Math 623, F 2005: Homework 1. Solutions.

(1) (a) The exact solution is $u(x) = e^{-x^2/2}$.

(b) The discretization is

$$\frac{u_{n+1} - 2u_n + u_{n-1}}{(\Delta x)^2} + n\Delta x \frac{u_{n+1} - u_n}{\Delta x} + u_n = 0$$

so we get

$$\begin{cases} u_{n+1} = \frac{2+(n-1)(\Delta x)^2}{1+n(\Delta x)^2} u_n - \frac{1}{1+n(\Delta x)^2} u_{n-1} \\ u_0 = 1 \\ u_1 = 1. \end{cases}$$

(c) The discretization is

$$\frac{u_{n+1} - 2u_n + u_{n-1}}{(\Delta x)^2} + n\Delta x \frac{u_n - u_{n-1}}{\Delta x} + u_n = 0$$

so we get

$$\begin{cases} u_{n+1} = (2 - (n+1)(\Delta x)^2) u_n - (1 - n(\Delta x)^2) u_{n-1} \\ u_0 = 1 \\ u_1 = 1. \end{cases}$$

(d) The discretization is

$$\frac{u_{n+1} - 2u_n + u_{n-1}}{(\Delta x)^2} + n\Delta x \frac{u_{n+1} - u_{n-1}}{2\Delta x} + u_n = 0$$

so we get

$$\begin{cases} u_{n+1} = \frac{2-(\Delta x)^2}{1+\frac{1}{2}n(\Delta x)^2} u_n - \frac{1-\frac{1}{2}n(\Delta x)^2}{1+\frac{1}{2}n(\Delta x)^2} u_{n-1} \\ u_0 = 1 \\ u_1 = 1. \end{cases}$$

(e) Example of Matlab code (notice the change in indices as matlab doesn't like indices to run from 0 to N). There is no major difference in performance between the three algorithm. The central difference would have worked much better with the initial conditions $u_0 = 1$ and $u_1 = \exp(-\frac{1}{2}(\Delta x)^2)$.

```
clear;
K=12;
epsilon=zeros(3,K);
uN=zeros(3,K);

% forward differences
for k=1:K
  u=zeros(1,2^k+1);
  N=2^k;
  dx=1/N;
  u(1)=1;
% u(2)=exp(-1/2*dx^2);
  u(2)=1;
  for n=1:N-1
    u(n+2)=(2+(n-1)*dx^2)/(1+n*dx^2)*u(n+1)-1/(1+n*dx^2)*u(n);
  end
  uN(1,k)=u(N+1);
  epsilon(1,k)=abs(exp(-1/2)-u(N+1));
end

% backward differences
for k=1:K
  u=zeros(1,2^k+1);
  N=2^k;
  dx=1/N;
  u(1)=1;
% u(2)=exp(-1/2*dx^2);
  u(2)=1;
  for n=1:N-1
    u(n+2)=(2-(n+1)*dx^2)*u(n+1)-(1-n*dx^2)*u(n);
  end
  uN(2,k)=u(N+1);
  epsilon(2,k)=abs(exp(-1/2)-u(N+1));
end

% central differences
for k=1:K
  u=zeros(1,2^k+1);
  N=2^k;
  dx=1/N;
  u(1)=1;
% u(2)=exp(-1/2*dx^2);
  u(2)=1;
```

```
  for n=1:N-1
    u(n+2)=(2-dx^2)/(1+1/2*n*dx^2)*u(n+1)-...
           (1-1/2*n*dx^2)/(1+1/2*n*dx^2)*u(n);
  end
  uN(3,k)=u(N+1);
  epsilon(3,k)=abs(exp(-1/2)-u(N+1));
end

uN

clf;
hold on;
plot(1:K,-log(epsilon(1,:)),'*');
plot(1:K,-log(epsilon(2,:)),'+');
plot(1:K,-log(epsilon(3,:)),'.');
```

(2) (a) The grid is

$$\begin{cases} t = 0, \Delta t, \dots, M\Delta t = 1 \\ x = -1 = -N\Delta x, \dots, 0, \Delta x, \dots, N\Delta x = 1 \end{cases}$$

The explicit dicretization is

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} = (1 + (n\Delta x)^2)\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2}$$

so we get

$$u_n^{m+1} = \alpha(1 + (n\Delta x)^2)u_{n+1}^m + (1 - 2\alpha(1 + (n\Delta x)^2)u_n^m + \alpha(1 + (n\Delta x)^2)u_{n-1}^m$$

for $0 \le m < M$ and $-N < n < N$. We also have the initial condition

$$u_n^0 = (n\Delta x)^4 \quad -N \le n \le N$$

and the boundary conditions

$$u_{-N}^m = u_N^m = 1 \quad 1 \le m \le M$$

(b) The matlab code is given below. As usual one has to be careful with the indices. The values $\alpha = 0.25$ and $\alpha = 0.125$ give reasonable results, but $\alpha = 0.5$ gives nonsense.

```
clear;
K=6;
xvec=[-1:0.1:1];
for i=1:3
  a=2^(-i);
  fprintf('alpha=%f\n\n',a);
  U=zeros(21,K);
  for k=1:K
    N=2^k;
    uold=zeros(1,2*N+1);
    unew=zeros(1,2*N+1);
    dx=1/N;
    dt=a*dx^2;
    M=1/dt;
    x=[-1:dx:1];
    unew=x.^4;
    for m=0:M-1
      uold=unew;
      unew(1)=1;
      unew(2*N+1)=1;
      unew(2:2*N)=a*(1+x(2:2*N).^2).*(uold(1:2*N-1)+...
        uold(3:2*N+1))+...
        (1-2*a*(1+x(2:2*N).^2)).*uold(2:2*N);
    end
    U(:,k)=interp1(x,unew,xvec)';
  end
  [0 2.^(-[1:K]); xvec' U]
end
```

(3)

$$\begin{cases} u_n^m = & \text{Solution to the FD scheme;} \\ \hat{u}_n^m = u(n\Delta x, m\Delta t) & \text{Exact solution to PDE} \\ \epsilon_n^m = \hat{u}_n^m - u_n^m & \text{error.} \end{cases}$$

We must show that $\epsilon_n^m \to 0$ (uniformly, for all m, n) as $\Delta t \to 0$?

We know that

$$\frac{u_n^{m+1} - u_n^m}{\Delta t} - (1 + (n\Delta x)^2)\frac{u_{n+1}^m - 2u_n^m + u_{n-1}^m}{(\Delta x)^2} = 0$$

for $0 \le m < M$ and $-N < n < N$.

From Taylor expansion we have:

$$\frac{\hat{u}_n^{m+1} - \hat{u}_n^m}{\Delta t} = u_t(n\Delta x, m\Delta t) + C_n^m \Delta t \tag{1}$$

and

$$\frac{\hat{u}_{n+1}^m - 2\hat{u}_n^m + \hat{u}_{n-1}^m}{(\Delta x)^2} = u_{xx}(n\Delta x, m\Delta t) + D_n^m (\Delta x)^2 \tag{2}$$

Here C_n^m, D_n^m are uniformly bounded in m, n, Δt and Δx: say $|C_n^m| < C, |D_n^m| < D$. Subtracting and using $u_t = (1 + x^2)u_{xx}$ gives:

$$\frac{\hat{u}_n^{m+1} - \hat{u}_n^m}{\Delta t} - (1 + (n\Delta x)^2)\frac{\hat{u}_{n+1}^m - 2\hat{u}_n^m + \hat{u}_{n-1}^m}{(\Delta x)^2} = C_n^m \Delta t - (1 + (n\Delta x)^2)D_n^m (\Delta x)^2$$

Thus

$$\frac{\epsilon_n^{m+1} - \epsilon_n^m}{\Delta t} - (1 + (n\Delta x)^2)\frac{\epsilon_{n+1}^m - 2\epsilon_n^m + \epsilon_{n-1}^m}{(\Delta x)^2} = C_n^m \Delta t - (1 + (n\Delta x)^2)D_n^m (\Delta x)^2$$

This gives:

$$\begin{aligned} \epsilon_n^{m+1} &= \alpha(1 + (n\Delta x)^2)(\epsilon_{n+1}^m + \epsilon_{n-1}^m) + (1 - 2\alpha(1 + (n\Delta x)^2)\epsilon_n^m \\ &\quad + \Delta t(C_n^m \Delta t - (1 + (n\Delta x)^2)D_n^m (\Delta x)^2). \end{aligned}$$

for $0 \le m < M$ and $0 < n < N$. Note that $\epsilon_n^m = 0$ for $m = 0$ or $n = N^{\pm}$.

Write $\epsilon^m = \max_n |\epsilon_n^m|$ for $m = 0, 1, \ldots M$. Then we get from the triangle inequality that

$$\epsilon^{m+1} \le 2\alpha(1 + (n\Delta x)^2)\epsilon^m + |1 - 2\alpha(1 + (n\Delta x)^2)|\epsilon^m + \Delta t[C\Delta t + D(\Delta x)^2].$$

Since $n(\Delta x) \le 1$ and $\alpha \le 1/4$ we can remove the absolute value sign in the second term in the right hand side. Thus

$$\begin{cases} \epsilon^{m+1} \le \epsilon^m + \Delta t[C\Delta t + D(\Delta x)^2] \\ \epsilon^0 = 0 \end{cases}$$

By induction we get

$$\epsilon^m \le \underbrace{m\Delta t}_{\le 1}[C\Delta t + \underbrace{D(\Delta x)^2}_{=\alpha^{-1}\Delta t}] \le (C + \frac{D}{\alpha})\Delta t \to 0 \text{ as } \Delta t \to 0.$$

(4) (a) Define $v(x,t) = u(e^x, t)$. Then $u_t = v_t$, $yu_y = v_x$ and $y^2 v_{yy} = u_{xx} - u_x$. Thus we get

$$\begin{cases} v_t = 3v_{xx} - 6v_x, & 0 < x < 1, 0 < t < 1 \\ v(x,0) = 0, & 0 \le x \le 1 \\ v(0,t) = t, & 0 \le t \le 1 \\ v(1,t) = t^2 & 0 \le t \le 1 \end{cases} \tag{3}$$

(b) We essentially have the heat equation. The grid is

$$\begin{cases} t = 0, \Delta t, \dots, M\Delta t = 1 \\ x = 0, \Delta x, \dots, N\Delta x = 1. \end{cases}$$

The scheme is

$$\frac{v_n^m - v_n^{m-1}}{\Delta t} = 3\frac{v_{n+1}^m - 2v_n^m + v_{n-1}^m}{(\Delta x)^2} - 6\frac{v_{n+1}^m - v_{n-1}^m}{2\Delta x}$$

which can be written as

$$v_n^m = \frac{1}{1+6\alpha}\left(v_n^{m-1} + 3\alpha(1+\Delta x)v_{n-1}^m + 3\alpha(1-\Delta x)v_{n+1}^m\right)$$

for $0 \le m < M$ and $0 < n < N$. We also have the initial condition

$$v_n^0 = 0 \quad 0 \le n \le N$$

and the boundary conditions

$$u_0^m = m\Delta t \quad \text{and} \quad u_N^m = (m\Delta t)^2 \quad 1 \le m \le M$$

(c) See the code below. The SOR step becomes:

$$\begin{cases} \tilde{v}_n^{m,k+1} &= \frac{1}{1+6\alpha}\left(v_n^{m-1} + 3\alpha(1+\Delta x)v_{n-1}^{m,k+1} + 3\alpha(1-\Delta x)v_{n+1}^{m,k}\right) \\ v_n^{m,k+1} &= v_n^{m,k} + \omega(\tilde{v}_n^{m,k+1} - v_n^{m,k}). \end{cases}$$

Notice that it is hard to make the SOR step vectorized in matlab, so the code becomes quite slow.

```
clear;
J=4;
eps=1e-9;
xvec=[0:0.1:1];
a=0.25;           %may change this
omega=1.1;        %may change this
U=zeros(length(xvec),J);
for j=1:J
  N=2^j;
  uold=zeros(1,N+1);
  unew=zeros(1,N+1);
  utilde=zeros(1,N+1);
  dx=1/N;
  dt=a*dx^2;
  M=1/dt;
  x=[0:dx:1];
  unew=zeros(1,N+1);     %initial condition
  for m=1:M
    uold=unew;
    unew(1)=m*dt;        %boundary condition
    unew(N+1)=(m*dt)^2;  %boundary condition
    unew(2:N)=uold(2:N);
    error=Inf;
    while(error>eps)
      ucomp=unew;
      utilde(1)=unew(1);
      for n=2:N
        utilde(n)=1/(1+6*a)*(uold(n)+...
                   3*a*(1+dx)*utilde(n-1)+...
                   3*a*(1-dx)*unew(n+1));
        unew(n)=ucomp(n)+omega*(utilde(n)-ucomp(n));
      end
      error=sum((unew-ucomp).^2);
    end
    U(:,j)=interp1(x,unew,xvec)';
  end
end
[0 2.^(-[1:J]); xvec' U]
```