**Báo Cáo Bài Tập Lớn**

**Sử dụng phương trình Laplace 2D-Jacobi để tính tiềm năng điện trên mặt phẳng 2D**

**Giáo viên hướng dẫn :** TS. Vũ Văn Thiệu

**Bộ môn :** Tính toán khoa học

**Nhóm 4:**

Đỗ Hoàng Minh Hiếu 20225837

Trịnh Minh Đạt 20225701

Trần Phúc Sơn 20225666

Hoàng Trường Giang 20225710

Ninh Lê Gia Bảo 20225693

Đặng Huy Hoàng 20225843

Mục lục

[Tóm tắt 2](#_Toc168660700)

[Mô tả bài toán 3](#_Toc168660701)

[Thuật toán 5](#_Toc168660702)

[Mã nguồn : 9](#_Toc168660703)

[Kết quả 14](#_Toc168660704)

[Nhận xét 16](#_Toc168660705)

# Tóm tắt

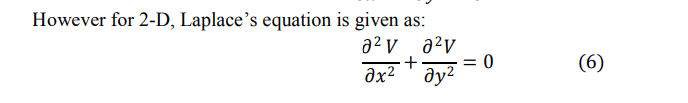
Nhiều ứng dụng trong Khoa học và Kỹ thuật đã nhận thấy phương trình Laplace rất hữu ích. Việc giải phương trình này bằng số có thể hữu ích trong việc tìm phân bố nhiệt độ trong một vật thể rắn, phân bố tiềm năng điện trong 1 khu vực. Trong bài tập này, phương pháp sai phân hữu hạn (Finite Difference Method - FDM) đã được sử dụng để rời rạc hóa phương trình Laplace để chứng minh được nghiệm số Numericals và sử dụng phương pháp lặp Jacobi để tính tiềm năng điện trên 1 hình chữ nhật. Kết quả thu được cho thấy sự phân bố tiềm năng điện trên 1 mặt phẳng 2D hình chữ nhật và sai số sau mỗi lần lặp sử dụng phương pháp lặp jacobi khi sử dụng sai phân hữu hạn để chia mặt phẳng 2D thành lưới các điểm

# Mô tả bài toán

Phương trình Laplace là một phương trình quan trọng liên quan đến nhiều ứng dụng trong lĩnh vực Khoa học và Kỹ thuật. Các ứng dụng này không chỉ giới hạn trong tĩnh điện học mà còn liên quan đến động lực học chất lỏng và dẫn nhiệt ở trạng thái ổn định. Đây là một phương trình vi phân riêng (PDE) bậc hai có tính chất elip . Các nghiệm của phương trình Laplace thường được biết đến là các hàm điều hòa như có thể thấy trong và rất hữu ích trong khoa học và kỹ thuật như đã đề cập trước đó. Phương trình Laplace có cả nghiệm phân tích và nghiệm số. Nghiệm số của phương trình Laplace được tìm bằng các phương pháp khác nhau áp dụng cho nhiều PDE tuyến tính. Bao gồm phương pháp sai phân hữu hạn (FDM),được đề cập trong đề tài này. Cách tiếp cận của nhóm để giải phương trình Laplace tập trung vào FDM. Ở đây, vấn đề chỉ sử dụng nghiệm số (Numericals) của phương trình Laplace. Ta cần phải set điều kiện biên của phương trình và có thể lấy điều kiện biên khi có 1 phương trình sẵn của hàm tiềm năng điện trong phương pháp Analytical. Nghiệm Numericals của phương trình Laplace được so sánh với nghiệm Analytical và cho thấy các kết quả tương tự nhau, chính vì vậy có thể dùng điều kiện biên chung. Phương pháp sai phân hữu hạn (FDM) đã được khai thác bằng cách sử dụng phép lặp Jacobi . Điều này được thực hiện để kiểm tra tính hiệu quả của phương pháp lặp Jacobi để tính tiềm năng điện trên 1 mặt phẳng 2D khi cho ra kết quả chuẩn xác

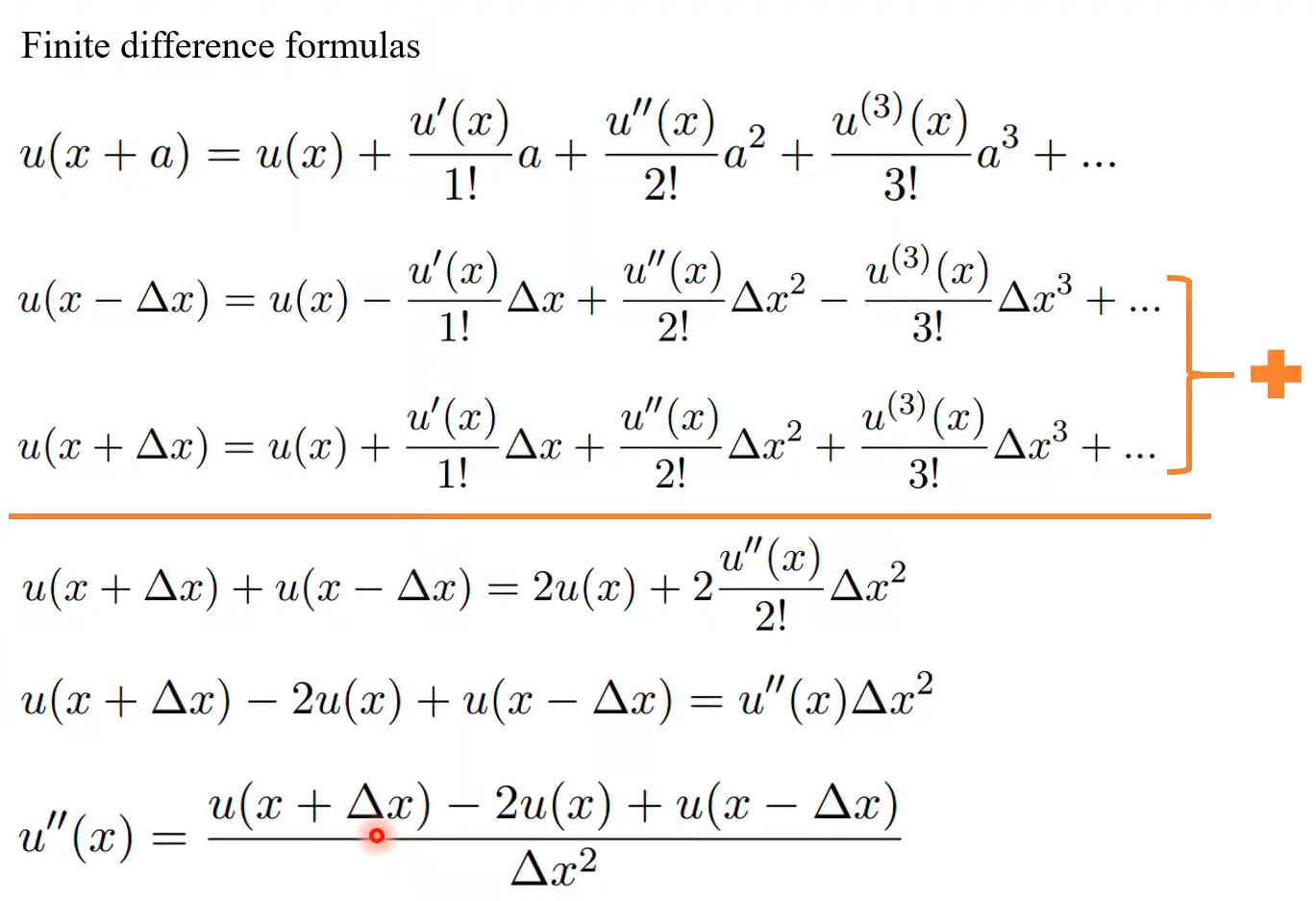
# Thuật toán

**B1.Giải thích PT laplace 2d và cách dùng phương pháp sai phân hữu hạn chia lưới các điểm :**

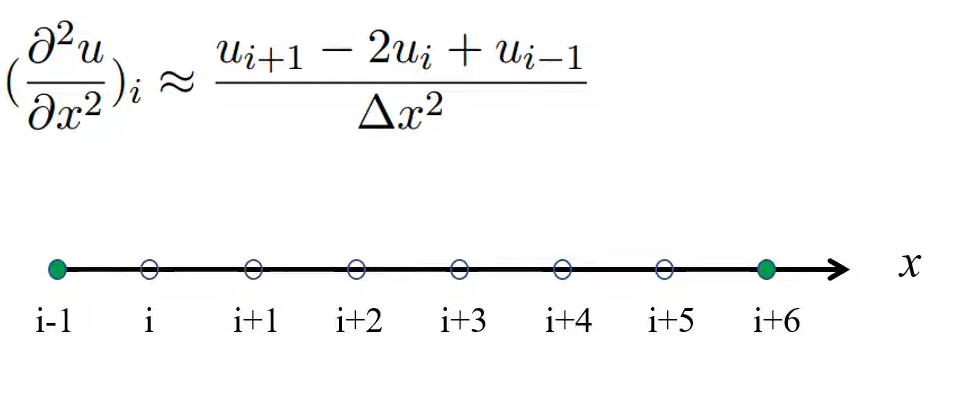


**Trong pt tiềm năng điện U=V sử dụng phương pháp sai phân hữu hạn để chia hình thành lưới các điểm với số lưới chiều dài là a, số lưới chiều rộng là b , ta được a\*b điểm cần tính tiềm năng điện**

Khi sử dụng khai triển Taylor\_Mauclaurin ta chứng minh được công thức xấp xỉ sau :

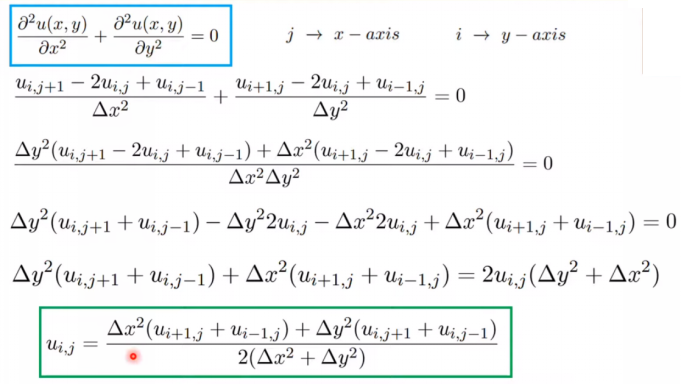
****

Trong thực tế, khi dùng phương pháp sai phân hữu hạn để chia theo trục x thì tiềm năng điện có đạo hàm bậc 2 theo x xấp xỉ công thức :

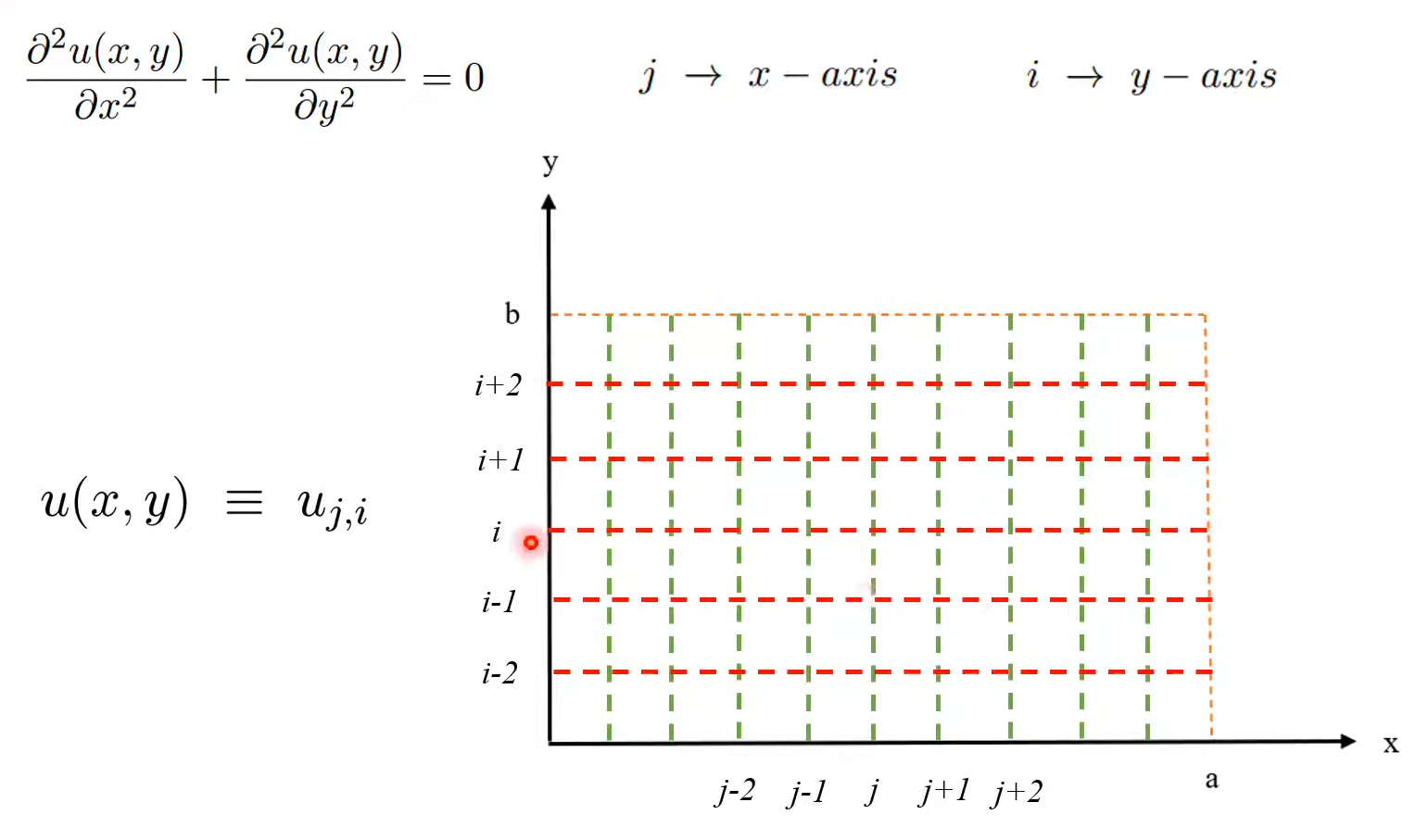


Tương tự với đạo hàm bậc 2 theo y

**Như vậy ta cm dc công thức nghiệm số Numericals sau :**

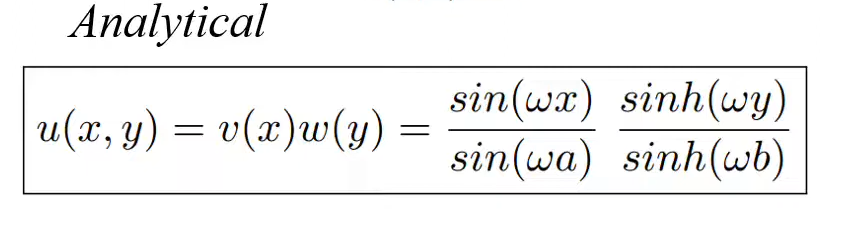
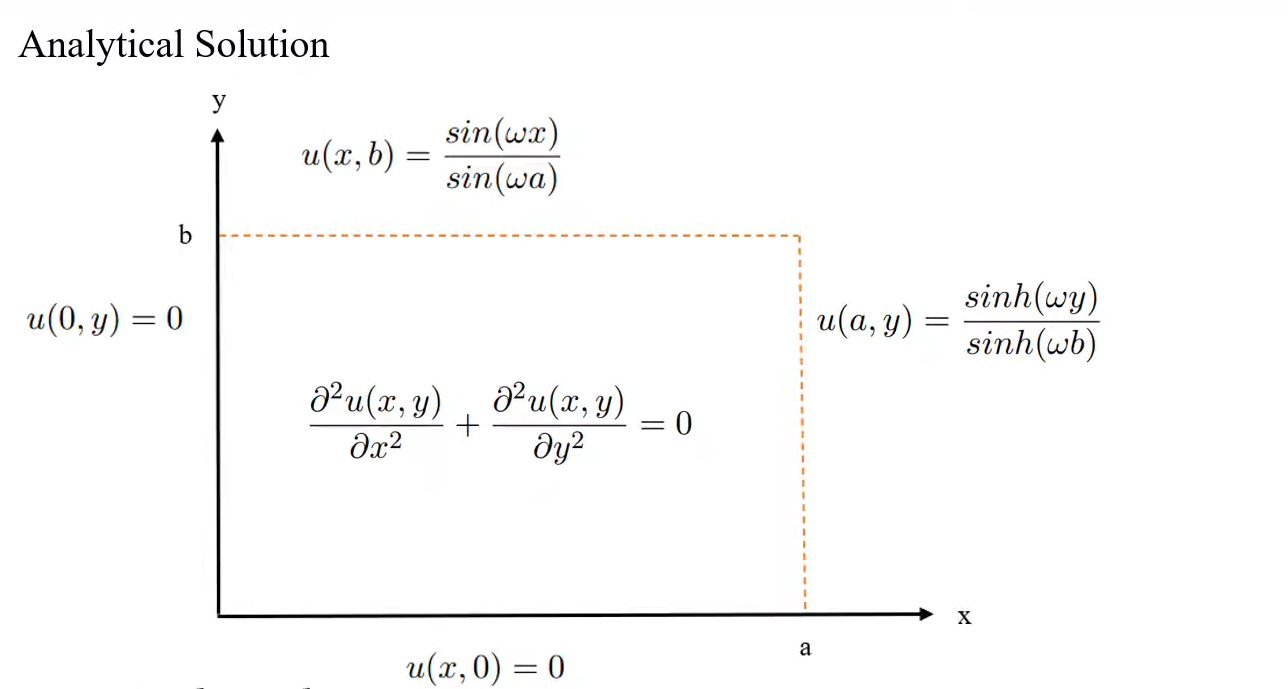


Trong hệ trục tọa độ ta có :



Trong thực tế, ngoài sử dụng phương pháp numerical , chúng ta cũng có thể sử dụng phương pháp analytical để tính tiềm năng điện. Về bản chất các biên và tiềm năng được tính trên các điểm của 2 phương pháp là xấp xỉ bằng nhau

Tại phương pháp analytical solution với công thức u(x,y) được định nghĩa sẵn. Ta biết về điều kiện của 4 biên của mặt phẳng để có thể sử dụng trong phần code



Đặt Omega=1 ta được 4 biên là: **u(x,b)= Sinx/sina**

**u(a,y)= sinh(y)/sinh(b)**

**u(x,0)=0**

**u(0,y)=0**

Ta sẽ sử dụng 4 biên ấy để set điều kiện biên khi làm ma trận U trong phần code tiếp theo

Trong phần code ,set tử số của nghiệm số Numerical =1 , như vậy ta được 1 hệ số courants cho nghiệm số Numerical như sau :

**Co = 1/ (2\*(dx^2+dy^2))**

**B2. Sử dụng phương pháp lặp jacobi để tính điểm tiếp theo**

đ/n :

Công thức lặp Jacobi được cho bởi:



Trong đó:

* ​ là giá trị tại điểm lưới (k,l) trong lần lặp thứ q.
* ​ là giá trị cập nhật tại điểm lưới (k,l) trong lần lặp thứ (q+1).
* Chỉ số k và l là các chỉ số của các điểm lưới theo hướng x và y tương ứng.

**Quá trình lặp:**

**Khởi tạo giá trị ban đầu:**

Giá trị khởi tạo ban đầu được thiết lập, thường là khi q=0.

**Cập nhật giá trị:**

Sử dụng công thức lặp, giá trị tại mỗi điểm lưới (k,l) được cập nhật. Việc cập nhật này dựa trên trung bình của các giá trị tại các điểm lân cận từ lần lặp trước q.

Các điểm lân cận được xét đến là các điểm trực tiếp kề nhau

theo hướng ngang và dọc: (k−1,l), (k+1,l), (k,l−1), và (k,l+1).

**Kiểm tra hội tụ:**

* + Sau khi cập nhật tất cả các điểm lưới, sự khác biệt giữa các vectơ của lần lặp tiếp theo và lần lặp trước đó được tính toán.
  + Quá trình lặp kết thúc khi mức dung sai định trước được đạt: ∣−∣< dung sai
  + Nếu sự khác biệt nhỏ hơn mức dung sai, điều này có nghĩa là nghiệm đã hội tụ và được coi là nghiệm.
  + Nếu không, quá trình lặp tiếp tục.

# Mã nguồn :

clc % Xóa cửa sổ lệnh

clear % Xóa tất cả các biến khỏi workspace

close all % Đóng tất cả các figure đang mở

% Nhập chiều dài chiều rộng của miền

a = input('Nhập chiều dài:');

b = input('Nhập chiều rộng:');

% Nhập số điểm lưới và sai số

nx = input('Nhập số lượng điểm lưới theo trục x:');

ny = input('Nhập số lượng điểm lưới theo trục y:');

tolerance = input('Nhập ngưỡng sai số:');

% Đặt kích thước của miền hình chữ nhật

a = a\*pi;

b = b\*pi;

% Khoảng cách giữa các điểm lưới theo chiều x và y

dx = a/nx;

dy = b/ny;

% Tạo các vector chứa tọa độ x và y

x = 0:dx:a;

y = 0:dy:b;

% Thiết lập các điều kiện biên cho ma trận U

Co = 1 / (2 \* (dx^2 + dy^2)); % Hệ số Courant cho phương pháp số

U = zeros(ny+1, nx+1); % Khởi tạo ma trận U

U(1, :) = 0;

U(ny+1, :) = sin(x) / sin(a);

U(:, 1) = 0;

U(:, nx+1) = sinh(y) / sinh(b);

% Thêm biến sai số và giá trị ngưỡng

error = Inf; % Khởi tạo sai số ban đầu là vô cùng

iter\_number = 0; % Khởi tạo số lần lặp

max\_iterations = 10000; % Số lần lặp tối đa để tránh vòng lặp vô hạn

errors = []; % Mảng để lưu giá trị error sau mỗi lần lặp

while error > tolerance && iter\_number < max\_iterations

U\_old = U; % Lưu lại ma trận U của lần lặp trước

maxU\_old = max(U\_old, [], 'all');

nU\_old = U\_old / maxU\_old;

% Tính giá trị của hàm U tại mỗi điểm lưới dựa trên phương pháp lặp số cho phương trình Laplace

for i = 2:ny

for j = 2:nx

U(i, j) = Co \* (dx^2 \* (U(i+1, j) + U(i-1, j)) + dy^2 \* (U(i, j+1) + U(i, j-1)));

end

end

maxU = max(U, [], 'all'); % Tìm giá trị lớn nhất trong ma trận U

nU = U / maxU; % Chuẩn hóa ma trận U để giá trị lớn nhất bằng 1

E = nU\_old - nU; % Tính ma trận sai số

% Tính sai số giữa hai lần lặp liên tiếp

error = max(max(E));

errors = [errors, error]; % Lưu giá trị error vào mảng

iter\_number = iter\_number + 1; % Tăng số lần lặp

end

% Vẽ biểu đồ Numerical

subplot(2, 1, 1);

contourf(nU, 200, 'linecolor', 'none')

title('Numerical')

xlabel('Chiều dài(mm)')

ylabel('Chiều rộng(mm)')

colormap(jet(256))

colorbar

clim([-1, 1])

% Vẽ đồ thị biểu diễn giá trị error sau mỗi lần lặp

subplot(2, 1, 2);

plot(errors, 'LineWidth', 2);

xlabel('Iteration Number');

ylabel('Error');

title('Sai số qua từng lần lặp(V)');

grid on;

% Vẽ đồ thị Numerical 3D

figure();

[X, Y] = meshgrid(1:size(nU, 2), 1:size(nU, 1));

surf(X, Y, nU, 'EdgeColor', 'none');

title('Numerical');

xlabel('Chiều dài(mm)');

ylabel('Chiều rộng(mm)');

zlabel('Điện thế');

colormap(jet(256));

colorbar;

clim([-1, 1]);

% Hiển thị số lần lặp và sai số cuối cùng

disp(['Số lần lặp: ', num2str(iter\_number)])

disp(['Sai số cuối cùng: ', num2str(error)])

# Kết quả

Đầu vào :

* Chiều dài hình chữ nhật
* Chiều rộng hình chữ nhật
* Số lượng điểm lưới theo trục x
* Số lượng điểm lưới theo trục y
* Ngưỡng sai số

A black screen with white text

Description automatically generated

Đầu ra :

* Số lần lặp
* Sai số cuối cùng

A black background with white text

Description automatically generated

Đồ thị biểu diễn trên hệ tọa độ Oxy và trục tọa độ Oxyz và đồ thị sai số  
A graph and diagram of a number

Description automatically generated with medium confidence  
A graph of a function

Description automatically generated with medium confidence

# Nhận xét

Đồ thị sai số qua mỗi lần lặp là hình Hypebol giảm dần. Tại lần lặp thứ 4437 thì sai số đạt mức nhỏ hơn 10^-12. Với mức sai số 10^-12 ta thấy có sự chênh lệch điện tích trên 4437 điểm/ 5000 điểm. Phân bố không đều: Phân bố điện tích trên hình chữ nhật thường không đều. Điện tích sẽ tập trung nhiều hơn ở các cạnh dài so với các cạnh ngắn. Ta thấy khi thay đổi độ dài rộng của hình chữ nhật thì phân bố điện tích tại các cạnh sẽ càng chênh lệch rõ rệt