HUST

ĐẠI HỌC BÁCH KHOA HÀ NỘI

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

ONE LOVE. ONE FUTURE.

2D Laplace Equations Successive Over Relaxation Method

Nhóm 6

Mục lục

I. Giới thiệu
II. Lý thuyết
III. Thực hiện bài toán
IV. Kết quả và giải thích
V. Tổng kết

I. GIỚI THIỆU

1. Giải phương trình 2D Laplace bằng phương pháp Successive Over Ralaxation (SOR)
2. Ứng dụng thực tế
3. Đặt vấn đề

1. Giải phương trình 2D Laplace bằng phương pháp SOR

- **Phương trình Laplace 2D** là một phương trình vi phân từng phần (PDE) mô tả nhiều hiện tượng vật lý như dòng nhiệt, điện thế trong một vùng không gian. Phương trình Laplace 2D có dạng:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

- **Phương pháp SOR** là một phương pháp số để giải các hệ phương trình tuyến tính, đặc biệt là các bài toán liên quan đến phương trình Laplace trong hai chiều. Phương pháp SOR kết hợp các lợi thế của phương pháp Jacobi và Gauss-Seidel bằng cách sử dụng hệ số lặp để tăng tốc quá trình hội tụ

2. Ứng dụng thực tế

- **Bài toán 2D Laplace – SOR** có ứng dụng thực tế trong nhiều lĩnh vực khác nhau như vật lý, hóa học, sinh học, khoa học máy tính và trí tuệ nhân tạo. Nó thường được sử dụng để mô phỏng sự phân bố điện thế, nhiệt độ, trường điện từ trong các môi trường tương ứng. Ngoài ra nó còn được áp dụng để tối ưu hóa hình học và phân tích hình học.
- Việc nghiên cứu bài toán này góp phần nâng cao hiệu quả và chất lượng trong thiết kế, mô phỏng và tối ưu các hệ thống phức tạp.

3. Đặt vấn đề

- Báo cáo này tập trung vào việc mô phỏng sự phân bố nhiệt độ khi đã ổn định trên miền hình vuông khi biết một số điều kiện biên cho trước (Dirichlet Boundary Condition and Mixed (Dirichlet & Neumann) Boundary Condition). Đồng thời tìm giá trị w sao cho số bước lặp là ít nhất. Thông tin này có thể có ý nghĩa trong các lĩnh vực khác nhau, bao gồm vật lý, cơ khí và khoa học vật liệu.

- Sử dụng phương pháp SOR để giải các phương trình nhiệt 2D ổn định

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

II. CƠ SỞ LÝ THUYẾT

1. 2D Laplace
2. Phương pháp Successive Over Ralaxation (SOR)
3. Ứng dụng lý thuyết vào bài toán

1. 2D Laplace

- Ta sẽ giải bài toán trên theo phương pháp FDM (Finite Difference Approximations)

Giả sử cần tính nhiệt độ tại điểm (x=i,y=j) -> T(x,y)=?
Xấp xỉ đạo hàm bậc 2 của T theo x, y tại điểm (i,j) trên lưới $\Delta x = \Delta y = h$ theo phương pháp sai phân trung tâm bậc 2:

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{h^2}$$

$$\frac{\partial^2 T}{\partial y^2} \approx \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{h^2}$$

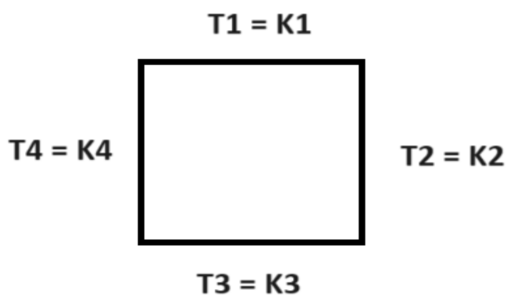
Do đó phương trình nhiệt ổn định trở thành: $\frac{T_{i+1,j}-2T_{i,j}+T_{i-1,j}}{h^2} + \frac{T_{i,j+1}-2T_{i,j}+T_{i,j-1}}{h^2} = 0$

suy ra: $T_{i,j} = \frac{1}{4}\left(T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1}\right)$

Ta sử dụng phương pháp SOR sẽ được nêu ở phần tiếp theo để giải ra T(i,j) để giải.

1. 2D Laplace

- Để giải phương trình trên chúng ta cần các điều kiện biên



+ Điều kiện biên Dirichlet: T(x,y)=g(x,y) trên biên của miền

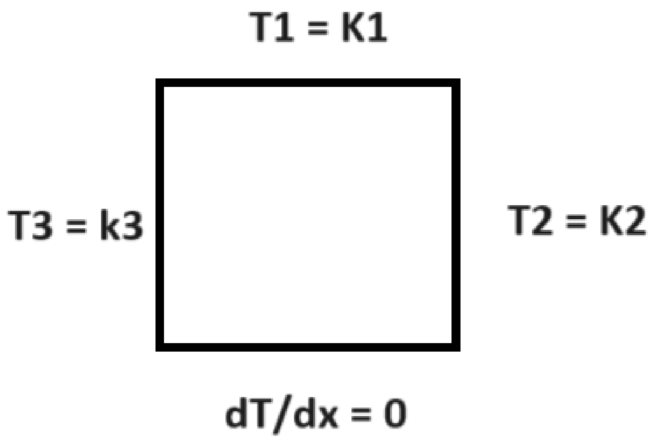
+ Điều kiện biên Dirichlet kết hợp với Neumann: điều kiện Dirichlet kết hợp các cạnh khác điều kiện sẽ là hàm vi phân của nhiệt độ theo dx hoặc dy

Đối với cạnh có điều kiện Neumann :

-> sử dụng công thức xấp xỉ đạo hàm $f'(x) \approx \frac{3f(x)-4f(x-\Delta x)+f(x-2\Delta x)}{2\Delta x}$

khi biết 3 điểm liên tiếp $\rightarrow \frac{3T(end)-4T(end-1)+T(end-2)}{2\Delta x} = 0$

$\rightarrow T(end) = \frac{4T(end-1)-T(end-2)}{3}$



2. Successive Over Ralaxation method (SOR method)

- Phương pháp Successive Over Relaxation (SOR) sử dụng công thức sau để tính $C_{i,j}^{k+1}$:

$$C_{i,j}^{k+1} = \frac{w}{4}\left(C_{i+1,j}^{(k)} + C_{i-1,j}^{(k+1)} + C_{i,j+1}^{(k)} + C_{i,j-1}^{(k+1)}\right) + (1-w)C_{i,j}^{k}$$

- Trong đó w là tham số thư giãn. Tham số w cho biết nồng độ tại một vị trí trong lần lặp trước ảnh hưởng đến lần lặp tiếp theo. Trong phương pháp SOR, chúng ta sử dụng giá trị w trong phạm vi $(1,2)$. Việc chọn w có thể tăng tốc độ hội tụ
- Cụ thể báo cáo này sẽ nghiên cứu phương pháp SOR với giá trị w được tối ưu.

3. Ứng dụng lý thuyết vào bài toán

Sau khi đã suy ra được công thức

$$T_{i,j} = \frac{1}{4}(T_{i+1,j} + T_{i-1,j} + T_{i,j+1} + T_{i,j-1})$$

- Ta áp dụng phương pháp SOR như sau để giải phương trình này.
- Đồng thời khởi tạo vecto $w = (0.5; 0.1; 1.9)$ và vecto $iter$ để lưu số vòng lặp khi chạy với mỗi w để tìm ra w có số vòng lặp nhỏ nhất.

3. Ứng dụng lý thuyết vào bài toán

+ Với điều kiện biên Dirichlet:

1. Khởi tạo ma trận t_init là ma trận 0 để lưu kết quả, đồng thời khởi tạo nhiệt độ cho các điểm tại biên, sai số tolerance = 10^{-5}

2. Lặp lại cho đến khi hội tụ:

* Với mỗi điểm (i,j) :

$$T_{i,j}^{new} = (1-w)T_{i,j}^{old} + \frac{w}{4}\left(T_{i+1,j}^{old} + T_{i-1,j}^{new} + T_{i,j+1}^{old} + T_{i,j-1}^{new}\right)$$

* Cập nhật giá trị mới cho tất cả các điểm lưới.

* Kiểm tra điều kiện hội tụ đến khi nhỏ hơn giá trị tolerance dừng.

+ Với điều kiện biên Neumann:

Sau khi đã suy ra được $T(end) = \frac{4T(end-1)-T(end-2)}{3}$

Tại mỗi vòng lặp cập nhật lại điều kiện trên cạnh có điều kiện Neumann theo công thức trên cụ thể ở ví dụ nêu trên là cạnh cuối cùng.

3. Ứng dụng lý thuyết vào bài toán

3. Vector thông lượng nhiệt

- Vector thông lượng nhiệt (heat flux vector) là biểu diễn tốc độ, hướng, lượng của dòng nhiệt qua một đơn vị diện tích trong một khoảng thời gian đơn vị

$$\vec{q} = -k\nabla T$$

- Trong đó:

  $\vec{q}$ là vector thông lượng nhiệt

  k là hệ số dẫn nhiệt (Trong bài toán sau sẽ áp dụng với k = 1)

  ∇T là gradient của nhiệt độ T

HUST
ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

III. Thực hiện bài toán 2D Laplace SOR

hust.edu.vn fb.com/dhbkhn

Code

```
% Dimensions of the simulation grid in x (xdim) and y (ydim) directions
xdim=100;
ydim=100;

% convergence tolerance
tolerance = 1e-5;

% create vector x, y
x=1:1:xdim+1;
y=1:1:ydim+1;

%--------DIRICHLET BOUNDARY CONDITION-----------------%

% x coordinates for boundary
i=1:1:xdim+1;

% Values of omega tested to find the optimal one
omega=0.5:0.1:1.9;

% Initializing initial temperature matrix used for all values of omega
t_init=zeros(xdim+1,ydim+1);

% A temperature of 100°C is applied on one boundary, the remaining boundaries are going to remain at zero
t_init(i,ydim+1)=100; % right boundary
t_init(1,i)=0; % top boundary
t_init(i,1)=0; % left boundary
t_init(xdim+1,i)=0; % bottom boundary

% Matrix of iterations for various values of omega
iter=zeros(1,length(omega));

```

Code

```
% Running for loop to compute number of iterations for different omega
for range=1:1:length(omega)

   % Initializing previous (t_prev) and present (t_now) iterations' temperature matrix
   t_now=t_init;
   t_prev=t_init;

   % Giving initial difference between t_now and t_prev to start the iterations
   t_prev(2, ydim)=1;

   % Iteration loop
   while(max(max(abs(t_now-t_prev)))>tolerance) % Run this until convergence

       % Updating previous iteration matrix as the present iteration matrix to continue iterations
       t_prev=t_now;

       % Iteration counter increment for each omega
       iter(range)=iter(range)+1;

       % Updating
       for i=2:1:xdim
           for j=2:1:ydim
               t_now(i,j)=t_now(i,j)+omega(range)*(((t_now(i-1,j)+t_now(i+1,j)+t_now(i,j-1)+t_now(i,j+1))/4)-t_now(i,j));
           end
       end

```

HUST
ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Code

```
        % Movie type colour scaled image plot to see how solution progresses for omega=1.9
        if range==length(omega)
            figure(1);
            imagesc(t_now);
            colorbar;
            colormap(jet);
         title(['\fontsize{7}Temperature distribution at iteration no ',int2str(iter(range)),' for omega=1.9 (Dirichlet BC)'],'Color','k');
            xlabel('x-axis','FontSize',10);
            ylabel('y-axis','FontSize',10);
            set(gca,'FontSize',10);
            getframe;
        end
    end
end
```

HUST
ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Code

```
% Compute gradient to get the heat flux
[Tx, Ty] = gradient(t_now);

% Plot heat flux vectors
figure(2);
quiver(x, y, -Tx, -Ty);
title('Heat flux vectors (Dirichlet BC)');
xlabel('x');
ylabel('y');
axis tight equal;

% Plot image
figure(3);
surf(x,y,t_now);
title('Temperature distribution (Dirichlet BC)');
xlabel('x');
ylabel('y');
zlabel('Temperature');

%Plotting alpha v/s iterations
figure(4);
plot(omega,iter);
title(['\fontsize{7}Plot of omega v/s no. of iterations on a grid of ',int2str(xdim),' x ',int2str(ydim), ' Dirichlet BC'],'color','k');
xlabel('omega','FontSize',10);
ylabel('No of iterations','FontSize',10);

```

Code

```
%--------MIXED BOUNDARY CONDITION (DIRICHLET AND NEUMANN)----------------%

% x coordinates for boundary
i=1:1:xdim+1;

% Values of omega tested to find the optimal one
omega=0.5:0.1:1.9;

% Initializing initial temperature matrix used for all values of omega
t_init=zeros(xdim+1,ydim+1);

% Boundary conditions
t_init(i,ydim+1)=50; % right boundary (Dirichlet BC)
t_init(1,i)=100; % top boundary (Dirichlet BC)
t_init(i,1)=75; % left boundary (Dirichlet BC)

% bottom boundary is Neumann BC

% Matrix of iterations for various values of omega
iter=zeros(1,length(omega));
```

Code

```
% Running for loop to compute number of iterations for different omega
for range=1:1:length(omega)

     % Initializing previous (t_prev) and present (t_now) iterations' temperature matrix
    t_now=t_init;
    t_prev=t_init;

    % Giving initial difference between t_now and t_prev to start the iterations
    t_prev(2, ydim)=1;

    % Iteration loop
    while(max(max(abs(t_now-t_prev)))>tolerance) % Run this until convergence

        % Updating previous iteration matrix as the present iteration matrix to continue iterations
        t_prev=t_now;

        % Iteration counter increment for each omega
        iter(range)=iter(range)+1;

        % Updating
        for i=2:1:xdim
            for j=2:1:ydim
                t_now(i,j)=t_now(i,j)+omega(range)*(((t_now(i-1,j)+t_now(i+1,j)+t_now(i,j-1)+t_now(i,j+1))/4)-t_now(i,j));
            end
        end

        % Update Neumann BC
        t_now(xdim + 1,:) = (4.*t_now(xdim,:)-t_now(xdim-1,:))/3;

```

HUST
ĐẠI HỌC BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

Code

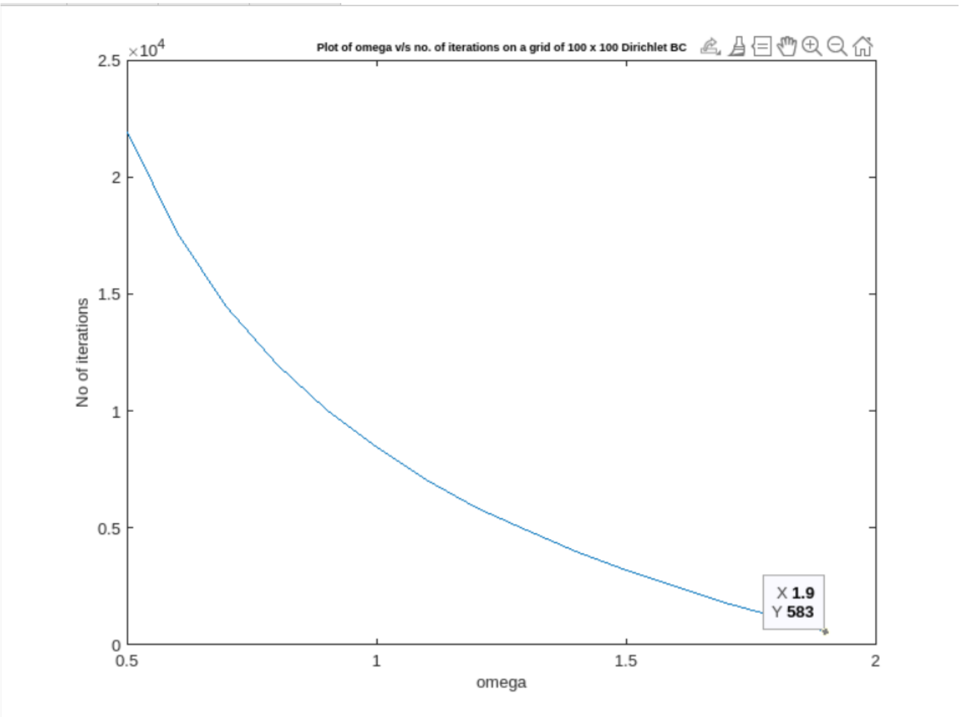
```
            % Movie type colour scaled image plot to see how solution progresses for omega=1.9
            if range==length(omega)
                figure(5);
                imagesc(t_now);
                colorbar;
                colormap(jet);
        title(['\fontsize{7}Temperature distribution on a at iteration no ',int2str(iter(range)),' for an alpha=1.9 (Mixed BC)'],'Color','k');
                xlabel('x-axis','FontSize',10);
                ylabel('y-axis','FontSize',10);
                set(gca,'FontSize',10);
                getframe;
            end
        end
    end

    % Compute gradient to get the heat flux
    [Tx, Ty] = gradient(t_now);

    % Plot heat flux vectors
    figure(6);
    quiver(x, y, -Tx, -Ty);
    title('Heat flux vectors (Mixed BC)');
    xlabel('x');
    ylabel('y');
    axis tight equal;

    % Plot image
    figure(7);
    surf(x,y,t_now);
    title('Temperature distribution (Mixed BC)');
    xlabel('x');
    ylabel('y');
    zlabel('Temperature');

    %Plotting alpha v/s iterations
    figure(8);
    plot(omega,iter);
    title(['\fontsize{7}Plot of omega v/s no. of iterations on a grid of ',int2str(xdim),' x ',int2str(ydim), ' Mixed BC'],'color','k');
    xlabel('omega','FontSize',10);
    ylabel('No of iterations','FontSize',10);
```

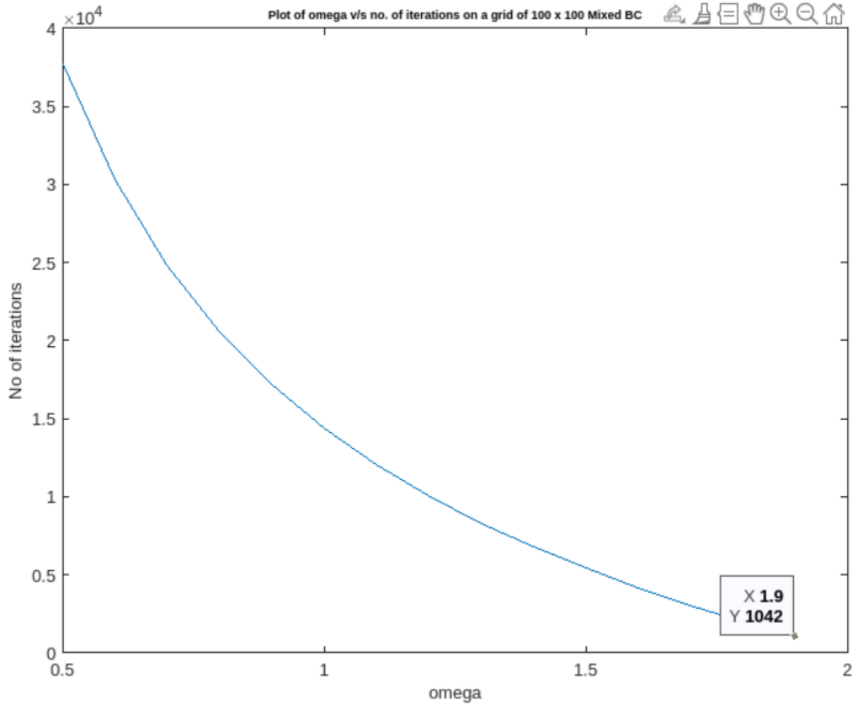
IV. Kết quả và Giải thích

HUST

hust.edu.vn fb.com/dhbkhn

Kết quả

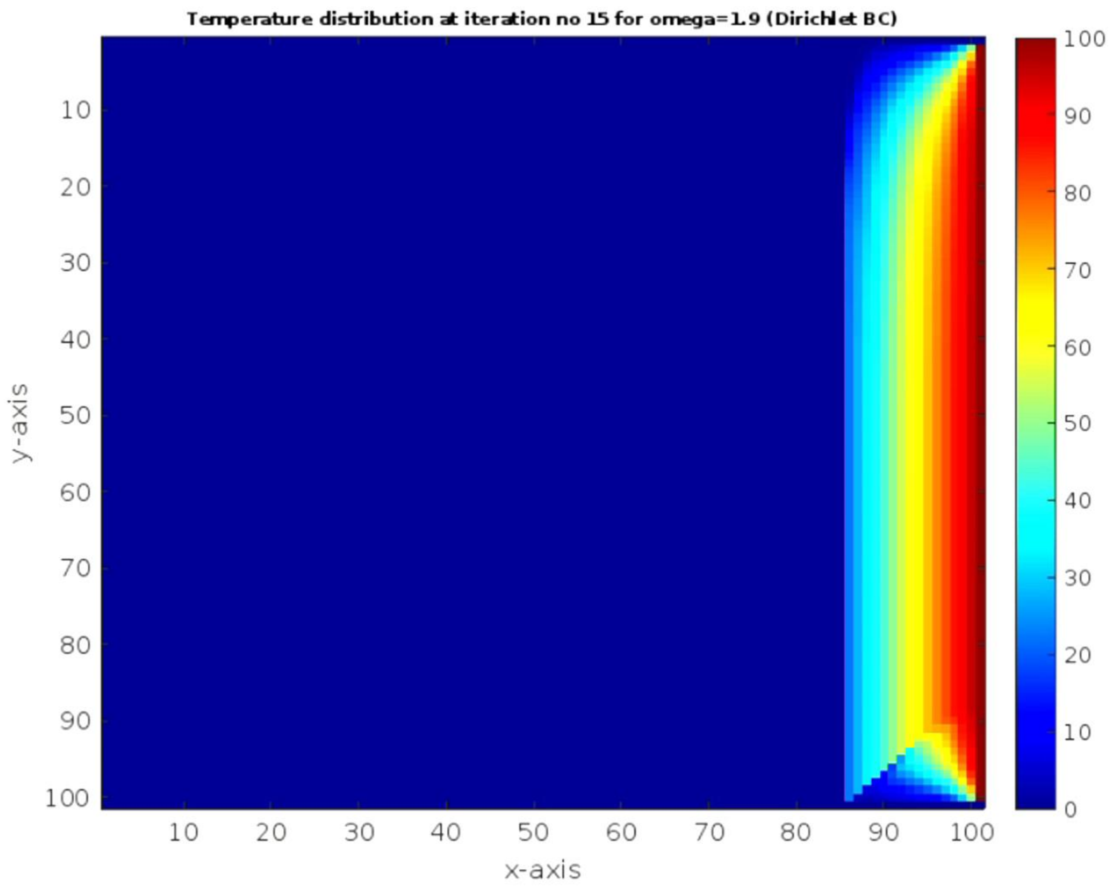




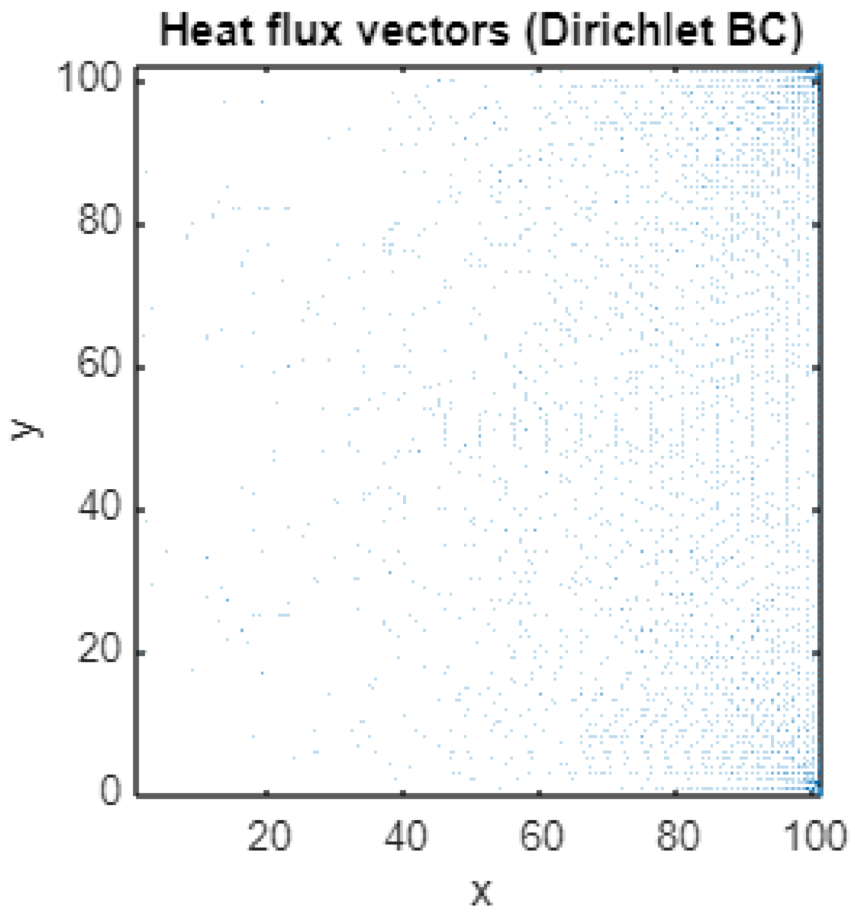
Hệ số w: Từ biểu đồ w - số vòng lặp của cả 2 trường hợp biên, khi $w = 1.9$ thì đều cho ra số vòng lặp là nhỏ nhất (số vòng lặp lần lượt là 583 với biên Dirichlet, 1042 với biên Dirichlet và Neuman).

-> Vậy nên với $w = 1.9$ thuật toán chạy tối ưu nhất.

Kết quả

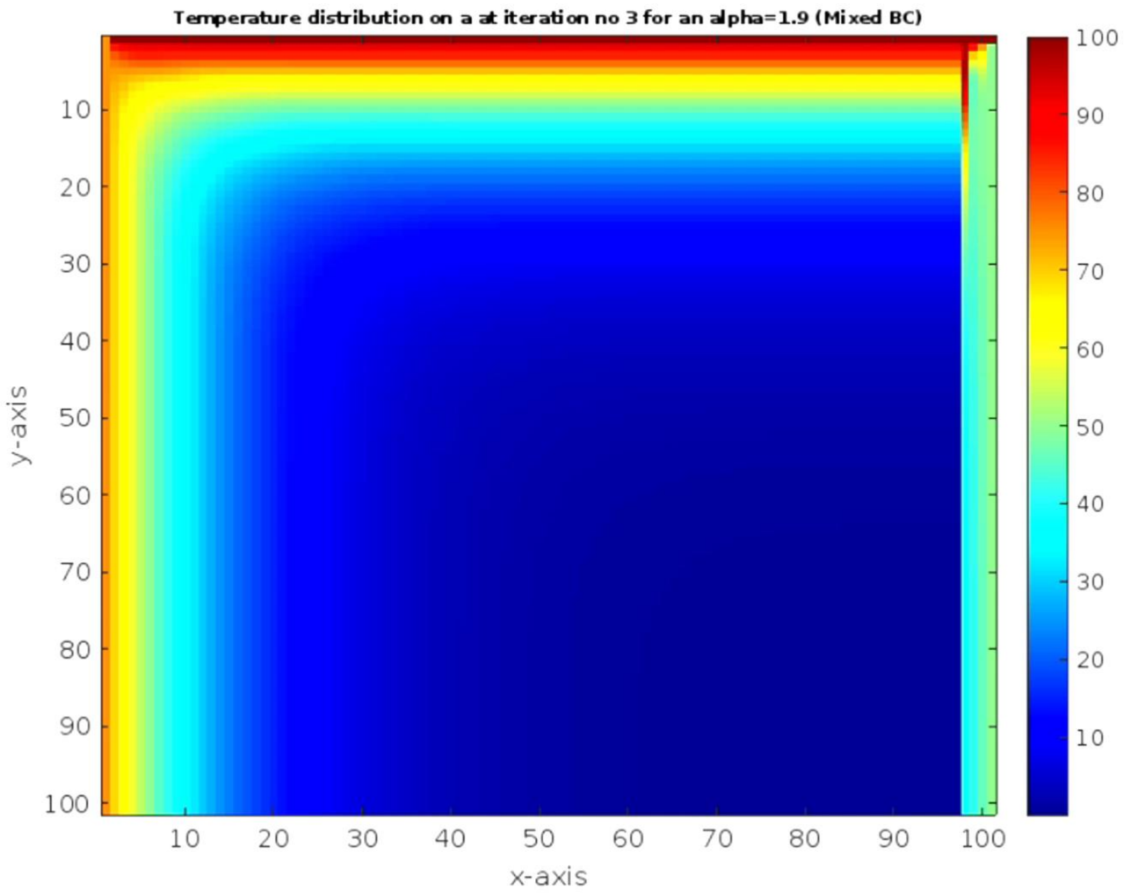


Temperature distribution at iteration no 15 for omega=1.9 (Dirichlet BC)
y-axis
x-axis
10
20
30
40
50
60
70
80
90
100
0

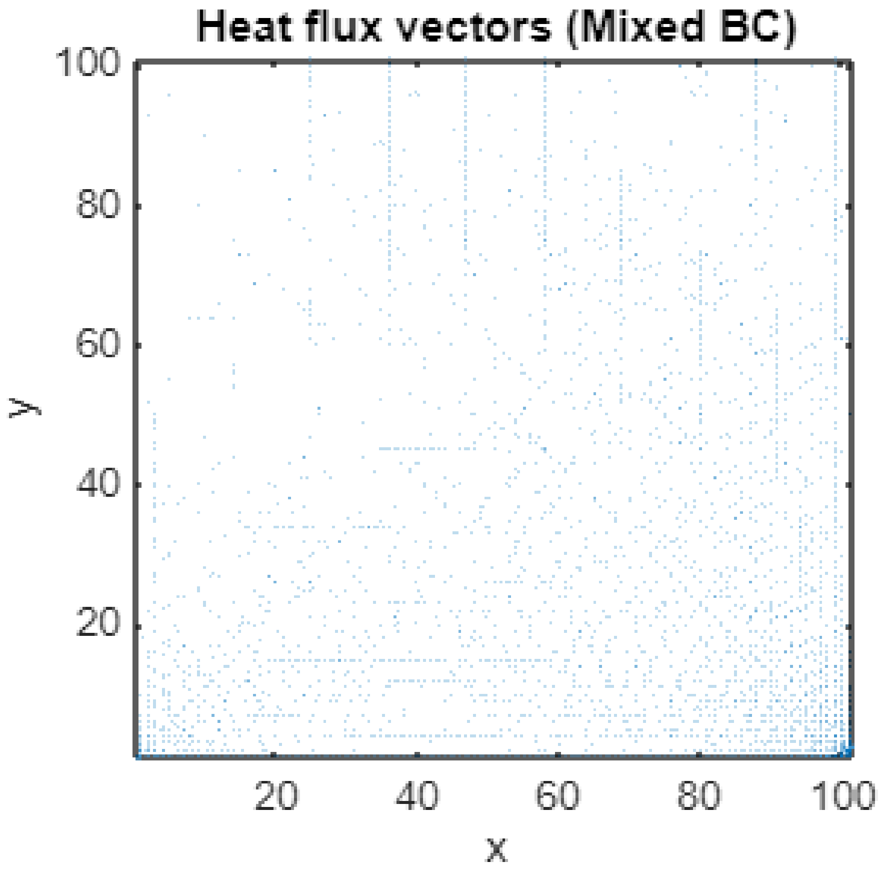


Heat flux vectors (Dirichlet BC)
y
x
0
20
40
60
80
100

Kết quả

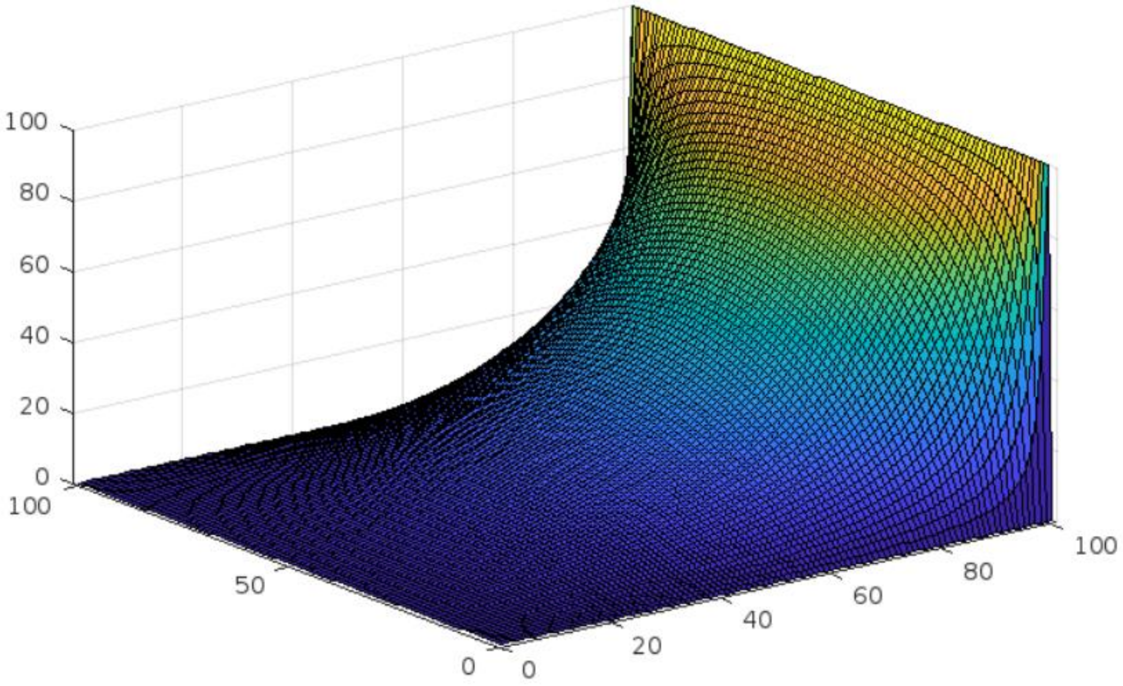


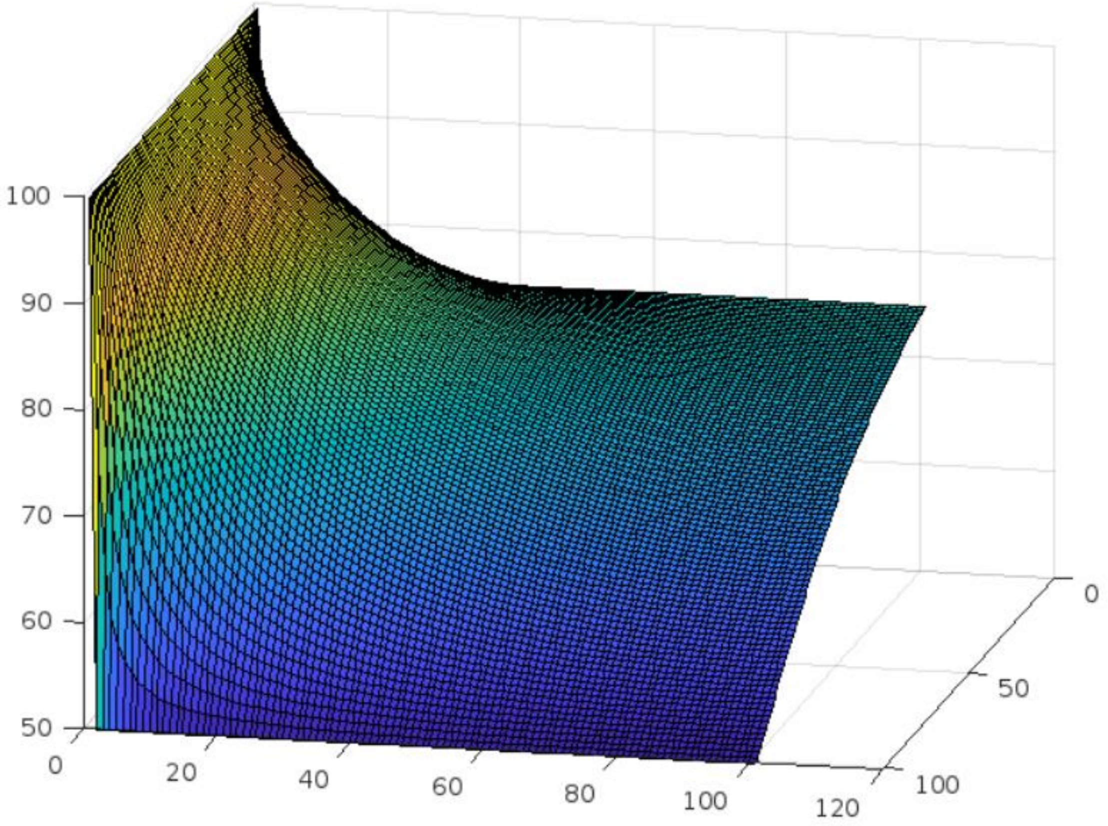
Temperature distribution on a at iteration no 3 for an alpha=1.9 (Mixed BC)
y-axis
x-axis
10
20
30
40
50
60
70
80
90
100



Heat flux vectors (Mixed BC)
y
x
20
40
60
80
100

Phân bố nhiệt độ trên 3 trục vector của điều kiện biên Diriclet





HUST

hust.edu.vn fb.com/dhbkhn

IV. Kết luận

Kết luận

Mô phỏng nhiệt độ bằng phương pháp SOR để giải phương trình Laplace 2D với 2 điều kiện biên là Dirichlet và Mixed (Dirichlet & Neumann) là phương pháp tốt nhất để giải bài toán đã đặt ra. Tối ưu hóa w để giảm số lần lặp ít nhất có thể để thuật toán kết thúc nhanh hơn.

HUST

hust.edu.vn fb.com/dhbkhn

THANK YOU !