therefore need to look at incoming and outgoing characteristics. The boundary condition $u^1(0,t) = u^1(1,t)$ implies that we need to have $w^1(0,t) - w^2(0,t) = 0$ and $w^1(1,t) - w^2(1,t) = 0$. This shows that the initial condition for the incoming characteristic is given by the value of w^1 or w^2 on the outgoing characteristic.

The characteristics for w^1 move with speed one to the right and provide us with initial data for w^2 on the line $\{1\} \times (0,1)$. Similarly, the characteristics for w^2 move with speed one to the right and provide us with initial data for w^1 on the line $\{0\} \times (0,1)$, see also Figure 3. The function w^1 is given by $w^1(x,t) = w_0^1(x-t)$ and thus $w^1(x,t) = .5(x - t + 1)$ in the lower right triangle. Similarly, $w^2(x,t) = w_0^2(x+t) = .5(-x - t + 1)$ in the lower left triangle. The function w^1 gives the boundary data $w^2(1,t) = .5(2-t)$ for w^2 and we only need to determine where the characteristic through (x,t) intersects the line $\{1\} \times (0,1)$. This happens if

$$(1-(t-t0),t) = (x,t) \quad \Leftrightarrow \quad t_0 = x+t-1.$$

Thus

$$w^2(x,t) = \frac{1}{2}\big(2-(x+t-1)\big) = \frac{1}{2}(3-x-t)$$

in the upper right triangle. Similarly w^2 allows us to find $w^1(0,t) = .5(1-t)$. The characteristic though (x,t) passes thorough the line $\{0\} \times (0,1)$ if

$$(t-t_0,t) = (x,t) \quad \Leftrightarrow \quad t_0 = t-x.$$

Hence

$$w^1(x,t) = \frac{1}{2}\big(-(t-x)+1\big) = \frac{1}{2}(x-t+1)$$

in the upper left triangle. We finally use the fact that $u = P^{-1}w$ to obtain the expressions for w^1 and w^2 in Figure 3.

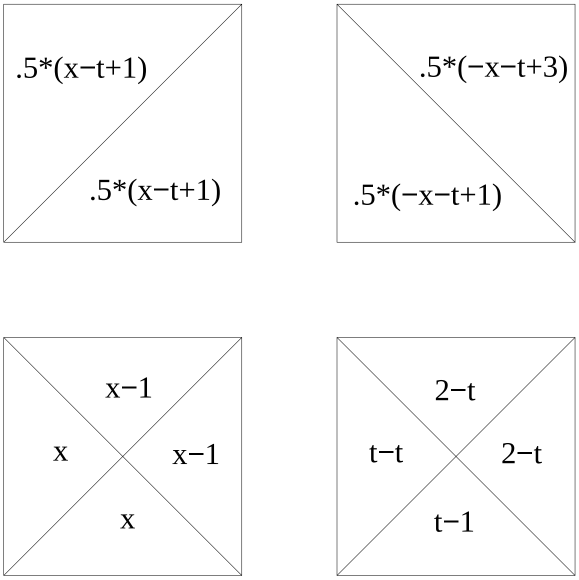

FIGURE 3. Construction of the solution of the hyperbolic system. The values for w^1 and w^2 can be found in the two panels on the left and on the right, respectively.

Problem 3: Find the leading order term of the truncation error for the Lax–Friedrichs scheme,

$$u_t + au_x \sim \frac{u_j^{n+1} - \frac{1}{2}(u_{j+1}^n + u_{j-1}^n)}{\Delta t} + a\,\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x}.$$

Show that the scheme is not unconditionally consistent. What condition in terms of Δx and Δt do you need to get consistency?

Solution: We use a Taylor series expansion to find

$$u_{j\pm1}^{n} = \left[u \pm \Delta x u_x + \frac{1}{2}\,(\Delta x)^2 u_{xx} \pm \frac{1}{6}\,(\Delta x)^3 u_{xxx}\right]_j^n + \mathcal{O}\big((\Delta x)^4\big)$$

and thus

$$\frac{1}{2}(u_{j+1}^{n} + u_{j-1}^{n}) = \left[u + \frac{1}{2}\,(\Delta x)^2 u_{xx}\right]_j^n + \mathcal{O}\big((\Delta x)^4\big).$$

Similarly

$$\frac{1}{2}\,(u_{j+1}^{n} - u_{j-1}^{n}) = \left[\Delta x u_x + \frac{1}{6}\,(\Delta x)^3 u_{xxx}\right]_j^n + \mathcal{O}\big((\Delta x)^5\big)$$

Substituting these expressions into the Lax–Friedrichs scheme we find

$$T_j^n = \left[u_t + a u_x + \frac{1}{2}\,\Delta t u_{tt} - \frac{1}{2}\,\frac{(\Delta x)^2}{\Delta t}\,u_{xx} + \frac{1}{6}\,a(\Delta x)^2 u_{xxx}\right]_j^n + \text{H.O.T.}$$

The leading order term of the truncation error is given by

$$T_j^n = \frac{1}{2}\,\Delta t u_{tt} - \frac{1}{2}\,\frac{(\Delta x)^2}{\Delta t}\,u_{xx} + \frac{1}{6}\,a(\Delta x)^2 u_{xxx}\big]_j^n$$

and we conclude that the scheme is only consistent if $(\Delta x)^2/\Delta t \to 0$ along the refinement path.