MATH-6840
Vignesh Ramakrishnan
Due: Tuesday February 22, 2022

Problem Set 5

1. (15 pts.) Consider the IBVP

$$
\begin{aligned}
u_t &= \nu u_{xx}, \qquad x \in (0,1), \qquad 0 < t \le T_f \\
u(x,0) &= f(x), \qquad x \in (0,1) \\
u_x(0,t) &= \alpha(t), \qquad u(1,t) = \beta(t), \qquad t \ge 0.
\end{aligned}
$$

Using the grid $x_j = j\Delta x$, $j = -1, 0, \ldots, N+1$, $\Delta x = 1/N$, apply the following discretization

$$
\begin{aligned}
D_{+t} v_j^n &= \nu D_{+x} D_{-x} \left(\theta v_j^{n+1} + (1-\theta) v_j^n \right), && \text{for } j = 0, 1, \ldots, N,\ n = 1, 2, \ldots \\
v_j^0 &= f(x_j) && \text{for } j = 0, 1, \ldots, N \\
D_{0x} v_0^n &= \alpha(t_n) && \text{for } n = 0, 1, \ldots \\
\nu D_{+x} D_{-x} v_N^n &= \beta^{'}(t_n) && \text{for } n = 0, 1, \ldots,
\end{aligned}
$$

where $\theta \in [0,1]$is a parameter (note $\theta = 1$corresponds to backward Euler, $\theta = \frac{1}{2}$corresponds to the trapezoidal rule, and $\theta = 0$ corresponds to forward Euler.

(a) Determine the order-of-accuracy (consistency) of the scheme including both the interior discretization and the boundary conditions. A Taylor's series expansion of the exact solution using the given discretization will help in determining the order of accuracy of this scheme given to us.

$$
\begin{aligned}
D_{+t} u_j^n &= \frac{u_j^{n+1} - u_j^n}{\Delta t} \\
u_j^{n+1} - u_j^n &= \left[\cancel{u} + \Delta t\ u_t + \frac{\Delta t^2}{2!}\ u_{tt} \right]_j^n - \cancel{u_j^n} \\
D_{+t} u_j^n &= \left[u_t + \frac{\Delta t}{2!}\ u_{tt} + \mathcal{O}(\Delta t^2) \right]_j^n
\end{aligned}
$$

$$
\begin{aligned}
\nu\ D_{+x} D_{-x} \left(\theta u_j^{n+1} + (1-\theta) u_j^n \right) &= \nu\theta D_{+x} D_{-x} \left[u + \Delta t\ u_t + \mathcal{O}(\Delta t^2) \right]_j^n + \nu(1-\theta) D_{+x} D_{-x} u_j^n \\
&= \cancel{\nu\theta\ D_{+x} D_{-x} u_j^n} + \nu\theta D_{+x} D_{-x} \left[\Delta t u_t + \mathcal{O}(\Delta t^2) \right]_j^n \\
&\quad + \nu D_{+x} D_{-x} u_j^n - \cancel{\nu\theta D_{+x} D_{-x} u_j^n} \\
&= \nu\theta D_{+x} D_{-x} \left[\Delta t u_t + \mathcal{O}(\Delta t^2) \right]_j^n + \nu D_{+x} D_{-x} u_j^n
\end{aligned}
$$

Here,

$$
\begin{aligned}
\nu D_{+x} D_{-x} u_j^n &= \nu \ \left[u_{xx} + \mathcal{O}(\Delta x^2)\right]_j^n \\
\nu\theta D_{+x} D_{-x} \left[\Delta t \ u_t + \mathcal{O}(\Delta t^2)\right]_j^n &= \nu\theta \ \Delta t D_{+x} D_{-x} \left[u_t\right]_j^n + \mathcal{O}(\Delta t^2) \\
&= \nu\theta \ \Delta t \frac{\left([u_t]_{j+1}^n - 2[u_t]_j^n + [u_t]_{j-1}^n\right)}{\Delta x^2} \\
&= \nu\theta \ \Delta t \left[u_{xxt} + \frac{2\Delta x^2}{4!} u_{xxxxt} + \mathcal{O}(\Delta x^4)\right]_j^n \\
&= \nu\theta \ \Delta t \left[u_{xxt} + \mathcal{O}(\Delta x^2)\right]_j^n
\end{aligned}
$$

Substituting them in the above equation results in,

$$
\begin{aligned}
\nu \ D_{+x} D_{-x} \ \left(\theta u_j^{n+1} + (1-\theta) u_j^n \ \right) &= \nu \ \left[u_{xx} + \mathcal{O}(\Delta x^2)\right]_j^n + \nu\theta \ \Delta t \left[u_{xxt} + \mathcal{O}(\Delta x^2)\right]_j^n \\
\tau_j^n = \ &\left[u_t + \frac{\Delta t}{2!} \ u_{tt} + \mathcal{O}(\Delta t^2)\right]_j^n - \nu \ \left[u_{xx} + \mathcal{O}(\Delta x^2)\right]_j^n + \nu\theta \ \Delta t \left[u_{xxt} + \mathcal{O}(\Delta x^2)\right]_j^n
\end{aligned}
$$

$$
\begin{aligned}
\tau_j^n &= \cancel{u_t} + \Delta t \left(\frac{1}{2} u_{tt} - \nu\theta u_{xxt}\right) + \mathcal{O}(\Delta t^2) - \ \cancel{\nu \ u_{xx}} + \mathcal{O}(\Delta x^2) \\
\tau_j^n &= \Delta t u_{tt} \left(\frac{1}{2} - \theta\right) + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x^2)
\end{aligned}
$$

If $\theta = \frac{1}{2}$, this scheme is second order accurate in both time and space, but if it isn't, then it is first order accurate in time and second order accurate in space.

(b) Using normal mode stability theory, determine the stability of the scheme taking account of the boundary conditions. Please use the notation that $r = \frac{\nu \Delta t}{\Delta x^2}$. Hint: for stability you need only consider the error equation so that the boundary conditions can be taken as homogeneous.

Expanding this discretization, we get,

$$
-r\theta \ v_{j+1}^{n+1} + (1+2r\theta) \, v_j^{n+1} - r\theta \ v_{j-1}^{n+1} = r\,(1-\theta)\, v_{j+1}^n + (1 - 2r\,(1-\theta))\, v_j^n + r\,(1-\theta)\, v_{j-1}^n
$$

Let $v_j^n = c a^n k^j$,

$$
\begin{aligned}
-r\theta\,(ak)\,\cancel{v_j^n} + (1-2r\theta)\, a\cancel{v_j^n} - r\theta \left(\frac{a}{k}\right) \cancel{v_j^n} &= r\,(1-\theta)\, k\cancel{v_j^n} + (1-2r\,(1-\theta))\, \cancel{v_j^n} + r\,(1-\theta)\, \frac{1}{k} \cancel{v_j^n} \\
-r\theta a \left(k + \frac{1}{k}\right) + (1-2r\theta)\, a &= r\,(1-\theta) \left(k + \frac{1}{k}\right) + (1 - 2r\,(1-\theta)) \\
(a - 2ra\theta) - (1 - 2r\,(1-\theta)) &= \ \left(k + \frac{1}{k}\right) (r\,(1-\theta) + ra\theta) \\
\left(k + \frac{1}{k}\right) &= \frac{(a-1)\,(1-2r) - 2r\theta}{r + r\theta\,(a-1)} = \mu
\end{aligned}
$$

$$
\begin{aligned}
\frac{k^2+1}{k} &= \mu \\
k^2 - \mu k + 1 &= 0 \\
\implies k = \frac{\mu \pm \sqrt{\mu^2-4}}{2} &= \frac{\mu}{2} \pm \sqrt{\left(\frac{\mu}{2}\right)^2 - 1}
\end{aligned}
$$

Let $\frac{\mu}{2} = \cos(\xi)$,

$$
k = \cos(\xi) \pm i \sin(\xi) = e^{\pm\, i\xi}
$$

$$
\begin{aligned}
v_j^n &= c_1 e^{i\xi j} + c_2 e^{-i\xi j} \\
v_0^n &= c_1 + c_2 = 0 \,(\text{Assumption - Homogenous BC}) \\
v_N^n &= c_1 \left(e^{i\xi N} - e^{-i\xi N}\right) = i\ 2c_1 \sin(N\xi) = 0
\end{aligned}
$$

$$
\begin{aligned}
\implies N\xi &= p\pi \\
\xi &= \frac{p\pi}{N} \\
\mu &= 2\cos\left(\frac{p\pi}{N}\right) = \frac{(a-1)(1-2r) - 2r\theta}{r + r\theta(a-1)} \\
\text{Let, } \cos\left(\frac{p\pi}{N}\right) &= \delta \\
\implies 2\delta\left(r + r\theta(a-1)\right) &= (a-1)(1-2r) - 2r\theta \\
2r\delta + 2\delta r\theta(a-1) &= (a-1)(1-2r) - 2r\theta \\
2r\delta + 2r\theta &= (a-1)(1-2r-2\delta r\theta) \\
a &= 1 + \frac{2r\delta + 2r\theta}{1-2r-2r\delta\theta}
\end{aligned}
$$

If $|a| \leq 1$, then,

$$
-1 \leq 1 + \frac{2r\delta + 2r\theta}{1-2r-2r\delta\theta} \leq 1
$$

From the right hand side inequality,

$$
\begin{aligned}
2r\delta + 2r\theta &\leq 1 - 2r - 2r\delta\theta \\
2r(\delta+\theta) &\leq 1 - 2r(1+\delta\theta) \\
2r(1+\delta+\theta+\delta\theta) &\leq 1 \\
r &\leq \frac{1}{2(1+\delta+\theta+\delta\theta)}
\end{aligned}
$$

This makes sense because when $\delta = 0, \theta = 0$, r becomes what we already know, $\leq \frac{1}{2}$.

(c) Based on the above, how do you expect the scheme converge with respect to grid parameters? Why?

The scheme will converge well as long as,

$$
\Delta t \leq \frac{\Delta x^2}{2\nu(1+\delta+\theta+\delta\theta)}
$$

2. (20 pts.) Here you will take steps to implement the discretization described in #1.

   (a) Carefully write down the $N+3$ linear equations that must be solved at each time step. Present this linear system.

$$-r\theta\ v_{j+1}^{n+1} + (1+2r\theta)\, v_j^{n+1} - r\theta\ v_{j-1}^{n+1} = r\,(1-\theta)\, v_{j+1}^n + (1-2r\,(1-\theta))\, v_j^n + r\,(1-\theta)\, v_{j-1}^n$$

$$\begin{aligned} D_{0x} v_0^n &= \alpha(t_n) && \text{for } n = 0,1,\ldots \\ D_{0x} v_0^{n+1} &= \alpha(t_{n+1}) && \text{for } n = 0,1,\ldots \\ \nu D_{+x} D_{-x} v_N^n &= \beta'(t_n) && \text{for } n = 0,1,\ldots, \\ \nu D_{+x} D_{-x} v_N^{n+1} &= \beta'(t_{n+1}) && \text{for } n = 0,1,\ldots, \end{aligned}$$

$$\begin{aligned} -v_{-1}^{n+1} + v_1^{n+1} &= 2\Delta x \alpha((n+1)\,\Delta t) \\ v_{N-1}^{n+1} - 2v_N^{n+1} + v_{N+1}^{n+1} &= \frac{\Delta x^2}{\nu} \beta'((n+1)\,\Delta t) \end{aligned}$$

$$\begin{bmatrix} -1 & 0 & 1 & \cdots & \cdots & \cdots & \cdots & \cdots \\ -r\theta & (1+2r\theta) & -r\theta & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & -r\theta & (1+2r\theta) & -r\theta & \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & -r\theta & (1+2r\theta) & -r\theta \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} v_{-1}^{n+1} \\ v_0^{n+1} \\ v_1^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ v_N^{n+1} \\ v_{N+1}^{n+1} \end{bmatrix} =$$

$$\begin{bmatrix} 2\Delta x\ \alpha\,((n+1)\,\Delta t) \\ r\,(1-\theta)\, v_1^n + (1-2r\,(1-\theta))\, v_0^n + r\,(1-\theta)\, v_{-1}^n \\ r\,(1-\theta)\, v_2^n + (1-2r\,(1-\theta))\, v_1^n + r\,(1-\theta)\, v_0^n \\ \vdots \\ \vdots \\ \vdots \\ r\,(1-\theta)\, v_{N-1}^n + (1-2r\,(1-\theta))\, v_N^n + r\,(1-\theta)\, v_{N+1}^n \\ \frac{\Delta x^2}{\nu}\beta'\,((n+1)\,\Delta t) \end{bmatrix}$$

   There are $N+1$ equations for solution variables v_0^{n+1} to v_N^{n+1}. Since there are two ghost nodes added, two extra equations are added at the top and at the bottom row of the linear system of equations and this constitutes to a total of $N+3$ equations.

   (b) Now implement the scheme in code using the solution $u_{ex} = e^{-\nu k^2 t}\sin(kx)$, from which you must determine $f(x)$, $\alpha(t)$, and $\beta(t)$. Note that much of the infrastructure can be adopted from the solution to PS #3 problem #4.

Through method of manufactured solutions, if $u_{ex} = e^{-\nu k^2 t}\sin(kx)$, then

$$
\begin{aligned}
u(x,t=0) &= \sin(kx) = f(x) \\
u_x(x=0,t) &= ke^{-\nu k^2 t}\cos(0) = ke^{-\nu k^2 t} = \alpha(t) \\
u(x=1,t) &= e^{-\nu k^2 t}\sin(k) = \beta(t) \\
u_t(x=1,t) &= -\nu k^2 \sin(k)\, e^{-\nu k^2 t} = \beta'(t)
\end{aligned}
$$

The code written for this question is included below.

Listing 1: Heat Equation - weighted Implicit scheme

```
function [err_norm,x,uhat,u_ex,A] = HeatEqn_ImplicitForcing(N,r,xlim1,xlim2
                                                    tlim1,tlim2,nu,k,theta)
% $Author: Vignesh Ramakrishnan$
% $RIN: 662028006$
% u_t - \nu u_{xx} = f(x,t)
% s.t u(x,0) = f(x)
% u_x(0,t)    = k\exp{-\nu*k^2*t} = \alpha(x)
% u(1,t)      = \beta(t)
% u_{ex} = e^{\nu*k^2*t}\sin(kx)
% This function is a method to prove that the Difference methods work and
% will be a good approximate to the exact solution. The task is to find
% functions f(x,t), u_x(0,t), \alpha(t) and \beta(t), plug it in
% and solve using the scheme: D+t v^n_j = \nu D+xD-xv^{n+1}_j + f^{n+1}_j
% Inputs: N        - Number of elements
%         r        - CFL number
%         xlim1    - left end of the spatial boundary
%         xlim2    - right end of the spatial boundary
%         tlim1    - start time of simulation
%         tmin2    - end time of simulation
%         nu       - Co-efficient of heat conduction
%         k        - wave number
%         theta    - weight of implicitness
% Output: err_norm - infinity norm of error between the exact solution and
%                    numerical solution

% nu  = 1;                                  % Co-efficient of heat conduction
% k   = 2;                                  % wave number

u0  = @(x) sin(k*x);                        % Initial condition
al  = @(t) k*exp(-nu*k^2*t);                % Neumann BC on Left Boundary
be  = @(t) sin(k)*exp(-nu*k^2*t);           % Dirchlet BC on Right Boundary
u1t = @(t) -nu*k^2*sin(k)*exp(-nu*k^2*t);% u_t @ x=1

dx  = (xlim2-xlim1)/N;                      % dx -spatial discretization

dt    = r*dx^2/nu;                          % r = CFL number
nStep= ceil((tlim2-tlim1)/dt);
```

```matlab
dt     = (tlim2-tlim1)/nStep;
r      = nu*dt/dx^2;                     % change r based on new dt

ng   = 1   ;                             % number of ghost points at BC
NP   = N+1+2*ng;                         % Total number of spatial points
ja   = ng+1;                             % xlim1's index number
jb   = NP-ng;                            % xlim2's index number

x    = (xlim1:dx:xlim2);                 % Spatial locations  x
t    = (tlim1:dt:tlim2);                 % Temporal locations t

u_prev   = zeros(NP,1);                  % Solution at previous tstep
u_curr   = zeros(NP,1);                  % Solution at current tstep

% set initial conditions for the spatial grid
u_prev(ja:jb) = u0(x);

% Set Neumann boundary condition
u_prev(ng)     = u_prev(ja+1) - 2*dx*al(tlim1);

% Set Compatability boundary condition
u_prev(NP)     = 2*u_prev(jb) - u_prev(jb-1) +...
                        (dx^2/nu)*u1t(tlim1);

% create Matrix A
A = zeros(NP);

for j = ng:NP
    if (j==ng)
        A(j,ng)    = -1;
        A(j,ja)    = 0;
        A(j,ja+1)  = 1;
    elseif (j==NP)
        A(j,jb-1)  = 1;
        A(j,jb)    = -2;
        A(j,NP)    = 1;
    else
        A(j,j-1)   = -r*theta;
        A(j,j)     = 1+2*r*theta;
        A(j,j+1)   = -r*theta;
    end
end

RHS = zeros(NP,1);

% find implicit solution each time step
for i=2:length(t)
    for j=ng:NP
```

```
            if j==ng
                RHS(j) = 2*dx*al(t(i));
            elseif j==NP
                RHS(j) = dx^2*u1t(t(i))/nu;
            else
                RHS(j) = r*(1-theta)*u_prev(j-1) + ...
                    (1-2*r*(1-theta))*u_prev(j) + r*(1-theta)*u_prev(j+1);
            end
        end
        u_curr = A\RHS;
        u_prev = u_curr;
end

u_ex = (exp(-nu*k^2*tlim2)*sin(k*x))';
uhat = u_curr(ja:jb);
err      = abs(u_ex - u_prev(ja:jb));
err_norm = max(err);

end
```

(c) Taking $\nu = 1$, $k = 2$ and $T_f = .4$, perform a convergence study with $N = 20, 40, 80, 160$ using $\theta = 1$ and $r = \frac{\nu\Delta t}{\Delta x} \approx 0.9$ (as usual, the time step may be slightly modified so the simulation actually attains the final time). Present plots of the solution and plots of the error at the final time for each grid resolution. Also present a log-log plot of the maximum error vs. the grid size, as well as a reference line indicating the expected convergence rate.

(d) Taking $\nu = 1$, $k = 2$ and $T_f = .4$, perform a convergence study with $N = 20, 40, 80, 160$ using $\theta = \frac{1}{2}$ and $r = \frac{\nu\Delta t}{\Delta x} \approx 0.9$ (as usual, the time step may be slightly modified so the simulation actually attains the final time). Present plots of the solution and plots of the error at the final time for each grid resolution. Also present a log-log plot of the maximum error vs. the grid size, as well as a reference line indicating the expected convergence rate.

Combining both part (c) and part (d), the results are presented below in Fig 1, Fig 2, Fig 3

3. (10 pts.) In HW #2 problem #2, we investigated the leapfrog scheme with a centered spatial discretization for the heat equation and experienced some difficulty in computing solutions.

(a) Determine the amplification factor of the discrete operator, and make surface plots of the amplitude of the amplification factor as a function of discrete wave number and the parameter $r = \frac{\nu\Delta t}{\Delta x^2}$. Note there are two roots and you should produce one plot for each root.

The Leapfrog scheme is written as,

$$\begin{aligned} D_{0t} v_j^n &= \nu \ D_{+x} D_{-x} v_j^n \\ v_j^{n+1} &= v_j^{n-1} + 2r \left(v_{j+1}^n - 2v_j^n + v_{j-1}^n \right) \end{aligned}$$

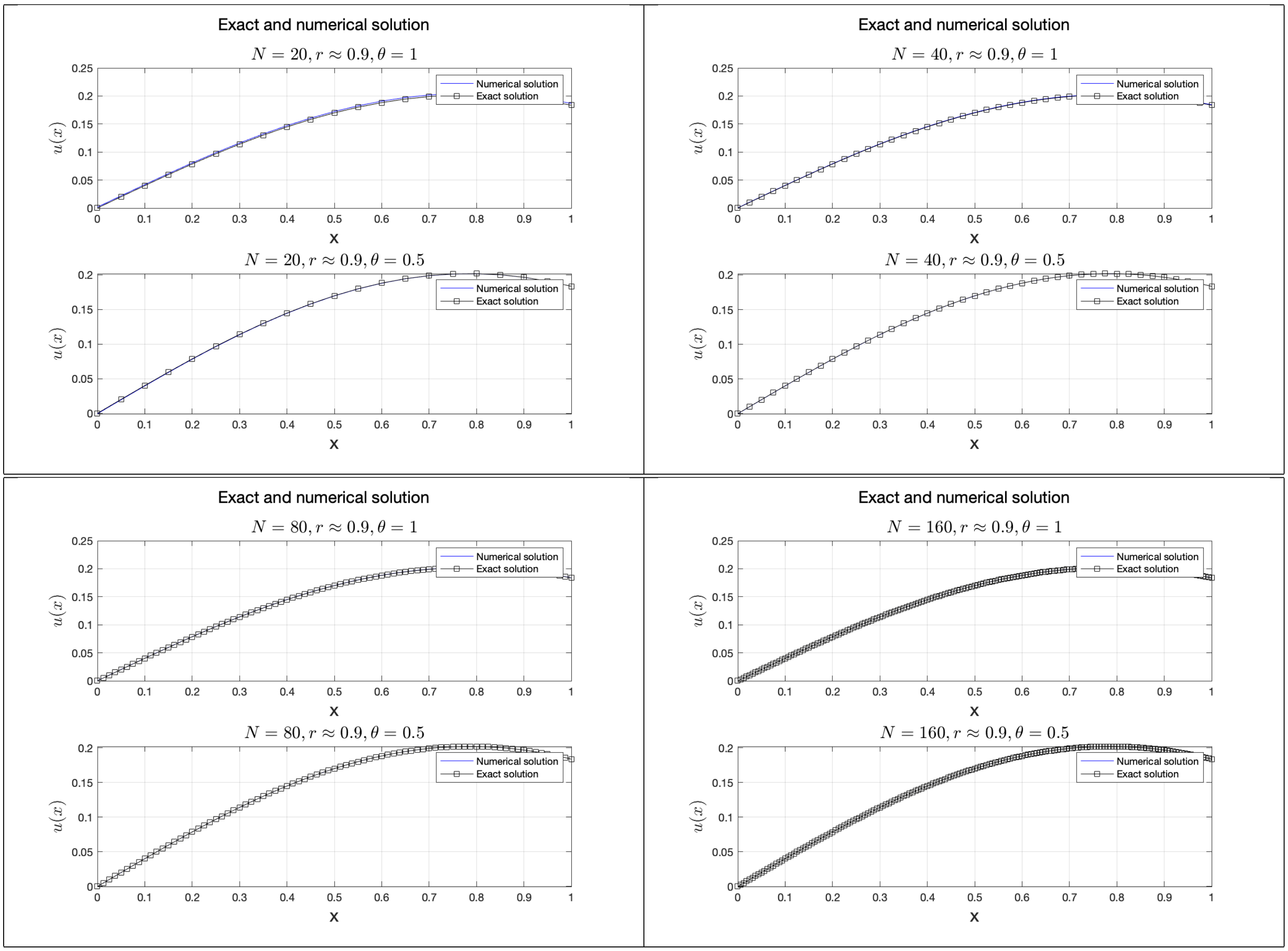

Figure 1: Solution plots

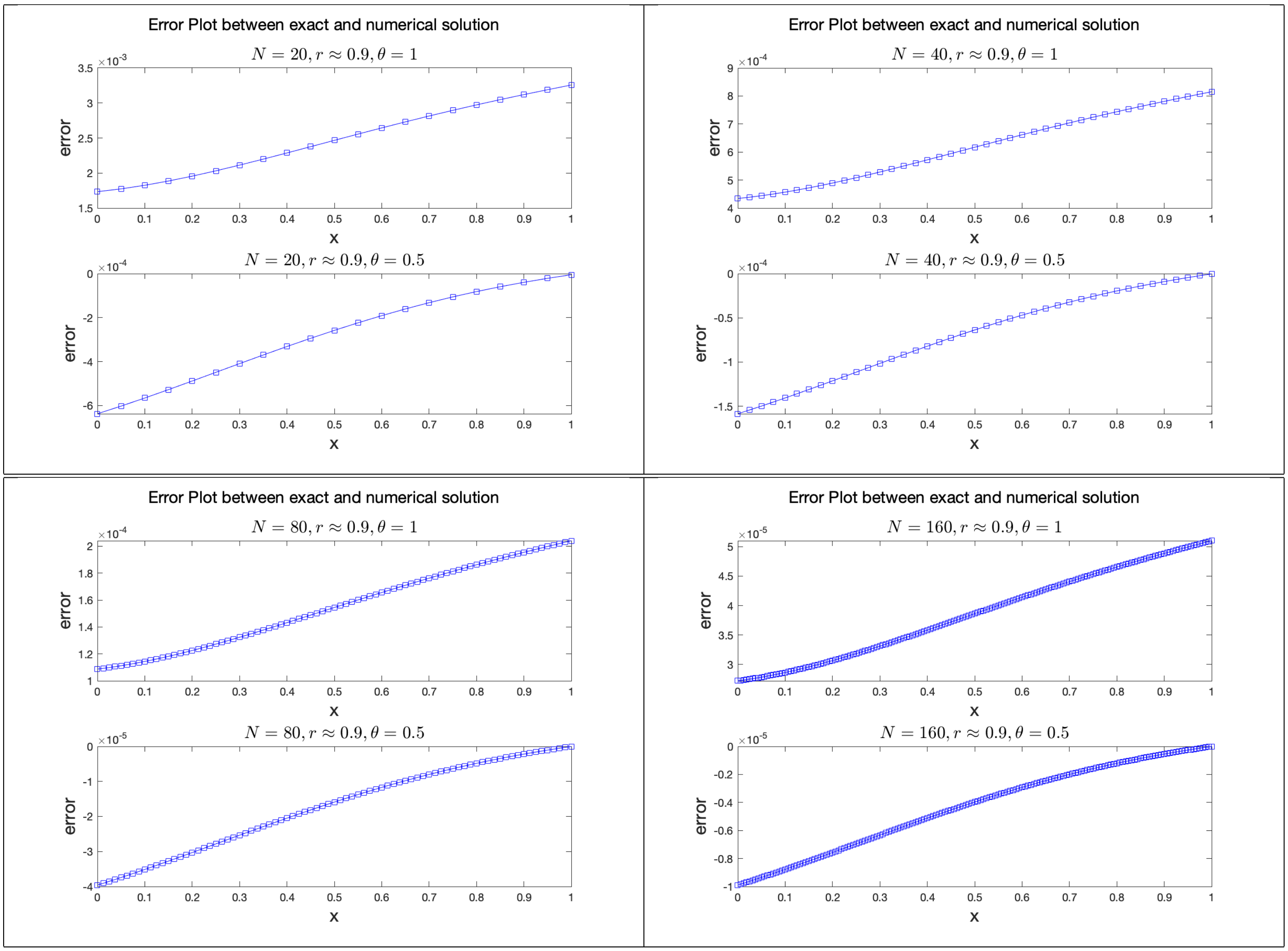

Figure 2: Error plots at different resolutions

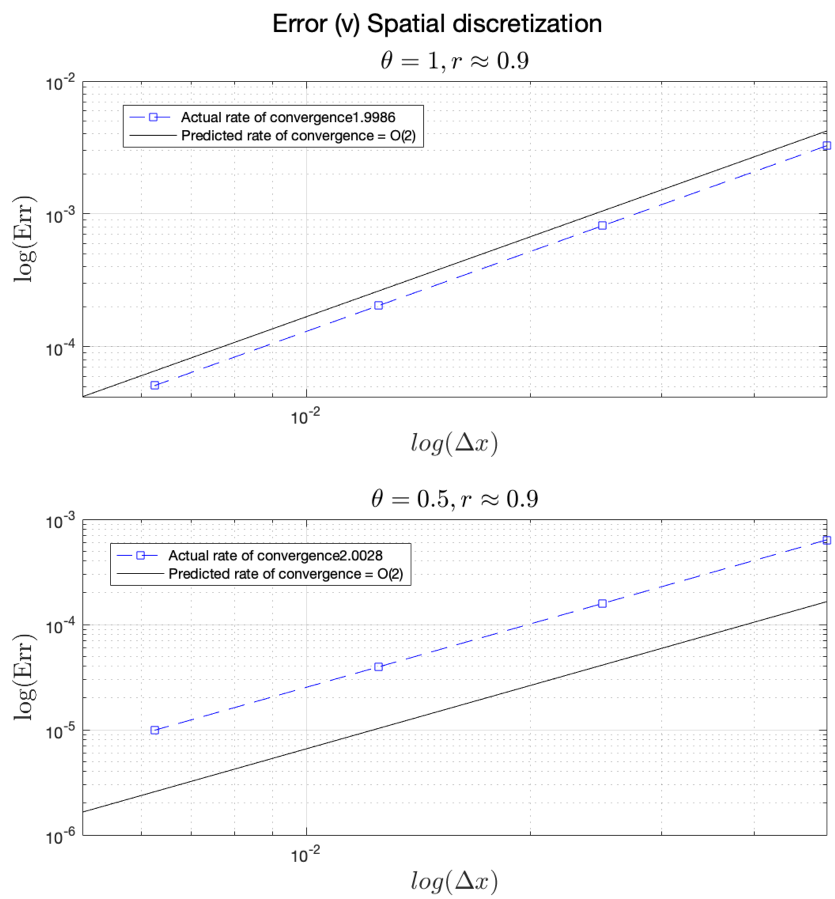

Figure 3: Log-Log plot of the infinity norm of error and spatial discretization

Let $v_j^n = ca^n k^j$, then the equation reduces to

$$a = \frac{1}{a} + 2r\left(k - 2 + \frac{1}{k}\right)$$
$$\left(a - \frac{1}{a}\right) = 2r\left(k + \frac{1}{k}\right) - 4r$$

$$\mu = \left(\frac{a^2 - 1}{2ar} + 2\right) = \left(k + \frac{1}{k}\right)$$

$$k = \frac{-\mu \pm \sqrt{\mu^2 - 4}}{2} = \frac{\mu}{2} \pm \sqrt{\left(\frac{\mu}{2}\right)^2 - 1}$$

If $\cos\xi = \frac{\mu}{2}$, then,

$$k = e^{\pm i\xi}$$

Assuming Homogeneous Boundary conditions,

$$v_0^n = c_1 + c_2 = 0$$
$$v_N^n = c_1\left(e^{i\xi N} - e^{-i\xi N}\right) = 0$$

Solving for ξ yields, $\xi = \frac{p\pi}{N}$

$$\frac{\mu}{2} = \cos\left(\frac{p\pi}{N}\right) = \delta$$

$$a^2 + 4ar(1 - \delta) - 1 = 0$$

Solving for a results in two solutions

$$a_1 = \frac{-4r(1-\delta) + \sqrt{16r^2(1-\delta)^2 + 4}}{2}$$
$$a_2 = \frac{-4r(1-\delta) - \sqrt{16r^2(1-\delta)^2 + 4}}{2}$$

The surface plots are shown below, in Fig 4

(b) Determine if the scheme is stable for any choice of the parameter $r > 0$ (hint: you may find it useful to use the plots from (a) as a guide). How do these results help to explain the behavior that we experienced in HW #2 problem #2.

There is one root of the amplification factor a where when, $r >$ somewhere around 0.5, the magnitude of the amplification factor becomes > 1 and hence the scheme becomes unstable.

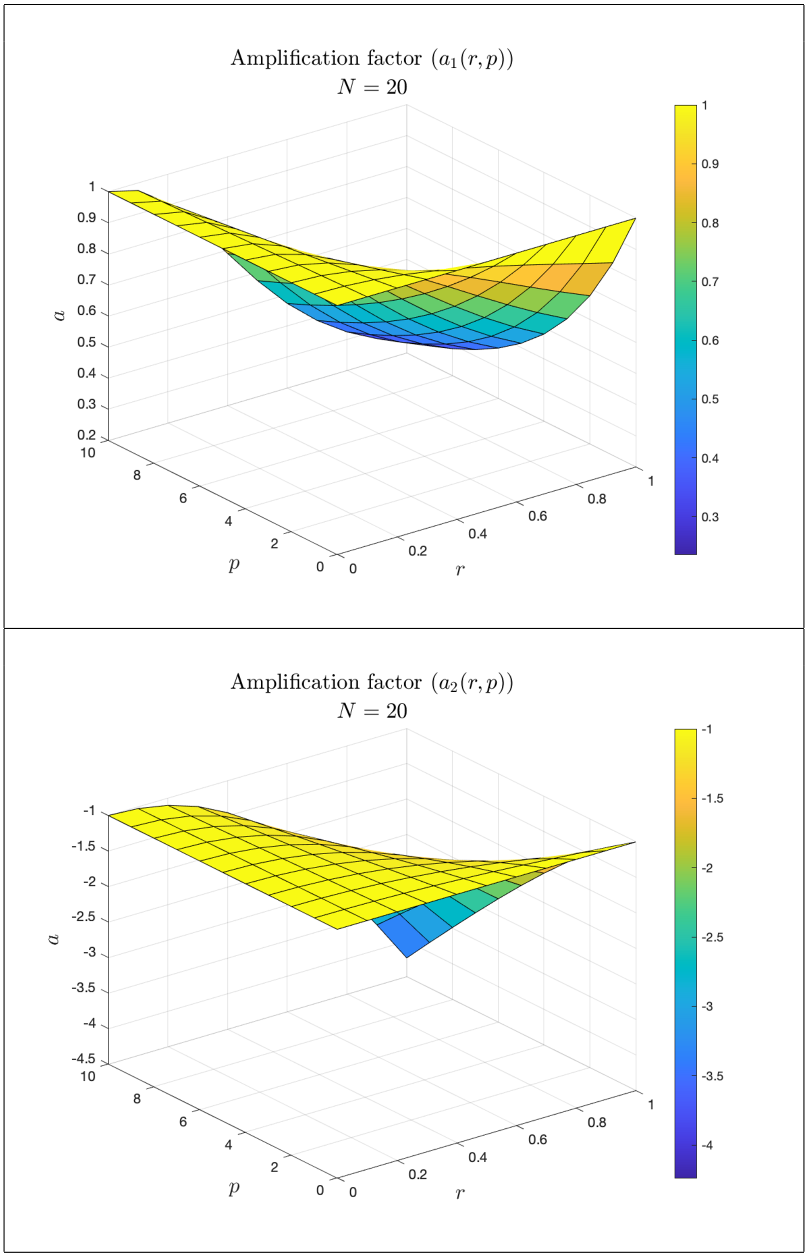

Figure 4: Amplification factor