Numerical Methods for Partial Differential Equations H.W.2

Name: 廖家緯 / Student ID: 309652008

February 12, 2022

H.W. 2-1

- **Exercise**

  Let $\bar{x} = 0.31, m = 15, 31, 63, 127, \quad$ i.e., $h = \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}$

  and let

$$\begin{cases} \dfrac{U_{j+1} - 2U_j + U_{j-1}}{h^2} = \delta_h(x - \bar{x}), & \forall j = 1, 2, \ldots, m, \\ U_0 = 0, \quad U_{m+1} = 0 \end{cases}$$

  where $\delta_h(x) = \begin{cases} \dfrac{1}{h}\left(1 - \dfrac{|x|}{h}\right), & |x| \leq h \\ 0, & |x| \geq h \end{cases}$.

  Compare U_j with $G(x_j, \bar{x})$, and compute $\|\cdot\|_\infty$.

- **Code**

  1. **Thomas Algorithm function**

```
function x = Thomas(A, d)
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

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

```

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

2. Finite Difference function

```
function U = FDM(m, x, f, alpha, beta)
h = (x(end)-x(1))/(m+1);
U = ones(m+2,1);
U(1) = alpha; U(end) = beta;
A = diag(-2*ones(m,1)) + diag(ones(m-1,1), 1) + diag(ones(m-1,1), -1);

F = f(x(2:end-1))';
F(1) = F(1)-alpha/h^2;
F(end) = F(end)-beta/h^2;

U(2:end-1) = Thomas(A, F)*h^2;

end
```

3. Delta function

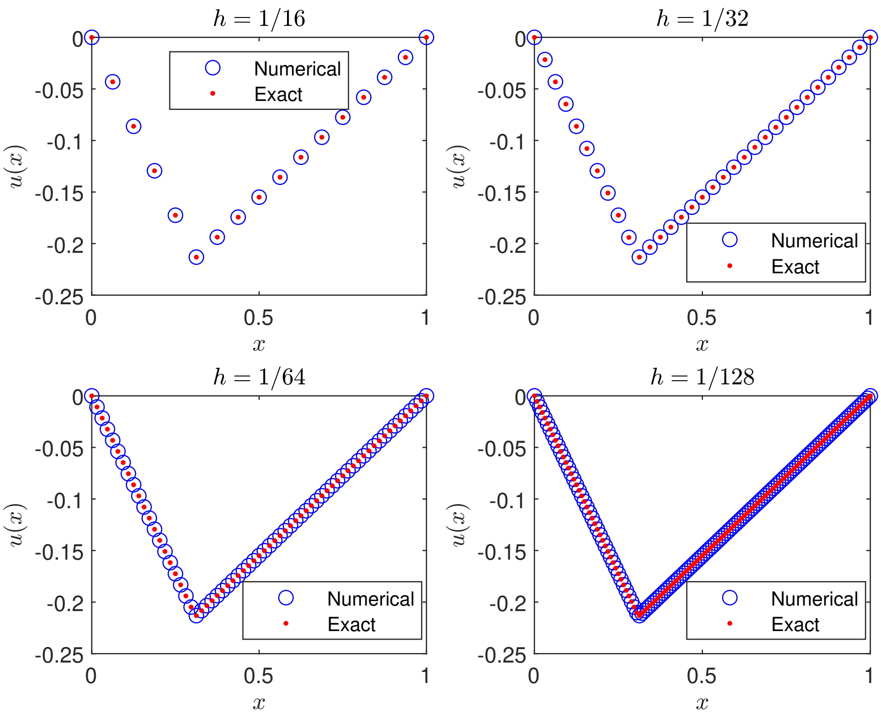
```
function f = delta_eps(eps, c)
f =@(x) (heaviside(x+eps)-heaviside(x)).*(eps+x)/eps^2 ...
    + (heaviside(x)-heaviside(x-eps)).*(eps-x)/eps^2;

if nargin == 2
    f = @(x) f(x-c);
end

end
```

4. Main function

```
clc; clear; close all;

G = @(x, c) (heaviside(x)-heaviside(x-c)).*(c-1).*x + ...
    + (heaviside(x-c)-heaviside(x-1)).*c.*(x-1);
alpha = 0; beta = 0; c = 0.31;

x0 = 0; xm_1 = 1;
mList = [15 ,31, 63, 127];
mListLength = length(mList);
ErrorList = zeros(mListLength,1);
RatioList = zeros(mListLength-1,1);

figure(1);
for i = 1:mListLength
    m = mList(i);
    h = (xm_1-x0)/(m+1); x = x0:h:xm_1;
    f = delta_eps(h, c);

```

```
    U = FDM(m, x, f, alpha, beta); % numerical solution
    U_hat = G(x', c); % exact solution
    ErrorList(i) = max(abs(U-U_hat)); % infinity norm

    % Plot the numerical solution and exact solution
    subplot(2,mListLength/2,i);
    plot(x, U, 'bo', x, U_hat, 'r.');
    title(['$h=1/$', int2str(m+1)], 'interpreter', 'latex');
    xlabel('$x$', 'interpreter', 'latex');
    ylabel('$u(x)$', 'interpreter', 'latex');
    legend('Numerical', 'Exact', 'Location', 'best');

    if i > 1
        RatioList(i) = log2(ErrorList(i-1)/ErrorList(i));
    end

end
hold off;
```

- **Numerical result**



h	Error: $\|U - \hat{U}\|_\infty$	$\log_2$ Ratio
$\frac{1}{16}$	5.5511e-17	
$\frac{1}{32}$	3.6082e-16	-2.7004
$\frac{1}{64}$	1.8319e-15	-2.3440
$\frac{1}{128}$	1.9706e-15	-0.1054

- **Experience**

  error 大概落在 10^{-17} 到 10^{-15} 之間，為機器誤差之範圍，故這裡不畫誤差的圖。會有此現象是因為我們使用的方法為二階收斂，誤差和 $u''(x)$ 有關，而 $u(x)$ 為 piecewise linear，微分兩次後就會消失。

H.W.2-2

Exercise.

Given $A \in \mathbb{R}^{n\times n}$ and A is singular. If $b \in N(A^\top)^\perp$, then $Ax = b$ is solvable.

Proof. First, we prove $N(A^\top) = CS(A)^\perp$.

$$
\begin{aligned}
& x \in N(A^\top) \\
\iff\quad & A^\top x = 0 \\
\iff\quad & \begin{bmatrix} a_1^\top \\ \vdots \\ a_n^\top \end{bmatrix} x = 0 \\
\iff\quad & a_i^\top x = 0,\ i = 1, \dots, n \\
\iff\quad & \langle x, a_i \rangle = 0,\ i = 1, \dots, n \\
\iff\quad & x \in CS(A) \text{ since } \{a_1, \dots, a_n\} \text{ is a basis of } CS(A).
\end{aligned}
$$

Note that $N(A^\top) = CS(A)^\perp \Longrightarrow N(A^\top)^\perp = CS(A)$.

Hence $b \in N(A^\top)^\perp \Longrightarrow b \in CS(A)$. Then there exits $x_1, \dots, x_n \in \mathbb{R}$ such that

$$
\begin{aligned}
b &= x_1 a_1 + \cdots + x_n a_n \\
&= \begin{bmatrix} a_1 & \dots & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \\
&= Ax.
\end{aligned}
$$

Therefore, there exits $x \in \mathbb{R}^n$ such that $Ax = b$. $\square$

H.W.2-3

- **Exercise**

  Given

$$\begin{cases} u'' = \pi^2 \sin \pi x, \\ u'(0) = \pi, \quad u'(1) = -\pi \end{cases}$$

  Use method 1 and method 2 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. **Finite Difference function**

```
function U = FDM(m, x, f, alpha, beta, um_1, method)
h = (x(end)-x(1))/(m+1);
F = f(x)';

% First order convergence
if strcmp(method, 'OneSideDiff')
    A = diag([-h; -2*ones(m,1); h]) ...
    + diag([ones(m,1); -h], -1) ...
    + diag([h; ones(m,1)], 1);
    F(1) = alpha; F(end) = beta;

% Second order convergence
```

```
elseif strcmp(method, 'CenterDiff')
    A = diag([-1; -2*ones(m,1); -1]) ...
    + diag(ones(m+1,1), -1) ...
    + diag(ones(m+1,1), 1);
    F(1) = F(1)/2 + alpha/h;
    F(end) = F(end)/2 - beta/h;

end

Ah = A/h^2;
U = Thomas(Ah, F, um_1);

end
```

3. PlotLogError

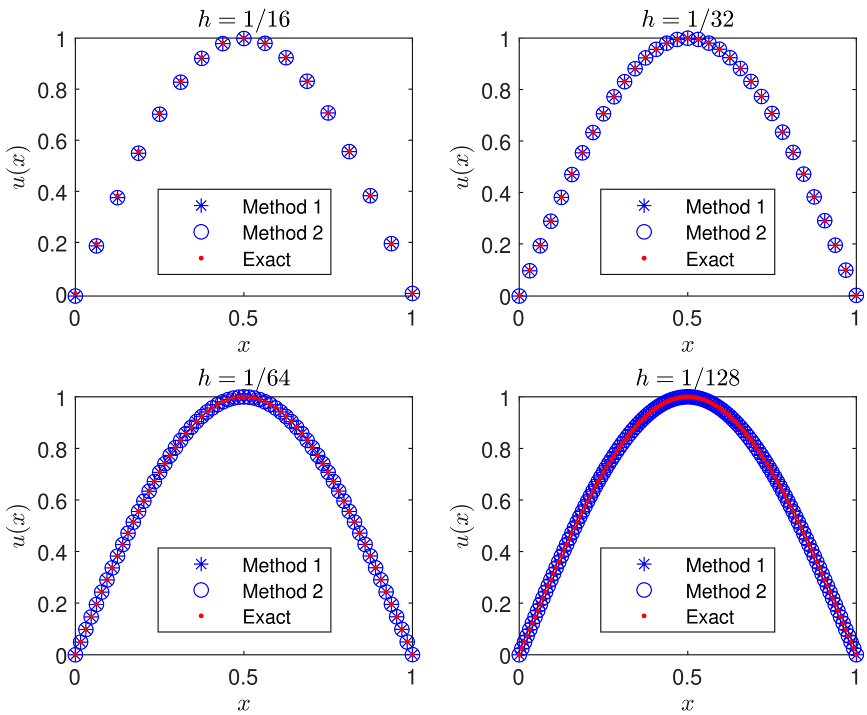
```
function PlotLogError(mList, ErrorList)
loglog(mList, ErrorList, '-ro', 'LineWidth', 1.2);
axis([min(mList), max(mList), ...
    min(ErrorList), max(ErrorList)]);
xlabel('$\log h $', 'interpreter', 'latex');
ylabel('$\log e$', 'interpreter', 'latex');

end
```

4. Main function

```
clc; clear; close all;
u = @(x) sin(pi*x);
f = @(x) -pi^2*sin(pi*x);
x0 = 0;    xm_1 = 1;
alpha = pi; beta = -pi;


%% FDM
mList = [15 ,31, 63, 127];
mListLength = length(mList);
ErrorList = zeros(mListLength,2);
RatioList = zeros(mListLength-1,2);

figure(1);
for i = 1:mListLength
    m = mList(i);
    h = (xm_1-x0)/(m+1); x = x0:h:xm_1;

    % Exact solution
    U_hat = u(x');

    % Numerical solution of method 1
    U1 = FDM(m, x, f, alpha, beta, u(xm_1), 'OneSideDiff');

    % Numerical solution of method 2
    U2 = FDM(m, x, f, alpha, beta, u(xm_1), 'CenterDiff');
```

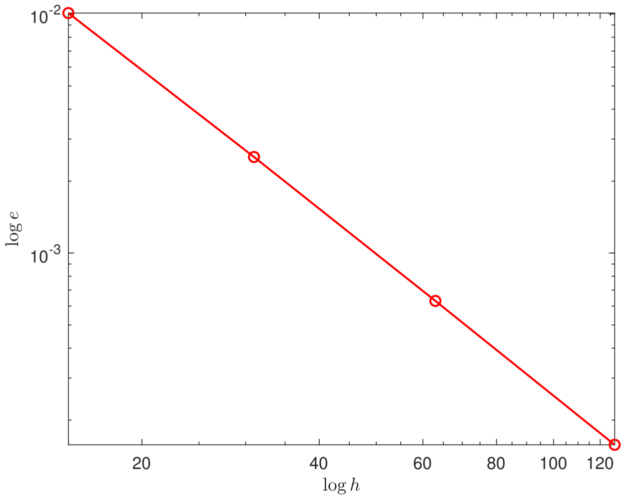
```

    % Compute error with infinity norm
    ErrorList(i,:) = max(abs([U1-U_hat, U2-U_hat]));

    % Plot the numerical solution and exact solution
    subplot(2,mListLength/2,i);
    plot(x, U1, 'b*',x , U2, 'bo', x, U_hat, 'r.');
    title(['$h=1/$', int2str(m+1)], 'interpreter', 'latex');
    xlabel('$x$', 'interpreter', 'latex');
    ylabel('$u(x)$', 'interpreter', 'latex');
    legend('Method 1', 'Method 2', 'Exact', 'Location', 'best');

    if i > 1
        RatioList(i,:) = log2(ErrorList(i-1,:)./ErrorList(i,:));
    end

end
hold off;

% Plot error with loglog
figure(2);
subplot(121); PlotLogError(mList, ErrorList(:,1))
subplot(122); PlotLogError(mList, ErrorList(:,2))
```

- **Numerical result**



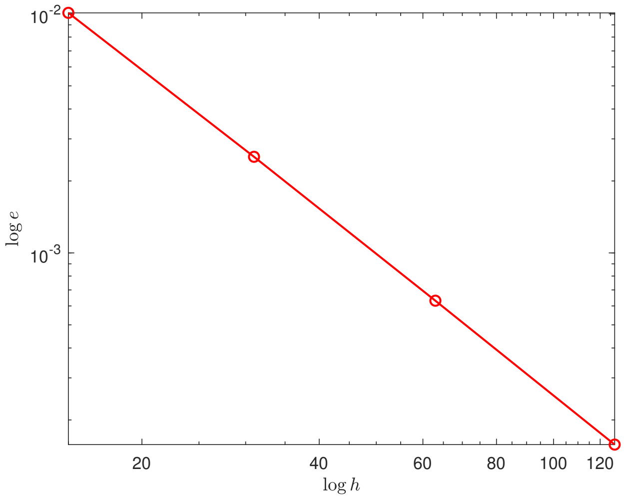
1. **Method 1:**

h	Error: $\|U - \hat{U}\|_\infty$	$\log_2$ Ratio
$\frac{1}{16}$	0.0101	
$\frac{1}{32}$	0.0025	2.0007
$\frac{1}{64}$	6.3085e-04	2.0002
$\frac{1}{128}$	1.5771e-04	2.0000



2. **Method 2:**

h	Error: $\|U - \hat{U}\|_\infty$	$\log_2$ Ratio
$\frac{1}{16}$	0.0101	
$\frac{1}{32}$	0.0025	2.0007
$\frac{1}{64}$	6.3085e-04	2.0002
$\frac{1}{128}$	1.5771e-04	2.0000



- **Experience**

  Method 1 為一階收斂，Method 2 為二階收斂，理論上這兩個方法的收斂速度不一樣，但這個例子中，$\sin x$ 是奇函數，泰勒展開後只會有奇數項，沒有偶數項，故在這個例子中，Method 1 也是二階收斂。