

# Numerical Methods for Partial Differential Equations

## H.W.1

Name: 廖家緯 / Student ID: 309652008

February 12, 2022

- Thomas Algorithm function

```
1 function x = Thomas(A, d)
2 n = size(A,1);
3 a = [0; diag(A,-1)];
4 b = diag(A);
5 c = diag(A,1);
6 x = zeros(n,1);
7
8 for i = 2:n
9     b(i) = b(i)-a(i)*c(i-1)/b(i-1);
10    d(i) = d(i)-a(i)*d(i-1)/b(i-1);
11 end
12
13 x(n) = d(n)/b(n);
14
15 for i = n-1:-1:1
16     x(i) = (d(i)-c(i)*x(i+1))/b(i);
17 end
```

- Finite Difference function

```
1 function U = FiniteDifference(nu, x, f, alpha, beta)
2 h = 1/nu;
3 U = ones(nu+1,1);
4 U(1) = alpha; U(end) = beta;
5 A = diag(-2*ones(nu-1,1)) + diag(ones(nu-2,1), 1) + diag(ones(nu-2,1), -1);
6
7 F = f(x(2:end-1))';
8 F(1) = F(1)-alpha/h^2;
9 F(end) = F(end)-beta/h^2;
10
11 U(2:end-1) = Thomas(A, F)*h^2;
12
13 end
```

- Main function

```

1  clc; clear; close all;
2  format long e
3
4  f1 = @(x) sin(pi*x);
5  u1 = @(x) -sin(pi*x)/pi^2;
6  alpha1 = 0; beta1 = 0;
7
8  f2 = @(x) 0*x+1;
9  u2 = @(x) x.^2/2;
10 alpha2 = 0; beta2 = 1/2;
11
12 f3 = @(x) (heaviside(x-1/2)-heaviside(x-1)).*x.^2 ...
13         + (heaviside(x)-heaviside(x-1/2)).*x/2;
14 u3 = @(x) (heaviside(x-1/2)-heaviside(x-1)).*(x.^4/12-5*x/64-1/192) ...
15         +(heaviside(x)-heaviside(x-1/2)).*(x.^3/12 -19*x/192);
16 alpha3 = 0; beta3 = 0;
17
18 %% Exercise 1
19 close all;
20
21 NList = [8 ,16, 32, 64];
22 ErrorList = zeros(4,1);
23 RatioList = zeros(3,1);
24
25 figure(1);
26 for i = 1:length(NList)
27     nu = NList(i);
28     x0 = 0; xn = 1; dx = (xn-x0)/nu; x = x0:dx:xn;
29
30     U = FiniteDifference(nu, x, f1, alpha1, beta1); % numerical solution
31     U_hat = u1(x'); % exact solution
32     ErrorList(i) = max(abs(U-U_hat)); % infinity norm
33
34     subplot(2,2,i);
35     plot(x, U, 'r*', x, U_hat, 'bo');
36     title(['h=1/', int2str(nu)]);
37     xlabel('x'); ylabel('u(x)');
38     legend('Numerical', 'Exact', 'Location', 'N');
39     hold on;
40
41     if i > 1
42         RatioList(i) = log2(ErrorList(i-1)/ErrorList(i));
43     end
44
45 end
46 hold off;
47
48 figure(2);
49 loglog(NList, ErrorList, '-ro');
50 axis([min(NList), max(NList), ...
51     min(ErrorList), max(ErrorList)]);
52 xlabel('log(h)');

```

```

53 ylabel('log(error)');
54
55 %% Exercise 2
56 close all;
57
58 NList = [8 ,16, 32, 64];
59 ErrorList = zeros(1,4);
60 RatioList = zeros(1,3);
61
62 figure(1);
63 for i = 1:length(NList)
64     nu = NList(i);
65     x0 = 0; xn = 1; dx = (xn-x0)/nu; x = x0:dx:xn;
66
67     U = FiniteDifference(nu, x, f2, alpha2, beta2); % numerical solution
68     U_hat = u2(x'); % exact solution
69     ErrorList(i) = max(abs(U-U_hat)); % infinity norm
70
71     subplot(2,2,i)
72     plot(x, U, 'r*', x, U_hat, 'bo');
73     title(['h=1/', int2str(nu)]);
74     xlabel('x'); ylabel('u(x)');
75     legend('Numerical', 'Exact', 'Location', 'NW');
76     hold on;
77
78     if i > 1
79         RatioList(i) = log2(ErrorList(i-1)/ErrorList(i));
80     end
81
82 end
83
84 figure(2);
85 loglog(NList, ErrorList, '-ro');
86 axis([min(NList), max(NList), ...
87     min(ErrorList), max(ErrorList)]);
88 xlabel('log(h)');
89 ylabel('log(error)');
90
91 %% Exercise 3
92 close all;
93
94 NList = [8 ,16, 32, 64];
95 ErrorList = zeros(1,4);
96 RatioList = zeros(1,3);
97
98 figure(1);
99 for i = 1:length(NList)
100     nu = NList(i);
101     x0 = 0; xn = 1; dx = (xn-x0)/nu; x = x0:dx:xn;
102
103     U = FiniteDifference(nu, x, f3, alpha3, beta3); % numerical solution
104     U_hat = u3(x'); % exact solution
105     ErrorList(i) = max(abs(U-U_hat)); % infinity norm

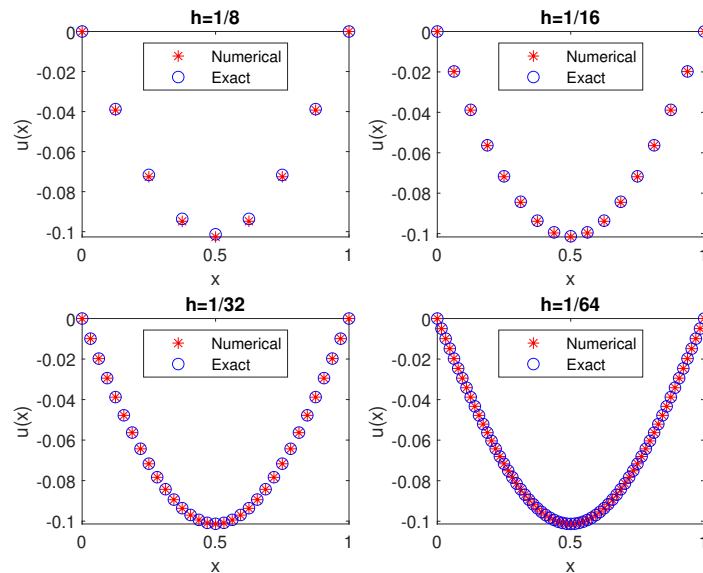
```

```

106
107     subplot(2,2,i)
108     plot(x, U, 'r*', x, U_hat, 'bo');
109     title(['h=1/', int2str(nu)]);
110     xlabel('x'); ylabel('u(x)');
111     legend('Numerical', 'Exact', 'Location', 'N');
112     hold on;
113
114     if i > 1
115         RatioList(i) = log2(ErrorList(i-1)/ErrorList(i));
116     end
117
118 end
119
120 figure(2);
121 loglog(NList, ErrorList, '-ro');
122 axis([min(NList), max(NList), ...
123     min(ErrorList), max(ErrorList)]);
124 xlabel('log(h)');
125 ylabel('log(error)');

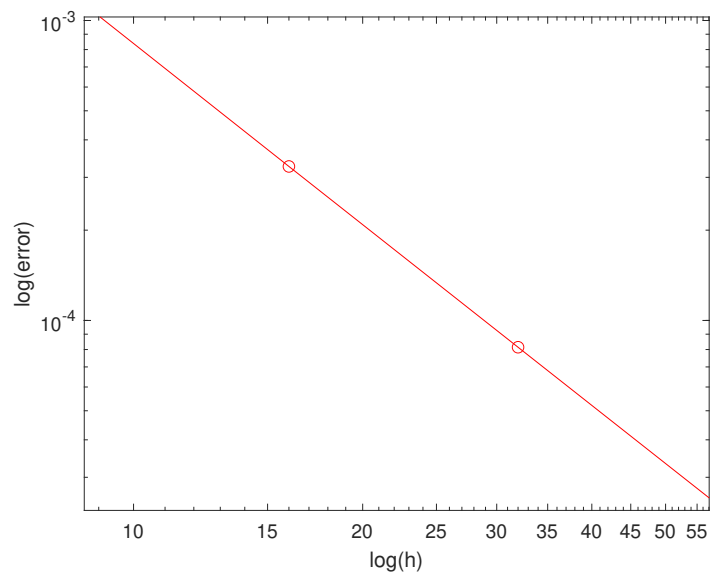
```

- (1)
- PDE:  $u'' = \sin \pi x$  on  $[0, 1]$ .
  - Boundary conditions:  $u(0) = 0, u(1) = 0$ .
  - Exact solution:  $u(x) = -\frac{1}{\pi^2} \sin \pi x$ .
  - Solution:

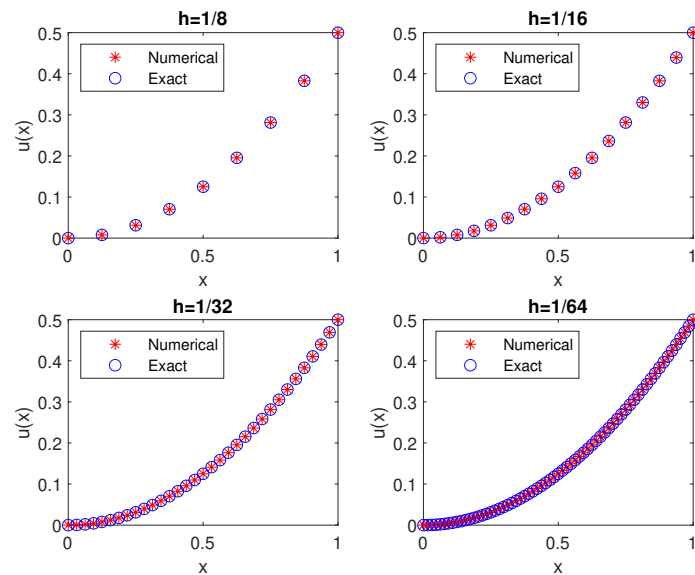


- Error analysis:

$h$	Error: $\ U - \hat{U}\ _\infty$	$\log_2$ Ratio
$\frac{1}{8}$	0.001312184986913	
$\frac{1}{16}$	3.261492871714650e-04	2.008366739526168
$\frac{1}{32}$	8.141944162229353e-05	2.002087242819560
$\frac{1}{64}$	2.034750345958347e-05	2.000521533817863

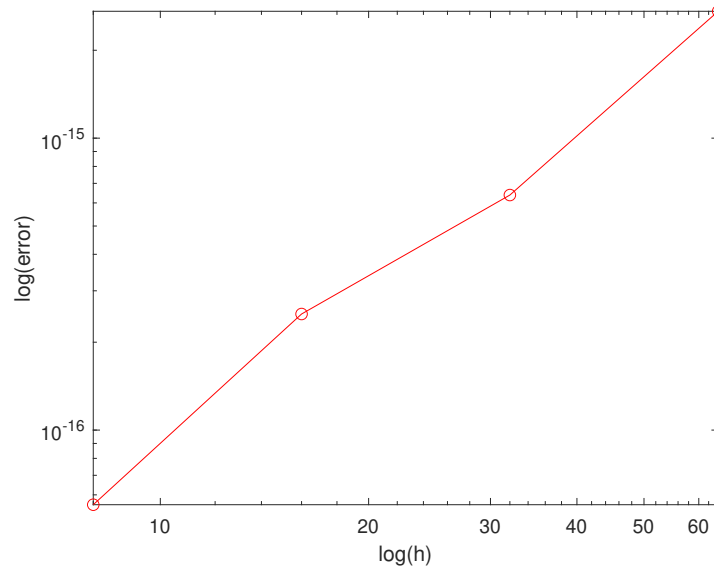


- (2)
- PDE:  $u'' = 1$  on  $[0, 1]$ .
  - Boundary conditions:  $u(0) = 0, u(1) = \frac{1}{2}$ .
  - Exact solution:  $u(x) = \frac{1}{2}x^2$ .
  - Solution:



- Error analysis:

$h$	Error: $\ U - \hat{U}\ _\infty$	$\log_2$ Ratio
$\frac{1}{8}$	5.551115123125783e-17	
$\frac{1}{16}$	2.498001805406602e-16	-2.169925001442313
$\frac{1}{32}$	6.383782391594650e-16	-1.353636954614701
$\frac{1}{64}$	2.720046410331634e-15	-2.091147888058195

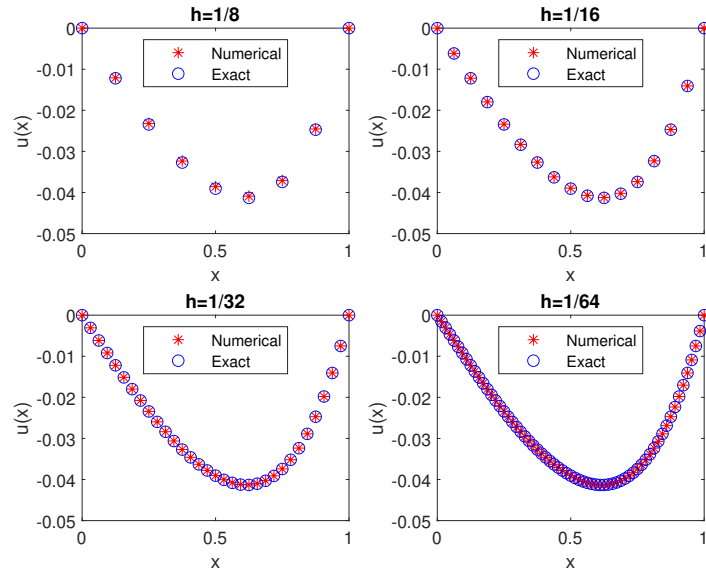


(3) • PDE:  $u'' = \begin{cases} \frac{x}{2}, & x < \frac{1}{2} \\ x^2, & x \geq \frac{1}{2} \end{cases}$  on  $[0, 1]$ .

- Boundary conditions:  $u(0) = 0, u(1) = 0$ .

• Exact solution:  $u(x) = \begin{cases} \frac{x^4}{12} - \frac{5x}{64} - \frac{1}{192}, & \frac{1}{2} \leq x \leq 1 \\ \frac{x^3}{12} - \frac{19x}{192}, & 0 \leq x < \frac{1}{2} \end{cases}$ .

- Solution:



- Error analysis:

$h$	Error: $\ U - \hat{U}\ _{\infty}$	$\log_2$ Ratio
$\frac{1}{8}$	4.882812499999861e-04	
$\frac{1}{16}$	1.220703125000069e-04	1.999999999999877
$\frac{1}{32}$	3.051757812511796e-05	1.9999999999994506
$\frac{1}{64}$	7.629394531590006e-06	1.9999999999941283

