Numerical Methods for Partial Differential Equations H.W.5

Name: 廖家緯 / Student ID: 309652008

February 12, 2022

- **Exercise 1**

  Let $\Omega = [0,1] \times [0,1]$ and $u_0(x,y) = x^2 + y^2$. Consider

$$\begin{cases} \Delta u = 4 \text{ on } \Omega, \\ u|_{\partial\Omega} = u_0|_{\partial\Omega} \end{cases}.$$

  Use method Five-point Laplacian and Nine-point Laplacian to compute U_j.

- **Exercise 2**

  Let $\Omega = [0,1] \times [0,1]$ and $u_0(x,y) = e^{x+y}$. Consider

$$\begin{cases} \Delta u = 2e^{x+y} \text{ on } \Omega, \\ u|_{\partial\Omega} = u_0|_{\partial\Omega} \end{cases}.$$

  Use method Five-point Laplacian and Nine-point Laplacian to compute U_j.

- **Code**

  1. **Thomas Algorithm function**

```
function x = Thomas(A, d, xn)
n = size(A,1);
a = [0; diag(A,-1)];
b = diag(A);
c = diag(A,1);
x = zeros(n,1);

for i = 2:n
    b(i) = b(i)-a(i)*c(i-1)/b(i-1);
    d(i) = d(i)-a(i)*d(i-1)/b(i-1);
end
```

```

if nargin ==3
    x(n) = xn;

else
    x(n) = d(n)/b(n);
end

for i = n-1:-1:1
    x(i) = (d(i)-c(i)*x(i+1))/b(i);
end
```

2. Tridiagonal matrix eigenvalue function

```
function Lambda = TridiagED(a, b, c, N)
dx = 1/(N+1);
Lambda = a+2*sqrt(b*c)*cos((1:N)'*pi*dx);
end
```

3. Five-point F function

```
function F = FivePointF(F, U, dx, dy)

F = (circshift(F,-1) + circshift(F,1) + ...
     circshift(F,[0 -1]) + circshift(F,[0 1]) + 8*F)/12;

F = F - ( 4*( circshift(U,1) + circshift(U,-1) + circshift(U, [0 1]) +
    circshift(U, [0 -1]) ) ...
        + 1*( circshift(U, [1 1]) + circshift(U, [1 -1]) + circshift(U,
            [-1 1]) + circshift(U, [-1 -1]) ))/(6*dx*dy);

F = F(2:end-1, 2:end-1);
end
```

4. Five-point Laplacian function

```
function U = FivePointLaplacian(M, N, F, U)
dx = 1/(M+1); dy = 1/(N+1);

F = F(2:end-1, 2:end-1);

F(1,:) = F(1,:) - U(1,2:end-1)/(dx*dy);
F(N,:) = F(N,:) - U(N+2,2:end-1)/(dx*dy);
F(:,1) = F(:,1) - U(2:end-1,1)/(dx*dy);
F(:,M) = F(:,M) - U(2:end-1,M+2)/(dx*dy);

Lambda = TridiagED(-4, 1, 1, M);
b = F*dx*dy;

bbar = zeros(N,M);
for i=1:M
    bbar(i, :) = dst(b(i,:))*sqrt(2/(N+1));
```

```
end

Ubar = zeros(N,M);
for k = 1:M
    A = diag(Lambda(k)*ones(N,1)) + diag(ones(N-1,1), 1) +
        diag(ones(N-1,1), -1);
    Ubar(:,k) = Thomas(A, bbar(:,k));
end

for i=1:M
    U(1+i, 2:end-1) = dst(Ubar(i,:))*sqrt(2/(N+1));
end
```

5. Nine-point Laplacian function

```
function U = NinePointLaplacian(M, N, F, U)
dx = 1/(M+1); dy = 1/(N+1);
Lambda = TridiagED(-20, 4, 4, M);
Gamma = TridiagED(4, 1, 1, M);

Fbar = FivePointF(F, U, dx, dy);
b = Fbar*6*dx*dy;

bbar = zeros(N,M);
for i=1:M
    bbar(i, :) = dst(b(i,:))*sqrt(2/(N+1));
end

Ubar = zeros(N,M);
for k = 1:M
    A = diag(Lambda(k)*ones(N,1)) + diag(Gamma(k)*ones(N-1,1), 1) +
        diag(Gamma(k)*ones(N-1,1), -1);
    Ubar(:,k) = Thomas(A, bbar(:,k));
end

for i=1:M
    U(1+i, 2:end-1) = dst(Ubar(i,:))*sqrt(2/(N+1));
end

end
```

6. PlotLogError

```
function PlotLogError(mList, ErrorList)
loglog(mList, ErrorList, '-ro', 'LineWidth', 1.2);
axis([min(mList), max(mList), ...
    min(ErrorList), max(ErrorList)]);
xlabel('$\log h $', 'interpreter', 'latex');
ylabel('$\log e$', 'interpreter', 'latex');

end
```

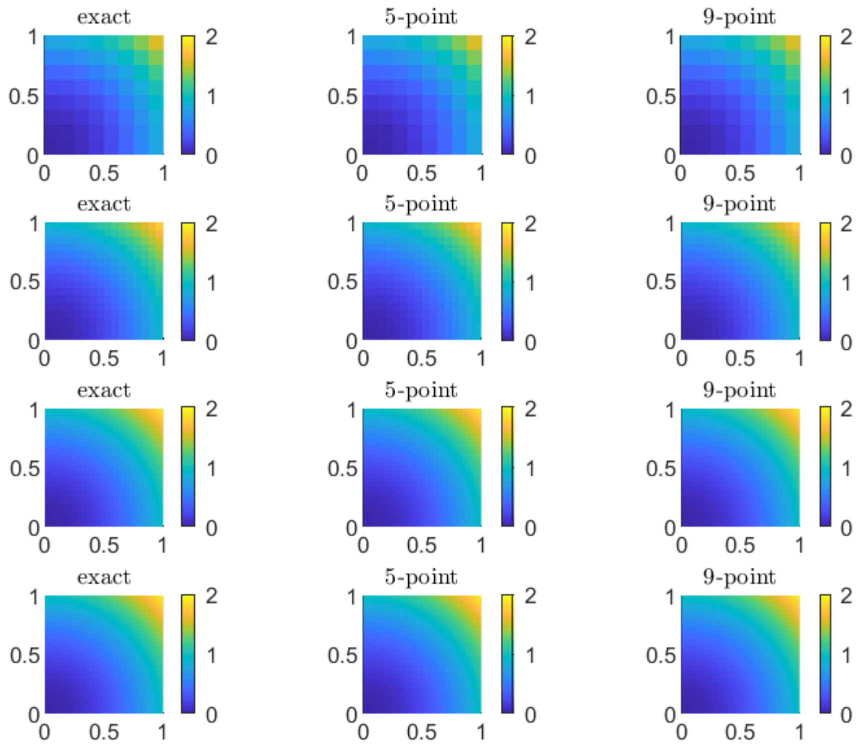
7. Main function

```
%% Exercise 1
clc; clear; close all;
u = @(x,y) x.^2+y.^2;
f = @(x, y) 4 + x.*0 + y.*0;

%%  Exercise 2
clc; clear; close all;
u = @(x, y) exp(1).^(x+y);
f = @(x, y) 2*exp(1).^(x+y);

%% FDM
figure(1)
mList = [7, 15 ,31, 63];
mListLength = length(mList);
ErrorList = zeros(mListLength,2);
RatioList = zeros(mListLength-1,2);

for ii = 1:mListLength
    M = mList(ii); N = mList(ii);
    dx = 1/(M+1); dy = 1/(N+1);
    x0 = 0; xM_1 = 1; x = x0:dx:xM_1;
    y0 = 0; yN_1 = 1; y = y0:dy:yN_1;
    [X, Y] = meshgrid(x,y);
    F = f(X,Y);

    % Exact solution
    Uhat = u(X,Y);

    % Boundary condiction
    U = zeros(N+2, M+2);
    U(1,:) = u(X(1,:), Y(1,:));
    U(N+2,:) = u(X(N+2,:), Y(N+2,:));
    U(:,1) = u(X(:,1), Y(:,1));
    U(:,M+2) = u(X(:,M+2), Y(:,M+2));

    % Five-point Laplacian method
    U1 = FivePointLaplacian(M, N, F, U);

    % Nine-point Laplacian method
    U2 = NinePointLaplacian(M, N, F, U);

    % Compute error
    ErrorList(ii,1) = max(abs(U1-Uhat), [], 'all');
    ErrorList(ii,2) = max(abs(U2-Uhat), [], 'all');
    if ii > 1
        RatioList(ii,:) = log2(ErrorList(ii-1,:)./ErrorList(ii,:));
    end

    subplot(4, 3, 1+3*(ii-1));
    surf(X,Y,Uhat, 'edgecolor', 'none');
    view([0,90]);
    title('Exact solution');
```

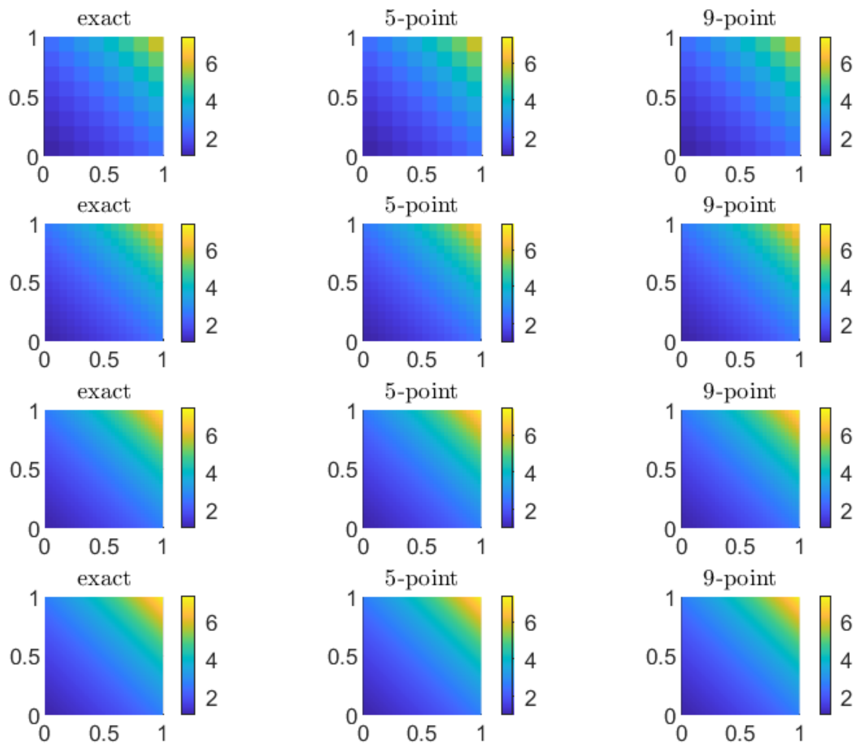
```
    colorbar; axis square;

    subplot(4, 3, 2+3*(ii-1));
    surf(X,Y,U1, 'edgecolor', 'none');
    view([0,90]);
    title('Five-point');
    colorbar; axis square;

    subplot(4, 3, 3+3*(ii-1));
    surf(X,Y,U2, 'edgecolor', 'none');
    view([0,90]);
    title('Nine-point');
    colorbar; axis square;
end

figure(2);
PlotLogError(mList, ErrorList(:,1))
title('Five-point');

figure(3);
PlotLogError(mList, ErrorList(:,2))
title('Nine-point');
```

- **Numerical result**

1. **Exercise 1 :**



2. **Exercise 2 :**



- **Error analysis**

1. **Exercise 1 :**

   (a) Error list

h	$\|U_5 - \hat{U}\|_\infty$	$\|U_9 - \hat{U}\|_\infty$
$\frac{1}{8}$	$2.2204e - 16$	$5.5511e - 16$
$\frac{1}{16}$	$1.5543e - 15$	$1.3323e - 15$
$\frac{1}{32}$	$5.2180e - 15$	$1.4655e - 14$
$\frac{1}{64}$	$1.1102e - 14$	$2.4425e - 14$

2. **Exercise 2 :**

   (a) Error list

h	$\|U_5 - \hat{U}\|_\infty$	$\|U_9 - \hat{U}\|_\infty$
$\frac{1}{8}$	$5.4405e - 04$	$2.8799e - 07$
$\frac{1}{16}$	$1.3995e - 04$	$1.8295e - 08$
$\frac{1}{32}$	$3.5085e - 05$	$1.1433e - 09$
$\frac{1}{64}$	$8.7856e - 06$	$7.1599e - 11$

   (b) Ratio list

$\log_2$ ratio	Five-point	Nine-point
$\log_2 \frac{e(8)}{e(16)}$	1.9588	3.9765
$\log_2 \frac{e(16)}{e(32)}$	1.9960	4.0002
$\log_2 \frac{e(32)}{e(64)}$	1.9977	3.9971

(c) Plot $\log - \log$ error