

BÀI TẬP THỰC HÀNH SỐ 4

THỂ TÍCH HỮU HẠN

PHƯƠNG TRÌNH NHIỆT

LƯU GIANG NAM^{1,2}

MỤC LỤC

1	Bài toán với điều kiện biên $u(a, t) = \phi_a(t)$, $u(b, t) = \phi_b(t)$	4
1.1	Giới thiệu bài toán	4
1.2	Chia lưới	4
1.3	Phân rã bài toán	5
1.3.1	Forward Euler method - Explicit method	7
1.3.2	Backward Euler method - Implicit Method	7
1.3.3	Crank - Nicolson method	8
1.3.4	Công thức xấp xỉ tổng quát	8
1.4	Xuất kết quả, so sánh và biểu diễn bậc hội tụ	8
1.4.1	Forward Euler method - Explicit method	10
1.4.2	Backward Euler method - Implicit Method	14
1.4.3	Crank - Nicolson method	18
1.4.4	Công thức xấp xỉ tổng quát	22
1.5	Kết luận và so sánh các phương pháp	26
2	Bài toán với điều kiện biên $\frac{\partial u(a, t)}{\partial x} = \psi_a(t)$, $\frac{\partial u(b, t)}{\partial x} = \psi_b(t)$	28
2.1	Giới thiệu bài toán	28
2.2	Chia lưới	28
2.3	Phân rã bài toán	28
2.3.1	Forward Euler method - Explicit method	31
2.3.2	Backward Euler method	31
2.3.3	Crank - Nicolson method	31
2.3.4	Công thức xấp xỉ tổng quát	31
2.4	Xuất kết quả, so sánh và biểu diễn bậc hội tụ	32
2.4.1	Forward Euler method - Explicit method	32
2.4.2	Backward Euler method - Implicit Method	34
2.4.3	Crank - Nicolson method	36
2.4.4	Công thức xấp xỉ tổng quát	38
2.5	Kết luận và so sánh các phương pháp	40

¹ Khoa Toán - Tin học, Đại học Khoa học Tự nhiên, TP Hồ Chí Minh

² Email: luugiangnam96@gmail.com

DANH SÁCH HÌNH VẼ

Hình 1	Bậc sai số với độc lập thời gian Dirichlet - Forward Method .	11
Hình 2	Bậc sai số với phụ thuộc thời gian Dirichlet - Forward Method	11
Hình 3	Xấp xỉ với biên Dirichlet bằng Forward Method	12
Hình 4	Bậc sai số với độc lập thời gian Dirichlet - Forward Method .	13
Hình 5	Bậc sai số với phụ thuộc thời gian Dirichlet - Forward Method	13
Hình 6	Xấp xỉ với biên Dirichlet - Forward Method	14
Hình 7	Bậc sai số với độc lập thời gian Dirichlet - Backward Method	15
Hình 8	Bậc sai số với phụ thuộc thời gian Dirichlet - Backward Method	15
Hình 9	Xấp xỉ với biên Dirichlet bằng Backward Method	16
Hình 10	Bậc sai số với độc lập thời gian Dirichlet - Backward Method	17
Hình 11	Bậc sai số với phụ thuộc thời gian Dirichlet - Backward Method	17
Hình 12	Xấp xỉ với biên Dirichlet - Backward Method	18
Hình 13	Bậc sai số với độc lập thời gian Dirichlet - Crank - Nicolson method	19
Hình 14	Bậc sai số với phụ thuộc thời gian Dirichlet - Crank - Nicolson method	19
Hình 15	Xấp xỉ với biên Dirichlet - Crank - Nicolson method	20
Hình 16	Bậc sai số với độc lập thời gian Dirichlet - Crank - Nicolson Method	21
Hình 17	Bậc sai số với phụ thuộc thời gian Dirichlet - Crank - Nicolson Method	21
Hình 18	Xấp xỉ với biên Dirichlet - Crank - Nicolson Method	22
Hình 19	Bậc sai số với độc lập thời gian Dirichlet - General method . .	23
Hình 20	Bậc sai số với phụ thuộc thời gian Dirichlet - General method	23
Hình 21	Xấp xỉ với biên Dirichlet - General method	24
Hình 22	Bậc sai số với độc lập thời gian Dirichlet - General Method .	25
Hình 23	Bậc sai số với phụ thuộc thời gian Dirichlet - General Method	25
Hình 24	Xấp xỉ với biên Dirichlet - General Method	26
Hình 25	Bậc sai số với độc lập thời gian Neumann - Forward Method .	32
Hình 26	Bậc sai số với phụ thuộc thời gian Neumann - Forward Method	33
Hình 27	Xấp xỉ với biên Neumann bằng Forward Method	33
Hình 28	Bậc sai số với độc lập thời gian Neumann - Backward Method	34
Hình 29	Bậc sai số với phụ thuộc thời gian Neumann - Backward Method	35
Hình 30	Xấp xỉ với biên Neumann bằng Backward Method	35
Hình 31	Bậc sai số với độc lập thời gian Neumann - Crank - Nicolson method	36
Hình 32	Bậc sai số với phụ thuộc thời gian Neumann - Crank - Nicolson method	37
Hình 33	Xấp xỉ với biên Neumann - Crank - Nicolson method	37
Hình 34	Bậc sai số với độc lập thời gian Neumann - General method .	38
Hình 35	Bậc sai số với phụ thuộc thời gian Neumann - General method	39
Hình 36	Xấp xỉ với biên Neumann - General method	39

DANH SÁCH BẢNG

Bảng 1	Bảng sai số ví dụ 1 biên Dirichlet - Forward Method.	12
Bảng 2	Bảng sai số ví dụ 2 biên Dirichlet bằng Forward Method. . . .	14
Bảng 3	Bảng sai số ví dụ 1 biên Dirichlet - Backward Method.	16
Bảng 4	Bảng sai số ví dụ 2 biên Dirichlet bằng Backward Method. . .	18
Bảng 5	Bảng sai số bài toán biên Dirichlet bằng Crank - Nicolson method.	20
Bảng 6	Bảng sai số ví dụ 2 biên Dirichlet bằng Crank - Nicolson Method.	22
Bảng 7	Bảng sai số bài toán biên Dirichlet bằng General method. . .	24
Bảng 8	Bảng sai số ví dụ 2 biên Dirichlet bằng General Method. . . .	26
Bảng 9	Bảng sai số ví dụ 1 biên Neumann - Forward Method.	34
Bảng 10	Bảng sai số ví dụ 1 biên Neumann - Backward Method. . . .	36
Bảng 11	Bảng sai số bài toán biên Neumann bằng Crank - Nicolson method.	38
Bảng 12	Bảng sai số bài toán biên Neumann bằng General method. . .	40

1 BÀI TOÁN VỚI ĐIỀU KIỆN BIÊN $u(a, t) = \phi_a(t)$, $u(b, t) = \phi_b(t)$

1.1 Giới thiệu bài toán

Xét phương trình khuếch tán:

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + f(x), \quad t > 0, \quad a < x < b \quad (1.1)$$

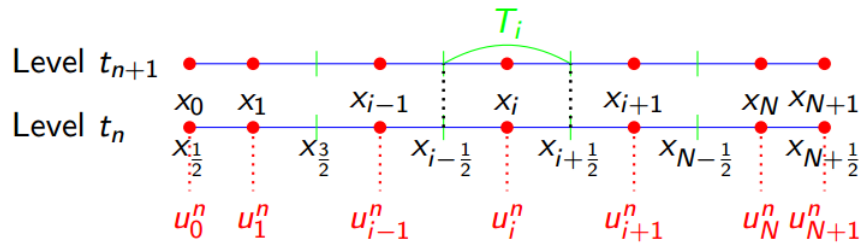
với κ là hệ số khuếch tán. Điều kiện đầu là:

$$u(x, 0) = u_0(x) \quad (1.2)$$

Điều kiện biên là:

$$u(a, t) = \phi_a(t), \quad u(b, t) = \phi_b(t) \quad (1.3)$$

1.2 Chia lưới



Chúng ta chọn $N + 1$ điểm $\{x_{i+1/2}\}_{i \in \overline{0, N}}$ trên $[0, 1]$ thỏa mãn:

$$a = x_{1/2} < x_{3/2} < \dots < x_{N-1/2} < x_{N+1/2} = b$$

Ta đặt $T_i = [x_{i-1/2}, x_{i+1/2}]$, $h = \max_{i \in \overline{1, N}} \{|T_i|\}$

$$x_0 = a, \quad x_{N+1} = b, \quad x_i \in T_i, \quad \forall i \in \overline{1, N}$$

Chúng ta gọi $\{T_i\}_{i \in \overline{1, N}}$ là control volume và $\{x_i\}_{i \in \overline{0, N+1}}$ là control point.

Chúng ta cũng chia đoạn $[0, T]$ thành $N_k + 1$ đoạn con với chiều dài cố định là k và kí hiệu $t_n = nk$. Kí hiệu u_i^n là giá trị $u(x_i, t_n)$.

Code cho công đoạn chia lưới:

```
1 a=0.0;
2 b=1.0;
3 time=0.5;
4 kappa = 1; % Choose a number of kappa
5 N=6; % Number of control volume
6 T=300; % Number of time iterior
7 M=4; % Number of iteration when refine mesh
8
9 %% Create the mesh point
10 dx=(b-a)/N;
11 k=time/T;
12 x=zeros(N+1,1); % x_(i+1/2)
13 t=0:k:time;
14 for i_iterx=1:N+1
15     x(i_iterx)=a+(i_iterx-1)*dx;
16 end
```

```

17
18 %% create control point
19 x_cp=zeros(N+2,1); % x(i)
20 for i_iterx=1:N+2
21     if(i_iterx==1)
22         x_cp(i_iterx)=x(i_iterx);
23     else
24         if(i_iterx==N+2)
25             x_cp(i_iterx)=x(i_iterx-1);
26         else
27             x_cp(i_iterx)=(x(i_iterx-1)+x(i_iterx))/2.0;
28         end
29     end
30 end
    
```

1.3 Phân rã bài toán

Lấy tích phân (1.1) trên T_i ta được:

$$\frac{1}{|T_i|} \int_{T_i} u_t(x, t) dx + \frac{\kappa}{|T_i|} \int_{T_i} -u_{xx}(x, t) dx = \frac{1}{|T_i|} \int_{T_i} f(x, t) dx \quad (1.4)$$

Áp dụng công thức Green ta có:

$$\frac{\kappa}{|T_i|} \int_{T_i} -u_{xx}(x, t) dx = \kappa \frac{-u_x(x_{i+1/2}, t) + u_x(x_{i-1/2}, t)}{|T_i|} \quad (1.5)$$

Đặt

$$f_i(t) = \frac{1}{|T_i|} \int_{T_i} f(x, t) dx$$

là giá trị trung bình của f trên T_i .

Khi đó (1.4) ta có thành:

$$\frac{du_i}{dt}(t) + \kappa \frac{-u_x(x_{i+1/2}, t) + u_x(x_{i-1/2}, t)}{|T_i|} = f_i(t) \quad (1.6)$$

Áp dụng khai triển Taylor ta có:

$$\begin{aligned} u(x_{i+1}, t) - u(x_i, t) &= (x_{i+1} - x_i) u_x(x_{i+1/2}, t) \\ &+ \left((x_{i+1} - x_{i+1/2})^2 - (x_i - x_{i+1/2})^2 \right) \frac{u_{xx}(x_{i+1/2}, t)}{2!} + O(h^3) \end{aligned}$$

Ta sẽ xét trường hợp $x_{i+1/2}$ là trung điểm của $[x_i, x_{i+1}]$, khi đó:

$$u_x(x_{i+1/2}, t) = \frac{u(x_{i+1}, t) - u(x_i, t)}{x_{i+1} - x_i} + O(h^2)$$

Khi đó ta có (1.6) thành:

$$\begin{aligned} \frac{du_i(t)}{dt} - \kappa \frac{u_{i-1}(t)}{(x_i - x_{i-1})|T_i|} + \kappa \left(\frac{1}{(x_{i+1} - x_i)|T_i|} + \frac{1}{(x_i - x_{i-1})|T_i|} \right) u_i(t) \\ - \kappa \frac{u_{i+1}(t)}{(x_{i+1} - x_i)|T_i|} = f_i(t) \end{aligned}$$

Đặt

$$\begin{aligned} \alpha_i &= \frac{\kappa}{(x_i - x_{i-1})|T_i|} \\ \beta_i &= \frac{-\kappa}{(x_{i+1} - x_i)|T_i|} + \frac{-\kappa}{(x_i - x_{i-1})|T_i|} \end{aligned}$$

$$\alpha_i = \frac{\kappa}{(x_{i+1} - x_i)|T_i|}$$

Khi đó ta có:

$$\frac{du_i(t)}{dt} - \alpha_i u_{i-1}(t) - \beta_i u_i(t) - \gamma u_{i+1}(t) = f_i(t), \quad \forall i \in \overline{2, N-1} \quad (1.7)$$

và

$$\frac{du_1(t)}{dt} - \beta_1 u_1(t) - \gamma u_2(t) = f_1(t) + \alpha_1 \phi_a(t) \quad (1.8)$$

$$\frac{du_N(t)}{dt} - \alpha_N u_{N-1}(t) - \beta_N u_N(t) = f_N(t) + \gamma \phi_b(t) \quad (1.9)$$

Hay ta có hệ ODE như sau:

$$\frac{dU(t)}{dt} = AU(t) + F(t) \quad (1.10)$$

với

$$A = \begin{cases} \begin{matrix} \beta_1 & \gamma_1 & & & \\ \alpha_2 & \beta_2 & \gamma_2 & & \\ & \alpha_3 & \beta_3 & \gamma_3 & \\ & & & \dots\dots\dots & \\ & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} \\ & & & \alpha_N & \beta_N \end{matrix} & \begin{matrix} = f_1(t) - \alpha_1 \phi_a(t) \\ = f_2(t) \\ = f_3(t) \\ \\ = f_{N-1}(t) \\ = f_N(t) - \gamma_N \phi_b(t) \end{matrix} \end{cases} \quad (1.11)$$

Code Matlab cho ma trận A, vector F khi thay thời gian vào và các giá trị đã cho và cần tính.

```

1  %% Create form for computing
2  A = zeros(N,N); % Matrix A
3  u = zeros(N,T); % Matrix value of u
4  u0 = zeros(N,1); % u(x,0)
5  phi_a = zeros(1,T); % u(a,t)
6  phi_b = zeros(1,T); % u(b,t)
7
8  %% Input interior
9  for i_iterx=1:N
10     u0(i_iterx) = u_ex(x_cp(i_iterx+1),0);
11 end
12
13 for i_itert = 1:T
14     phi_a(i_itert) = u_ex(a,t(i_itert));
15 end
16
17 for i_itert = 1:T
18     phi_b(1,i_itert) = u_ex(b,t(i_itert));
19 end
20
21 %% Matrix of function f
22 F = zeros(N,T);
23 for i_itert = 1:T
24     for i_iterx = 1:N
25         if(i_iterx ==1)
26             alpha = ...
27                 kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+1)-x_cp(i_iterx))));
28             F(i_iterx,i_itert) = ...
29                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2 ...
30                 + alpha*phi_a(1,i_itert));
31         elseif(i_iterx ==N)
32             gamma = ...
33                 kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+2)-x_cp(i_iterx+1))));
34             F(i_iterx,i_itert) = ...
35                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2 ...
36                 + gamma*phi_b(1,i_itert));
37         else
38             F(i_iterx,i_itert) = ...
39                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2 ...
40                 + alpha*phi_a(1,i_itert) + gamma*phi_b(1,i_itert));
41         end
42     end
43 end

```

```

31         else
32             F(i_iterx,i_ittert) = ...
                 (fun(x(i_iterx+1),t(i_ittert))+fun(x(i_iterx),t(i_ittert)))/2;
33
34         end
35     end
36 end
37
38 %% Matrix A
39 for i_iterx = 1:N
40     alpha = ...
         kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+1)-x_cp(i_iterx)));
41     gamma = ...
         kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+2)-x_cp(i_iterx+1)));
42     if(i_iterx ==1)
43         A(i_iterx,i_iterx) = -alpha - gamma;
44         A(i_iterx,i_iterx+1) = gamma;
45     elseif(i_iterx == N)
46         A(i_iterx,i_iterx) = -alpha - gamma;
47         A(i_iterx,i_iterx-1) = alpha;
48     else
49         A(i_iterx,i_iterx) = -alpha - gamma;
50         A(i_iterx,i_iterx-1) = alpha;
51         A(i_iterx,i_iterx+1) = gamma;
52     end
53 end
    
```

Để giải (1.10) ta sẽ sử dụng một số phương pháp sau:

1.3.1 Forward Euler method - Explicit method

Ta có công thức Forward Euler method như sau:

$$U^{n+1} = U^n + k(AU^n + F^n), \quad U^n \approx U(nk) \quad (1.12)$$

với $\Delta t = k$, hay

$$U^{n+1} = (I + kA)U^n + kF^n, \quad U^n \approx U(nk) \quad (1.13)$$

"Time step restriction" là :

$$r = \frac{k}{h^2} \leq \frac{1}{2}$$

Khi đó ta sẽ code bài toán bằng Matlab như sau:

```

1  %% Forward Euler method
2  u(:,1) = u0(:,1);
3  for i_ittert = 2:T
4      u(:,i_ittert) = (eye(N,N)+k*A)*u(:,i_ittert-1) + k*F(:,i_ittert-1);
5  end
    
```

Tuy nhiên chú ý đến điều kiện

$$\frac{k}{h^2} \leq \frac{1}{2} \Leftrightarrow \frac{\text{time}/T}{((b-a)/N)^2} \leq \frac{1}{2} \Leftrightarrow N^2 \leq T$$

vì đã chọn $\text{time} = \frac{1}{2}$ và $a = 0$, $b = 1$.

Vậy như cách chọn trên $N = 6$, $T = 300$ thì sau 4 lần gấp đôi lên ta vẫn thỏa điều kiện "Time step restriction" (ở lần cuối thì ta có: $48^2 = 2304 < 2400$).

1.3.2 Backward Euler method - Implicit Method

Ta có công thức Backward Euler method như sau:

$$U^{n+1} = U^n + kAU^{n+1} + kF^{n+1} \quad (1.14)$$

với $\Delta t = k$, hay

$$U^{n+1} = \frac{U^n + F^{n+1}}{I - kA} \quad (1.15)$$

Khi đó ta có code Matlab cho phương pháp này như sau:

```
1 %% Backward Euler method
2 u(:,1) = u0(:,1);
3 for i_iter = 2:T
4     u(:,i_iter) = (eye(N,N)-k*A)^(-1)*(u(:,i_iter-1) + k*F(:,i_iter));
5 end
```

1.3.3 Crank - Nicolson method

Ta có công thức θ method như sau:

$$U^{n+1} = U^n + \frac{kA}{2} [U^{n+1} + U^n] + \frac{k}{2} [F^{n+1} + F^n] \quad (1.16)$$

với $\Delta t = k$, hay

$$U^{n+1} = \frac{(I + 1/2 kA)U^n + kF_{\theta}^n}{1 - 1/2 kA} \quad (1.17)$$

với $F_{\theta}^n = \frac{1}{2}(F^{n+1} + F^n)$

```
1 %% Crank - Nicolson method
2 for i_iter = 2:T
3     u(:,i_iter) = ...
4         (eye(N,N)-1/2*k*A)^(-1)*((eye(N,N)+1/2*k*A)*u(:,i_iter-1) + ...
5         k/2*(F(:,i_iter)+F(:,i_iter-1)));
6 end
```

1.3.4 Công thức xấp xỉ tổng quát

Ta có công thức tổng quát θ method như sau:

$$U^{n+1} = U^n + kA [\theta U^{n+1} + (1 - \theta)U^n] + k[\theta F^{n+1} + (1 - \theta)F^n] \quad (1.18)$$

với $\Delta t = k$, hay

$$U^{n+1} = \frac{(I + (1 - \theta)kA)U^n + kF_{\theta}^n}{1 - \theta kA} \quad (1.19)$$

với $F_{\theta}^n = \theta F^{n+1} + (1 - \theta)F^n$

1.4 Xuất kết quả, so sánh và biểu diễn bậc hội tụ

Để so sánh được ta cần tạo một ma trận chứa các giá trị chính xác và ma trận chứa các giá trị xấp xỉ, cả hai đều thuộc $M_{N+2,T}$.

Dưới đây là đoạn code Matlab cho việc tạo hai ma trận.

```
1 %% Create a matrix for exact solution
2
3 u_exact = zeros(N+2,T);
4 t=0:k:time;
5 for i_iter=1:T
6     for i_iterx = 1:N+2
7         u_exact(i_iterx,i_iter) = u_ex(x_cp(i_iterx),t(i_iter));
8     end
9 end
10
```



```

11 %% Create a matrix for approximate solution
12 u_dis = zeros(N+2,T);
13 t=0:k:time;
14 for i_ittert=1:T
15     for i_iterx = 1:N+2
16         if(i_iterx == 1)
17             u_dis(i_iterx,i_ittert) = phi_a(1,i_ittert);
18         elseif(i_iterx == N+2)
19             u_dis(i_iterx,i_ittert) = phi_b(1,i_ittert);
20         else
21             u_dis(i_iterx,i_ittert) = u(i_iterx - 1, i_ittert);
22         end
23     end
24 end

```

Sau đó là vẽ hai hình lên cùng một figure để biểu hiện sự hội tụ của nghiệm xấp xỉ đến nghiệm chính xác.

Code:

```

1 figure
2 plot(x_cp,u_dis(:,T), 'b-*', x_cp,u_exact(:,T), 'r');
3 title(['Compare exact and approximate solution at h = ', num2str(dx), ...
4       ', k = ', num2str(k)]);
5 legend('Approximate solution', 'Exact solution')

```

Bài báo cáo sẽ tính toán và vẽ bậc hội tụ theo thời gian và độc lập với thời gian (tức chỉ đánh giá phụ thuộc x tại thời điểm T).

Code Matlab cho bậc hội tụ trên toàn thời gian:

```

1 %% Calculate error for all period
2 norml0=0;
3 for i_iterx = 1:N
4     norml0 = norml0 ...
5         +(u_dis(i_iterx+1,T)-u_exact(i_iterx+1,T))^2*(x(i_iterx+1)-x(i_iterx));
6 end
7 norml2(jj) = norml2(jj) + sqrt(norml0)*k;
8 norml2(jj) = sqrt(norml2(jj));
9 normh0 = 0;
10 for i_ittert = 1:T
11     for i_iterx = 1:N
12         normh0 = normh0 ...
13             +(u_dis(i_iterx+1,i_ittert)-u_exact(i_iterx+1,i_ittert)-...
14             -(u_dis(i_iterx,i_ittert)-u_exact(i_iterx,i_ittert)))^2...
15             /(x_cp(i_iterx+1)-x_cp(i_iterx));
16     end
17 end
18 normh1(jj) = normh1(jj) + sqrt(normh0)*k;
19 normh1(jj) = sqrt(normh1(jj));

```

Code Matlab cho xấp xỉ chỉ phụ thuộc và không gian tại thời điểm T .

```

1 %% Calculate error independent on t
2 for i_iterx = 1:N
3     norml2_x(jj) = norml2_x(jj) ...
4         +(u_dis(i_iterx+1,i_ittert)-u_exact(i_iterx+1,i_ittert))^2...
5         *(x(i_iterx+1)-x(i_iterx));
6 end
7 norml2_x(jj) = sqrt(norml2_x(jj));
8 for i_iterx = 1:N
9     normh1_x(jj) = normh1_x(jj) ...
10         +(u_dis(i_iterx+1,i_ittert)-u_exact(i_iterx+1,i_ittert)-...
11         -(u_dis(i_iterx,i_ittert)-u_exact(i_iterx,i_ittert)))^2...
12         /(x_cp(i_iterx+1)-x_cp(i_iterx));
13 end

```

```
13 normh1_x(jj) = sqrt(normh1_x(jj));
```

Trong đó jj chính là giá trị của vòng FOR để biết số lần lặp.

Sau khi ra khỏi vòng FOR của jj ta sẽ thực hiện công đoạn cuối là vẽ bậc hội tụ của cả hai sai số.

```
1 %% Plot error for all period
2 figure
3 plot(log(11.^(1/2)), -log(norml2), 'r', log(11.^(1/2)), log(11.^(1/2)), 'black', ...
4 log(11.^(1/2)), -log(normh1), 'b', log(11.^(1/2)), 1.5*log(11.^(1/2)), 'c');
5 title('Error in L2 and H1 norms');
6 legend('L2 Norm', 'x', 'H1 Norm', '3x/2' )
7
8 %% Plot error independent on t
9 figure
10 plot(log(11.^(1/2)), -log(norml2_x), 'r', log(11.^(1/2)), 2*log(11.^(1/2)), 'black', ...
11 log(11.^(1/2)), -log(normh1_x), 'b', log(11.^(1/2)), 1.5*log(11.^(1/2)), 'c');
12 title('Error in L2 and H1 norms independent on t');
13 legend('L2 Norm', '2x', 'H1 Norm', '3x/2' )
```

1.4.1 Forward Euler method - Explicit method

VÍ DỤ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (1.20)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

điều kiện biên là:

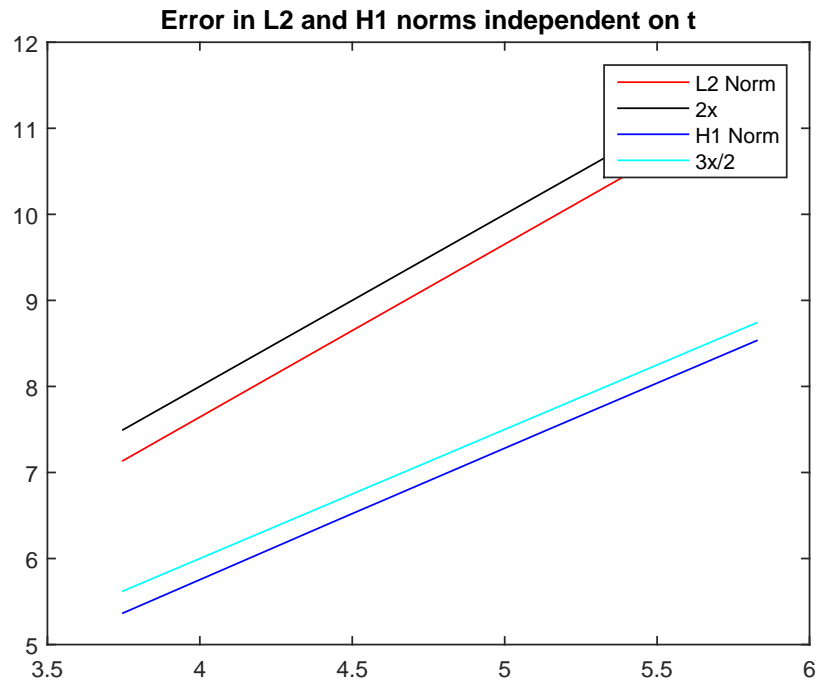
$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

Nghiệm chính xác là:

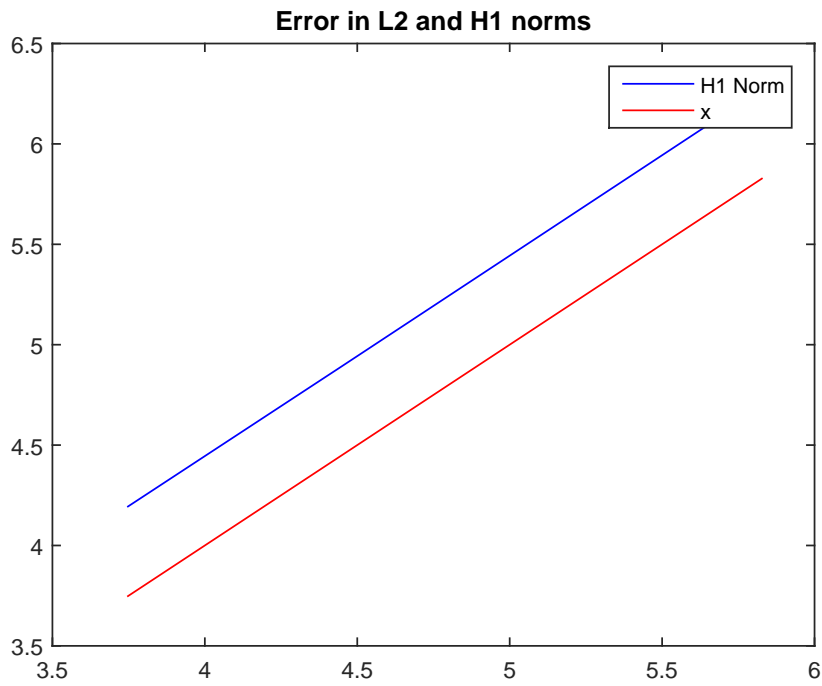
$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Điều kiện ổn định đã được xét như trên phần giới thiệu phương pháp.

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

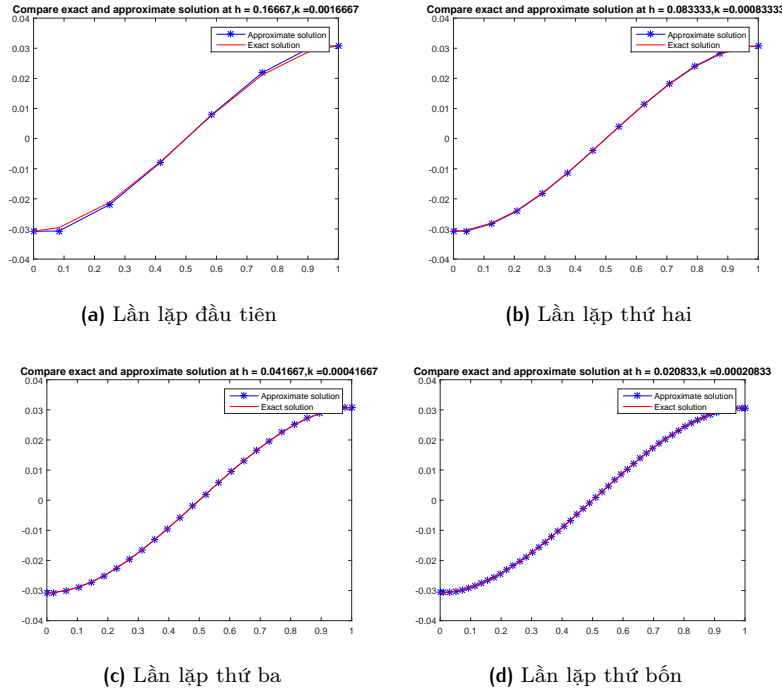


Hình 1: Ví dụ 1 Dirichlet - Forward Method, bậc hội tụ theo x



Hình 2: Ví dụ 1 Dirichlet - Forward Method, bậc hội tụ theo (x,t)

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ là bậc 1.



Hình 3: Ví dụ 1 Dirichlet - Forward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Forward Method.

		Thời gian chạy: 2.022962s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.015085108907841	$1.0e-03 * 0.795223327580421$	0.004672353840191
2	0.007566077012040	$1.0e-03 * 0.197345342416571$	0.001609751219762
3	0.003782313027741	$1.0e-03 * 0.049077294058822$	0.000560941528213
4	0.001890490439234	$1.0e-03 * 0.012176532195559$	0.000196809054990

Bảng 1: Bảng sai số ví dụ 1 biên Dirichlet - Forward Method.

VÍ DỤ 2: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t = \frac{1}{16}u_{xx}, \quad \forall x \in (0, 1), \quad t \in (0, T) \quad (1.21)$$

với điều kiện đầu

$$u_0(x) = \sin(2\pi x)$$

điều kiện biên là:

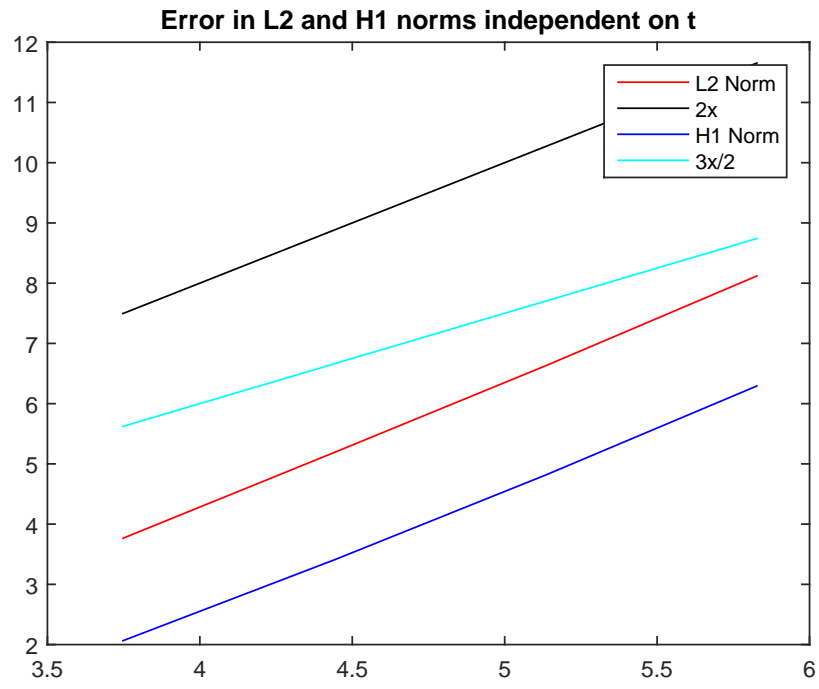
$$u(0, t) = u(1, t) = 0$$

Nghiệm chính xác là:

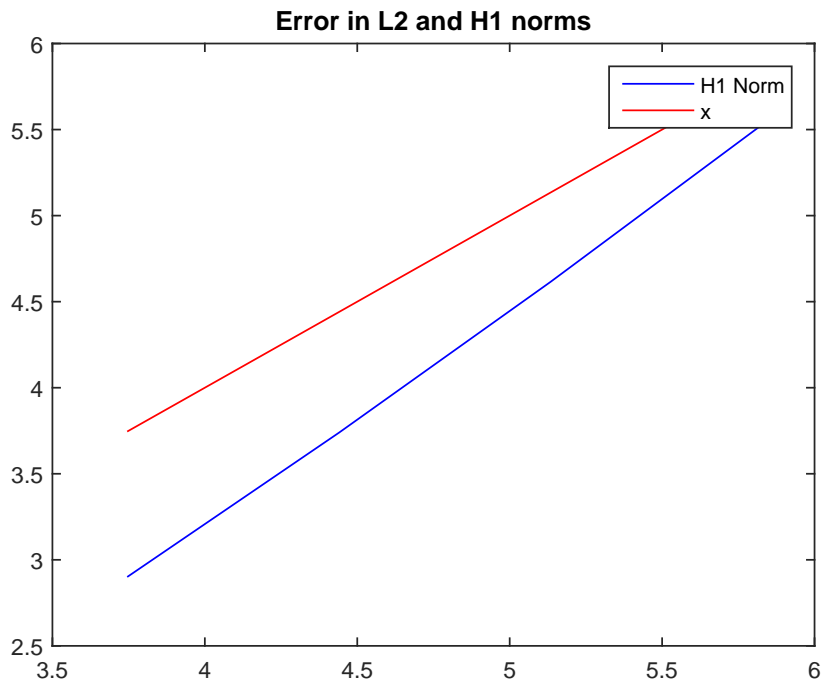
$$u(x, t) = e^{\frac{1}{4}\pi^2 t} \sin(2\pi x)$$

Điều kiện ổn định đã được xét như trên phần giới thiệu phương pháp.

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

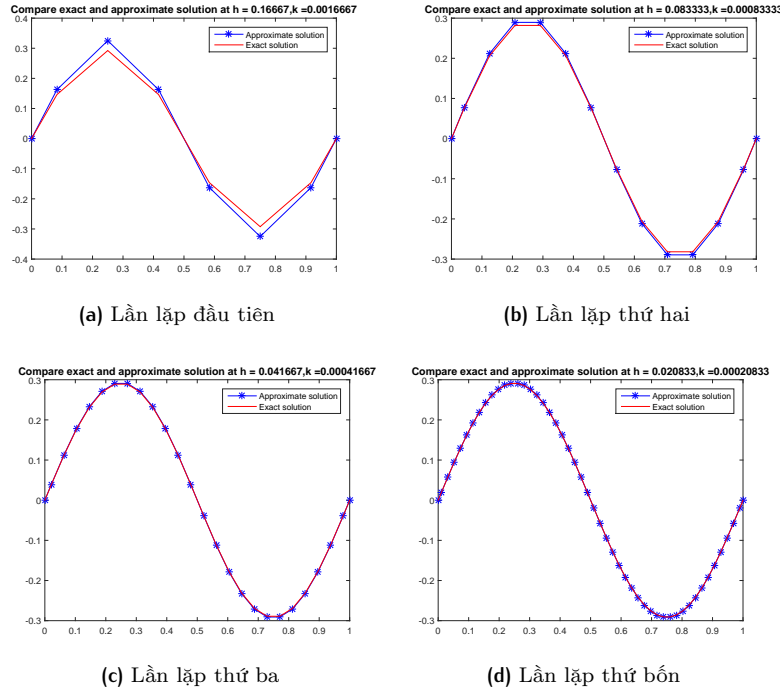


Hình 4: Ví dụ 2 Dirichlet - Forward Method, bậc hội tụ theo x



Hình 5: Ví dụ 1 Dirichlet - Forward Method, bậc hội tụ theo (x,t)

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ nhỏ hơn bậc 1.



Hình 6: Ví dụ 2 Dirichlet - Forward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Forward Method.

	Thời gian chạy: 1.847897s		
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.054844885177567	0.023174113645618	0.126929847938932
2	0.023757985750817	0.005577504927694	0.033170586805413
3	0.009901424402968	0.001322960629252	0.008114170389475
4	0.003977116323913	0.000297651459893	0.001849294542698

Bảng 2: Bảng sai số ví dụ 2 biên Dirichlet bằng Forward Method.

1.4.2 Backward Euler method - Implicit Method

Ví dụ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (1.22)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

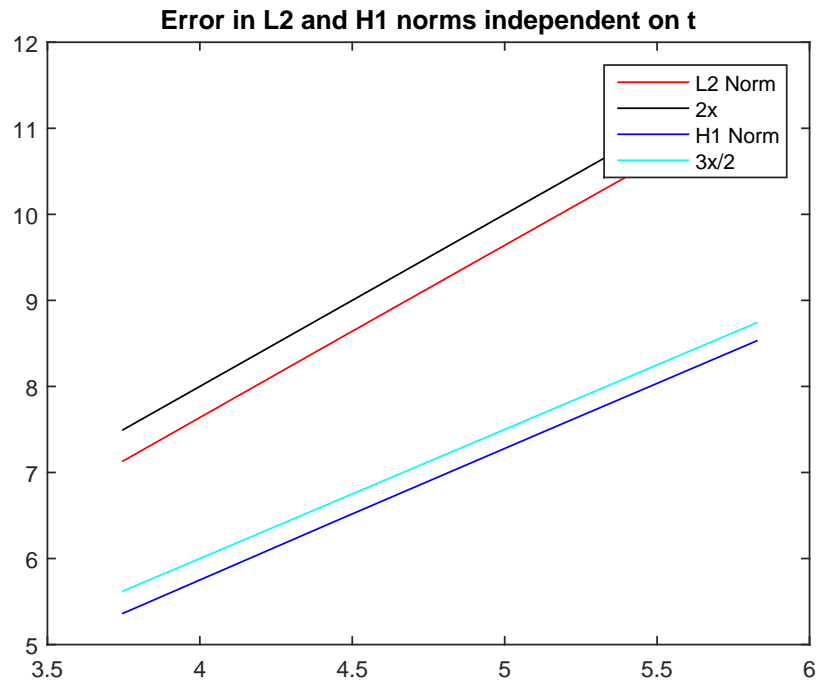
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

Nghiệm chính xác là:

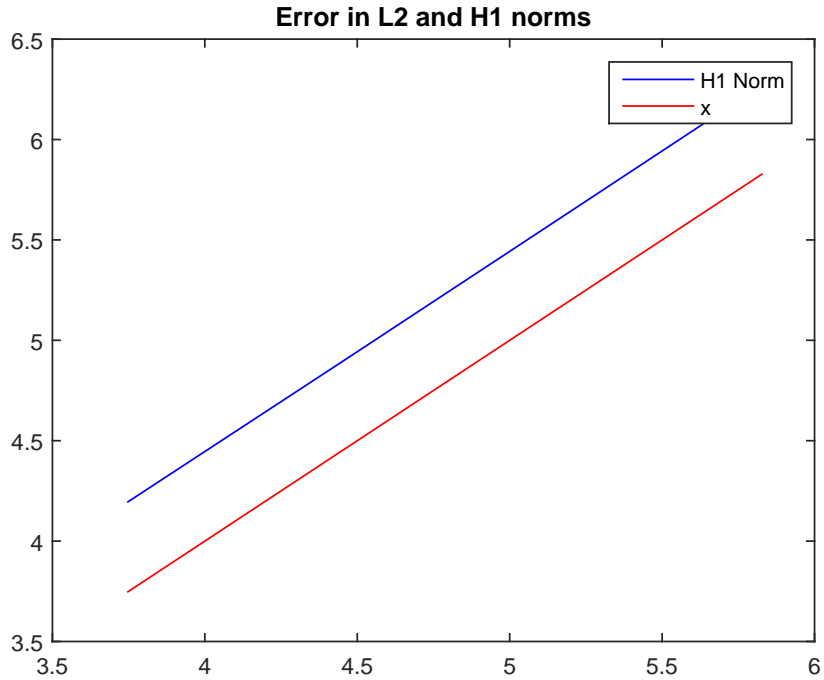
$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

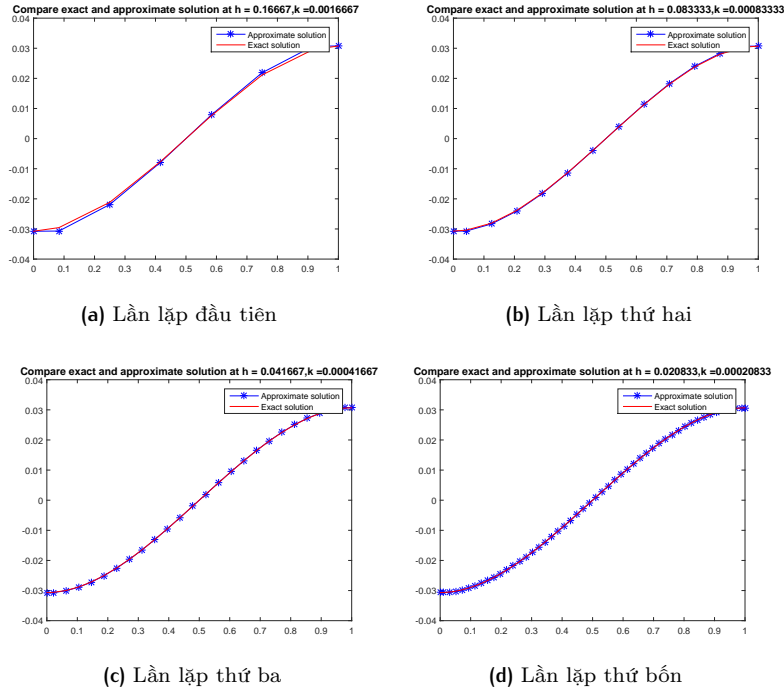


Hình 7: Ví dụ 1 Dirichlet - Backward Method, bậc hội tụ theo x

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ là bậc 1.



Hình 8: Ví dụ 1 Dirichlet - Backward Method, bậc hội tụ theo (x,t)



Hình 9: Ví dụ 1 Dirichlet - Backward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Backward Method.

		Thời gian chạy: 2.552187s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.015062468906261	$1.0e-03 * 0.798784804777040$	0.004688044876956
2	0.007564963469630	$1.0e-03 * 0.198812129127733$	0.001614921583357
3	0.003784399881796	$1.0e-03 * 0.049773136274160$	0.000562728742033
4	0.001891612405600	$1.0e-03 * 0.012519742610171$	0.000197435909413

Bảng 3: Bảng sai số ví dụ 1 biên Dirichlet - Backward Method.

Ví dụ 2: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t = \frac{1}{16}u_{xx}, \quad \forall x \in (0, 1), \quad t \in (0, T) \quad (1.23)$$

với điều kiện đầu

$$u_0(x) = \sin(2\pi x)$$

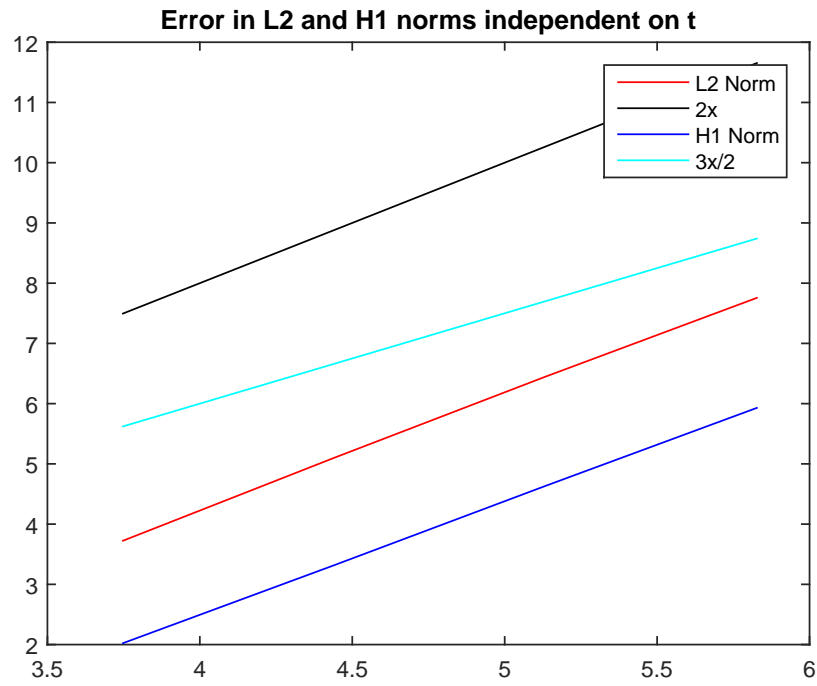
điều kiện biên là:

$$u(0, t) = u(1, t) = 0$$

Nghiệm chính xác là:

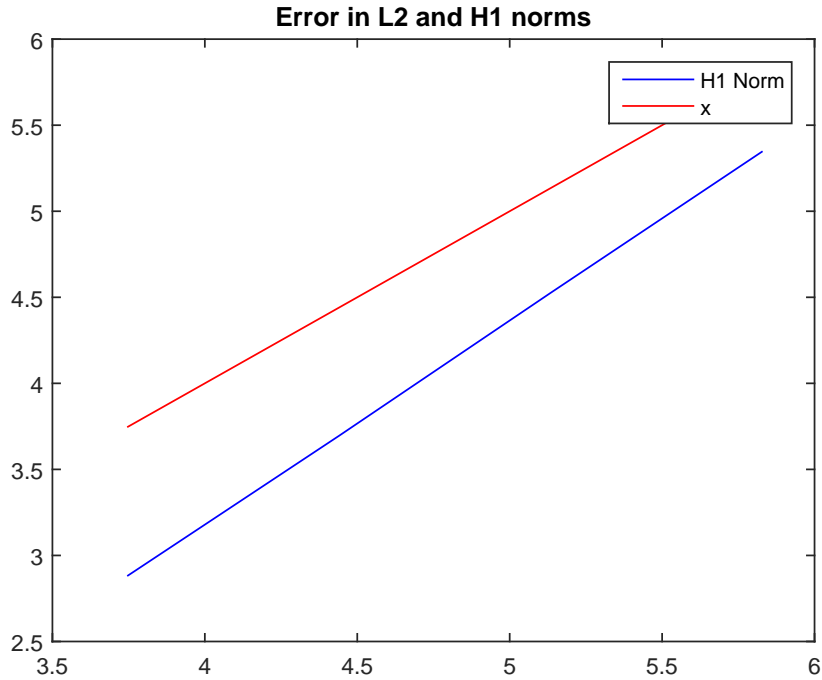
$$u(x, t) = e^{\frac{1}{4}\pi^2 t} \sin(2\pi x)$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

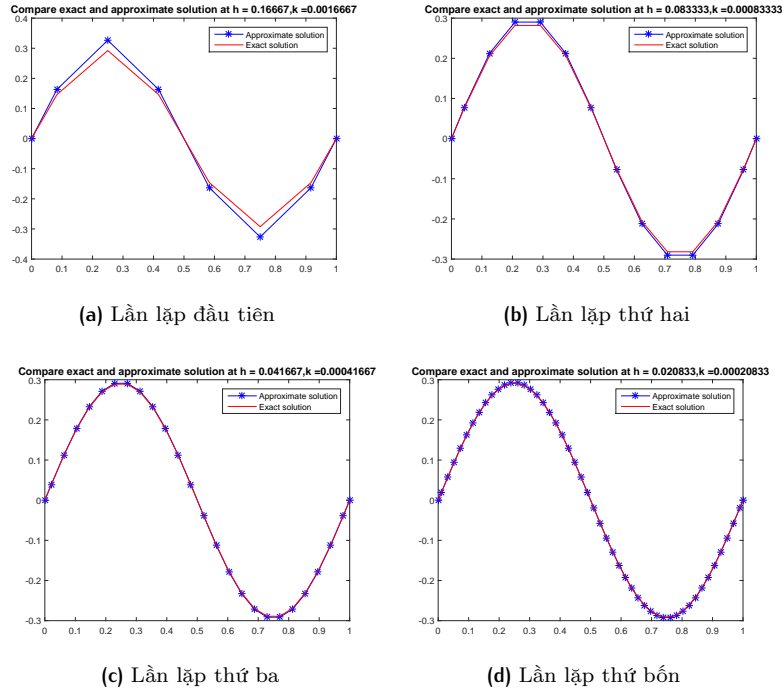


Hình 10: Ví dụ 2 Dirichlet - Backward Method, bậc hội tụ theo x

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ nhỏ hơn bậc 1.



Hình 11: Ví dụ 2 Dirichlet - Backward Method, bậc hội tụ theo (x,t)



Hình 12: Ví dụ 2 Dirichlet - Backward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Backward Method.

		Thời gian chạy: 2.358219s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.055958259088924	0.024142987260616	0.132236587281988
2	0.024821690529608	0.006090782692619	0.036223156884234
3	0.010829815465354	0.001583039610516	0.009709323806773
4	0.004769434129439	0.000428109265784	0.002659822764443

Bảng 4: Bảng sai số ví dụ 2 biên Dirichlet bằng Backward Method.

1.4.3 Crank - Nicolson method

VÍ DỤ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (1.24)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

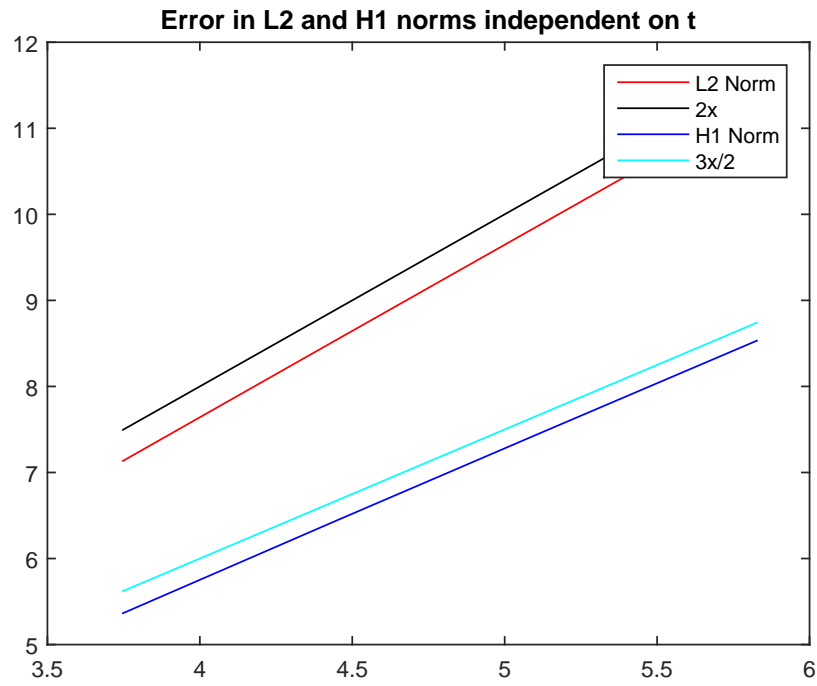
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

Nghiệm chính xác là:

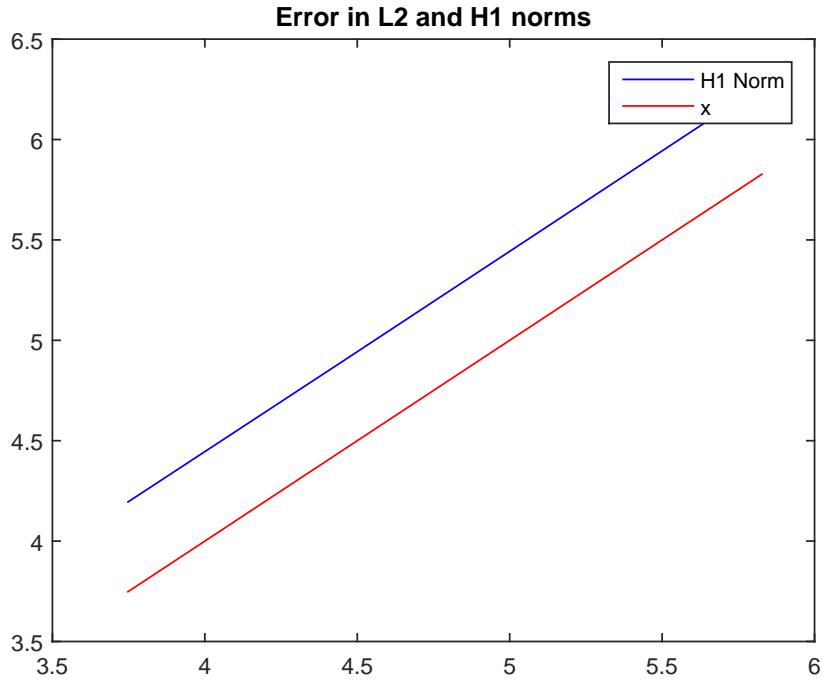
$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

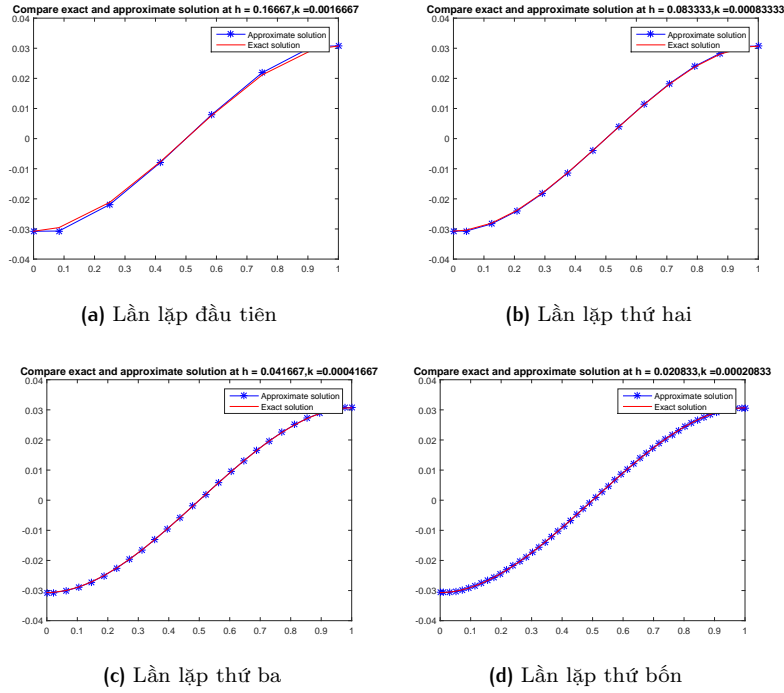


Hình 13: Ví dụ 1 Dirichlet - Crank - Nicolson Method, bậc hội tụ theo x

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ là bậc 1.



Hình 14: Ví dụ 1 Dirichlet - Crank - Nicolson Method, bậc hội tụ theo (x,t)



Hình 15: Ví dụ 1 Dirichlet - Crank - Nicolson method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Crank - Nicolson method.

		Thời gian chạy: 2.547538s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.015073744615328	$1.0e-03 * 0.797000592957527$	0.004680181375254
2	0.007565479144359	$1.0e-03 * 0.198077662341201$	0.001612328817645
3	0.003783304696663	$1.0e-03 * 0.049424611758117$	0.000561829952800
4	0.001890714837808	$1.0e-03 * 0.012347643837993$	0.000197118615877

Bảng 5: Bảng sai số bài toán biên Dirichlet bằng Crank - Nicolson method.

VÍ DỤ 2: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t = \frac{1}{16}u_{xx}, \quad \forall x \in (0, 1), \quad t \in (0, T) \quad (1.25)$$

với điều kiện đầu

$$u_0(x) = \sin(2\pi x)$$

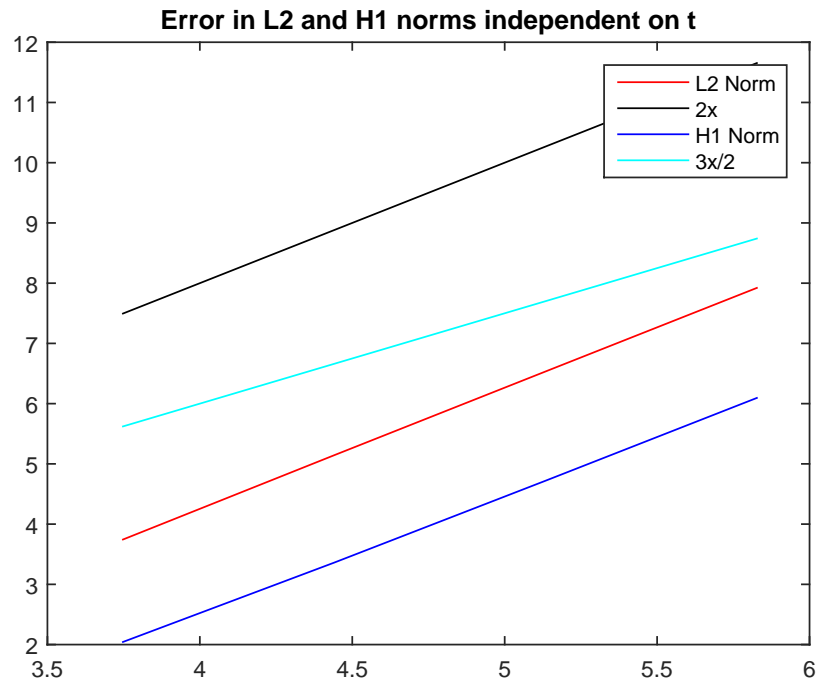
điều kiện biên là:

$$u(0, t) = u(1, t) = 0$$

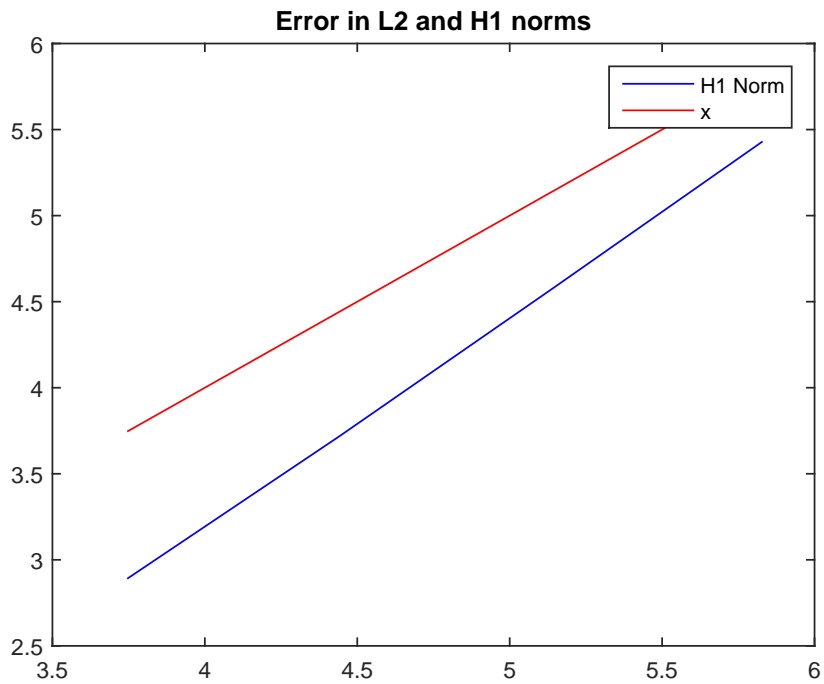
Nghiệm chính xác là:

$$u(x, t) = e^{\frac{1}{4}\pi^2 t} \sin(2\pi x)$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

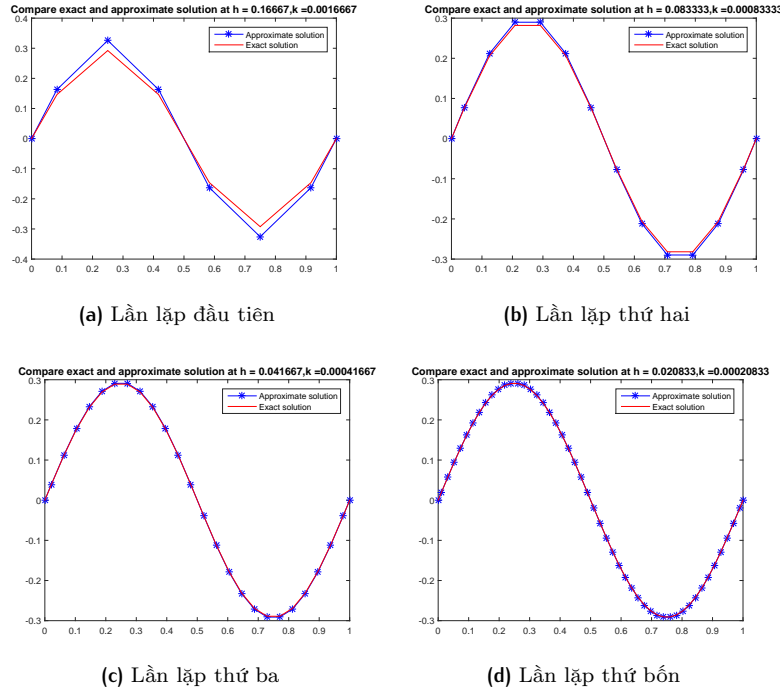


Hình 16: Ví dụ 2 Dirichlet - Crank - Nicolson Method, bậc hội tụ theo x



Hình 17: Ví dụ 2 Dirichlet - Crank - Nicolson Method, bậc hội tụ theo (x,t)

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ nhỏ hơn bậc 1.



Hình 18: Ví dụ 2 Dirichlet - Crank - Nicolson Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng Crank - Nicolson Method.

		Thời gian chạy: 2.411232s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.055405032094089	0.023658949546710	0.129585403536099
2	0.024295987851292	0.005834246474451	0.034697482410827
3	0.010376152594029	0.001453025853506	0.008911904931235
4	0.004391243232513	0.000362886796231	0.002254598623939

Bảng 6: Bảng sai số ví dụ 2 biên Dirichlet bằng Crank - Nicolson Method.

1.4.4 Công thức xấp xỉ tổng quát

Với phương pháp này ta sẽ chọn $\theta = \frac{1}{4}$.

VÍ DỤ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (1.26)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

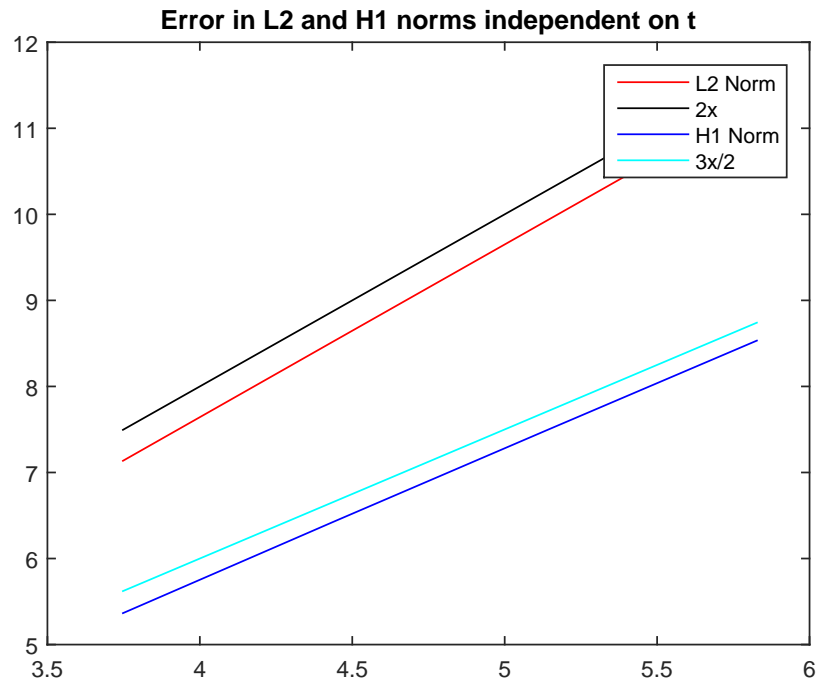
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

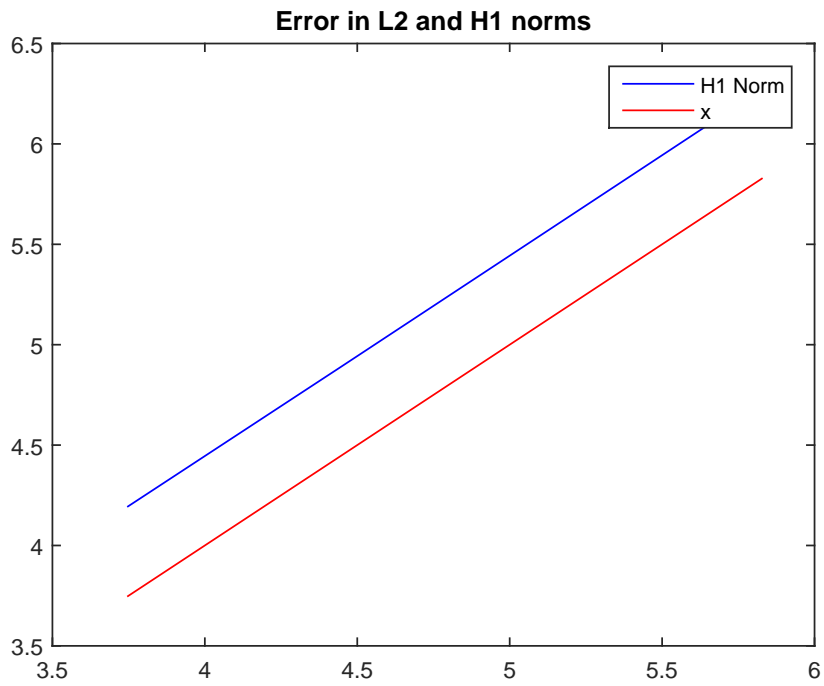
Nghiệm chính xác là:

$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

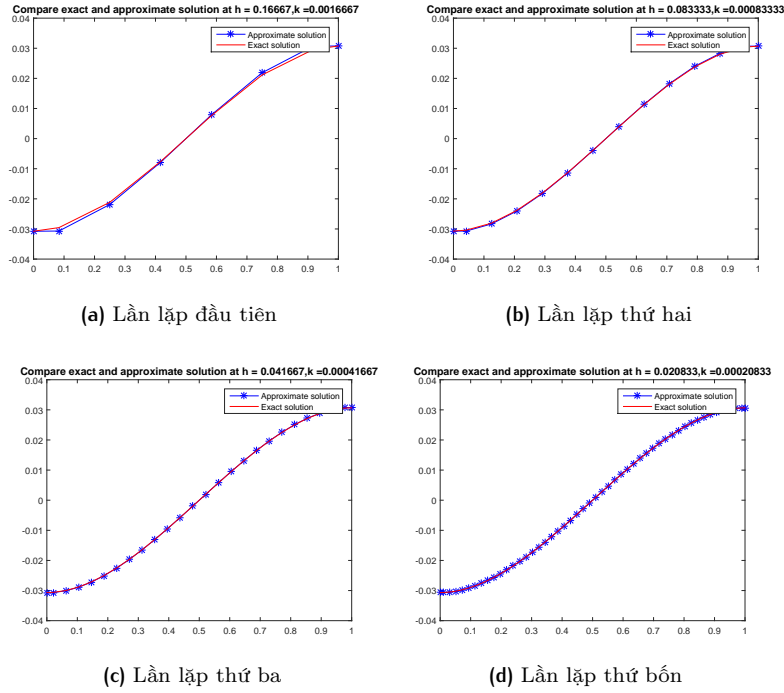


Hình 19: Ví dụ 1 Dirichlet - General Method, bậc hội tụ theo x



Hình 20: Ví dụ 1 Dirichlet - General Method, bậc hội tụ theo (x,t)

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ là bậc 1.



Hình 21: Ví dụ 1 Dirichlet - General method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng General method.

		Thời gian chạy: 3.176595s	
Lần lặp	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.015079414750219	$1.0e-03 * 0.796111094036233$	0.004676263123315
2	0.007565766062878	$1.0e-03 * 0.197711233917538$	0.001611038123775
3	0.003782789160192	$1.0e-03 * 0.049250801036795$	0.000561384443711
4	0.001890313747905	$1.0e-03 * 0.012261962313321$	0.000196962867223

Bảng 7: Bảng sai số bài toán biên Dirichlet bằng General method.

Ví dụ 2: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t = \frac{1}{16}u_{xx}, \quad \forall x \in (0, 1), \quad t \in (0, T) \quad (1.27)$$

với điều kiện đầu

$$u_0(x) = \sin(2\pi x)$$

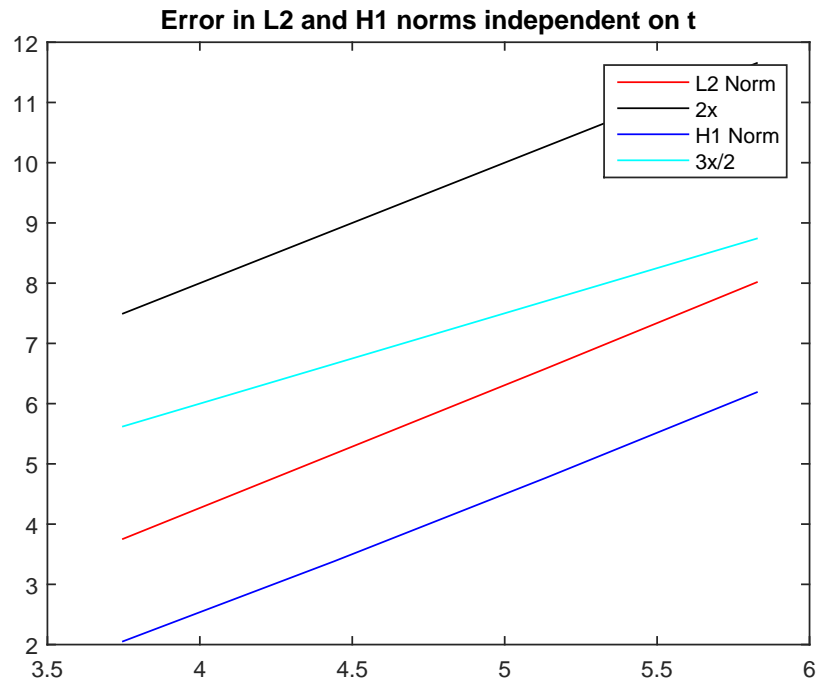
điều kiện biên là:

$$u(0, t) = u(1, t) = 0$$

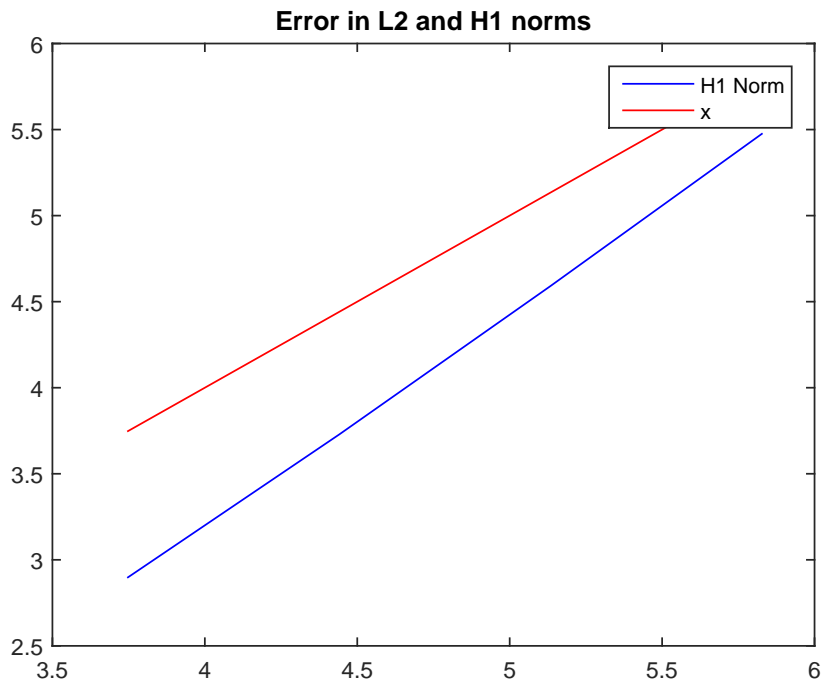
Nghiệm chính xác là:

$$u(x, t) = e^{-\frac{1}{4}\pi^2 t} \sin(2\pi x)$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

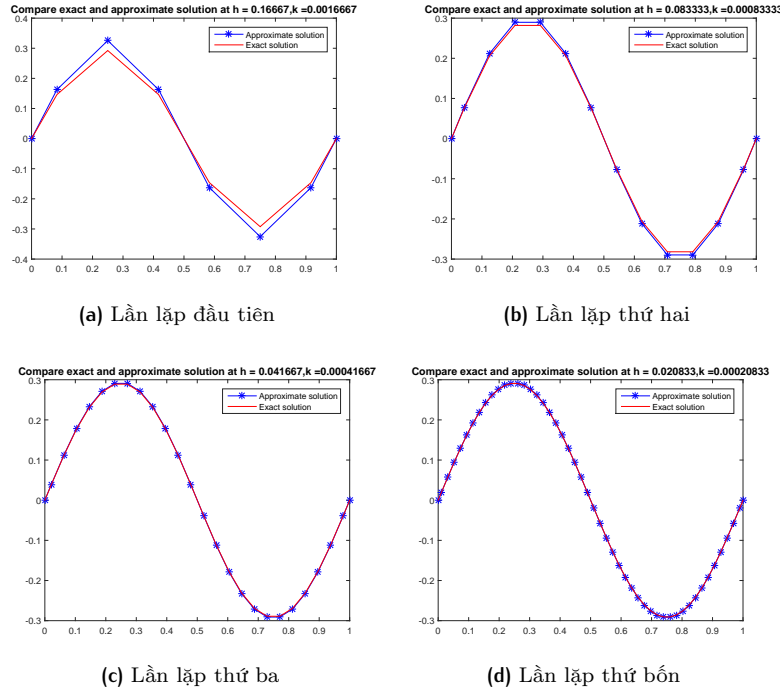


Hình 22: Ví dụ 2 Dirichlet - General Method, bậc hội tụ theo x



Hình 23: Ví dụ 2 Dirichlet - General Method, bậc hội tụ theo (x,t)

Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ nhỏ bậc 1.



Hình 24: Ví dụ 2 Dirichlet - General Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Dirichlet bằng General Method.

Lần lặp	Chuẩn H^1 thời gian	Thời gian chạy: 2.658202s	
		Chuẩn L^2	Chuẩn H^1
1	0.055125837028074	0.023416631425872	0.128258172527341
2	0.024028575491830	0.005705901370825	0.033934187271504
3	0.010141603491887	0.001387999675080	0.008513077120450
4	0.004189316231178	0.000330270736531	0.002051956576667

Bảng 8: Bảng sai số ví dụ 2 biên Dirichlet bằng General Method.

1.5 Kết luận và so sánh các phương pháp

Ta có một số kết luận như sau:

- Bậc hội tụ của cả hai bài toán trong H^1 nếu phụ thuộc vào thời gian đều bé hơn hoặc bằng 1.
- Bậc hội tụ của cả hai bài toán trong H^1 nếu không phụ thuộc vào thời gian đều bằng $3/2$ còn trong L^2 sẽ là 2.
- Thời gian chạy code khá nhanh, hầu hết chỉ dưới 3 giây.
- Kết quả xấp xỉ tốt và độ bền vững của hội tụ cũng tốt (cả 2 ví dụ đều cho kết quả xấp xỉ khá giống nhau).

Qua các bảng đánh giá thời gian tính toán và sai số của ví dụ 1 (Bảng 1, Bảng 3, Bảng 5, Bảng 7) ta thấy:

- Thời gian chạy kết quả nhanh nhất là Forward Method chỉ với 2.022s, trong khi nếu ta chọn $\theta = 1/4$ cho trường hợp tổng quát thì kết quả chạy ra hơn 3s. Hai phương pháp còn lại chạy gần như nhau.
- Về kết quả sai số thì Forward cũng là phương pháp tính chính xác nhất cũng với phương pháp tổng quát. Hai phương pháp còn lại cho kết quả khá giống nhau.

Ví dụ 2 (Bảng 2, Bảng 4, Bảng 6, Bảng 8) ta thấy:

- Thời gian chạy kết quả nhanh nhất là Forward Method chỉ với 1.847s, trong khi ba phương pháp còn lại chạy gần như nhau.
- Về kết quả sai số thì Forward vẫn là phương pháp tốt nhất.

Vậy có thể thấy trong 4 phương pháp làm ở hai ví dụ trên, Forward Method là phương pháp tốt nhất và cũng đảm bảo tính bền vững của hội tụ.

2 BÀI TOÁN VỚI ĐIỀU KIỆN BIÊN $\frac{\partial u(a,t)}{\partial x} = \psi_a(t)$, $\frac{\partial u(b,t)}{\partial x} = \psi_b(t)$

2.1 Giới thiệu bài toán

Xét phương trình khuếch tán:

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2} + f(x), \quad t > 0, \quad a < x < b \quad (2.1)$$

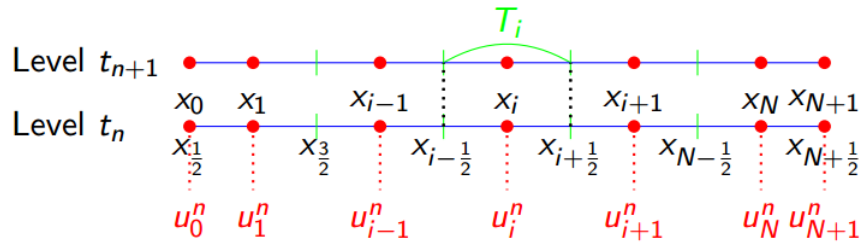
với κ là hệ số khuếch tán. Điều kiện đầu là:

$$u(x, 0) = u_0(x) \quad (2.2)$$

Điều kiện biên là:

$$\frac{\partial u(a,t)}{\partial x} = \psi_a(t), \quad \frac{\partial u(b,t)}{\partial x} = \psi_b(t) \quad (2.3)$$

2.2 Chia lưới



Chúng ta chọn $N + 1$ điểm $\{x_{i+1/2}\}_{i \in \overline{0, N}}$ trên $[0, 1]$ thỏa mãn:

$$a = x_{1/2} < x_{3/2} < \dots < x_{N-1/2} < x_{N+1/2} = b$$

Ta đặt $T_i = [x_{i-1/2}, x_{i+1/2}]$, $h = \max_{i \in \overline{1, N}} \{|T_i|\}$

$$x_0 = a, \quad x_{N+1} = b, \quad x_i \in T_i, \quad \forall i \in \overline{1, N}$$

Chúng ta gọi $\{T_i\}_{i \in \overline{1, N}}$ là control volume và $\{x_i\}_{i \in \overline{0, N+1}}$ là control point.

Chúng ta cũng chia đoạn $[0, T]$ thành $N_k + 1$ đoạn con với chiều dài cố định là k và kí hiệu $t_n = nk$. Kí hiệu u_i^n là giá trị $u(x_i, t_n)$.

2.3 Phân rã bài toán

Lấy tích phân (2.1) trên T_i ta được:

$$\frac{1}{|T_i|} \int_{T_i} u_t(x, t) dx + \frac{\kappa}{|T_i|} \int_{T_i} -u_{xx}(x, t) dx = \frac{1}{|T_i|} \int_{T_i} f(x, t) dx \quad (2.4)$$

Áp dụng công thức Green ta có:

$$\frac{\kappa}{|T_i|} \int_{T_i} -u_{xx}(x, t) dx = \kappa \frac{-u_x(x_{i+1/2}, t) + u_x(x_{i-1/2}, t)}{|T_i|} \quad (2.5)$$

Đặt

$$f_i(t) = \frac{1}{|T_i|} \int_{T_i} f(x, t) dx$$

là giá trị trung bình của f trên T_i .

Khi đó (2.4) ta có thành:

$$\frac{du_i}{dt}(t) + \kappa \frac{-u_x(x_{i+1/2}, t) + u_x(x_{i-1/2}, t)}{|T_i|} = f_i(t) \quad (2.6)$$

Áp dụng khai triển Taylor ta có:

$$\begin{aligned} u(x_{i+1}, t) - u(x_i, t) &= (x_{i+1} - x_i)u_x(x_{i+1/2}, t) \\ &+ \left((x_{i+1} - x_{i+1/2})^2 - (x_i - x_{i+1/2})^2 \right) \frac{u_{xx}(x_{i+1/2}, t)}{2!} + O(h^3) \end{aligned}$$

Ta sẽ xét trường hợp $x_{i+1/2}$ là trung điểm của $[x_i, x_{i+1}]$, khi đó:

$$u_x(x_{i+1/2}, t) = \frac{u(x_{i+1}, t) - u(x_i, t)}{x_{i+1} - x_i} + O(h^2)$$

Khi đó ta có (2.6) thành:

$$\begin{aligned} \frac{du_i}{dt}(t) - \kappa \frac{u_{i-1}(t)}{(x_i - x_{i-1})|T_i|} + \kappa \left(\frac{1}{(x_{i+1} - x_i)|T_i|} + \frac{1}{(x_i - x_{i-1})|T_i|} \right) u_i(t) \\ - \kappa \frac{u_{i+1}(t)}{(x_{i+1} - x_i)|T_i|} = f_i(t) \end{aligned}$$

Đặt

$$\begin{aligned} \alpha_i &= \frac{\kappa}{(x_i - x_{i-1})|T_i|} \\ \beta_i &= \frac{-\kappa}{(x_{i+1} - x_i)|T_i|} + \frac{-\kappa}{(x_i - x_{i-1})|T_i|} \\ \alpha_i &= \frac{\kappa}{(x_{i+1} - x_i)|T_i|} \end{aligned}$$

Khi đó ta có:

$$\frac{du_i}{dt}(t) - \alpha_i u_{i-1}(t) - \beta_i u_i(t) - \gamma_i u_{i+1}(t) = f_i(t), \quad \forall i \in \overline{1, N} \quad (2.7)$$

và phân rã điều kiện biên như sau:

$$\frac{u_1(t) - u_0(t)}{x_1 - x_0} = \psi_a(t) \Leftrightarrow u_0(t) = u_1(t) - (x_1 - x_0)\psi_a(t) \quad (2.8)$$

$$\frac{u_{N+1}(t) - u_N(t)}{x_{N+1} - x_N} = \psi_b(t) \Leftrightarrow u_{N+1}(t) = u_N(t) + (x_{N+1} - x_N)\psi_b(t) \quad (2.9)$$

Vậy ta có hệ:

$$\begin{cases} \frac{du_i}{dt}(t) = \alpha_i u_{i-1}(t) + \beta_i u_i(t) + \gamma_i u_{i+1}(t) + f_i(t), \quad \forall i \in \overline{2, N-1} \\ \frac{du_1}{dt}(t) = -\gamma_1 u_1(t) + \gamma_1 u_2(t) + f_1(t) - \alpha_1 (x_1 - a)\psi_a(t) \\ \frac{du_N}{dt}(t) = \alpha_N u_{N-1}(t) - \alpha_N u_N(t) + f_N(t) + \gamma_N (b - x_N)\psi_b(t) \end{cases} \quad (2.10)$$

Tuy nhiên chứng minh được hệ vô (số) nghiệm. Thay (2.8) và (2.9) vào ta có:

Hay ta có hệ ODE như sau:

$$\frac{dU}{dt} = AU(t) + F(t) \quad (2.11)$$

với

$$A = \begin{cases} \begin{matrix} -\gamma_1 & \gamma_1 & & & \\ \alpha_2 & \beta_2 & \gamma_2 & & \\ & \alpha_3 & \beta_3 & \gamma_3 & \\ & & & & \dots\dots\dots \\ & & \alpha_{N-1} & \beta_{N-1} & \gamma_{N-1} \\ & & & \alpha_N & -\alpha_N \end{matrix} & \begin{matrix} = f_1(t) - \alpha_1(x_1 - a)\phi_a(t) \\ = f_2(t) \\ = f_3(t) \\ \\ = f_{N-1}(t) \\ = f_N(t) + \gamma_N(b - x_N)\phi_b(t) \end{matrix} \end{cases} \quad (2.12)$$

Đây là phần khác biệt giữa biên Neumann và biên Dirichlet nên bài viết sẽ trình bày đoạn code Matlab cho phần này:

```

1  %% Create for computing
2  A = zeros(N,N); % Matrix A
3  u = zeros(N,T); % Matrix value of u
4  u0 = zeros(N,1); % u(x,0)
5  psi_a = zeros(T); %psi_(a,t)
6  psi_b = zeros(T); %psi_(b,t)
7
8  %% Input iterior
9  for i_iterx=1:N
10     u0(i_iterx) = u_ex(x_cp(i_iterx+1),0); % u_0(x,0)
11 end
12
13 for i_itert = 1:T
14     psi_a(i_itert) = ps(a,t(i_itert)); %psi_a(t)
15 end
16
17 for i_itert = 1:T
18     psi_b(i_itert) = ps(b,t(i_itert)); % psi_b(t)
19 end
20
21 %% Matrix of function f
22 F = zeros(N,T);
23 for i_itert =1:T
24     for i_iterx =1:N
25         if(i_iterx ==1)
26             alpha = ...
27                 kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+1)-x_cp(i_iterx)));
28             F(i_iterx,i_itert) = ...
29                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2 ...
30                 - alpha*(x_cp(i_iterx+1)-x_cp(i_iterx))*psi_a(i_itert);
31         elseif (i_iterx ==N)
32             gamma = ...
33                 kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+2)-x_cp(i_iterx+1)));
34             F(i_iterx,i_itert) = ...
35                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2 ...
36                 + gamma*(x_cp(i_iterx+2)-x_cp(i_iterx+1))*psi_b(i_itert);
37         else
38             F(i_iterx,i_itert) = ...
39                 (fun(x(i_iterx+1),t(i_itert))+fun(x(i_iterx),t(i_itert)))/2;
40         end
41     end
42 end
43
44 %% Matrix A
45 for i_iterx = 1:N
46     alpha = ...
47         kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+1)-x_cp(i_iterx)));
48     gamma = ...
49         kappa/((x(i_iterx+1)-x(i_iterx))*(x_cp(i_iterx+2)-x_cp(i_iterx+1)));
50     if(i_iterx ==1)
51         A(i_iterx,i_iterx) = - gamma;
52         A(i_iterx,i_iterx+1) = gamma;
53     elseif(i_iterx == N)
54         A(i_iterx,i_iterx) = -alpha;
55         A(i_iterx,i_iterx-1) = alpha;
56     end
57 end

```

```

48
49     else
50         A(i_iterx,i_iterx) = -alpha - gamma;
51         A(i_iterx,i_iterx-1) = alpha;
52         A(i_iterx,i_iterx+1) = gamma;
53     end
54 end

```

Để giải (2.11) ta sẽ sử dụng một số phương pháp sau:

2.3.1 Forward Euler method - Explicit method

Ta có công thức Forward Euler method như sau:

$$U^{n+1} = U^n + k(AU^n + F^n), \quad U^n \approx U(nk) \quad (2.13)$$

với $\Delta t = k$, hay

$$U^{n+1} = (I + kA)U^n + kF^n, \quad U^n \approx U(nk) \quad (2.14)$$

"Time step restriction" là :

$$r = \frac{k}{h^2} \leq \frac{1}{2}$$

2.3.2 Backward Euler method

Ta có công thức Backward Euler method như sau:

$$U^{n+1} = U^n + kAU^{n+1} + kF^{n+1} \quad (2.15)$$

với $\Delta t = k$, hay

$$U^{n+1} = (I - kA)^{-1}(U^n + kF^{n+1}) \quad (2.16)$$

2.3.3 Crank - Nicolson method

Ta có công thức θ method như sau:

$$U^{n+1} = U^n + \frac{kA}{2} [U^{n+1} + U^n] + \frac{k}{2} [F^{n+1} + F^n] \quad (2.17)$$

với $\Delta t = k$, hay

$$U^{n+1} = (1 - 1/2kA)^{-1}((I + 1/2kA)U^n + kF_{\theta}^n) \quad (2.18)$$

$$\text{với } F_{\theta}^n = \frac{1}{2}(F^{n+1} + F^n)$$

2.3.4 Công thức xấp xỉ tổng quát

Ta có công thức tổng quát θ method như sau:

$$U^{n+1} = U^n + kA [\theta U^{n+1} + (1 - \theta)U^n] + k [\theta F^{n+1} + (1 - \theta)F^n] \quad (2.19)$$

với $\Delta t = k$, hay

$$U^{n+1} = \frac{(I + (1 - \theta)kA)U^n + kF_{\theta}^n}{1 - \theta kA} \quad (2.20)$$

$$\text{với } F_{\theta}^n = \theta F^{n+1} + (1 - \theta)F^n$$

2.4 Xuất kết quả, so sánh và biểu diễn bậc hội tụ

2.4.1 Forward Euler method - Explicit method

VÍ DỤ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (2.21)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

điều kiện biên là:

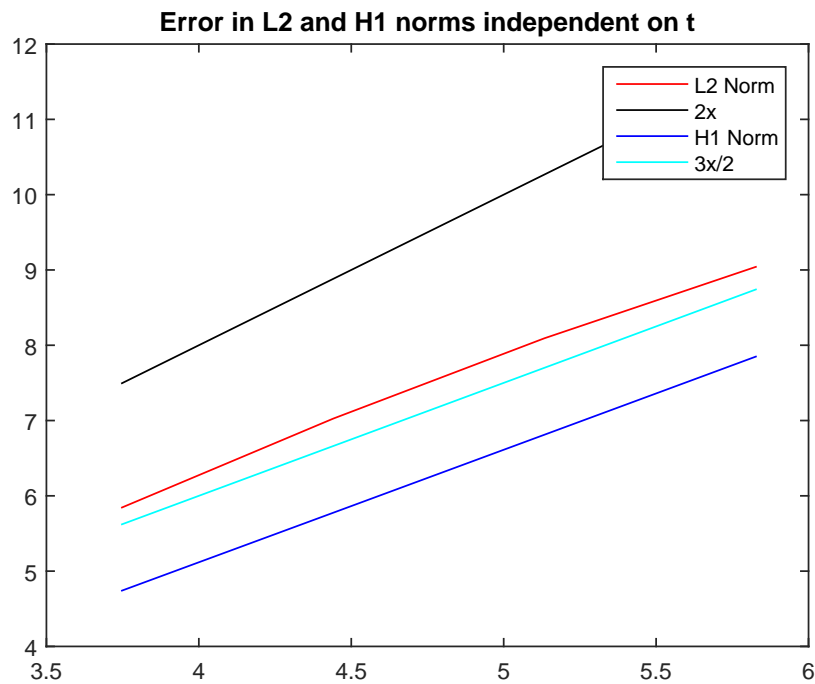
$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

Nghiệm chính xác là:

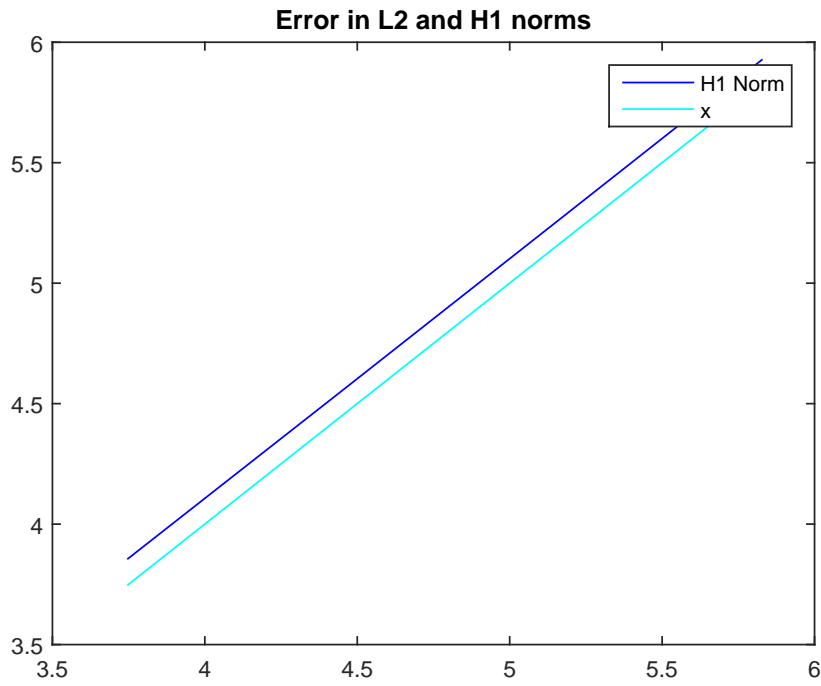
$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Điều kiện ổn định đã được xét như trên phần giới thiệu phương pháp.

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

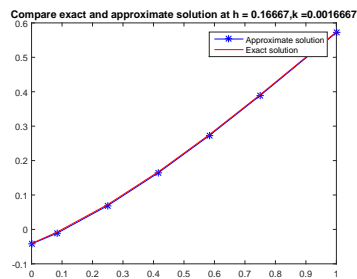


Hình 25: Ví dụ 1 Neumann - Forward Method, bậc hội tụ theo x

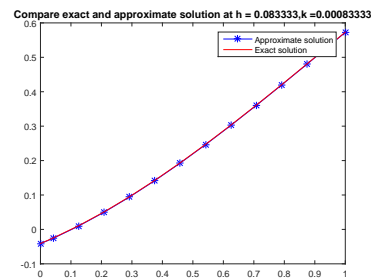


Hình 26: Ví dụ 1 Neumann - Forward Method, bậc hội tụ theo (x,t)

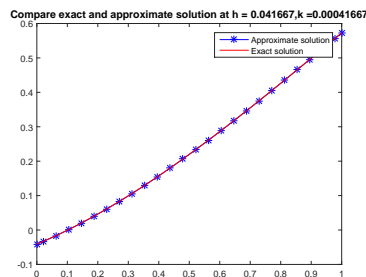
Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 sẽ là 1.



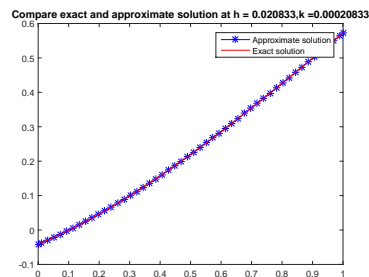
(a) Lần lặp đầu tiên



(b) Lần lặp thứ hai



(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

Hình 27: Ví dụ 1 Neumann - Forward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Neumann bằng Forward Method.

		Thời gian chạy: 2.093670s	
Lần lập	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.021143558142583	0.002894282177586	0.008728345512461
2	0.010624627748370	0.000888751494074	0.003102496805273
3	0.005325768137096	0.000305587690349	0.001100246394562
4	0.002666871270713	0.000118427243591	0.000389988817722

Bảng 9: Bảng sai số ví dụ 1 biên Neumann - Forward Method.

2.4.2 Backward Euler method - Implicit Method

ví dụ : Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (2.22)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

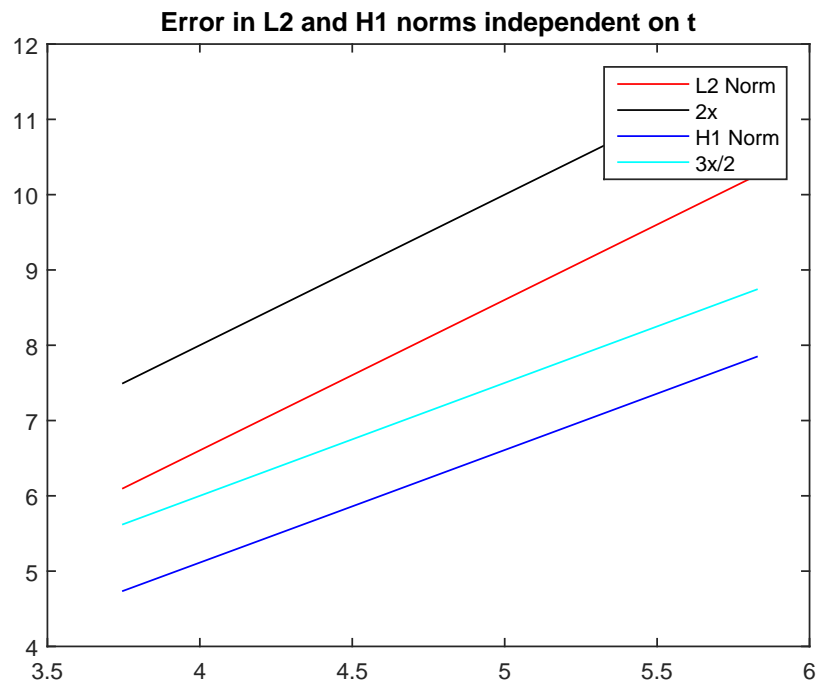
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

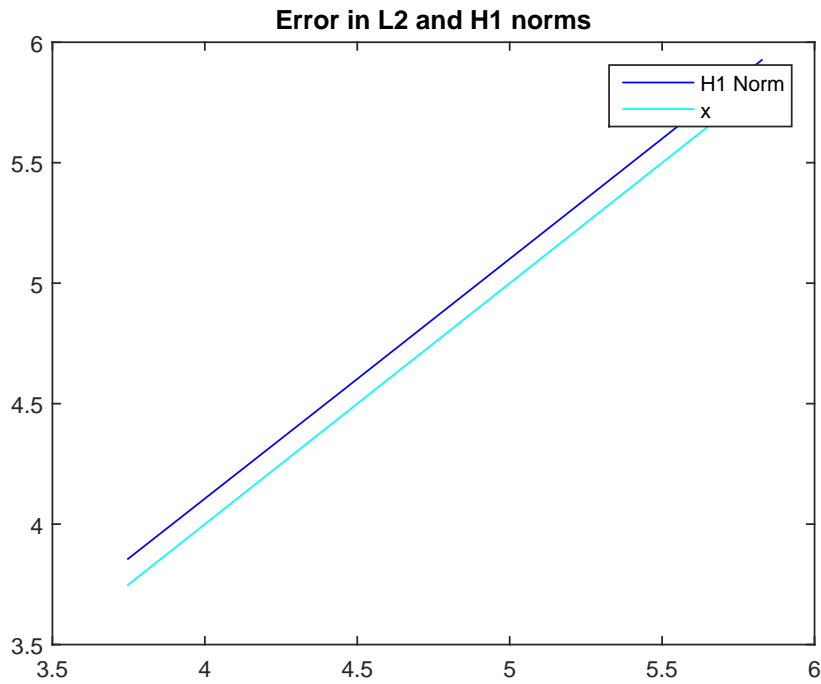
Nghiệm chính xác là:

$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

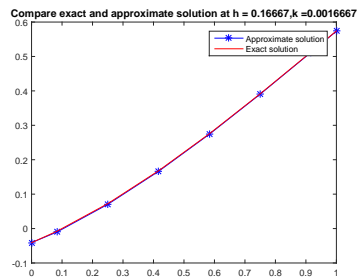


Hình 28: Ví dụ 1 Neumann - Backward Method, bậc hội tụ theo x

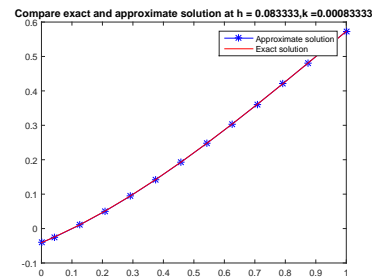


Hình 29: Ví dụ 1 Neumann - Backward Method, bậc hội tụ theo (x,t)

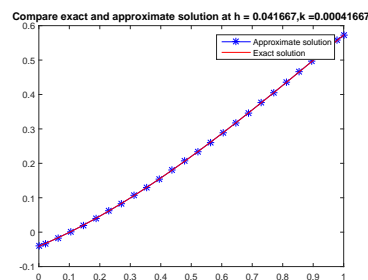
Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 là bậc 1.



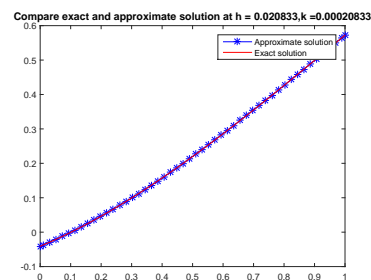
(a) Lần lặp đầu tiên



(b) Lần lặp thứ hai



(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

Hình 30: Ví dụ 1 Neumann - Backward Method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Neumann bằng Backward Method.

		Thời gian chạy: 2.583929s	
Lần lập	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.021189217691424	0.001617083207546	0.008809411309052
2	0.010647935063282	0.000264508817209	0.003131157948552
3	0.005337439666518	0.000057948343380	0.001110353271980
4	0.002672694719994	0.000052609263824	0.000393552094862

Bảng 10: Bảng sai số ví dụ 1 biên Neumann - Backward Method.

2.4.3 Crank - Nicolson method

Với phương pháp này ta sẽ chọn $\theta = \frac{1}{5}$.

ví dụ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (2.23)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

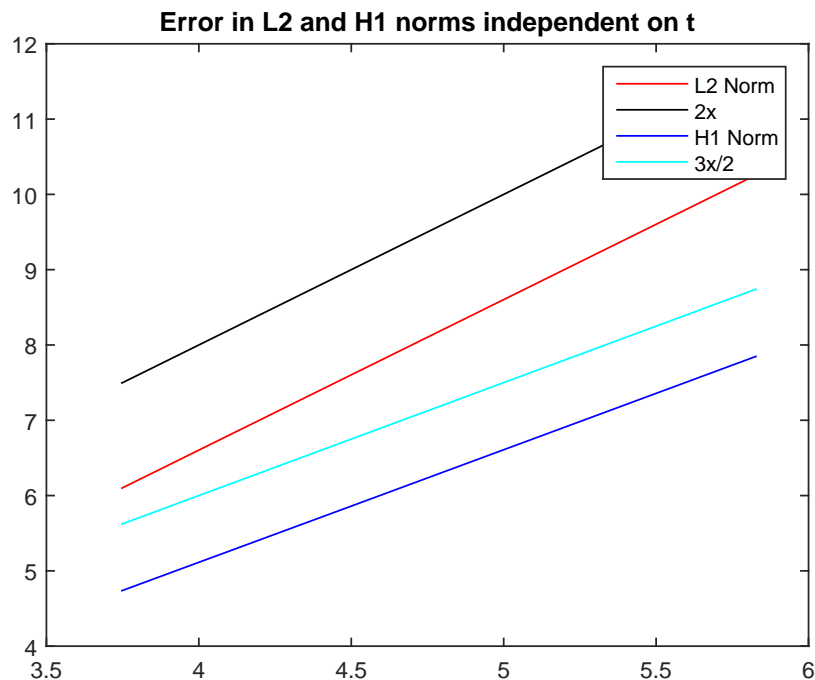
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

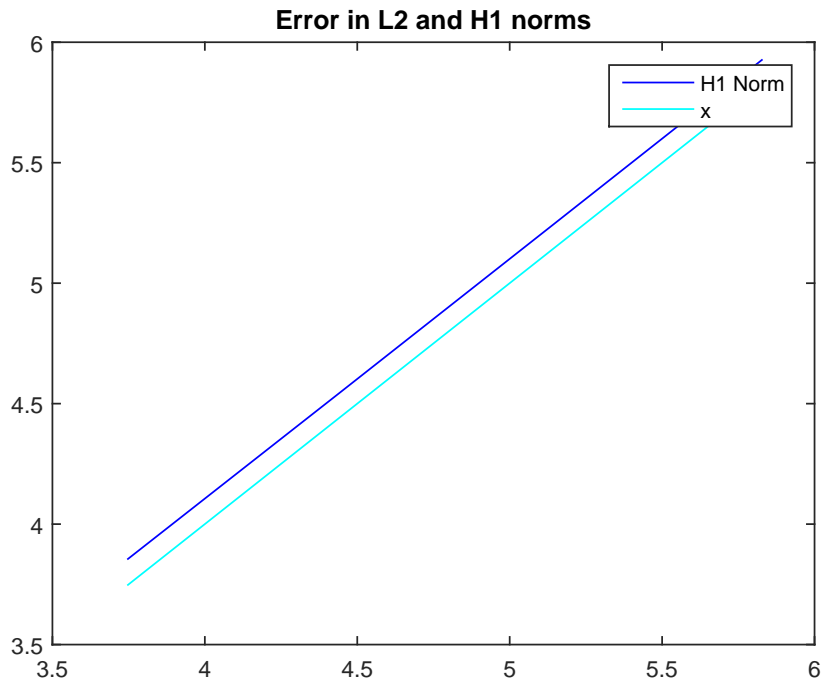
Nghiệm chính xác là:

$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

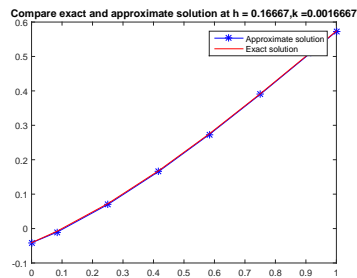


Hình 31: Ví dụ 1 Neumann - Crank - Nicolson Method, bậc hội tụ theo x

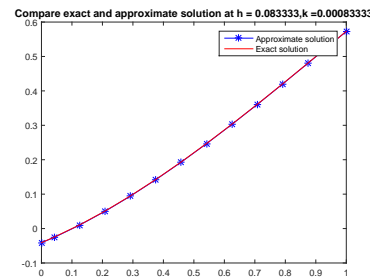


Hình 32: Ví dụ 1 Neumann - Crank - Nicolson Method, bậc hội tụ theo (x,t)

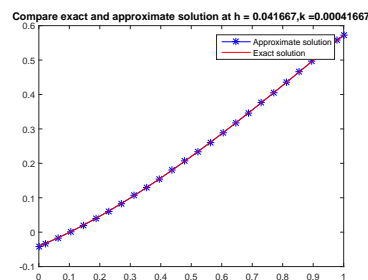
Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 là bậc 1.



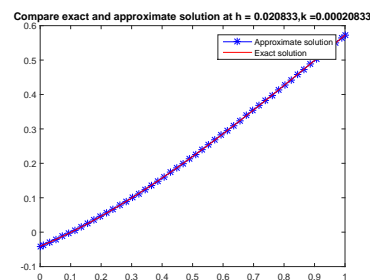
(a) Lần lặp đầu tiên



(b) Lần lặp thứ hai



(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

Hình 33: Ví dụ 1 Neumann - Crank - Nicolson method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Neumann bằng Crank - Nicolson method.

		Thời gian chạy: 2.651012s	
Lần lập	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.021164712358915	0.002243995339823	0.008765760589604
2	0.010634523005057	0.000561484397920	0.003114637723412
3	0.005329825884909	0.000140431131325	0.001103752370694
4	0.002668001373565	0.000035115279172	0.000390676068515

Bảng 11: Bảng sai số bài toán biên Neumann bằng Crank - Nicolson method.

2.4.4 Công thức xấp xỉ tổng quát

ví dụ 1: Xét bài toán cụ thể trên $[0, 1] \times [0, 1]$:

$$u_t - u_{xx} = \left(\frac{2x^3}{3} - x^2 + 2x - \frac{5}{6} \right) e^{-2t} \quad (2.24)$$

với điều kiện đầu

$$u_0(x) = -\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12}$$

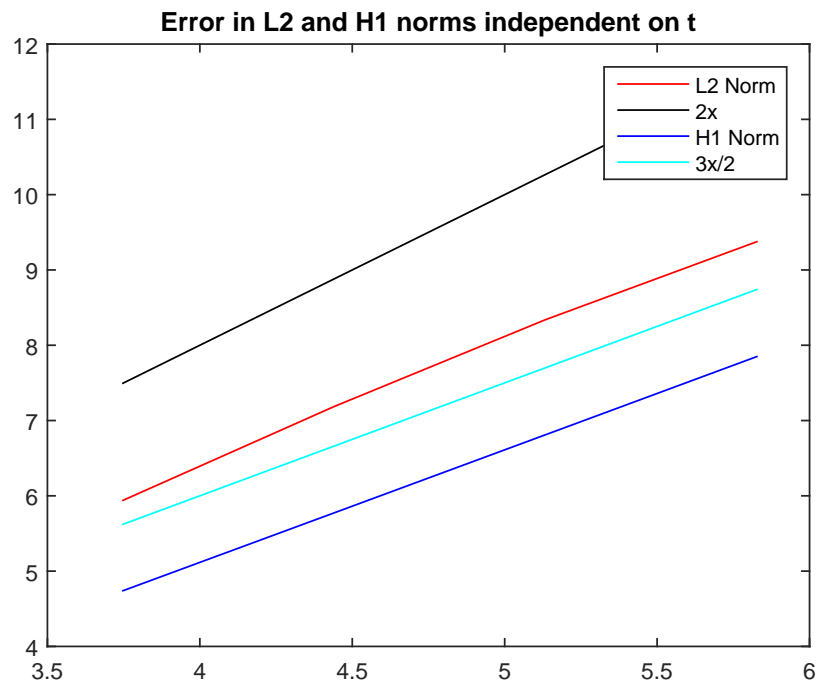
điều kiện biên là:

$$u(0, t) = -\frac{e^{-2t}}{12}, \quad u(1, t) = \frac{e^{-2t}}{12}$$

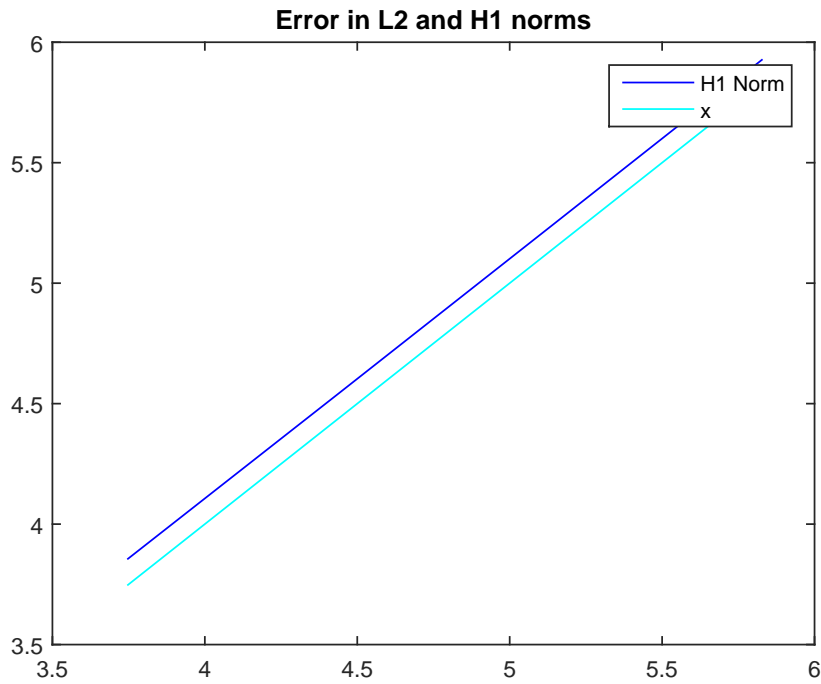
Nghiệm chính xác là:

$$u(x, t) = \left(-\frac{x^3}{3} + \frac{x^2}{2} - \frac{1}{12} \right) e^{-2t}$$

Khi đó ta có đồ thị biểu diễn bậc hội tụ:

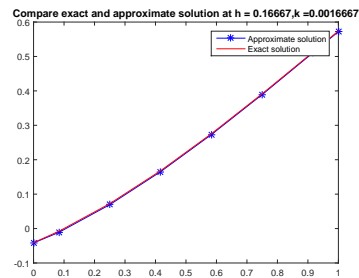


Hình 34: Ví dụ 1 Neumann - General Method, bậc hội tụ theo x

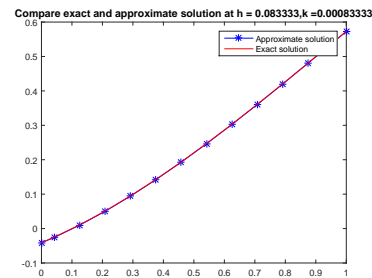


Hình 35: Ví dụ 1 Neumann - General Method, bậc hội tụ theo (x,t)

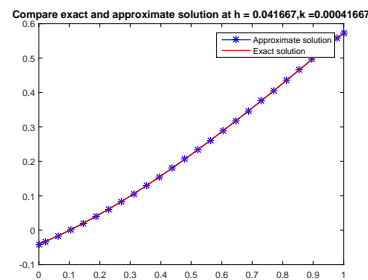
Hai hình trên cho thấy nếu không phụ thuộc thời gian thì sai số trong L^2 sẽ là bậc 2 và sai số trong H^1 sẽ là bậc $3/2$. Còn nếu phụ thuộc vào thời gian thì bậc hội tụ của H^1 là bậc 1.



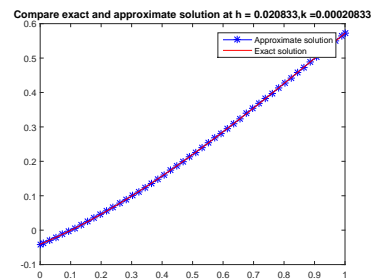
(a) Lần lặp đầu tiên



(b) Lần lặp thứ hai



(c) Lần lặp thứ ba



(d) Lần lặp thứ bốn

Hình 36: Ví dụ 1 Neumann - General method, xấp xỉ

Dưới đây là kết quả tính xấp xỉ sau các lần chia lưới mịn hơn của bài toán biên Neumann bằng General method.

		Thời gian chạy: 2.789961s	
Lần lập	Chuẩn H^1 thời gian	Chuẩn L^2	Chuẩn H^1
1	0.021151613559046	0.002632458344813	0.008765760589604
2	0.010628160652582	0.000756648089385	0.003114637723412
3	0.005326961942254	0.000238836226249	0.001103752370694
4	0.002666893627014	0.000084788332373	0.000390676068515

Bảng 12: Bảng sai số bài toán biên Neumann bằng General method.

2.5 Kết luận và so sánh các phương pháp

Ta có một số kết luận như sau:

- Bậc hội tụ của cả hai bài toán trong H^1 nếu phụ thuộc vào thời gian đều bằng 1.
- Bậc hội tụ của cả hai bài toán trong H^1 nếu không phụ thuộc vào thời gian đều bằng $3/2$ còn trong L^2 sẽ là 2.
- Thời gian chạy code khá nhanh, hầu hết chỉ dưới 3 giây.
- Kết quả xấp xỉ tốt và độ bền vững của hội tụ cũng tốt (cả 2 ví dụ đều cho kết quả xấp xỉ khá giống nhau).

Qua các bảng đánh giá thời gian tính toán và sai số của ví dụ (Bảng 9, Bảng 10, Bảng 11, Bảng 12) ta thấy:

- Thời gian chạy kết quả nhanh nhất là Forward Method chỉ với 2.09s, trong khi nếu ta chọn $\theta = 1/4$ cho trường hợp tổng quát thì kết quả chạy ra gần 3s. Hai phương pháp còn lại chạy gần như nhau.
- Về kết quả sai số thì Crank - Nicolson Method là phương pháp tính chính xác nhất. Hai phương pháp còn lại cho kết quả khá giống nhau và cũng gần tốt bằng Crank - Nicolson Method.

TÀI LIỆU

- [1] Finite Volume Methods in 1D, Parabolic Le Anh Ha (Lecture Note). Khoa Toán - Tin, Đại học Khoa học tự nhiên TP HCM, 2017.
- [2] Finite Volume Methods in 1D, Elliptic Le Anh Ha (Lecture Note). Khoa Toán - Tin, Đại học Khoa học tự nhiên TP HCM, 2017.
- [3] Finite Volume Methods Robert Eymard, Thierry Gallowet, Rapphaele herbin. [On the electrodynamics of moving bodies]. Annalen der Physik, 322(10):891–921, 1905.
- [4] Introduction to Numerical Integration
<http://www.ece.utah.edu/ece6340/LECTURES/Jan30/NumericalIntegration.pdf>
- [5] 1D Integration using the Trapezoidal and Simpson Rules
<http://mathfaculty.fullerton.edu/mathews/n2003/SimpsonsRule2DMod.html>