BÀI THỰC HÀNH LẦN 1 FINITE DIFFERENCE METHOD

LƯU GIANG NAM[1,2]

MỤC LỤC

---

1 Mã số sinh viên: 1411174
2 Email: luugiangnam96@gmail.com

DANH SÁCH HÌNH VẼ

DANH SÁCH BẢNG

LỜI NÓI ĐẦU

Bài báo cáo môn học Phương pháp sai phân hữu hạn (Finite Different Methods, FDM) sẽ bao gồm 3 phần, tương ứng với 3 bài tập.

Về chi tiết của từng phần sẽ là:

1. Bài 1 bao gồm 2 ý: Lưới đều và lưới không đều cho biên Dirichlet (cả thuần nhất và không thuần nhất).
2. Bài 2 là sự kết hợp giữ biên Dirichlet và biên Neumann, tất cả đều làm cho lưới đều. Bài này sẽ mở rộng ra thêm để so sánh các phương pháp xấp xỉ đạo hàm ở biên và đưa ra nhận xét về bậc hội tụ và sự hội tụ.
3. Bài toán với điều kiện biên Neumann vẫn sẽ so sánh 2 xấp xỉ trên biên và đưa ra nhận xét về bậc hội tụ và sự hội tụ.

1 BÀI TOÁN VỚI ĐIỀU KIỆN BIÊN DIRICHLET

Xét bài toán đạo hàm riêng sau:

$$\begin{cases} -u_{xx} &= f(x), \quad \forall x \in \ ]0,1[\\ u(0) &= \alpha \\ u(1) &= \beta \end{cases} \tag{1}$$

Bài toán sẽ được giải với 2 dạng lưới là lưới đều và lưới không đều.

Trước hết, ta sẽ có một số đánh giá cho các đạo hàm bậc 1 cả các điểm trong, điểm biên và đạo hàm bậc 2.

1.1 Xấp xỉ đạo hàm

Sử dụng khai triển Taylor ta có 3 kết quả sau cho các đạo hàm của các điểm trong:

$$\frac{\partial u}{\partial x}(x_i) = \frac{u_{i+1} - u_i}{\Delta x} + O(\Delta x) \tag{2}$$

$$\frac{\partial u}{\partial x}(x_i) = \frac{u_i - u_{i-1}}{\Delta x} + O(\Delta x) \tag{3}$$

$$\frac{\partial u}{\partial x}(x_i) = \frac{u_{i+1} - u_{i-1}}{2\Delta x} + O(\Delta^2 x) \tag{4}$$

Công thức (2) được gọi là xấp xỉ forward difference, (3) được gọi là xấp xỉ backward difference, (4) được gọi là xấp xỉ central difference.

Ngoài ra cũng sử dụng khai triển Taylor ta có thể xấp xỉ được đạo hàm tại biên như sau:

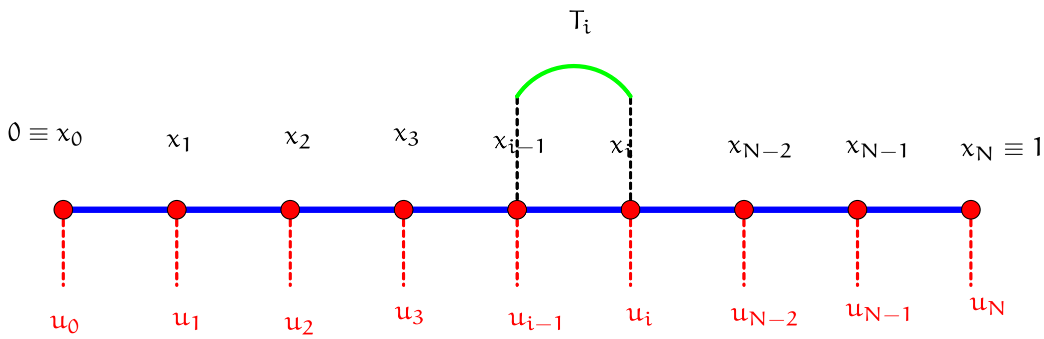
$$\frac{\partial u}{\partial x}(x_0) = \frac{u_1 - u_0}{\Delta x} + O(\Delta x) \tag{5}$$

$$\frac{\partial u}{\partial x}(x_0) = \frac{-3u_0 + 4u_1 - u_2}{2\Delta x} + O(\Delta^2 x) \tag{6}$$

Xấp xỉ đạo hàm cấp 2:

$$\frac{\partial^2 u}{\partial x^2}(x_i) = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} + O(\Delta^2 x) \tag{7}$$

1.2 Lưới đều



Xét một lưới đều có $N+1$ điểm $(x_i)_{i\in\overline{0,N}}$ như hình và bán kính của lưới này là $\Delta x = \dfrac{1}{N}$. Khi đó ta có:

$$x_i = i\Delta x \tag{8}$$

Từ công thức đạo hàm bậc 2 (7) ta có bài toán (1) thành:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} = f_i, \quad \forall i \;=\; 1, 2, ..., N-1 \tag{9}$$

với $f_i = f(x_i)$.

Dưới đây là code cho việc tạo lưới đều và nhập các thông tin ban đầu:

```
%% Initial informations
ax=0.0;
bx=1.0;
cases=1; % We can choose other values: 2,3,...

%% Boundary values
al = exact_solution(0,cases);
be = exact_solution(1,cases);

%% Mesh
N=25;% number of mesh points of first mesh
number_mesh=4;
number_mesh_point=zeros(number_mesh,1);
norm_max=zeros(number_mesh,1);
norm_l2=zeros(number_mesh,1);
norm_maxh1=zeros(number_mesh,1);
norm_h1=zeros(number_mesh,1);
```

1.2.1 *Phân rã bài toán*

Vì bài toán cho điều kiện biên là $u(0) = u_0 = \alpha$ và $u(1) = u_N = \beta$ nên ta có:

$$\begin{cases} i = 1, & \dfrac{2u_1 - u_2}{\Delta^2 x} = f_1 + \dfrac{\alpha}{\Delta^2 x} \\ i = 2, & \dfrac{-u_1 + 2u_2 - u_3}{\Delta^2 x} = f_2 \\ i = 3, & \dfrac{-u_2 + 2u_3 - u_4}{\Delta^2 x} = f_3 \\ \vdots & \vdots \\ i = N-2, & \dfrac{-u_{N-3} + 2u_{N-2} - u_{N-1}}{\Delta^2 x} = f_{N-2} \\ i = N-1, & \dfrac{-u_{N-2} + 2u_{N-1}}{\Delta^2 x} = f_{N-1} + \dfrac{\beta}{\Delta^2 x} \end{cases} \tag{10}$$

Từ đó ta có hệ phương trình $AU = F$ với $A \in \mathbb{R}^N \times \mathbb{R}^N, \quad U, F \in \mathbb{R}^N$, có dạng:

$$A = \frac{1}{\Delta^2 x}\begin{bmatrix} 2 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \tag{11}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \qquad F = \begin{bmatrix} f_1 + \frac{\alpha}{\Delta^2 x} \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{\beta}{\Delta^2 x} \end{bmatrix} \tag{12}$$

Ta có A là ma trận 3 đường chéo và đối xứng xác định dương.

Dưới đây là code Matlab cho việc tạo ma trận A, vector F.

```
 %% Create matrix A
    A=sparse(N-1,N-1);
 %A = zeros(N-1,N-1);
    for i=1:N-1
        if (i==1)
            A(i,i)=2;
            A(i,i+1)=-1;
        elseif(i==N-1)
            A(i,i-1)=-1;
            A(i,i)=2;
        else
            A(i,i-1)=-1;
            A(i,i+1)=-1;
            A(i,i)=2;
        end
    end

    A=A/((del_x)^2);

%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        if (i==1)
            b(i)=functionf(i*del_x,cases) + al/((del_x)^2);
        elseif (i==N-1)
            b(i)=functionf(i*del_x,cases) + be/((del_x)^2);
        else
            b(i)=functionf(i*del_x,cases);
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
    u=A\b;
```

1.2.2 *Mô phỏng kết quả và tính xấp xỉ*

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=al;
        elseif(i==N+1)
            u_dis(i)=be;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

Còn nghiệm chính xác sẽ sử dụng 2 hàm số exact_solution.m và functionf.m và nhập lại vào hàm main như sau:

```
%% Get exact solution
    u_ex=zeros(N+1,1);
    for i=1:N+1
        u_ex(i)=exact_solution((i-1)*del_x,cases);
    end
```

Code sai số và bậc sai số:

```
%% Calculate the error on L^infinity
    norm_max(inumber_mesh)=0.0;
    for i=1:N+1
        if (abs(u_dis(i)-u_ex(i)) > norm_max(inumber_mesh))
            norm_max(inumber_mesh)=abs(u_dis(i)-u_ex(i));
        end
    end

    norm_max(inumber_mesh);
%%  Calculate the error on L^2

    norm_l2(inumber_mesh)=0;
    for i=1:N+1
        norm_l2(inumber_mesh)=norm_l2(inumber_mesh)+(u_dis(i)-u_ex(i))^2*del_x;
    end

    norm_l2(inumber_mesh)=(norm_l2(inumber_mesh))^(1/2);
    norm_l2(inumber_mesh);
%% Calculate the error on maxH1

    norm_maxh1(inumber_mesh)=0;
    for i=1:N
        if (abs(((u_dis(i+1)-u_ex(i+1))-(u_dis(i)-u_ex(i)))/del_x) > ...
            norm_maxh1(inumber_mesh))
            norm_maxh1(inumber_mesh)=...
                abs(((u_dis(i+1)-u_ex(i+1))-(u_dis(i)-u_ex(i)))/del_x);
        end
    end
    norm_maxh1(inumber_mesh);

%% Calculate the error on H1

    norm_h1(inumber_mesh)=0;
    for i=1:N
        norm_h1(inumber_mesh)=norm_h1(inumber_mesh)+...
            (((u_dis(i+1)-u_ex(i+1))-(u_dis(i)-u_ex(i)))/del_x)^2*del_x;
    end
    norm_h1(inumber_mesh)=(norm_h1(inumber_mesh))^(1/2);
```

Cuối cùng là vẽ hình cho từng trường hợp lưới (từ thô đến mịn hơn) và bậc xấp xỉ:

```
%% Figure exact and dicrete solutions
    figure
    plot(x,u_ex,'blue', x,u_dis,'red');
    xlabel('x');ylabel('value');
    title(['Comparison between exact and discrete solutions with N=', ...
        num2str(N)]);

%% Figure for errors respect to number of mesh point
figure
plot(log(number_mesh_point), -log(norm_max),'blue', ...
    log(number_mesh_point), -log(norm_l2), 'red',...
    log(number_mesh_point), -log(norm_maxh1), 'cyan', ...
        log(number_mesh_point), -log(norm_h1), 'magenta',...
    log(number_mesh_point), 2*log(number_mesh_point),'black');
xlabel('Log(MeshPoint)');ylabel('-Log(Error)');
title('Errors');
legend('norm_{max}','norm_{l_2}','norm_{max ...
    h_1}','norm_{h1}','2x','Location','NorthEastOutside');
```

1.2.3 *Ví dụ cụ thể*

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = -90x^8 - 20x^3 \tag{13}$$

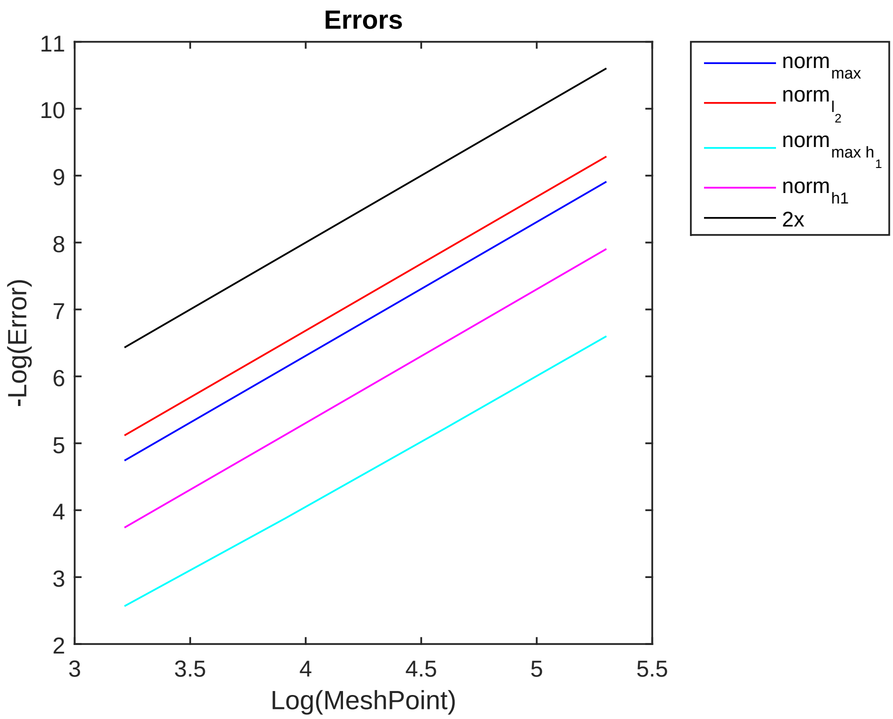
và điều kiện biên

$$u(0) = 1$$
$$u(1) = 2$$

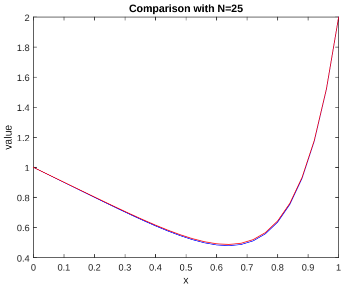
Ta có nghiệm chính xác là:

$$u(x) = x^{10} + x^5 - x + 1 \tag{14}$$

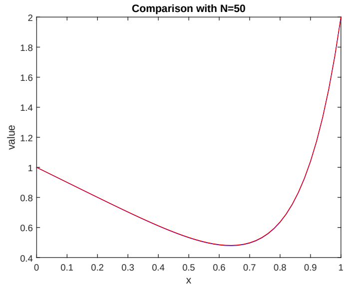
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1:



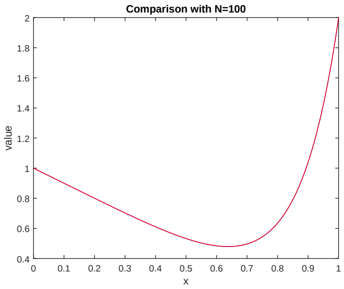
Hình 1: Bậc sai số của Ví dụ 1 - Dirichlet



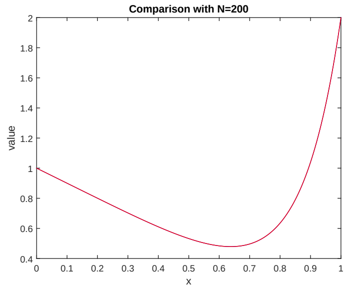
(a) Lần lặp đầu tiên



(b) Lần lặp thứ hai



(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

Hình 2: Ví dụ 1 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Dirichlet, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.008666855556105	0.005951619170475
2	0.002172556376344	0.001491953688419
3	5.437244844449518 e-04	3.732406582446289 e-04
4	1.359539034643253 e-04	9.332591634162775 e-05

Bảng 1: Bảng sai số cho L^∞ và L^2 biên Dirichlet, ví dụ 1

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	0.076362795980955	0.023599252449292
2	0.020663332743009	0.005934458993419
3	0.005371479074601	0.001485788229004
4	0.001369155594899	0.000371583026783

Bảng 2: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichlet, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = \frac{289\pi^2}{4} \sin\left(\frac{17\pi x}{2}\right) \tag{15}$$

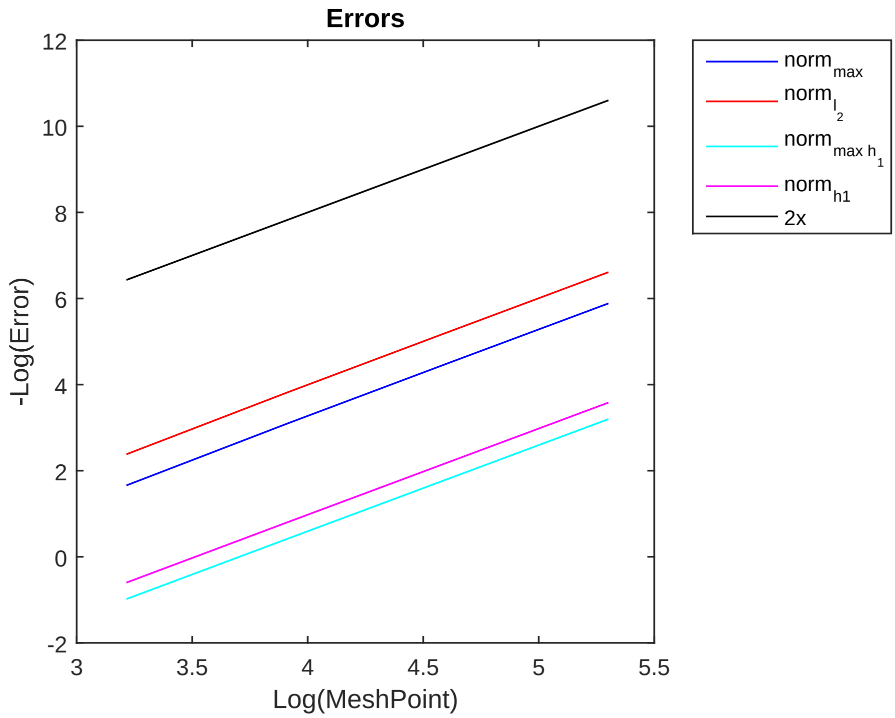
và điều kiện biên

$$u(0) = 1$$
$$u(1) = 2$$

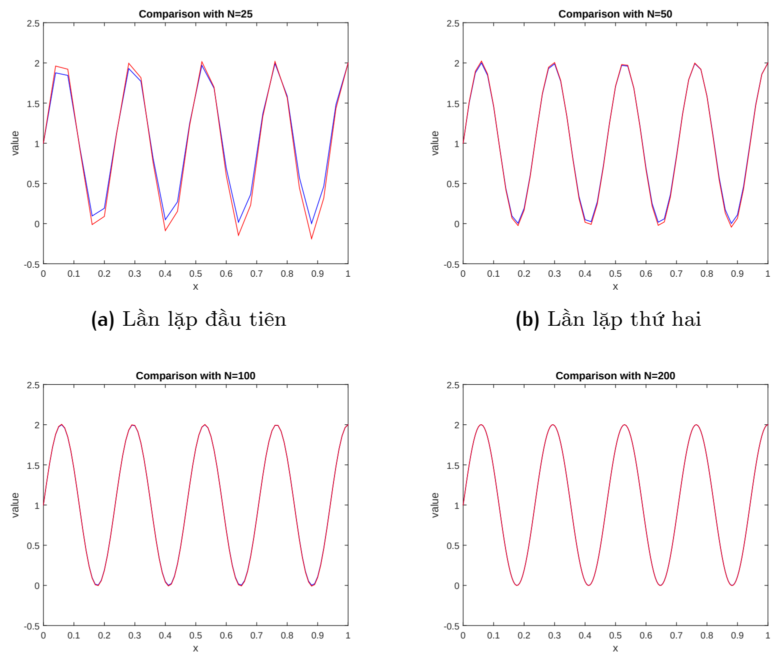
Ta có nghiệm chính xác là:

$$u(x) = \sin\left(\frac{17\pi x}{2}\right) + 1 \tag{16}$$

Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



Hình 3: Bậc sai số của Ví dụ 2 - Dirichlet



(a) Lần lặp đầu tiên

(b) Lần lặp thứ hai

(c) Lần lặp thứ ba

(d) Lần lặp thứ bốn

Hình 4: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Dirichlet, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.189223844158418	0.091822100215051
2	0.045283281951739	0.021974204893786
3	0.011199746014827	0.005434801470444
4	0.002799105055284	0.001355067291398

Bảng 3: Bảng sai số cho L^∞ và L^2 biên Dirichlet, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	2.653843127469559	1.810551045235416
2	0.659660771973064	0.449253574521370
3	0.164616875579493	0.112112887826098
4	0.041162769055303	0.028015857922963

Bảng 4: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichlet, ví dụ 2.

1.2.4 *Nhận xét*

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.

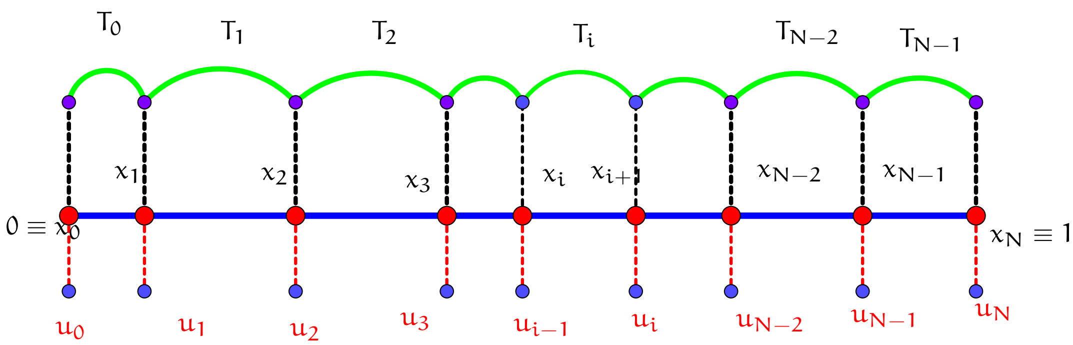
Bậc hội tụ:

- Qua hai hình (1) và (3) ta có thể thấy bậc hội tụ của bài toán biên Dirichlet có bậc hội tụ là bậc 2 (song song với đường 2x).
- Điều này hợp lí vì khi xây dựng, chỉ có 1 lần ta xấp xỉ đó chính là xấp xỉ đạo hàm bậc 2 và xấp xỉ đó có sai số bậc 2 $O(h^2)$.

Sự hội tụ:

- Qua bảng ghi lại kết quả (1), (2), (3) và (4), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (2) và (4) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khia triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

1.3 Lưới bất kì



1.3.1 *Phân rã bài toán*

Xét một lưới đều có $N+1$ điểm $(x_i)_{i\in\overline{0,N}}$ như hình và bán kính của lưới này là $h = \max_{i=\overline{0,N}} |x_i - x_{i-1}|$.

Ta sẽ phân rã bài toán (vì các công thức xấp xỉ đạo hàm trên chỉ sủ dụng được cho lưới đều):

Áp dụng công thứ khai triển Taylor ta có:

$$u(x_{i+1}) = u(x_i) + u'(x_i)(x_{i+1} - x_i) + \frac{1}{2}u''(x_i)(x_{i+1} - x_i)^2 + O(h^3) \tag{17}$$

$$u(x_{i-1}) = u(x_i) + u'(x_i)(x_{i-1} - x_i) + \frac{1}{2}u''(x_i)(x_{i-1} - x_i)^2 + O(h^3) \tag{18}$$

Cộng (17) và (18) ta có:

$$u(x_{i+1}) - 2u(x_i) + u(x_{i-1}) = u'(x_i)(x_{i+1} - 2x_i + x_{i-1}) + \frac{1}{2}u''(x_i)\left((x_{i+1} - x_i)^2 + (x_i - x_{i-1})^2\right) + O(h^3) \tag{19}$$

Mà ta có công thức xấp xỉ đạo hàm bậc 1 dạng trung tâm như sau:

$$u'(x_i) = \frac{u(x_{i+1}) - u(x_{i-1})}{x_{i+1} - x_{i-1}} \tag{20}$$

Thay (20) vào (21) ta có:

$$\begin{aligned} u(x_{i+1}) - 2u(x_i) + u(x_{i-1}) &= \frac{x_{i+1} - 2x_i + x_{i-1}}{x_{i+1} - x_{i-1}}u(x_{i+1}) - \frac{x_{i+1} - 2x_i + x_{i-1}}{x_{i+1} - x_{i-1}}u(x_{i-1}) \\ &+ \frac{1}{2}u''(x_i)\left((x_{i+1} - x_i)^2 + (x_i - x_{i-1})^2\right) + O(h^3) \end{aligned}$$

Khi đó biến đổi lại ta được:

$$\frac{\partial^2 u}{\partial x^2}(x_i) = \alpha_i u_{i+1} + \beta_i u_i + \gamma_i u_{i-1} + O(h) \tag{21}$$

với các giá trị α_i, β_i, γ_i được định nghĩa như sau:

$$\alpha_i = \frac{4(x_i - x_{i-1})}{(x_{i+1} - x_{i-1})\left((x_{i+1} - x_i)^2 + (x_i - x_{i-1})^2\right)} \tag{22}$$

$$\beta_i = \frac{-4}{(x_{i+1} - x_i)^2 + (x_i - x_{i-1})^2} \tag{23}$$

$$\gamma_i = \frac{4(x_{i+1} - x_i)}{(x_{i+1} - x_{i-1})\left((x_{i+1} - x_i)^2 + (x_i - x_{i-1})^2\right)} \tag{24}$$

Khi đó, vì có thêm điều kiện $u(0) = u_0 = 0$ và $u(1) = u_N = 0$ nên ta có:

$$\begin{cases} i = 1: \ -\beta_1 u_1 - \alpha_1 u_2 & = f_1 \\ i = 2: \ -\gamma_2 u_1 - \beta_2 u_2 - \alpha_2 u_3 & = f_2 \\ i = 3: \qquad -\gamma_3 u_2 - \beta_3 u_3 - \alpha_3 u_4 & = f_3 \\ \vdots & \vdots \\ i = N-2: \qquad -\gamma_{N-2} u_{N-3} - \beta_{N-2} u_{N-2} - \alpha_{N-2} u_{N-1} & = f_{N-2} \\ i = N-2: \qquad -\gamma_{N-1} u_{N-2} - \beta_{N-1} u_{N-1} & = f_{N-1} \end{cases} \tag{25}$$

Hệ phương trình trên có thể được viết lại dưới dạng $AU = F$ với $A \in \mathbb{R}^{(N-1)\times(N-1)}$ còn U, F là hai vector thuộc $\mathbb{R}^{N-1}$ có dạng:

$$A = \begin{bmatrix} -\beta_1 & -\alpha_1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\gamma_2 & -\beta_2 & -\alpha_2 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ldots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -\gamma_{N-2} & -\beta_{N-2} & -\alpha_{N-2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & -\gamma_{N-1} & -\beta_{N-1} \end{bmatrix} \tag{26}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix}, \quad F = \begin{bmatrix} f_1 + c.\alpha \\ f_2 \\ \vdots \\ f_{N-2} \\ f_{N-1} + a * \beta \end{bmatrix} \tag{27}$$

với $f_i = f(x_i)$.

Ta có A là ma trận 3 đường chéo và đối xứng xác định dương.

Dưới đây là code cho việc tạo lưới đều và nhập các thông tin ban đầu:

```
ax=0.0; %a=0
bx=1.0; %b=1

al =1; %alpha
be = 2; %beta
%% Create mesh point
    x=zeros(N+1,1);
    for i=1:N+1
        x(i)=1-cos(pi*(i-1)/(2*N));
    end
```

Dưới đây là code cho việc tạo ma trận A, vector F.

```
 %% Create matrix A
    A=sparse(N-1,N-1);
%A = zeros(N-1,N-1);
    for i=1:N-1
        a = ...
            (4*(x(i+1)-x(i)))/((x(i+2)-x(i))*((x(i+1)-x(i))^2+(x(i+2)-x(i+1))^2));
        b = -4/((x(i+1)-x(i))^2+(x(i+2)-x(i+1))^2);
        c = ...
            (4*(x(i+2)-x(i+1)))/((x(i+2)-x(i))*((x(i+1)-x(i))^2+(x(i+2)-x(i+1))^2));
        if (i==1)
            A(i,i)=-b;
            A(i,i+1)=-a;
        elseif(i==N-1)
            A(i,i)=-b;
            A(i,i-1)=-c;
        else
            A(i,i+1)=-a;
            A(i,i)=-b;
            A(i,i-1)=-c;
        end
    end

%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        a = ...
            (4*(x(i+1)-x(i)))/((x(i+2)-x(i))*((x(i+1)-x(i))^2+(x(i+2)-x(i+1))^2));
        c = ...
            (4*(x(i+2)-x(i+1)))/((x(i+2)-x(i))*((x(i+1)-x(i))^2+(x(i+2)-x(i+1))^2));
        if(i==1)
            b(i)=functionf(x(i,1),cases) + c*al;
        elseif(i==N-1)
            b(i)=functionf(x(i,1),cases) + a*be;
        else
            b(i)=functionf(x(i,1),cases);
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
```

```
u=A\b;
```

1.3.2 Mô phỏng kết quả và tính xấp xỉ

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=al;
        elseif(i==N+1)
            u_dis(i)=be;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

Còn nghiệm chính xác sẽ sử dụng 2 hàm số exact_solution.m và functionf.m và nhập lại vào hàm main như sau:

```
%% Get exact solution
    u_ex=zeros(N+1,1);
    for i=1:N+1
        u_ex(i)=exact_solution((i-1)*del_x,cases);
    end
```

1.3.3 Ví dụ cụ thể

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = -90x^8 - 20x^3 \tag{28}$$

và điều kiện biên

$$u(0) = 1$$
$$u(1) = 2$$

Ta có nghiệm chính xác là:

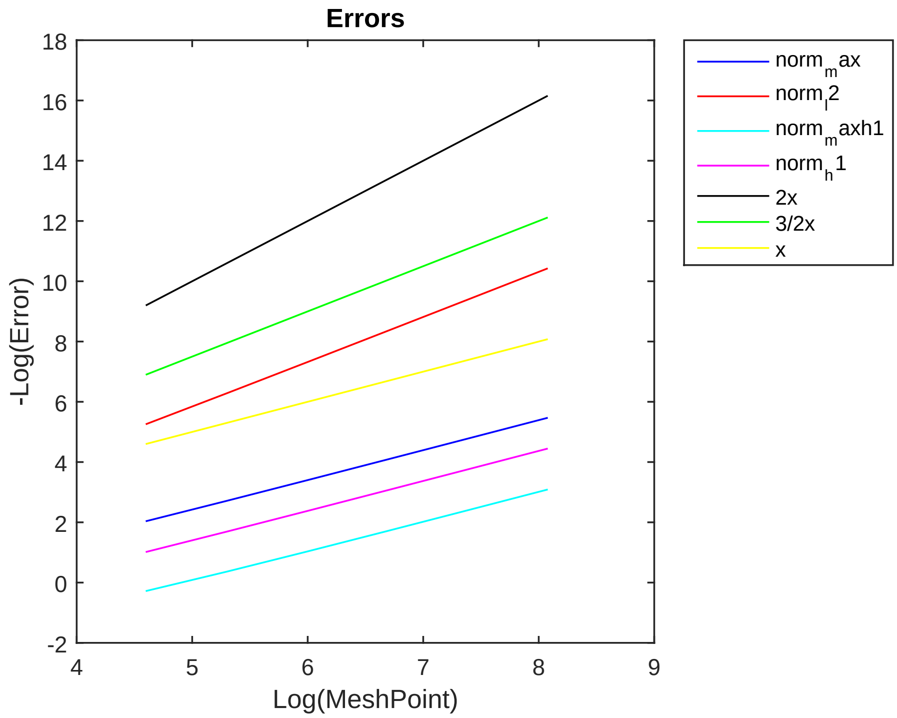
$$u(x) = x^{10} + x^5 - x + 1 \tag{29}$$

Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1:

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Dirichlet, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.497818512856293	0.018835442246434
2	0.232798971102586	0.006456402640924
3	0.112530801076521	0.002250321476273
4	0.055270145115739	7.902155384214215 e-04
5	0.027389080540972	2.784596389693171 e-04
6	0.013632558095971	9.828970447329202 e-05

Bảng 5: Bảng sai số cho L^∞ và L^2 biên Dirichlet, ví dụ 1



Hình 5: Bậc sai số của Ví dụ 1 - Dirichlet

Lần lập	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	10.965932276358538	6.461864053844932
2	5.602288530624390	3.247841466138639
3	2.808038578484399	1.626510686405056
4	1.405269218508132	0.813697922179073
5	0.702887772301379	0.406934009914774
6	0.351499759413090	0.203485069791087

Bảng 6: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichet, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = \frac{289\pi^2}{4} \sin\left(\frac{17\pi x}{2}\right) \tag{30}$$

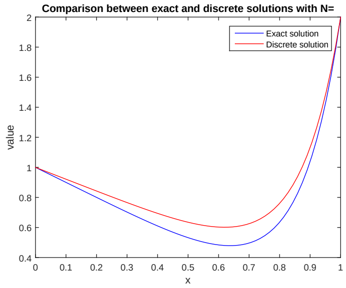
và điều kiện biên

$$u(0) = 1$$
$$u(1) = 2$$

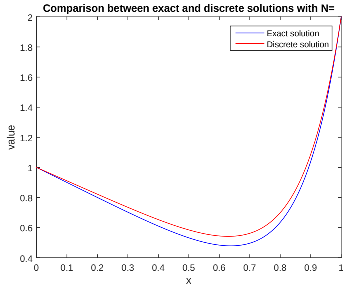
Ta có nghiệm chính xác là:

$$u(x) = \sin\left(\frac{17\pi x}{2}\right) + 1 \tag{31}$$

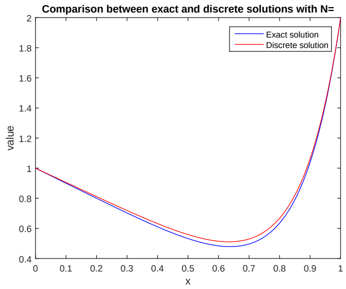
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



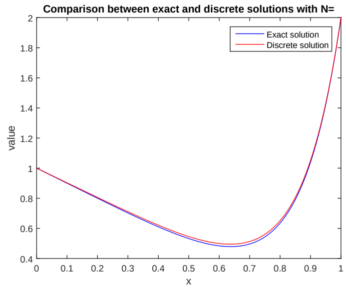
(a) Lần lặp đầu tiên



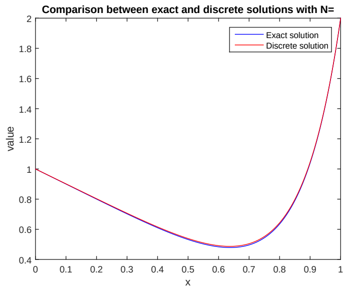
(b) Lần lặp thứ hai



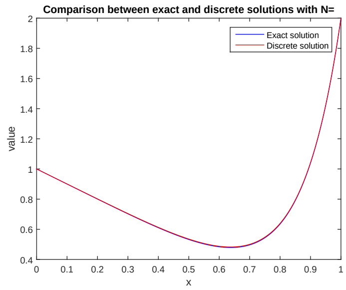
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

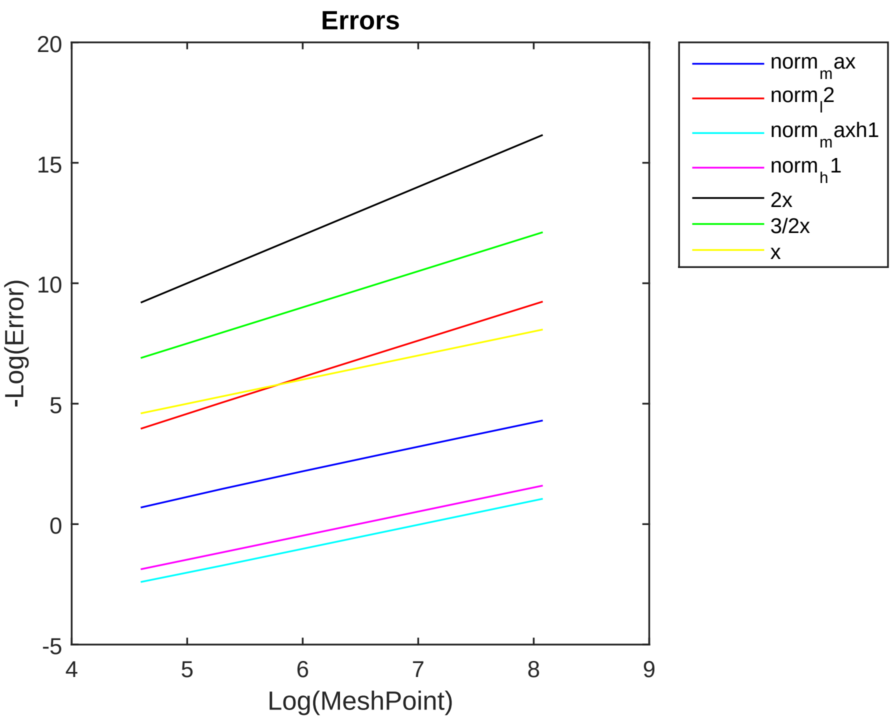


(e) Lần lặp thứ năm

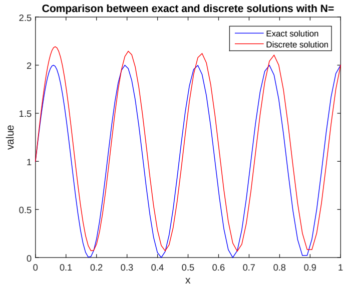


(f) Lần lặp thứ sáu

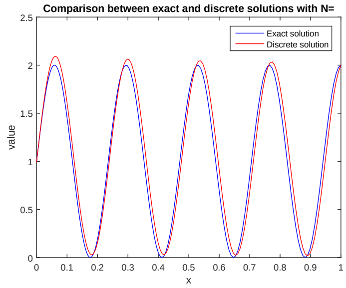
Hình 6: Ví dụ 1 được mô phỏng qua 4 lần chia lưới mịn hơn



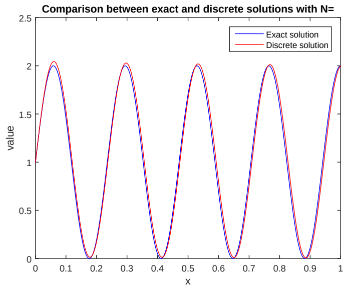
Hình 7: Bậc sai số của Ví dụ 2 - Dirichlet



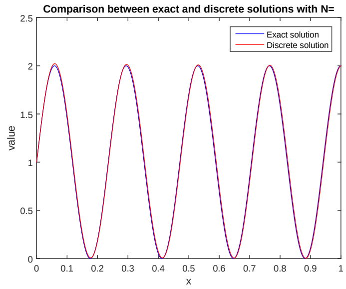
(a) Lần lặp đầu tiên



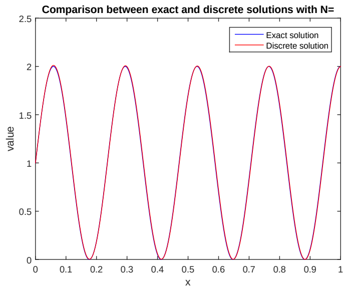
(b) Lần lặp thứ hai



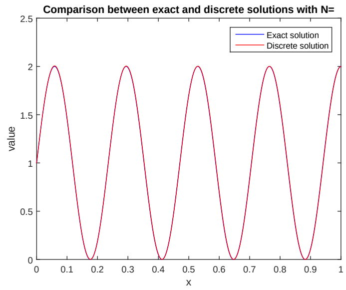
(c) Lần lặp thứ ba



(d) Lần lặp thứ buốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 8: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Dirichlet, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.189223844158418	0.091822100215051
2	0.045283281951739	0.021974204893786
3	0.011199746014827	0.005434801470444
4	0.002799105055284	0.001355067291398

Bảng 7: Bảng sai số cho L^∞ và L^2 biên Dirichet, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	2.653843127469559	1.810551045235416
2	0.659660771973064	0.449253574521370
3	0.164616875579493	0.112112887826098
4	0.041162769055303	0.028015857922963

Bảng 8: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichet, ví dụ 2.

1.3.4 Nhận xét

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.

Bậc hội tụ:

- Qua hai hình (5) và (7) ta có thể thấy bậc hội tụ với chuẩn L^2 của bài toán biên Dirichlet có bậc hội tụ là bậc $\frac{3}{2}$ (song song với đường $\frac{3}{2}x$)
- Trong khi các chuẩn còn lại đều có bậc hội tụ bậc 1 (song song với đường x).
- Điều này hợp lí vì khi xây dựng, có 2 lần ta xấp xỉ đó chính là (20) và (21) ta đều sử dụng xấp xỉ đạo hàm bậc 1, $O(h)$.

Sự hội tụ:

- Qua bảng ghi lại kết quả (5), (6), (7) và (8), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (6) và (8) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 (quá chuận và lệch) vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khia triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

1.4 Nhận xét chung

Bậc hội tụ:

- Đối với lưới đều, do sử dụng thẳng công thức xấp xỉ bậc 2 có xấp xỉ $O(h^2)$ nên bài toán có bậc xấp xỉ là bậc 2 đối với tất cả các chuẩn.
- Đối với lưới không đều, do trong quá trình xây dựng, chỉ sử dụng xấp xỉ đạo hàm có sai số $O(h)$ nên sai số trong L^2 bị tác động chuyển từ bậc 2 xuống bậc $\frac{3}{2}$, còn lại các chuẩn khác đều cho sai số bậc 1.

Sự hội tụ:

- Chuẩn L^2 luôn cho xấp xỉ tốt nhất, trong khi với phương pháp FDM thì các sai số trong các chuẩn Sobolev khá lớn.
- Lưới đều luôn cho kết quả tốt hơn (rất nhiều) so với lưới không đều vì bậc hội tụ của lưới đều là bậc 1 trong khi lưới đều là bậc 2.

2 BÀI TOÁN VỚI ĐIỀU KIỆN BIÊN DIRICHLET – NEUMANN

Xét bài toán đạo hàm riêng sau:

$$\begin{cases} -u_{xx} &= f(x), \quad \forall x \in \ ]0,1[\\ u'(0) &= \alpha \\ u(1) &= \beta \end{cases} \tag{32}$$

Bài toán sẽ được giải với dạng lưới đều.

Trước hết, ta sẽ có một số đánh giá cho các đạo hàm bậc 1 cả các điểm trong, điểm biên và đạo hàm bậc 2.

2.1 Sử dụng xấp xỉ biên (5)

2.1.1 *Phân rã bài toán*

Từ công thức đạo hàm bậc 2 (7) ta có bài toán (32) thành:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} = f_i, \quad \forall i \ = \ 1, 2, ..., N-1 \tag{33}$$

với $f_i = f(x_i)$.

Ngoài ra sử dụng phân rã tại biên (5) ta có:

$$\frac{\partial u}{\partial x}(x_0) = \frac{u_1 - u_0}{\Delta x} + O(\Delta x) \tag{34}$$

Phương trình trên tương đương:

$$u_0 = u_1 - \alpha \Delta x \tag{35}$$

Thay vào (33) với $i = 1$ ta có:

$$\frac{-u_1 + \alpha\Delta x + 2u_1 - u_2}{\Delta^2 x} = f_i$$
$$\frac{u_1 - u_2}{\Delta^2 x} = f_i - \frac{\alpha}{\Delta x}$$

Còn với $i = N-1$ của phương trình (33) ta vẫn có:

$$\frac{2u_{N-2} - u_{N-1}}{\Delta^2 x} = f_{N-1} + \frac{\beta}{\Delta^2 x} \tag{36}$$

Vì bài toán cho điều kiện biên là $u(0) = u_0 = \alpha$ và $u(1) = u_N = \beta$ nên ta có:

$$\begin{cases} i = 1, & \frac{u_1 - u_2}{\Delta^2 x} & = f_1 - \frac{\alpha}{\Delta x} \\ i = 2, & \frac{-u_1 + 2u_2 - u_3}{\Delta^2 x} & = f_2 \\ i = 3, & \frac{-u_2 + 2u_3 - u_4}{\Delta^2 x} & = f_3 \\ \vdots & & \vdots \\ i = N-2, & \frac{-u_{N-3} + 2u_{N-2} - u_{N-1}}{\Delta^2 x} & = f_{N-2} \\ i = N-1, & \frac{-u_{N-2} + 2u_{N-1}}{\Delta^2 x} & = f_{N-1} + \frac{\beta}{\Delta^2 x} \end{cases} \tag{37}$$

Từ đó ta có hệ phương trình $AU = F$ với $A \in \mathbb{R}^N \times \mathbb{R}^N$, $U, F \in \mathbb{R}^N$, có dạng:

$$A = \frac{1}{\Delta^2 x} \begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \tag{38}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \qquad F = \begin{bmatrix} f_1 - \frac{\alpha}{\Delta x} \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{\beta}{\Delta^2 x} \end{bmatrix} \tag{39}$$

2.1.2 Code matlab cho hệ

Dưới đây là code cho việc tạo ma trận A, vector F.

```
 %% Create matrix A   Choose 1st
    A=sparse(N-1,N-1);
    for i=1:N-1
        if (i==1)
            A(i,i)=1;
            A(i,i+1)=-1;
        elseif(i==N-1)
            A(i,i-1)=-1;
            A(i,i)=2;
        else
            A(i,i-1)=-1;
            A(i,i+1)=-1;
            A(i,i)=2;
        end
    end

    A=A/((del_x)^2);
%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        if(i==1)
            b(i)=functionf(i*del_x,cases) - al/(del_x);
        elseif(i==N-1)
            b(i)=functionf(i*del_x,cases) + be/((del_x)^2);
        else
            b(i)=functionf(i*del_x,cases);
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
    u=A\b;
```

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=u(1,1)-al*x(1);
        elseif(i==N+1)
            u_dis(i)=3;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

2.1.3 Ví dụ

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = -12x^2 - 2 \tag{40}$$

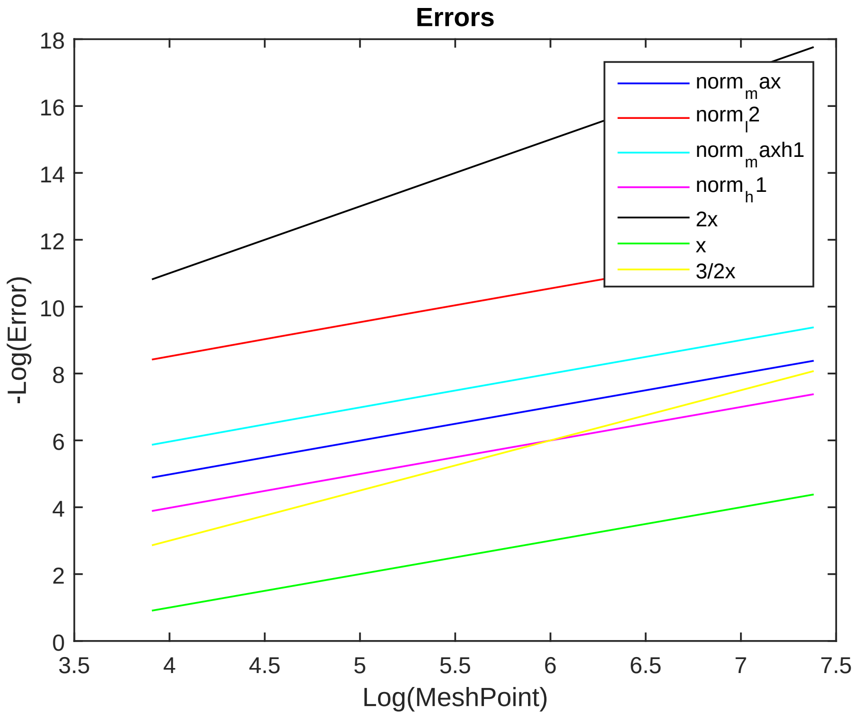
và điều kiện biên

$$u(0)' = 0$$
$$u(1) = 3$$

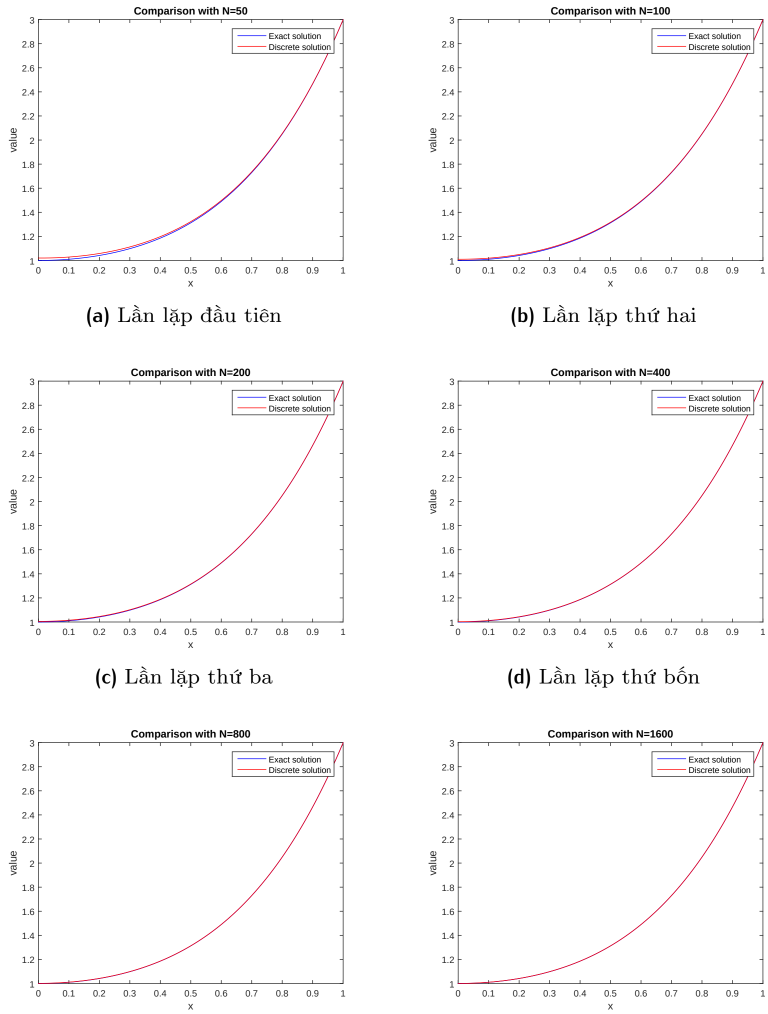
Ta có nghiệm chính xác là:

$$u(x) = x^4 + x^2 + 1 \tag{41}$$

Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1 với



Hình 9: Bậc sai số của Ví dụ 1 - Dirichlet - Neumann bậc 1



(a) Lần lặp đầu tiên

(b) Lần lặp thứ hai

(c) Lần lặp thứ ba

(d) Lần lặp thứ bốn

(e) Lần lặp thứ năm

(f) Lần lặp thứ sáu

Hình 10: Ví dụ 1 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Dirichlet - Neumann, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.020400000000001	0.012011416791223
2	0.010100000000000	0.005889289713822
3	0.005025000000003	0.002915658516244
4	0.002506249999995	0.001450597510686
5	0.001251562499996	7.234926761366144 e-04
6	6.253906249984453 e-04	3.612950506547524 e-04

Bảng 9: Bảng sai số cho L^∞ và L^2 biên Dirichlet - Neumann bậc 1, ví dụ 1

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	0.020792000000003	0.020401306624822
2	0.010199000000011	0.010100164998652
3	0.005049874999941	0.005025020729127
4	0.002512484374861	0.002506252597650
5	0.001253123046752	0.001251562825110
6	6.257810063914349 e-04	0.000625390665663

Bảng 10: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichlet - Neumann bậc 1, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = \frac{81\pi^2}{4} \cos\left(\frac{9\pi x}{2}\right) \tag{42}$$

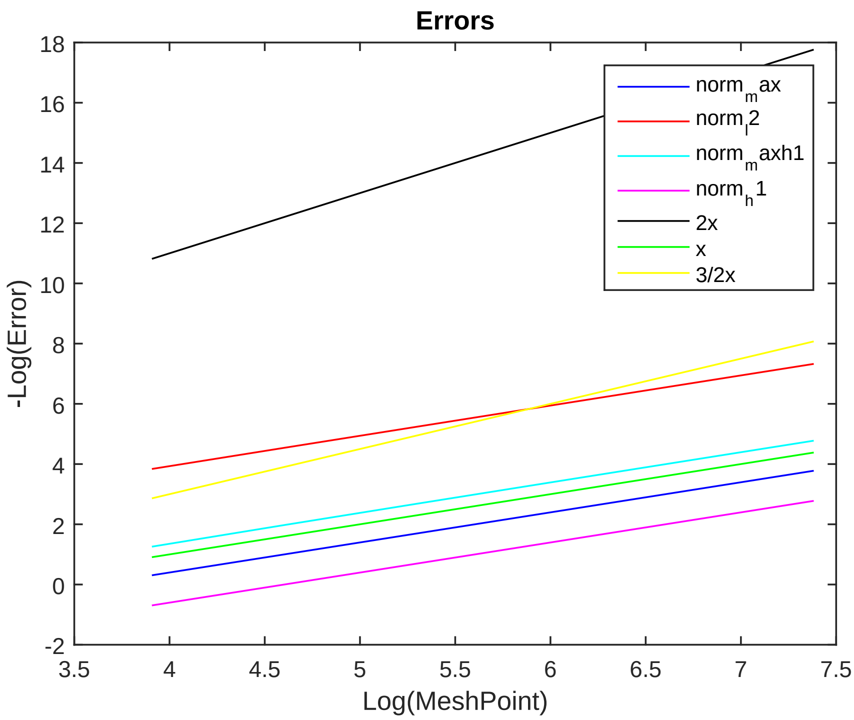
và điều kiện biên

$$u'(0) = 0$$
$$u(1) = 3$$

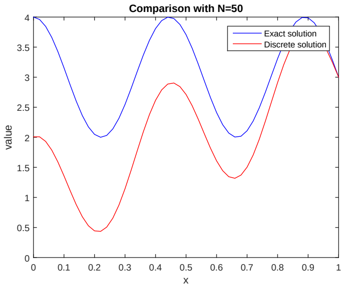
Ta có nghiệm chính xác là:

$$u(x) = \cos\left(\frac{9\pi x}{2}\right) + 3 \tag{43}$$

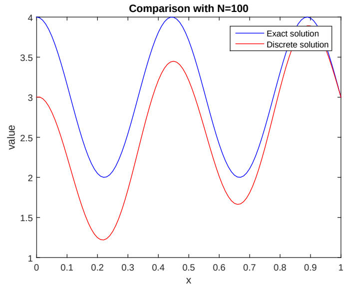
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



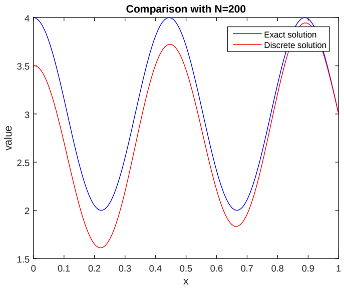
Hình 11: Bậc sai số của Ví dụ 2 - Dirichlet -Neumann bậc 1



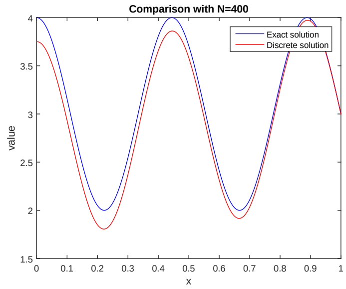
(a) Lần lặp đầu tiên



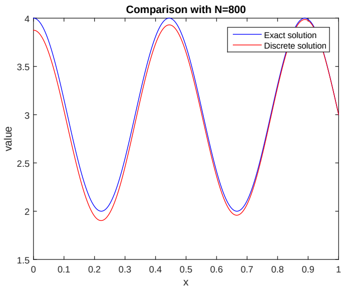
(b) Lần lặp thứ hai



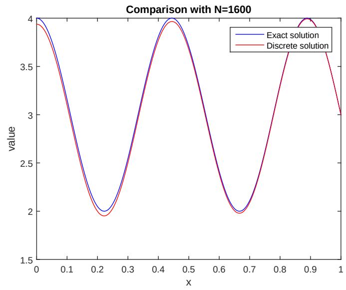
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 12: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Dirichlet - Neumann, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	1.991906194324834	1.171021523570504
2	0.997630284221080	0.581242456079671
3	0.499232244828501	0.289548627739725
4	0.249720261417054	0.144505470249412
5	0.124886156923963	0.072185437213112
6	0.062449584482497	0.036075883229437

Bảng 11: Bảng sai số cho L^∞ và L^2 biên Dirichet - Neumann bậc 1, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	2.092734831238263	1.993009424758493
2	1.022840225827748	0.997767853625641
3	0.505535270408597	0.499249423445966
4	0.251295938150875	0.249722407749141
5	0.125280076802881	0.124886425156450
6	0.062548064581591	0.062449618007968

Bảng 12: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichet - Neumann bậc 1, ví dụ 2.

2.1.4 Nhận xét

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.
Bậc hội tụ:

- Qua hai hình (9) và (11) ta có thể thấy bậc hội tụ của bài toán biên Dirichlet - Neumann nếu sử dụng xấp xỉ biên (5) có bậc hội tụ là bậc $\frac{3}{2}$ (song song với đường $3/_2x$) đối với chuẩn L^2 và các chuẩn còn lại đều là bậc 1.
- Điều này hợp lí vì khi xây dựng xấp xỉ ở biên, ta đã sử dụng xấp xỉ bậc 1 nên chuẩn L^2 đã bị kéo từ bậc 2 xuống bậc $3/_2$ còn các chuẩn còn lại đề là 1.

Sự hội tụ:

- Qua bảng ghi lại kết quả (9), (10), (11) và (12), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (10) và (12) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khia triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

2.2 Sử dụng xấp xỉ biên (6)

2.2.1 Phân rã bài toán

Sử dụng phân rã tại biên (6) ta có:

$$\frac{\partial u}{\partial x}(x_0) = \frac{-3u_0 + 4u_1 - u_2}{2\Delta x} + O(\Delta^2 x) \tag{44}$$

Phương trình (33) tương đương:

$$u_0 = \frac{4}{3}u_1 - \frac{1}{3}u_2 - \frac{2}{3}\alpha\Delta x \tag{45}$$

Thay vào (73) với $i = 1$ ta có:

$$\frac{-\frac{4}{3}u_1 + \frac{1}{3}u_2 + \frac{2}{3}\alpha\Delta x + 2u_1 - u_2}{\Delta^2 x} = f_i$$
$$\frac{\frac{2}{3}u_1 - \frac{2}{3}u_2}{\Delta^2 x} = f_i - \frac{2\alpha}{3\Delta x}$$

Còn với $i = N-1$ của phương trình (73) ta vẫn có:

$$\frac{2u_{N-2} - u_{N-1}}{\Delta^2 x} = f_{N-1} + \frac{\beta}{\Delta^2 x} \tag{46}$$

Vì bài toán cho điều kiện biên là $u(0) = u_0 = \alpha$ và $u(1) = u_N = \beta$ nên ta có:

$$\begin{cases} i = 1, & \dfrac{\frac{2}{3}u_1 - \frac{2}{3}u_2}{\Delta^2 x} & = f_1 - \dfrac{2\alpha}{3\Delta x} \\ i = 2, & \dfrac{-u_1 + 2u_2 - u_3}{\Delta^2 x} & = f_2 \\ i = 3, & \dfrac{-u_2 + 2u_3 - u_4}{\Delta^2 x} & = f_3 \\ \vdots & & \vdots \\ i = N-2, & \dfrac{-u_{N-3} + 2u_{N-2} - u_{N-1}}{\Delta^2 x} & = f_{N-2} \\ i = N-1, & \dfrac{-u_{N-2} + 2u_{N-1}}{\Delta^2 x} & = f_{N-1} + \dfrac{\beta}{\Delta^2 x} \end{cases} \tag{47}$$

Từ đó ta có hệ phương trình $AU = F$ với $A \in \mathbb{R}^N \times \mathbb{R}^N$, $U, F \in \mathbb{R}^N$, có dạng:

$$A = \frac{1}{\Delta^2 x}\begin{bmatrix} \frac{2}{3} & -\frac{2}{3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix} \tag{48}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \qquad F = \begin{bmatrix} f_1 - \frac{2\alpha}{3\Delta x} \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{\beta}{\Delta^2 x} \end{bmatrix} \tag{49}$$

2.2.2 *Code Matlab*

Dưới đây là code cho việc tạo ma trận A, vector F.

```
 %% Create matrix A   Choose 2nd
    A=sparse(N-1,N-1);
    for i=1:N-1
        if (i==1)
            A(i,i)=2/3;
            A(i,i+1)=-2/3;
        elseif(i==N-1)
            A(i,i-1)=-1;
            A(i,i)=2;
        else
            A(i,i-1)=-1;
            A(i,i+1)=-1;
            A(i,i)=2;
        end
    end

    A=A/((del_x)^2);

%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        if(i==1)
            b(i)=functionf(i*del_x,cases) - 2/3*al/(Δ_x);
        elseif(i==N-1)
            b(i)=functionf(i*del_x,cases) + be/((Δ_x)^2);
        else
            b(i)=functionf(i*del_x,cases);
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
    u=A\b;
```

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
```

```
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=al;
        elseif(i==N+1)
            u_dis(i)=be;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

2.2.3 *Ví dụ*

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = -12x^2 - 2 \tag{50}$$

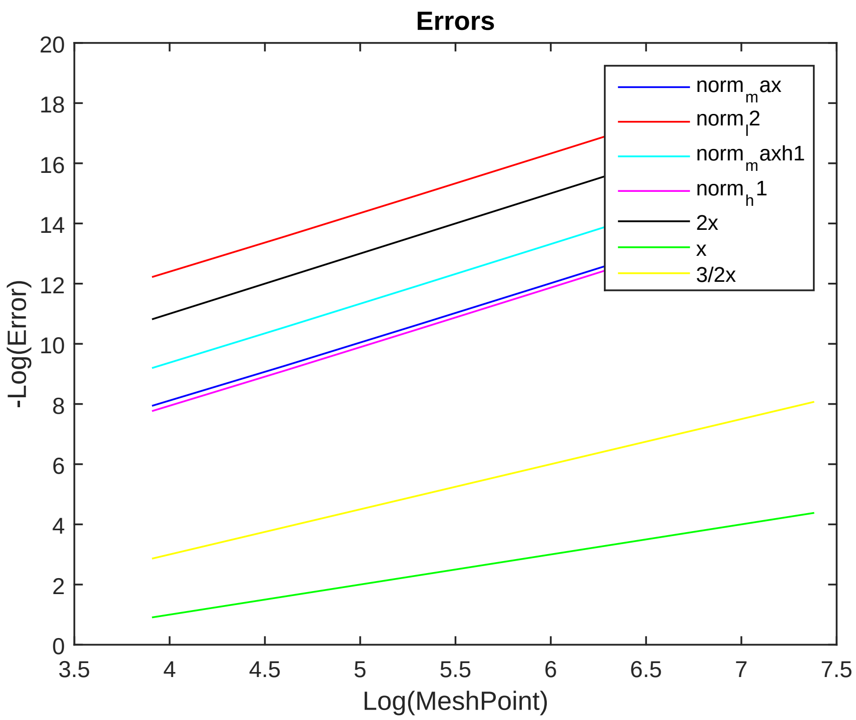
và điều kiện biên

$$u(0)' = 0$$
$$u(1) = 3$$

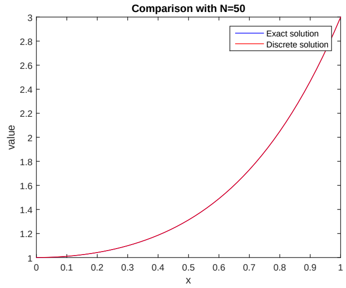
Ta có nghiệm chính xác là:

$$u(x) = x^4 + x^2 + 1 \tag{51}$$

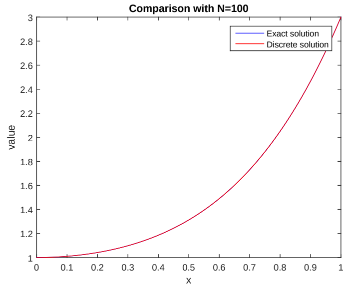
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1 với



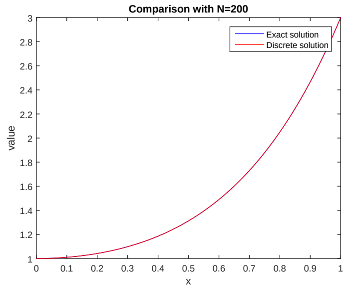
Hình 13: Bậc sai số của Ví dụ 1 - Dirichlet - Neumann bậc 2



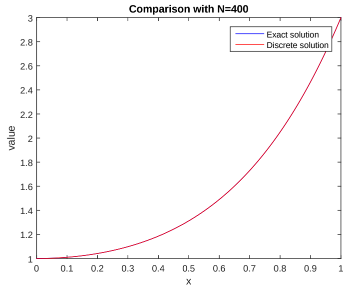
(a) Lần lặp đầu tiên



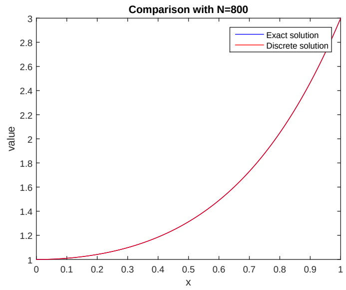
(b) Lần lặp thứ hai



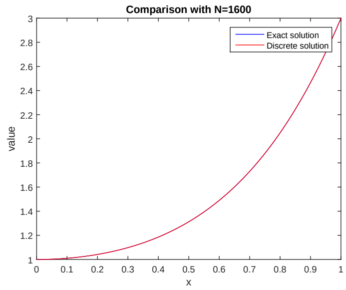
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 14: Ví dụ 1 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Dirichlet - Neumann, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	3.534400000109184 e-04	2.670940772133067 e-04
2	9.408999995041967 e-05	6.992498785753953 e-05
3	2.425562515262492 e-05	1.787081186268484 e-05
4	6.156600833140402 e-06	4.516121375165009 e-06
5	1.550805058370131 e-06	1.135066355444152 e-06
6	3.891603250760767 e-07	2.845183656817497 e-07

Bảng 13: Bảng sai số cho L^∞ và L^2 biên Dirichet - Neumann bậc 2, ví dụ 1

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	7.440000000080715 e-04	0.420970307746742 e-03
2	1.929999999816801 e-04	0.110313190476700 e-03
3	4.912499997900000 e-05	0.028220393895659 e-03
4	1.239062488167519 e-05	0.007135836256186 e-03
5	3.111328084060006 e-06	0.001794081523166 e-03
6	7.795414092015562 e-07	0.000449784955568 e-03

Bảng 14: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichet - Neumann bậc 2, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = \frac{81\pi^2}{4}\cos\left(\frac{9\pi x}{2}\right) \tag{52}$$

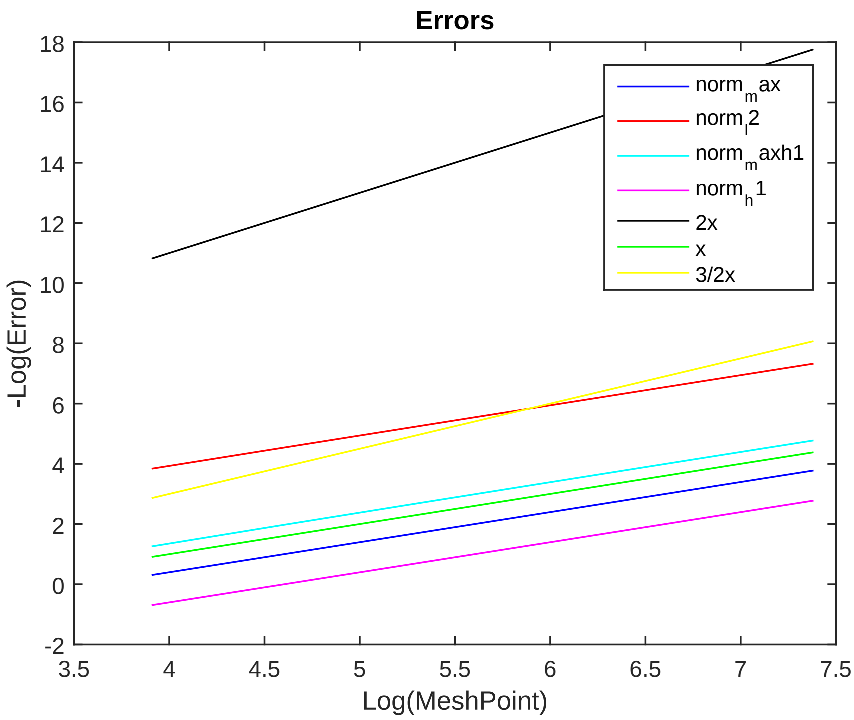
và điều kiện biên

$$u'(0) = 0$$
$$u(1) = 3$$

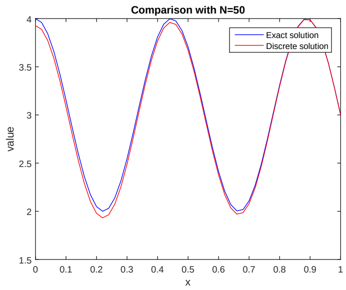
Ta có nghiệm chính xác là:

$$u(x) = \cos\left(\frac{9\pi x}{2}\right) + 3 \tag{53}$$

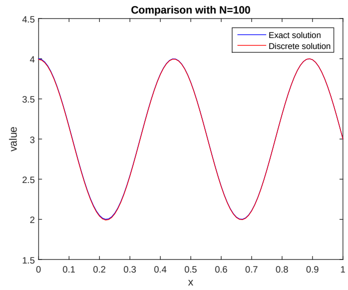
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



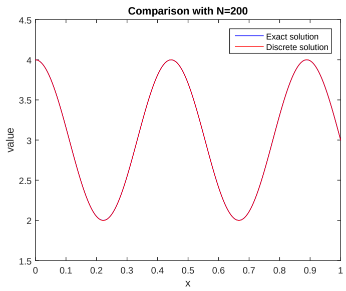
Hình 15: Bậc sai số của Ví dụ 2 - Dirichlet -Neumann bậc 2



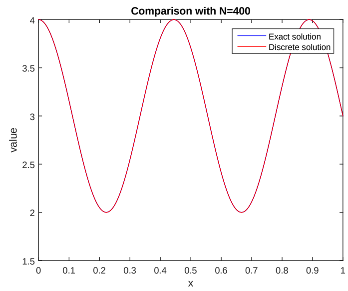
(a) Lần lặp đầu tiên



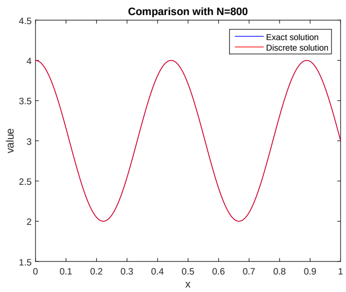
(b) Lần lặp thứ hai



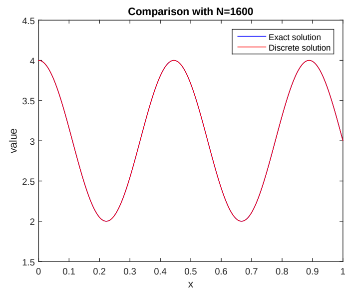
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 16: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Dirichlet - Neumann, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.072668140059511	0.046577008760272
2	0.009572316922677	0.005890552083176
3	0.001396135614606	7.760138054914678 e-04
4	2.260290631737050 e-04	1.156405712401638 e-04
5	4.122939346196120 e-05	2.145597889988141 e-05
6	8.404295870789724 e-06	4.794875291610648 e-06

Bảng 15: Bảng sai số cho L^∞ và L^2 biên Dirichet - Neumann bậc 2, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	0.173496776972959	0.098371432673542
2	0.033512113577938	0.018531912829635
3	0.007134272184572	0.004224157376689
4	0.001627591035813	0.001036660465632
5	3.873994149472537 e-04	0.000258920137169
6	9.441220214512214 e-05	0.000064837123698

Bảng 16: Bảng sai số cho $\max_{H_1}$ và H_1 biên Dirichet - Neumann bậc 2, ví dụ 2.

2.2.4 *Nhận xét*

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.

Bậc hội tụ:

- Qua hai hình (13) và (15) ta có thể thấy bậc hội tụ với tất cả các chuẩn của ví dụ 1 của bài toán biên Dirichlet -Neumann sử dụng xấp xỉ tại biên (6) có bậc hội tụ là bậc 2 (song song với đường $2x$).
- Điều này hợp lí vì khi xây dựng, có 2 lần ta xấp xỉ ta đều sử dụng xấp xỉ đạo hàm bậc 2, $O(h^2)$ (xấp xỉ đạo hàm cấp 2 và đạo hàm tại biên).
- Tuy nhiên đã có sự thay đổi khi ví dụ 2 tất cả các chuẩn đều chuyền về bậc 1, có thể là do ảnh hưởng của hàm lượng giác đến khai triển Taylor.

Sự hội tụ:

- Qua bảng ghi lại kết quả (13), (14), (15) và (16), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (14) và (16) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 (quá chuận và lệch) vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khia triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

2.3 Nhận xét chung

Bậc hội tụ:

- Việc sử dụng công thức xấp xỉ đạo hàm cấp 2 có sai số bậc 2 và xấp xỉ đạo hàm cấp 1 tại biên cũng có sai số bậc 2 sẽ tạo ra bậc sai số là 2 cho tất cả các chuẩn. Khi thay xấp xỉ tại biên xuống chỉ còn bậc 1, lập tức bậc sai số sẽ chuyển về 1 hoặc $3/2$ hoặc nằm giữa đoạn trên.
- Hàm lượng giác cũng tác động đến bậc hội tụ khi các hàm lượng giác đều không được xấp xỉ tốt lắm bằng khai triển Taylor. Bậc hội tụ thường giao động trong đoạn $[1, 3/2]$.

3 BÀI TOÁN VỚI ĐIỀU KIỆN BIÊN NEUMANN

Xét bài toán đạo hàm riêng sau:

$$\begin{cases} -u_{xx} &= f(x), \quad \forall x \in \ ]0,1[\\ u'(0) &= \alpha \\ u'(1) &= \beta \end{cases} \tag{54}$$

Bài toán sẽ được giải với dạng lưới đều.

Trước hết, ta sẽ có một số đánh giá cho các đạo hàm bậc 1 cả các điểm trong, điểm biên và đạo hàm bậc 2.

3.1 Sử dụng xấp xỉ biên (5)

3.1.1 *Phân rã bài toán*

Từ công thức đạo hàm bậc 2 (7) ta có bài toán (??) thành:

$$-\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta^2 x} = f_i, \quad \forall i \ = \ 1, 2, ..., N-1 \tag{55}$$

với $f_i = f(x_i)$.

Ngoài ra sử dụng phân rã tại biên (6) ta có:

$$\frac{\partial u}{\partial x}(x_0) = \frac{u_1 - u_0}{\Delta x} + O(\Delta x) \tag{56}$$

Phương trình trên tương đương:

$$u_0 = u_1 - \alpha \Delta x \tag{57}$$

Thay vào (55) với $i = 1$ ta có:

$$\frac{-u_2 + 2u_1 - u_1 + \alpha \Delta x}{\Delta^2 x} = f_1$$
$$\Leftrightarrow \frac{u_1 - u_2}{\Delta^2 x} = f_i - \frac{\alpha}{\Delta x}$$

Tương tự với biên bên phải: (6) ta có:

$$\frac{\partial u}{\partial x}(x_N) = \frac{u_N - u_{N-1}}{2\Delta x} + O(\Delta^2 x) \tag{58}$$

Phương trình trên tương đương:

$$u_N = u_{N-1} + \beta \Delta x \tag{59}$$

Thay vào (55) với $i = N - 1$ ta có:

$$\frac{-u_{N-1} + 2u_{N-1} - u_{N-2} - \beta \Delta x}{\Delta^2 x} = f_{N-1} \tag{60}$$
$$\Leftrightarrow \frac{u_{N-1} - u_{N-2}}{\Delta^2 x} = f_{N-1} + \frac{\beta}{\Delta x} \tag{61}$$

Tuy nhiên nếu giữ nguyên ma trận A trên thì sẽ không giải được vì A là một ma trận suy biến ($\det(A) = 0$). Trong tài liệu [2] trang 32 và 33 có giải thích cho điều này (Sự tồn tại và duy nhất của nghiệm). Nghiệm trên chỉ tồn tại với $\alpha = \beta = 0$ khi và chỉ khi $\int_0^1 f(0) = 0$. Nhưng đây sẽ là một lớp hàm, nên cần điều kiện $\int_0^1 u(x) = c$ (chọn $c = 0$ để tiện cho việc giải).

Phân rã điều kiện $\int_0^1 u(x) = 0$ ra với lưới đều ta có:

$$\sum_{i=0}^{N} = u_i = 0 \tag{62}$$

Thay

$$u_0 = u_1 - \alpha\Delta x$$
$$u_N = u_{N-1} + \beta\Delta x$$

vào (80) ta có:

$$2u_1 + u_2 + ... + u_{N-2} + 2u_{N-1} + \beta\Delta x = 0 \tag{63}$$

Tiếp tục từ (61) thay:

$$u_{N-1} = u_{N-2} + \beta\Delta x + f_{N-1}\Delta^2 x \tag{64}$$

vào (??) ta có:

$$2u_1 + u_2 + ... + u_{N-3} + 3u_{N-2} + 3\beta\Delta x + 2\Delta^2 x f_{N-1} = 0 \tag{65}$$

$$\Leftrightarrow f_{N-1} = \frac{-u_1 - \frac{1}{2}u_2 - ... - \frac{1}{2}u_{N-3} - \frac{3}{2}u_{N-2}}{\Delta^2 x} + \frac{\alpha}{2\Delta x} - \frac{3\beta}{2\Delta x} \tag{66}$$

Khi đó thu được kết quả:
Từ đó ta có hệ phương trình $AU = F$ với $A \in \mathbb{R}^N \times \mathbb{R}^N$, $U, F \in \mathbb{R}^N$, có dạng:

$$A = \frac{1}{\Delta^2 x}\begin{bmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ -1 & \frac{-1}{2} & \frac{-1}{2} & \frac{-1}{2} & \cdots & \frac{-1}{2} & \frac{-3}{2} & 0 \end{bmatrix} \tag{67}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \qquad F = \begin{bmatrix} f_1 - \frac{\alpha}{\Delta x} \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{3\beta}{2\Delta x} - \frac{\alpha}{2\Delta x} \end{bmatrix} \tag{68}$$

Dưới đây là code cho việc tạo ma trận A, vector F.

```
 %% Create matrix A
    A=sparse(N-1,N-1);
%A = zeros(N-1,N-1);
    for i=1:N-1
        if (i==1)
            A(i,i)=1;
            A(i,i+1)=-1;
        elseif(i==N-1)
            A(i,1) = -1;
            A(i,N-2) = -3/2;
            A(i,N-1)=0;
            for j=2:N-3
                A(i,j)= -1/2;
            end
```

```
        else
            A(i,i-1)=-1;
            A(i,i+1)=-1;
            A(i,i)=2;
        end
    end

    A=A/((del_x)^2);

%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        if (i==1)
            b(i)=functionf(i*del_x,cases) -al/del_x;
        elseif (i==N-1)
            b(i)=functionf(i*del_x,cases) + (3*be)/(2*del_x) - ...
                (al)/(2*del_x);
        else
            b(i)=functionf(i*del_x,cases);
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
    u=A\b;
```

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=4/3*u(1,1)-1/3*u(2,1) -2/3*al*del_x;
        elseif(i==N+1)
            u_dis(i)=4/3*u(N-1,1)-1/3*u(N-2,1) +2/3*be*del_x;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

3.1.2 *Ví dụ cụ thể với xấp xỉ* (5)

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = 3 - 6x \tag{69}$$

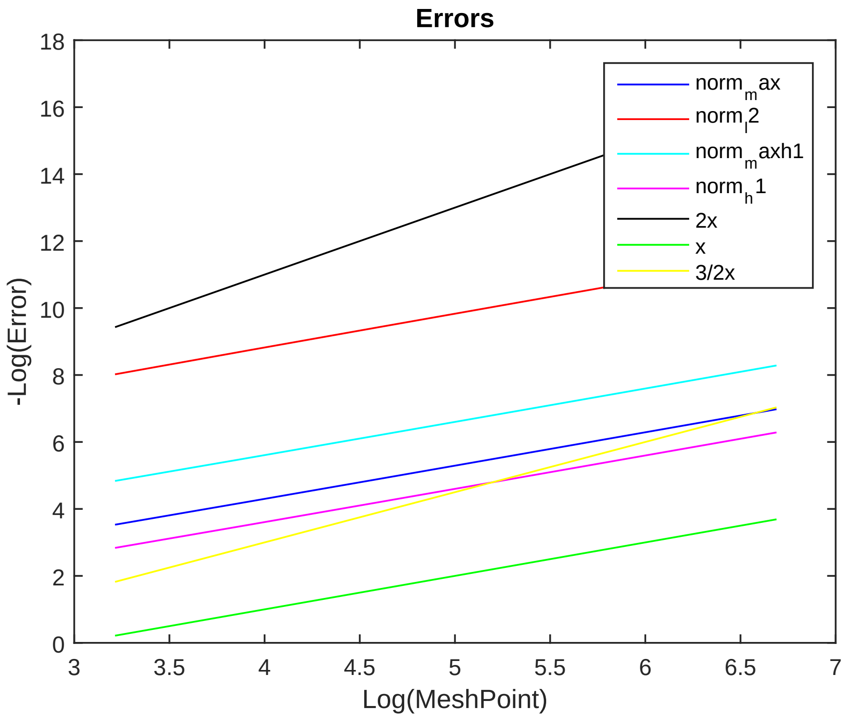
và điều kiện biên

$$u(0)' = 0$$
$$u'(1) = 0$$

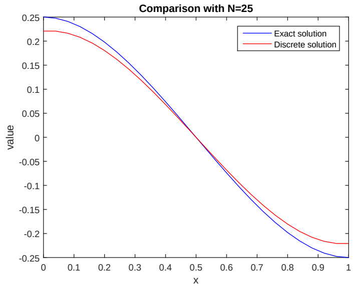
Ta có nghiệm chính xác là:

$$u(x) = x^4 \frac{3}{2} x^2 + \frac{1}{4} \tag{70}$$

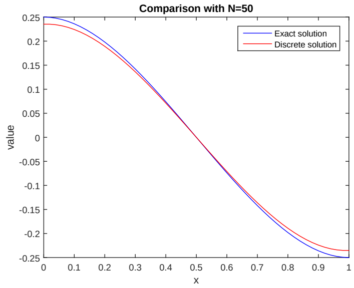
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1:



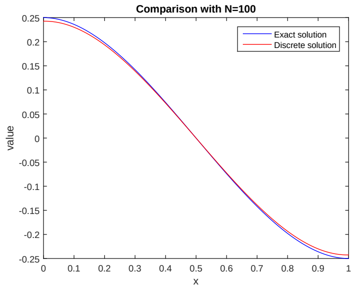
Hình 17: Bậc sai số của Ví dụ 1 - Neumann, bậc 1



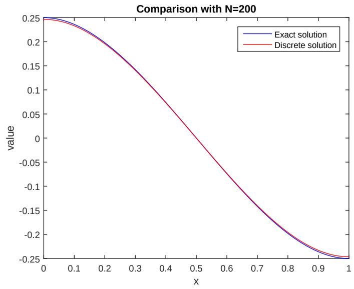
(a) Lần lặp đầu tiên



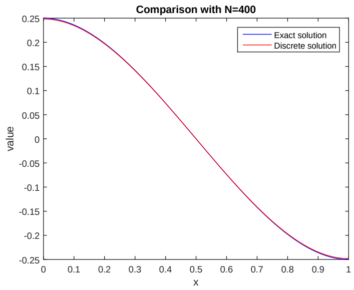
(b) Lần lặp thứ hai



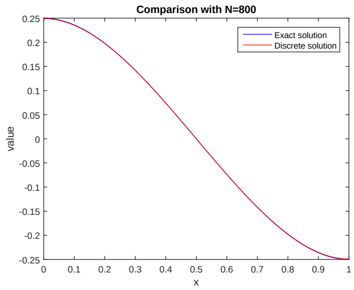
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 18: Ví dụ 1 được mô phỏng qua 6 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Neumann, bậc 2, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.029200000000000	0.017866964375629
2	0.014800000000002	0.008800712698414 e-04
3	0.007450000000000	0.004365725426547
4	0.003737500000000	0.002174023787747
5	0.001871874999996	0.001084779422230
6	9.367187500148311 e-04	5.418287448390509 e-04

Bảng 17: Bảng sai số cho L^∞ và L^2 biên Neumann bậc 1, ví dụ 1

Lần lặp	Chuẩn max_{H_1}	Chuẩn H^1
1	0.058400000000006	0.058400000000001
2	0.029600000000016	0.029600000000004
3 0.014900000000020	0.014899999999999	
4	0.007475000000062	0.007475000000000
5	0.003743750000085	0.003743749999992
6	0.001873437500310	0.001873437500029

Bảng 18: Bảng sai số cho max_{H_1} và H_1 biên Neumann bậc 1, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = 60x - 270x^8 \tag{71}$$

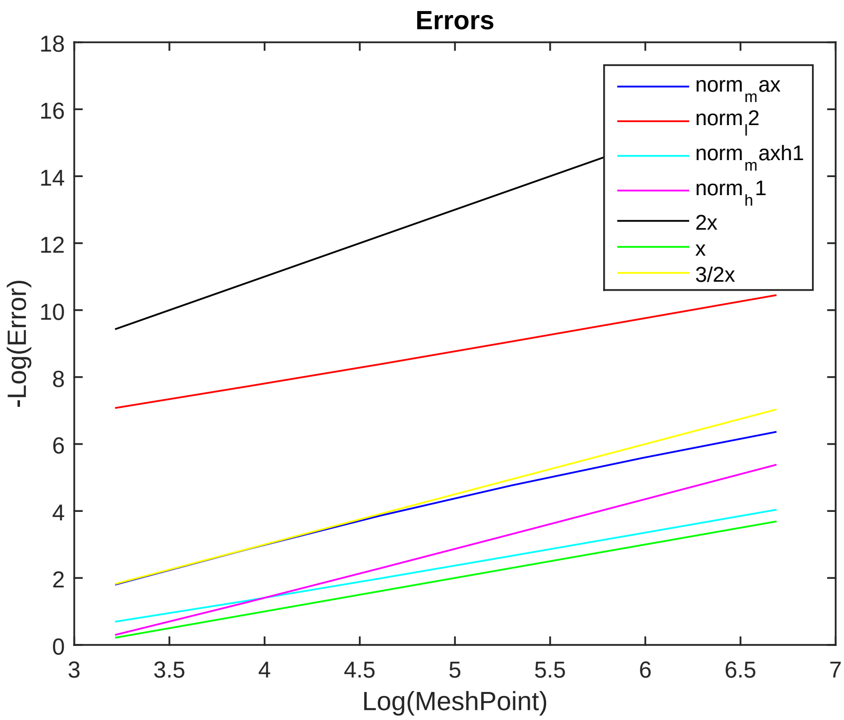
và điều kiện biên

$$u'(0) = 0$$
$$u'(1) = 0$$

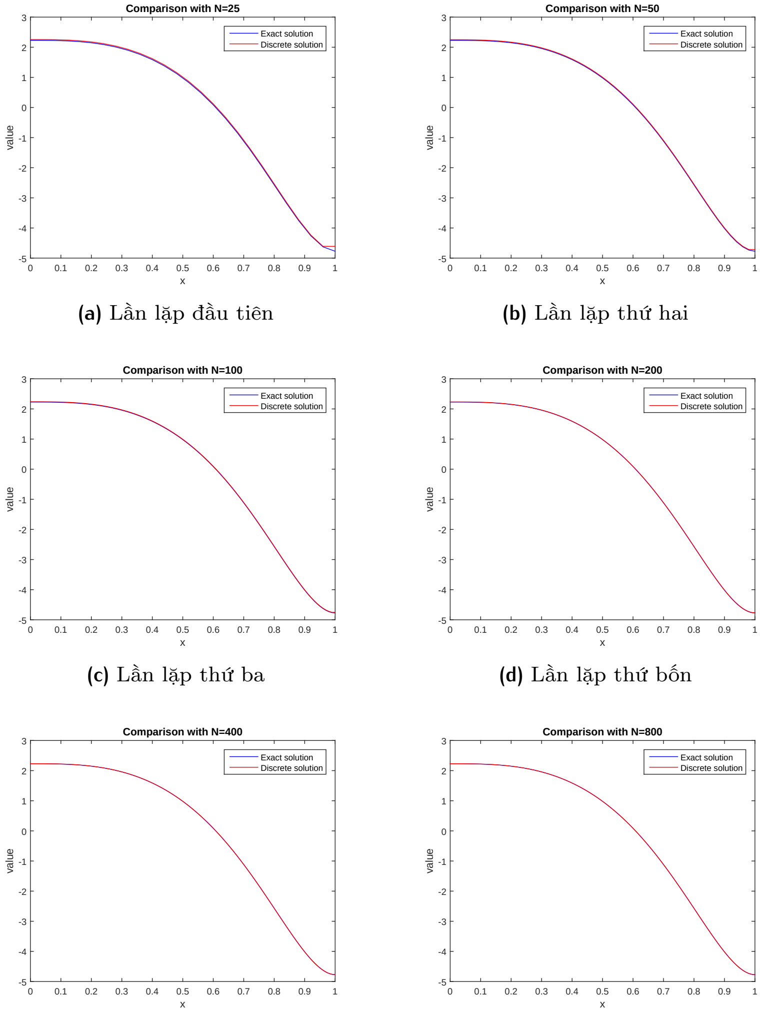
Ta có nghiệm chính xác là:

$$u(x) = 3x^{10} - 10x^3 + \frac{49}{22} \tag{72}$$

Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



Hình 19: Bậc sai số của Ví dụ 2 -Neumann, bậc 1



(a) Lần lặp đầu tiên

(b) Lần lặp thứ hai

(c) Lần lặp thứ ba

(d) Lần lặp thứ bốn

(e) Lần lặp thứ năm

(f) Lần lặp thứ sáu

Hình 20: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Neumann, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.164105444104037	0.045947398937399
2	0.057209947535710	0.024120382003431
3	0.021124116949764	0.012493920837821
4	0.008521766766443	0.006324338728074
5	0.003728821612026	0.003174779658401
6	0.001728600469645	0.001589523421004

Bảng 19: Bảng sai số cho L^∞ và L^2 biên Neumann bậc 1, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	3.678447699362586	0.737296444629932
2	1.964921033132150	0.278257120735407
3	1.015622502641378	0.101638027661988
4	0.516328279462996	0.036524216466249
5	0.260322314279904	0.013018724833048
6	0.130704353624367	0.004621565379800

Bảng 20: Bảng sai số cho $\max_{H_1}$ và H_1 biên Neumann b, ví dụ 2.

3.1.3 Nhận xét

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.
Bậc hội tụ:

- Qua hai hình (17) và (19) ta có thể thấy bậc hội tụ của bài toán biên Neumann nếu sử dụng xấp xỉ biên (5) có bậc hội tụ là bậc 1 (song song với đường x) đối với tất cả các chuẩn.
- Điều này hợp lí vì khi xây dựng xấp xỉ ở biên, ta đã sử dụng xấp xỉ bậc 1 nên chuẩn L^2 đã bị kéo xuống bậc 1.
- Tuy nhiên ví dụ 2 có sự hỗi tạp giữa các bậc hội tụ, chỉ có thể kết luận là bậc hội tụ nằm trong khoảng 1 đến $3/2$, vì câu này sử dụng hàm lượng giác.

Sự hội tụ:

- Qua bảng ghi lại kết quả (17), (18), (19) và (20), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (18) và (20) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khia triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

3.2 Sử dụng xấp xỉ biên (6)

3.2.1 Phân rã bài toán

Từ công thức đạo hàm bậc 2 (7) ta có:

$$-\frac{u_{i+1}-2u_i+u_{i-1}}{\Delta^2 x} = f_i, \quad \forall i = 1, 2, ..., N-1 \tag{73}$$

với $f_i = f(x_i)$.
Ngoài ra sử dụng phân rã tại biên (6) ta có:

$$\frac{\partial u}{\partial x}(x_0) = \frac{-3u_0 + 4u_1 - u_2}{2\Delta x} + O(\Delta^2 x) \tag{74}$$

Phương trình trên tương đương:

$$u_0 = \frac{4}{3}u_1 - \frac{1}{3}u_2 - \frac{2}{3}\alpha\Delta x \tag{75}$$

Thay vào (73) với $i = 1$ ta có:

$$\frac{-\frac{4}{3}u_1 + \frac{1}{3}u_2 + \frac{2}{3}\alpha\Delta x + 2u_1 - u_2}{\Delta^2 x} = f_1$$

$$\frac{\frac{2}{3}u_1 - \frac{2}{3}u_2}{\Delta^2 x} = f_i - \frac{2\alpha}{3\Delta x}$$

Tương tự với biên bên phải: (6) ta có:

$$\frac{\partial u}{\partial x}(x_N) = \frac{3u_N - 4u_{N-1} + u_{N-2}}{2\Delta x} + O(\Delta^2 x) \tag{76}$$

Phương trình trên tương đương:

$$u_N = \frac{4}{3}u_{N-1} - \frac{1}{3}u_{N-2} + \frac{2}{3}\beta\Delta x \tag{77}$$

Thay vào (73) với $i = N-1$ ta có:

$$\frac{-\frac{4}{3}u_{N-1} + \frac{1}{3}u_{N-2} - \frac{2}{3}\beta\Delta x + 2u_{N-1} - u_{N-2}}{\Delta^2 x} = f_{N-1} \tag{78}$$

$$\Leftrightarrow \frac{\frac{2}{3}u_{N-1} - \frac{2}{3}u_{N-2}}{\Delta^2 x} = f_{N-1} + \frac{2\beta}{3\Delta x} \tag{79}$$

Tuy nhiên nếu giữ nguyên ma trận A trên thì sẽ không giải được vì A là một ma trận suy biến ($\det(A) = 0$). Trong tài liệu [2] trang 32 và 33 có giải thích cho điều này (Sự tồn tại và duy nhất của nghiệm). Nghiệm trên chỉ tồn tại với $\alpha = \beta = 0$ khi và chỉ khi $\int_0^1 f(0) = 0$. Nhưng đây sẽ là một lớp hàm, nên cần điều kiện $\int_0^1 u(x) = c$ (chọn $c = 0$ để tiện cho việc giải).

Phân rã điều kiện $\int_0^1 u(x) = 0$ ra với lưới đều ta có:

$$\sum_{i=0}^{N} = u_i = 0 \tag{80}$$

Thay

$$u_0 = \frac{4}{3}u_1 - \frac{1}{3}u_2 - \frac{2}{3}\alpha\Delta x$$
$$u_N = \frac{4}{3}u_{N-1} - \frac{1}{3}u_{N-2} + \frac{2}{3}b\Delta x$$

vào (80) ta có:

$$\frac{7}{3}u_1 + \frac{2}{3}u_2 + u_3 + ... + u_{N-3} + \frac{2}{3}u_{N-2} + \frac{7}{3}u_{N-1} = \frac{2}{3}\alpha\Delta x - \frac{2}{3}\beta\Delta x \tag{81}$$

Tiếp tục từ (79) thay:

$$u_{N-1} = u_{N-2} + \frac{2}{3}\Delta^2 x f_N + b\Delta x \tag{82}$$

vào (81) ta có:

$$\frac{7}{3}u_1 + \frac{2}{3}u_2 + u_3 + ... + u_{N-3} + 3u_{N-2} = \frac{14}{9}\Delta^2 x f_N - \frac{7}{3}\beta\Delta x - \frac{2}{3}\Delta x(b-a) \tag{83}$$

$$\Leftrightarrow \frac{-3}{2}u_1 - \frac{3}{7}u_2 - \frac{9}{14}u_3 - ... - \frac{9}{14}u_{N-3} - \frac{27}{14}u_{N-2} = f_{N-1} + \frac{15}{14}\frac{\beta}{\Delta x} + \frac{3}{7}\frac{\alpha}{\Delta x} \tag{84}$$

Khi đó thu được kết quả:

Từ đó ta có hệ phương trình $AU = F$ với $A \in \mathbb{R}^N \times \mathbb{R}^N$, $U, F \in \mathbb{R}^N$, có dạng:

$$A = \frac{1}{\Delta^2 x}\begin{bmatrix} \frac{2}{3} & -\frac{2}{3} & 0 & 0 & \cdots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 2 & -1 \\ \frac{-3}{2} & \frac{-3}{7} & \frac{-9}{14} & \frac{-9}{14} & \cdots & \frac{-9}{14} & \frac{-27}{14} & 0 \end{bmatrix} \tag{85}$$

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ u_{N-2} \\ u_{N-1} \end{bmatrix} \qquad F = \begin{bmatrix} f_1 - \frac{2\alpha}{3\Delta x} \\ f_2 \\ f_3 \\ \vdots \\ f_{N-2} \\ f_{N-1} + \frac{15}{14}\frac{\beta}{\Delta x} + \frac{3}{7}\frac{\alpha}{\Delta x} \end{bmatrix} \tag{86}$$

Ta có A là ma trận 3 đường chéo và đối xứng xác định dương.

Dưới đây là code cho việc tạo ma trận A, vector F.

```
 %% Create matrix A
    A=sparse(N-1,N-1);
%A = zeros(N-1,N-1);
    for i=1:N-1
        if (i==1)
            A(i,i)=2/3;
            A(i,i+1)=-2/3;
        elseif(i==N-1)
            A(i,1) = -3/2;
            A(i,2) = -3/7;
            A(i,N-2) = -27/14;
            A(i,N-1)=0;
            for j=2:N-3
                A(i,j)= -9/14;
            end
        else
            A(i,i-1)=-1;
            A(i,i+1)=-1;
            A(i,i)=2;
        end
    end

    A=A/((del_x)^2);
%% Create vector b
    b=zeros(N-1,1);
    for i=1:N-1
        if(i==1)
            b(i)=functionf(i*del_x,cases) -(2*al)/(3*del_x) ;
        elseif(i==N-1)
            b(i)=functionf(i*del_x,cases) + (15*be)/(14*del_x) ...
                -(3*al)/(7*del_x) ;
        else
            b(i)=functionf(i*del_x,cases) -(2*al)/(3*del_x) ;
        end
    end
```

Việc giải hệ trên sẽ sử dụng hàm có sẵn trong Matlab.

```
%% Solve discrete solution
    u=A\b;
```

Với kết quả U đã tính trên, ta sẽ plot trên Matlab và so sánh với nghiệm chính xác ta có:

Code cho nghiệm xấp xỉ vừa tính:

```
%% Create discrete solution with boundary
    u_dis=zeros(N+1,1);
    for i=1:N+1
        if (i==1)
            u_dis(i)=4/3*u(1,1)-1/3*u(2,1) -2/3*al*del_x;
        elseif(i==N+1)
            u_dis(i)=4/3*u(N-1,1)-1/3*u(N-2,1) +2/3*be*del_x;
        else
            u_dis(i)=u(i-1,1);
        end
    end
```

3.2.2 *Ví dụ*

Đầu tiên ta xét ví dụ:

VÍ DỤ 1 : Với hàm f là

$$f(x) = 3 - 6x \tag{87}$$

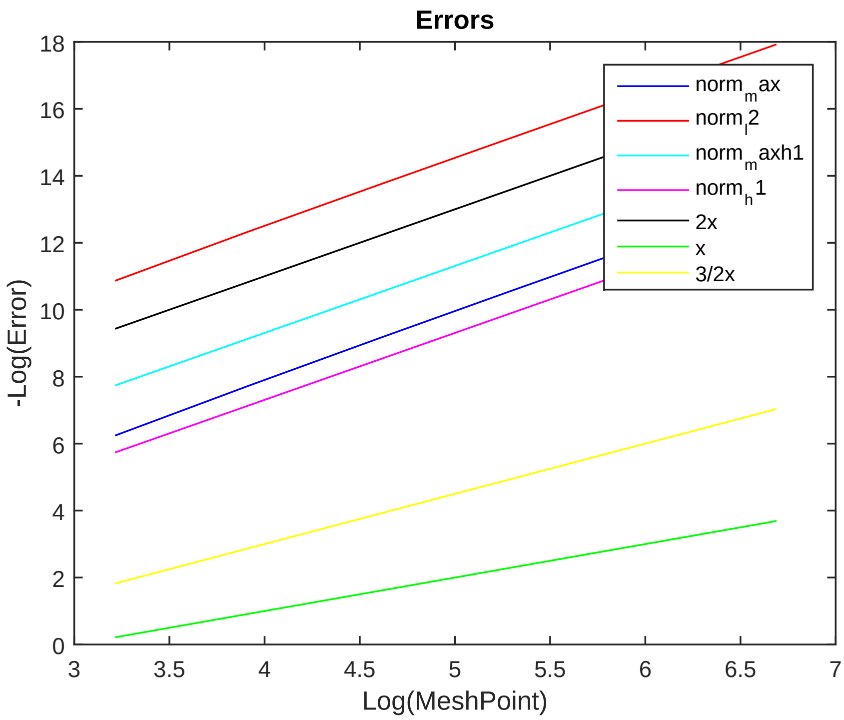
và điều kiện biên

$$u(0)' = 0$$
$$u'(1) = 0$$

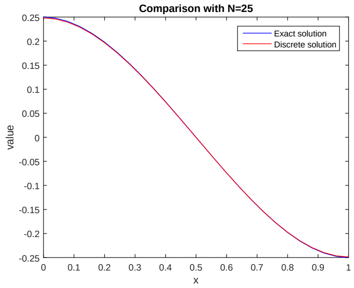
Ta có nghiệm chính xác là:

$$u(x) = x^4 \frac{3}{2}x^2 + \frac{1}{4} \quad (88)$$

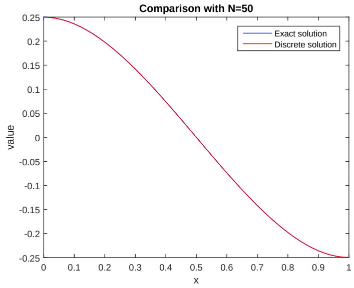
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 1:



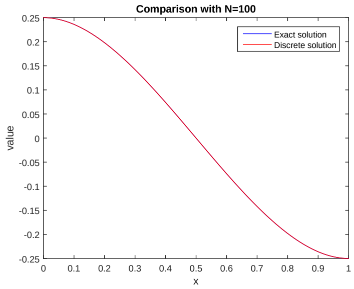
Hình 21: Bậc sai số của Ví dụ 1 - Neumann, bậc 2



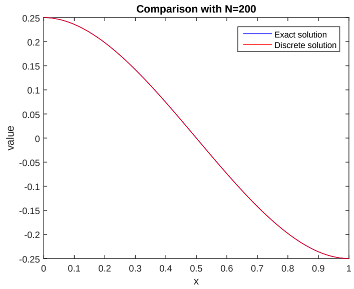
(a) Lần lặp đầu tiên



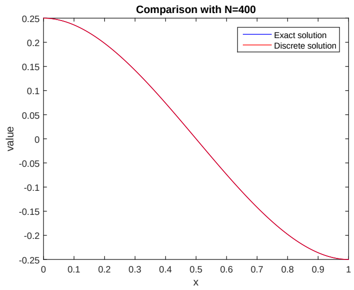
(b) Lần lặp thứ hai



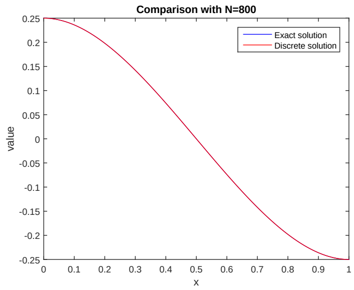
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 22: Ví dụ 1 được mô phỏng qua 6 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp biên Neumann bậc 2, ví dụ 1:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.001930256410256	0.001035324136865
2	4.439215686291020 e-04	2.419580258539983 e-04
3	1.056600660060691 e-04	5.887577354600821 e-05
4	2.571828357647576 e-05	1.455978230811000 e-05
5	6.340463940768260 e-06	3.623100401523676 e-06
6	1.573850586855663 e-06	9.038724525162393 e-07

Bảng 21: Bảng sai số cho L^∞ và L^2 biên Neumann, ví dụ 1

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	0.003200000000006	0.003200000000000
2	8.000000000146779 e-04	0.000800000000001
3 2.000000000168534 e-04	0.000199999999995	
4	5.000000006805117 e-05	0.000049999999992
5	1.250000007946284 e-05	0.000012499999990
6	3.125000302972580 e-06	0.000003125000055

Bảng 22: Bảng sai số cho $\max_{H_1}$ và H_1 biên Neumann, ví dụ 1

VÍ DỤ 2 : Với hàm f là

$$f(x) = 60x - 270x^8 \tag{89}$$

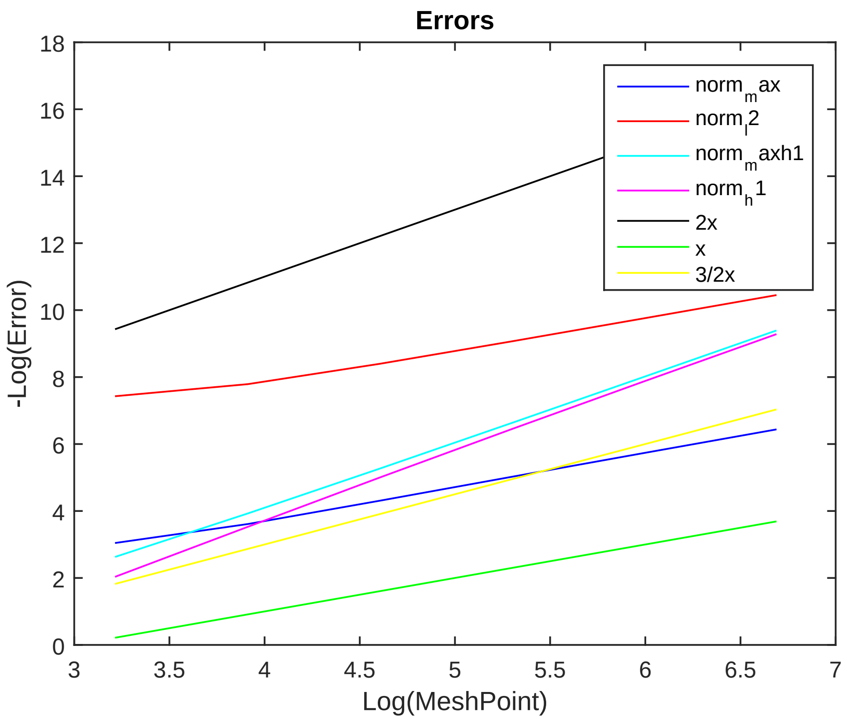
và điều kiện biên

$$u'(0) = 0$$
$$u'(1) = 0$$

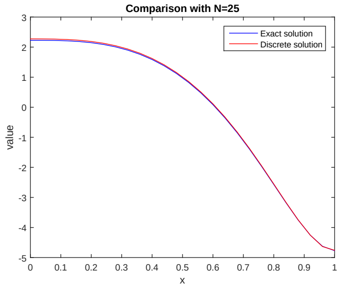
Ta có nghiệm chính xác là:

$$u(x) = 3x^{10} - 10x^3 + \frac{49}{22} \tag{90}$$

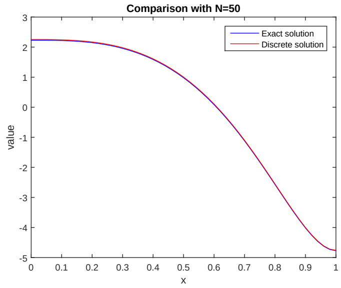
Khi đó một số ảnh minh họa và bậc xấp xỉ cho ví dụ 2:



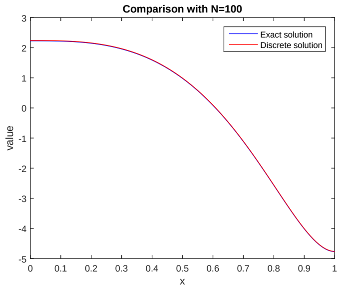
Hình 23: Bậc sai số của Ví dụ 2 -Neumann, bậc 2



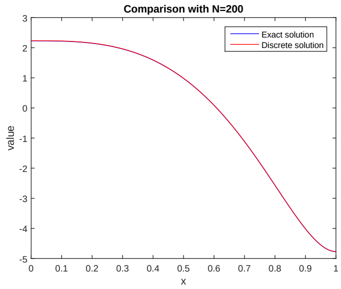
(a) Lần lặp đầu tiên



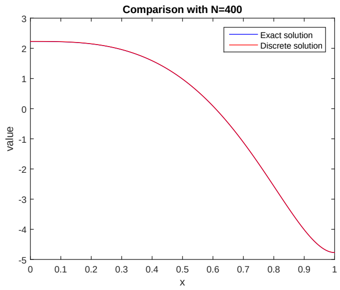
(b) Lần lặp thứ hai



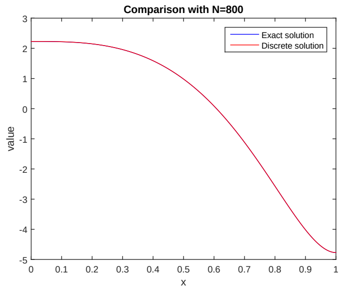
(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn



(e) Lần lặp thứ năm



(f) Lần lặp thứ sáu

Hình 24: Ví dụ 2 được mô phỏng qua 4 lần chia lưới mịn hơn

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới min hơn của trường hợp Neumann, ví dụ 2:

Lần lặp	Chuẩn L^∞	Chuẩn L^2
1	0.047478240078680	0.032371373283872
2	0.027029600745991	0.022611574635164
3	0.013469088917204	0.012317286580218
4	0.006595632492378	0.006302986943159
5	0.003245834476746	0.003172156296019
0.001589198374346	6.615692705315995 e-04	

Bảng 23: Bảng sai số cho L^∞ và L^2 Neumann bậc 2, ví dụ 2.

Lần lặp	Chuẩn $\max_{H_1}$	Chuẩn H^1
1	0.529337776932914	0.129588562518674
2	0.145540873764860	0.029376273519747
3	0.038151857676283	0.006736254579315
4	0.009766383604237	0.001586798284445
5	0.002470633900487	0.000382838728655
6	6.213189323034385 e-04	0.000093854201439

Bảng 24: Bảng sai số cho $\max_{H_1}$ và H_1 biên Neumann bậc 2, ví dụ 2.

3.2.3 Nhận xét

Có 2 nhận xét, một về bậc hội tụ, hai là sự hội tụ về nghiệm chính xác.
Bậc hội tụ:

- Qua hai hình (21) và (23) ta có thể thấy bậc hội tụ với tất cả các chuẩn của ví dụ 1 của bài toán biên Neumann sử dụng xấp xỉ tại biên (6) có bậc hội tụ là bậc 2 (song song với đường $2x$).
- Điều này hợp lí vì khi xây dựng, có 2 lần ta xấp xỉ ta đều sử dụng xấp xỉ đạo hàm bậc 2, $O(h^2)$ (xấp xỉ đạo hàm cấp 2 và đạo hàm tại biên).
- Tuy nhiên đã có sự thay đổi khi ví dụ 2 tất cả các chuẩn đều chuyền về bậc nằm trong khoảng 1 và $3/2$, có thể là do ảnh hưởng của hàm lượng giác đến khai triển Taylor.

Sự hội tụ:

- Qua bảng ghi lại kết quả (23), (??), (??) và (24), đồng thời là các ảnh so sánh kết quả sau cá bước lặp (22) và (??) ta thấy sau từng bước lặp, sai số dần nhỏ đi, đúng như dự định.
- Ngoài ra ví dụ 1 có kết quả sai số và sự hội tụ nhanh hơn ví dụ 2 vì ví dụ 1 có kết quả là một hàm đa thức trong khi ví dụ 2 liên quan đến lượng giác mà khai triển Taylor sẽ hội tụ nhanh cho hàm đa thức và khá chậm khi có liên quan đến lượng giác, Lôgarit.

3.3 Nhận xét chung

Bậc hội tụ:

- Việc sử dụng công thức xấp xỉ đạo hàm cấp 2 có sai số bậc 2 và xấp xỉ đạo hàm cấp 1 tại biên cũng có sai số bậc 2 sẽ tạo ra bậc sai số là 2 cho tất cả các chuẩn. Khi thay xấp xỉ tại biên xuống chỉ còn bậc 1, lập tức bậc sai số sẽ chuyển về 1 hoặc $3/2$ hoặc nằm giữa đoạn trên.
- Hàm lượng giác cũng tác động đến bậc hội tụ khi các hàm lượng giác đều không được xấp xỉ tốt lắm bằng khai triển Taylor. Bậc hội tụ thường giao động trong đoạn $[1, 3/2]$.

TÀI LIỆU

[1] Finite Different Methods in 1D, Le Anh Ha (Lecture Note). Khoa Toán - Tin, Đại học Khoa học tự nhiên TPHCM, 2017.

[2] Finite Different Methods for Ordinary and Partial Differential Equations, Randall J.LeVeque. [Steady - Stable and Time - Dependent Problems]. University of Washington, Seattle, Washington.