# AM 255: Problem Set 6

Douglas Lanman
5 December 2006

## Problem 1

---

Consider the following initial value problem.

$$\begin{aligned} u_t &= -u_{xxxx}, \quad -\infty < x < \infty,\ 0 \le t \\ u(x,0) &= f(x) = \sin(x), \quad -\infty < x < \infty \end{aligned} \tag{1}$$

Find the analytic solution and implement the Crank-Nicholson approximation for Equation 1. Evaluate the numerical solution at time $T = 2\pi$ with discrete grids of size $N = \{20, 40, 80, 160, 320\}$ and $k = h$. Graphically compare the exact solution to the numerical solution and tabulate the $L_2$-errors. Finally, estimate the order of approximation achieved.

---

Let's begin by deriving a closed-form solution for Equation 1. Following the derivation on pages 38 and 39 of [1] we assume a solution of the following form.

$$u(x,t) = \frac{1}{\sqrt{2\pi}} e^{i\omega x} \hat{u}(\omega, t) \tag{2}$$

Substituting Equation 2 into Equation 1 yields the following ordinary differential equation

$$\hat{u}_t(\omega, t) = -\omega^4 \hat{u}(\omega, t), \quad \hat{u}(\omega, 0) = \hat{f}(\omega),$$

which has the general solution

$$\hat{u}(\omega, t) = e^{-\omega^4 t} \hat{f}(\omega) \quad \Rightarrow \quad u(x,t) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} e^{i\omega x} e^{-\omega^4 t} \hat{f}(\omega). \tag{3}$$

At this point, we require the Fourier series for the initial condition. As was found in Problem 3 of Problem Set 4, the Fourier coefficients $\hat{f}(\omega)$ can be obtained by inspection.

$$f(x) = \sin(x) = \frac{e^{ix} - e^{-ix}}{2i} = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} e^{i\omega x} \hat{f}(\omega)$$

$$\Rightarrow \hat{f}(\omega) = \begin{cases} -i\sqrt{\frac{\pi}{2}} & \text{if } \omega = 1 \\ i\sqrt{\frac{\pi}{2}} & \text{if } \omega = -1 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Substituting Equation 4 into Equation 3 gives the desired analytic solution.

$$u(x,t) = e^{-t} \left( \frac{e^{ix} - e^{-ix}}{2i} \right) \quad \Rightarrow \quad \boxed{u(x,t) = e^{-t} \sin(x)}$$

Now we turn our attention to deriving the Crank-Nicholson approximation to Equation 1. Recall (from Equation 2.3.3 in [1]) that the Crank-Nicholson scheme for $u_t = u_x$ is given by

$$\left( I - \frac{k}{2} D_0 \right) v_j^{n+1} = \left( I + \frac{k}{2} D_0 \right) v_j^n, \ j = 0, 1, \ldots, N. \tag{5}$$

Similarly, recall (from Equation 2.5.19 in [1]) that the Crank-Nicholson scheme for $u_t = u_{xx}$ is given by

$$\left(I - \frac{k}{2} D_+ D_-\right) v_j^{n+1} = \left(I + \frac{k}{2} D_+ D_-\right) v_j^n,\ j = 0, 1, \ldots, N. \tag{6}$$

Finally, note that (according to Equation 2.7.7 in [1]) the most natural centered difference approximation to the fourth partial derivative is given by

$$\frac{\partial^4}{\partial x^4} \rightarrow Q_4 = (D_+ D_-)^2 = D_+ D_- D_+ D_-. \tag{7}$$

Combining Equations 5, 6, and 7, it is apparent that the corresponding Crank-Nicholson scheme for $u_t = -u_{xxxx}$ is given by

$$\boxed{\left(I + \frac{k}{2}\left(D_+ D_- D_+ D_-\right)\right) v_j^{n+1} = \left(I - \frac{k}{2}\left(D_+ D_- D_+ D_-\right)\right) v_j^n,\ j = 0, 1, \ldots, N.} \tag{8}$$

My implementation of the discrete difference approximation, as defined by Equation 8, was completed using Matlab and is included as `CrankNicholson.m`. Before presenting the results of my program, I will briefly outline the architecture of the source code. On lines 11-51 I select the values of $\{N, h, k\}$ and determine the resulting grid points $\{x, t\}$. (Note that on lines 41-43 I ensure that the last time is given by $T = 2\pi$.) Lines 53-83 implement Equation 8. Note that I directly solve for the amplification factor $Q$ on lines 65-67 using the difference operators $D_+$ and $D_-$ evaluated on lines 61 and 62. Finally, lines 85-123 create the tables and plots shown in this write-up.

Recall from class on 11/20/06 that we expect the Crank-Nicholson scheme in Equation 8 to be second-order in both space in time. As tabulated below, the approximation results for $k = h$ (i.e., equal space and time step sizes) confirm this expectation.

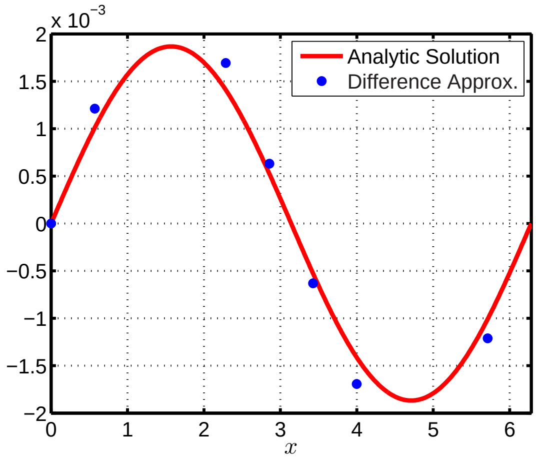
| $N$ | $L_2$-error | order |
|---|---|---|
| 10 | 6.593e-4 | NA |
| 20 | 1.617e-4 | 2.03 |
| 40 | 4.115e-5 | 1.97 |
| 80 | 1.046e-5 | 1.98 |
| 160 | 2.641e-6 | 1.99 |
| 320 | 6.642e-7 | 1.99 |

Note that the standard definition of the discrete $L_2$-norm was used to evaluate the total error as

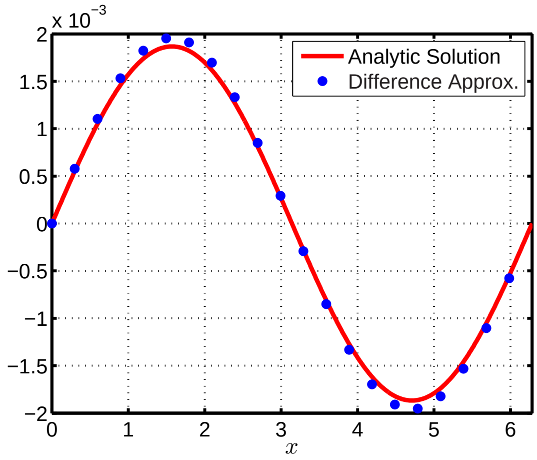
$$L_2\text{-error}(N) \triangleq \sqrt{\sum_{j=0}^{N} |u(x_j, t^n) - v_j^n|^2 h}.$$

In addition, the following definition of order of approximation was given in class.

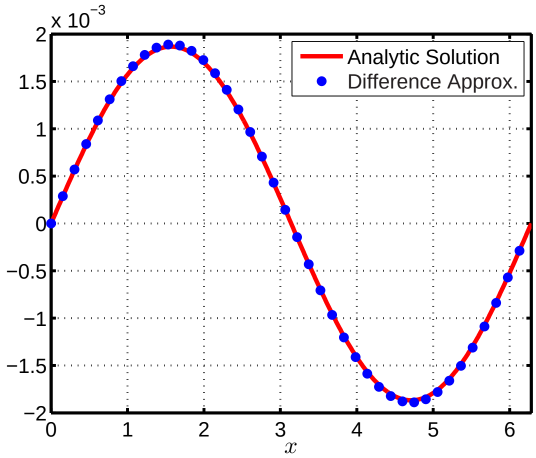
$$\text{order} \triangleq \log_2\left(\frac{L_2\text{-error}(N)}{L_2\text{-error}(2N)}\right)$$



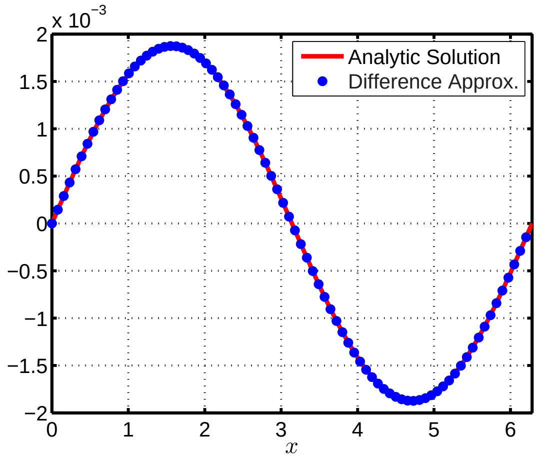

(a) $N = 10$



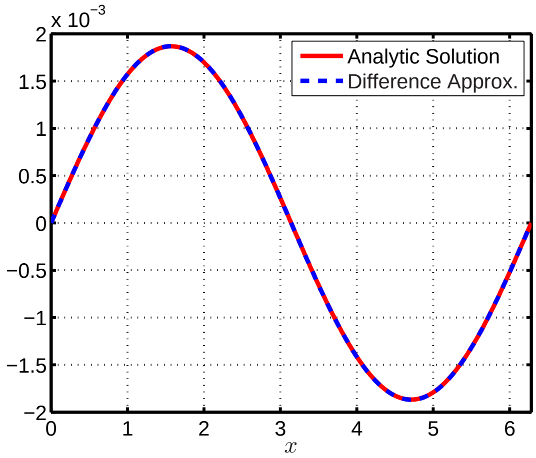

(b) $N = 20$



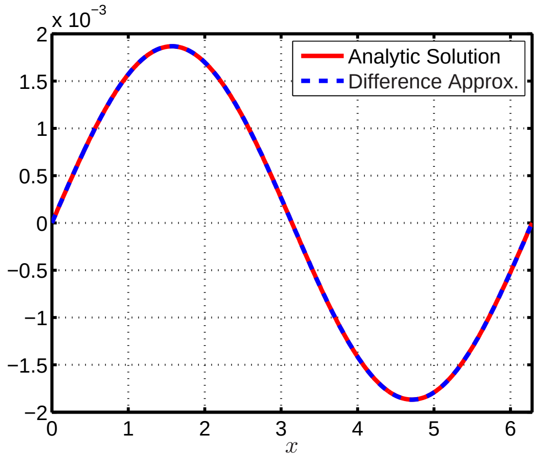

(c) $N = 40$




(d) $N = 80$




(e) $N = 160$




(f) $N = 320$

Figure 1: Comparison between the Crank-Nicholson difference approximation and the analytic solution of Equation 1 at time $T = 2\pi$, for $N = \{10, 20, 40, 80, 160, 320\}$ and $k = h$.

# Problem 2

Consider the following two-dimensional initial value problem.

$$\begin{aligned} u_t &= -u_{xxxx} - u_{yyyy}, \quad -\infty < x, y < \infty,\ 0 \le t \\ u(x,y,0) &= f(x,y) = \sin(x+y), \quad -\infty < x, y < \infty \end{aligned} \tag{9}$$

Find the analytic solution and implement the Crank-Nicholson approximation for Equation 9. Evaluate the numerical solution at time $T = 2\pi$ with discrete grids of size $N = \{20, 40, 80, 160, 320\}$ and $k = h$. Graphically compare the exact solution to the numerical solution and tabulate the $L_2$-errors. Finally, estimate the order of approximation achieved.

---

Let's begin by making the change of variables such that $z \triangleq x+y$. Furthermore, let's assume that the solution has the separable form $u(z,t) = Z(z)T(t)$, where $Z(z)$ is a function of a single variable $z = x + y$ and $T(t)$ is a function of time. Under this change of variables, Equation 9 is transformed as follows.

$$u_t = -2u_{zzzz}, \quad u(z,0) = f(z) = \sin(z)$$

Note that this expression has a similar form as Equation 1 – only differing by the constant multiplier on the right-hand side. As a result, the Fourier transform is given by

$$\hat{u}_t(\omega,t) = -2\omega^4 \hat{u}(\omega,t), \quad \hat{u}(\omega,0) = \hat{f}(\omega),$$

which has the general solution

$$\hat{u}(\omega,t) = e^{-2\omega^4 t}\hat{f}(\omega) \quad \Rightarrow \quad u(z,t) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} e^{i\omega z} e^{-2\omega^4 t}\hat{f}(\omega).$$

Substituting Equation 4 into this expression gives the analytic solution for Equation 9.

$$u(z,t) = e^{-2t}\left(\frac{e^{iz} - e^{-iz}}{2i}\right) \quad \Rightarrow \quad \boxed{u(x,y,t) = e^{-2t}\sin(x+y)}$$

Now we turn our attention to deriving a second-order approximation scheme for Equation 9. First, note that the full Crank-Nicholson scheme is given by

$$\left(I + \frac{k}{2}\left((D_{+x}D_{-x})^2 + (D_{+y}D_{-y})^2\right)\right)v^{n+1} = \left(I - \frac{k}{2}\left((D_{+x}D_{-x})^2 + (D_{+y}D_{-y})^2\right)\right)v^n.$$

Rather than directly implementing this scheme, we will use the *Stang-splitting* technique to reduce the computation complexity. As described on pages 195-200 in [1], Strang-splitting can be used to implement general one step methods for $u_t = (P_1 + P_2)u$, where $P_1$ and $P_2$ are linear differential operators in space. If we let $Q_1$ and $Q_2$ denote the amplification factors for each component, then $v^{n+1} = Q_1 v^n$ is an approximation of $v_t = P_1 v$, and $w^{n+1} = Q_2 w^n$ is an approximation of $w_t = P_2 w$. For this problem, $P_1 = -\partial/\partial_{xxxx}$ and $P_2 = -\partial/\partial_{yyyy}$. Using the one-dimensional Crank-Nicholson scheme, $Q_1$ and $Q_2$ have the following forms.

$$Q_1(k) = \left(I + \frac{k}{2}\left(D_{+x}D_{-x}D_{+x}D_{-x}\right)\right)^{-1}\left(I - \frac{k}{2}\left(D_{+x}D_{-x}D_{+x}D_{-x}\right)\right) \tag{10}$$

$$Q_2(k) = \left(I + \frac{k}{2}\left(D_{+y}D_{-y}D_{+y}D_{-y}\right)\right)^{-1}\left(I - \frac{k}{2}\left(D_{+y}D_{-y}D_{+y}D_{-y}\right)\right) \tag{11}$$

Substituting Equations 10 and 11 into Equation 5.4.12 in [1] provides the following second-order splitting scheme for Equation 9.

$$\boxed{v^{n+1} = Q_1\left(\frac{k}{2}, t_{n+1/2}\right) Q_2\left(k, t_n\right) Q_1\left(\frac{k}{2}, t_n\right) v^n} \tag{12}$$

My implementation of the second-order Stang-splitting scheme, as defined by Equation 12, was completed using Matlab and is included as `StrangSplitting.m`. Before presenting the results of my program, I will briefly outline the architecture of the source code. On lines 11-54 I select the values of $\{N, h, k\}$ and determine the resulting grid points $\{x, y, t\}$. (Note that on lines 44-46 I ensure that the last time is given by $T = 2\pi$.) Lines 56-95 implement Equation 12. Note that I directly solve for the amplification factors $Q_1$ and $Q_2$ on lines 67-75 using the difference operators $D_+$ and $D_-$ evaluated on lines 64 and 65. Finally, lines 97-147 create the tables and plots shown in this write-up.

Recall from class on 11/20/06 that we expect the Strang-splitting scheme in Equation 12 to be second-order in both space in time. As tabulated below, the approximation results for $k = h$ (i.e., equal space and time step sizes) confirm this expectation.

| $N$ | $L_2$-error | order |
|---|---|---|
| 10 | 9.506e-6 | NA |
| 20 | 2.144e-6 | 2.15 |
| 40 | 5.337e-7 | 2.01 |
| 80 | 1.349e-7 | 1.98 |
| 160 | 3.402e-8 | 1.99 |
| 320 | 8.551e-9 | 1.99 |

Note that the discrete $L_2$-norm was used to evaluate the total error as

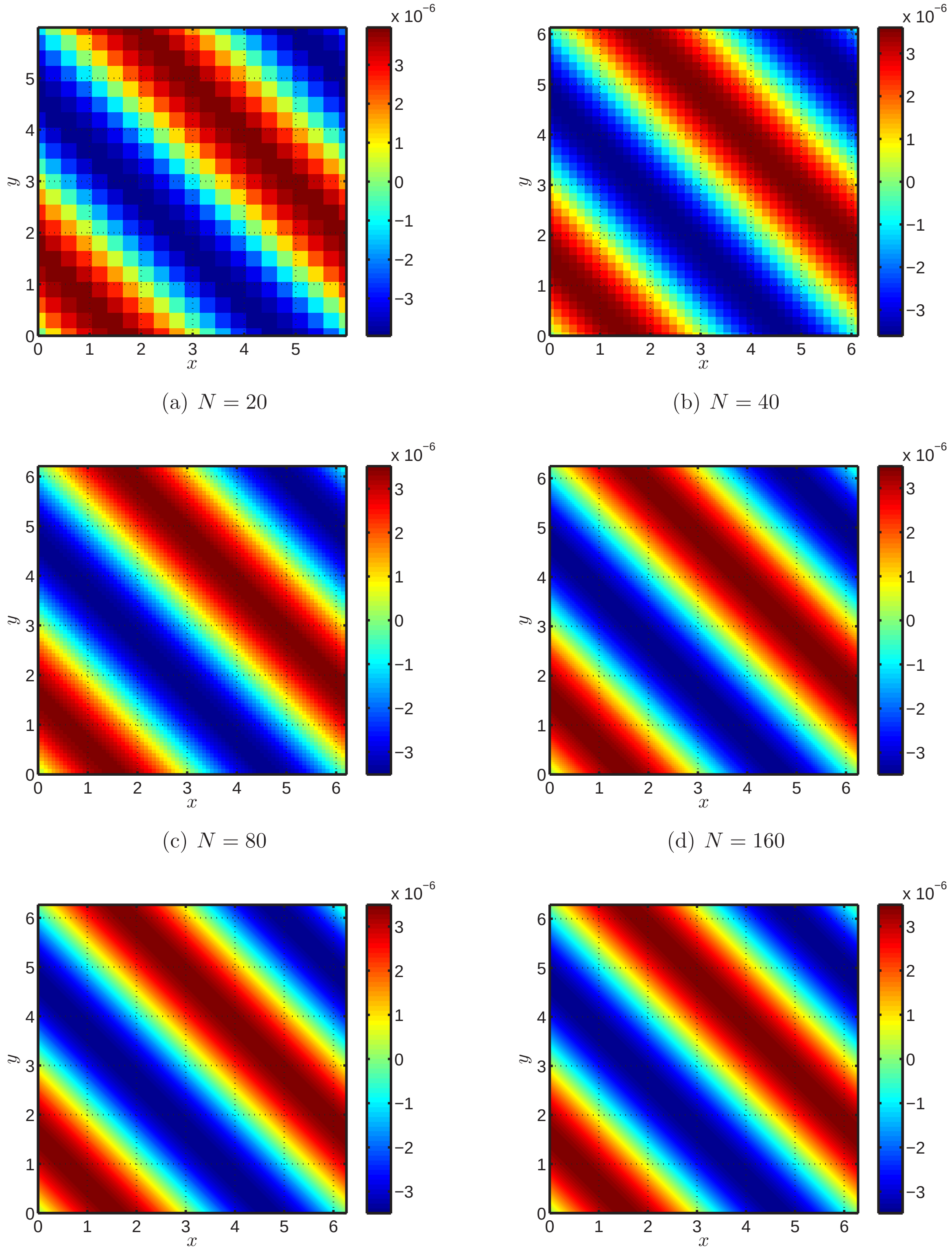
$$L_2\text{-error}(N) \triangleq \sqrt{\sum_{j=0}^{N}\sum_{k=0}^{N} |u(x_j, y_k, t^n) - v_{jk}^n|^2 h^2}.$$

As in Problem 1, the following definition of order of approximation was used in this analysis.

$$\text{order} \triangleq \log_2\left(\frac{L_2\text{-error}(N)}{L_2\text{-error}(2N)}\right)$$

# References

[1] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time Dependent Problems and Difference Methods*. John Wiley & Sons, 1995.




(a) $N = 20$

(b) $N = 40$

(c) $N = 80$

(d) $N = 160$

(e) $N = 320$

(f) exact solution

Figure 2: Comparison between the Strang-splitting difference approximation and the analytic solution of Equation 9 at time $T = 2\pi$, for $N = \{20, 40, 80, 160, 320\}$ and $k = h$.

```
% AM 255, Problem Set 6, Problem 1
%    Solves u_t = -u_xxxx IVP using the Crank-Nicholson
%    scheme. Results are displayed graphically and
%    tabulated for inclusion in the write-up.
%
% Douglas Lanman, Brown University, Dec. 2006

% Reset Matlab environment and command window.
clear all; clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part I: Specify discrete grid parameters.

% Define the initial condition.
IC = @(x) sin(x);

% Define the exact solution.
ES = @(x,t) exp(-t)*sin(x);

% Define space grid interval(s) for evaluation.
N = [10 20 40 80 160 320]; % #gridpoints s.t. N+2 on [0,2*pi]
h = 2*pi./(N+1);           % resulting space steps

% Select the final time for evaluation.
% Note: Initial time is assumed to be zero.
tf = 2*pi;

% Select time step.
% Note: This scheme is unconditionally stable.
k = h;

% Set discrete positions/time-steps for evaluation.
% Note: All time steps will be equal, except the
%       last; it will be adjusted so that the final
%       time will be exactly 'tf'.
x = cell(1,length(N));
t = cell(1,length(N));
for i = 1:length(N)
   x{i} = h(i)*(0:N(i));
   t{i} = (0:k(i):tf);
   if t{i}(end) ~= tf
      t{i}(end+1) = tf;
   end
end

% Initialize the numerical solution(s).
v = cell(1,length(N));
for i = 1:length(N)
   v{i} = zeros(length(t{i}),N(i)+1);
   v{i}(1,:) = IC(x{i}); % boundary values
```

```
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part II: Solve the IVP using Crank-Nicholson scheme.

% Update solution sequentially (beginning with I.C.).
for i = 1:length(N)

   % Store forward/backward difference operators.
   I  = eye(N(i)+1);
   Dp = (1/h(i))*(circshift(I,[0 1])-I);
   Dm = (1/h(i))*(I-circshift(I,[0 -1]));

   % Evaluate amplification factor.
   A = I + (k(i)/2)*(Dp*Dm*Dp*Dm);
   B = I - (k(i)/2)*(Dp*Dm*Dp*Dm);
   Q = B/A;

   % Calculate Crank-Nicholson solution.
   % Note: Modify amplication factor for the last time step.
   for n = 1:(length(t{i})-1)
      if n ~= (length(t{i})-1)
         v{i}(n+1,:) = (Q*v{i}(n,:)')';
      else
         kf = diff(t{i}(end-1:end));
         A  = I + (kf/2)*(Dp*Dm*Dp*Dm);
         B  = I - (kf/2)*(Dp*Dm*Dp*Dm);
         Q  = B/A;
         v{i}(n+1,:) = (Q*v{i}(n,:)')';
      end
   end

end % End of Crank-Nicholson solution.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part III: Plot/tabulate modeling results.

% Evaluate the exact solution.
xe = linspace(0,2*pi,1000);
fe = ES(xe,tf);

% Determine the L2-error and the approximation order.
L2_error = zeros(1,length(N));
order    = zeros(1,length(N));
for i = 1:length(N)
   L2_error(i) = sqrt(sum((abs(ES(x{i},t{i}(end))-v{i}(end,:)).^2)*h(i)));
   if i > 1
      order(i) = log2(L2_error(i-1)/L2_error(i));
   end
end
```

```
% Tabulate results.
disp(' N     L2-error     order');
disp('---------------------------');
for i = 1:length(N)
   if i > 1
      fprintf('%3d   %.5g   %+2.2f\n',N(i),L2_error(i),order(i));
   else
      fprintf('%3d   %.5g\n',N(i),L2_error(i));
   end
end

% Compare approximation to exact solution.
figure(1); clf;
plot(xe,fe,'r-','LineWidth',3);
hold on;
   plot(x{3},v{3}(end,:),'.','MarkerSize',20,'LineWidth',3);
hold off;
set(gca,'LineWidth',2,'FontSize',14,'FontWeight','normal');
xlabel('$x$','FontName','Times','Interpreter','Latex','FontSize',16);
%title('Difference Approximation vs. Analytic Solution');
grid on; xlim([0 2*pi]); ylim(2e-3*[-1 1]);
legend('Analytic Solution','Difference Approx.');
```

```
% AM 255, Problem Set 6, Problem 2
%    Solves u_t = -u_xxxx-u_yyyy IVP using Strang Splitting
%    and the Crank-Nicholson scheme. Results are displayed
%    graphically and tabulated for the write-up.
%
% Douglas Lanman, Brown University, Dec. 2006

% Reset Matlab environment and command window.
clear all; clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part I: Specify discrete grid parameters.

% Define the initial condition.
IC = @(x,y) sin(x+y);

% Define the exact solution.
ES = @(x,y,t) exp(-2*t)*sin(x+y);

% Define space grid interval(s) for evaluation.
% Note: Use equal number of points along x and y axes.
N = [10 20 40 80 160 320]; % #gridpoints s.t. N+2 on [0,2*pi]
h = 2*pi./(N+1);           % resulting space steps

% Select the final time for evaluation.
% Note: Initial time is assumed to be zero.
tf = 2*pi;

% Select time step.
% Note: This scheme is unconditionally stable.
k = h;

% Set discrete positions/time-steps for evaluation.
% Note: All time steps will be equal, except the
%       last; it will be adjusted so that the final
%       time will be exactly 'tf'.
x = cell(1,length(N));
y = cell(1,length(N));
t = cell(1,length(N));
for i = 1:length(N)
   x{i} = repmat(h(i)*(0:N(i)),N(i)+1,1);
   y{i} = repmat(h(i)*(0:N(i))',1,N(i)+1);
   t{i} = (0:k(i):tf);
   if t{i}(end) ~= tf
      t{i}(end+1) = tf;
   end
end

% Initialize the numerical solution(s).
v = cell(1,length(N));
```

```
for i = 1:length(N)
   v{i} = zeros(N(i)+1,N(i)+1,length(t{i}));
   v{i}(:,:,1) = IC(x{i},y{i}); % boundary values
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part II: Solve IVP using Strang Splitting with Crank-Nicholson scheme.

% Update solution sequentially (beginning with I.C.).
for i = 1:length(N)

   % Store forward/backward difference operators.
   I  = eye(N(i)+1);
   Dp = (1/h(i))*(circshift(I,[0 1])-I);
   Dm = (1/h(i))*(I-circshift(I,[0 -1]));

   % Evaluate amplification factors (for a full step).
   Af = I + (k(i)/2)*(Dp*Dm*Dp*Dm);
   Bf = I - (k(i)/2)*(Dp*Dm*Dp*Dm);
   Qf = Bf/Af;

   % Evaluate amplification factors (for a half step).
   Ah = I + (k(i)/4)*(Dp*Dm*Dp*Dm);
   Bh = I - (k(i)/4)*(Dp*Dm*Dp*Dm);
   Qh = Bh/Ah;

   % Calculate (second-order accurate) splitting solution.
   % Note: Use the 1D Crank-Nicholson amplification factors.
   %       Modify amplication factors for the last time step.
   for n = 1:(length(t{i})-1)
      if n ~= (length(t{i})-1)
         v{i}(:,:,n+1) = (Qh*(Qf*(Qh*v{i}(:,:,n)')')')';
      else
         kf = diff(t{i}(end-1:end));
         Af = I + (kf/2)*(Dp*Dm*Dp*Dm);
         Bf = I - (kf/2)*(Dp*Dm*Dp*Dm);
         Qf = Bf/Af;
         Ah = I + (kf/4)*(Dp*Dm*Dp*Dm);
         Bh = I - (kf/4)*(Dp*Dm*Dp*Dm);
         Qh = Bh/Ah;
         v{i}(:,:,n+1) = (Qh*(Qf*(Qh*v{i}(:,:,n)')')')';
      end
   end

end % End of Strang Splitting solution.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part III: Plot/tabulate modeling results.

% Evaluate the exact solution.
```

```
xe = repmat(linspace(0,2*pi,1000),1000,1);
ye = repmat(linspace(0,2*pi,1000)',1,1000);
fe = ES(xe,ye,tf);

% Determine the L2-error and the approximation order (in space and time).
L2_error = zeros(1,length(N));
order    = zeros(1,length(N));
for i = 1:length(N)
   L2_error(i) = ...
      sqrt(sum(sum((abs(ES(x{i},y{i},t{i}(end))-v{i}(:,:,end)).^2)*(h(i)^2))));
   if i > 1
      order(i) = log2(L2_error(i-1)/L2_error(i));
   end
end

% Tabulate results.
disp(' N     L2-error     order');
disp('---------------------------');
for i = 1:length(N)
   if i > 1
      fprintf('%3d   %.5g   %+2.2f\n',N(i),L2_error(i),order(i));
   else
      fprintf('%3d   %.5g\n',N(i),L2_error(i));
   end
end

% Display numerical approximation.
figure(1); clf; pInd = length(N);
imagesc(x{pInd}(1,:),y{pInd}(:,1),v{pInd}(:,:,end));
set(gca,'LineWidth',2,'FontSize',14,'FontWeight','normal','YDir','normal');
xlabel('$x$','FontName','Times','Interpreter','Latex','FontSize',16);
ylabel('$y$','FontName','Times','Interpreter','Latex','FontSize',16);
%title('Difference Approximation');
axis square; grid on; axis([0 x{pInd}(1,end) 0 y{pInd}(end,1)]);
set(gca,'XTick',0:1:6); set(gca,'YTick',0:1:6);
h = colorbar; set(h,'LineWidth',2,'FontSize',14,'FontWeight','normal');

% Display exact solution.
figure(2); clf;
imagesc(xe(1,:),ye(:,1),fe);
set(gca,'LineWidth',2,'FontSize',14,'FontWeight','normal','YDir','normal');
xlabel('$x$','FontName','Times','Interpreter','Latex','FontSize',16);
ylabel('$y$','FontName','Times','Interpreter','Latex','FontSize',16);
%title('Analytic Solution');
axis([0 2*pi 0 2*pi]); axis square; grid on;
set(gca,'XTick',0:1:6); set(gca,'YTick',0:1:6);
h = colorbar; set(h,'LineWidth',2,'FontSize',14,'FontWeight','normal');
```