ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
***** *****

BÁO CÁO BÀI TẬP LỚN
HỌC PHẦN: TÍNH TOÁN KHOA HỌC

Nhóm: 19
Mã lớp học: 147769
Giảng viên hướng dẫn: TS. Vũ Văn Thiệu

- Hà Nội, tháng 6 năm 2024 -

Thành viên trong nhóm

STT	Họ và tên	Mã sinh viên
01	Trịnh Hữu An	20225593
02	Phạm Quốc Cường	20225604
03	Vũ Quang Dũng	20225818
04	Lương Văn Khanh	20225728
05	Nguyễn Đức Tấn Sang	20225664
06	Nguyễn Ngọc Quân	20225662

Mục lục

TOPIC: 2D Laplace Equations - Successive Over Relaxation Method

1 Phát biểu bài toán

- Mô phỏng sự lan truyền virus trên một lưới (grid) trên một lưới 2D
- Sử dụng phương pháp Successive Over-Relaxation (SOR) để tính toán xác suất lan truyền và xác định các vị trí tiềm năng cho sự phát triển virus
- Mô phỏng này nhằm mục đích nghiên cứu sự phát triển của virus trên một lưới và kiểm tra sự ảnh hưởng của các tham số khác nhau đến quá trình lan truyền

2 Cơ sở lý thuyết

2.1 Phương trình Laplace

- Phương trình Laplace là một phương trình đạo hàm riêng bậc hai được sử dụng rộng rãi trong nhiều lĩnh vực khoa học và kỹ thuật, như cơ học chất lỏng, điện từ học và lý thuyết nhiệt.
- Dạng tổng quát:
$$\nabla^2\phi = 0$$
- Biểu diễn trong toạ độ hai chiều (2D):
$$\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} = 0$$
- Đây là một phương trình vi phân điều hòa, nghiệm của nó thường biểu diễn các hàm điều hòa, không thay đổi theo thời gian, và thường mô tả trạng thái cân bằng

2.2 Phương pháp số: Successive Over-Relaxation (SOR)

- SOR (Successive Over-Relaxation) là một phương pháp lặp số được sử dụng để giải các hệ phương trình tuyến tính bằng cách kết hợp các lần lặp của phương pháp lặp Gauss-Seidel với một yếu tố thêm vào để tăng tốc độ hội tụ.

- Phương pháp này đặc biệt hiệu quả khi áp dụng cho các hệ phương trình xuất hiện trong bài toán rời rạc hóa của phương trình Laplace trong lĩnh vực vật lý, toán học và kỹ thuật.

2.2.1 Rời rạc hóa phương trình Laplace

- Chia miền tính toán thành lưới vuông với các điểm lưới (grid points).
- Sử dụng phương pháp sai phân hữu hạn (finite difference method) để thay thế các đạo hàm bậc hai bằng các sai phân trung tâm.
- Giả sử miền tính toán được chia thành một lưới có kích thước $(n+1) \times (n+1)$ với bước lưới h. Phương trình Laplace tại một điểm lưới (i, j) có thể được rời rạc hóa như sau:

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} \approx \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{h^2} + \frac{\phi_{i,j+1} - 2\phi_{i,j} + \phi_{i,j-1}}{h^2} = 0$$

- Dẫn đến phương trình:

$$\phi_{i,j} = \frac{1}{4} \left(\phi_{i+1,j} + \phi_{i-1,j} + \phi_{i,j+1} + \phi_{i,j-1} \right)$$

2.2.2 Phương pháp Gauss-Seidel

- Phương pháp Gauss-Seidel là một phương pháp lặp để giải hệ phương trình tuyến tính, áp dụng các giá trị cập nhật ngay lập tức để tăng tốc độ hội tụ.
- Với phương trình Laplace rời rạc hóa, phương pháp Gauss-Seidel cập nhật giá trị $u_{i,j}$ theo công thức:

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

2.3 Xác suất và ngẫu nhiên

- Xác suất được sử dụng để quyết định việc phát triển virus tại các vị trí ứng viên dựa trên năng lượng thức ăn. Xác suất phát triển tại vị trí (i, j) được xác định bởi:

$$P(i, j) = \frac{(C_{i,j})^p}{\sum_{\text{candidates}} (C_{i,j})^p}$$

Trong đó:

- $C_{i,j}$ là năng lượng thức ăn tại vị trí (i, j).
- p là tham số xác suất.
- Tổng mẫu số là tổng năng lượng thức ăn tại tất cả các vị trí ứng viên.

Giới thiệu bài toán: Bài toán mô phỏng sự lan truyền của virus trên một lưới 2D sử dụng phương pháp Successive Over-Relaxation (SOR) để tính toán xác suất lan truyền và xác định các vị trí tiềm năng cho sự phát triển virus.

- Mục đích: Nghiên cứu sự phát triển của virus trên một lưới và kiểm tra sự ảnh hưởng của các tham số khác nhau đến quá trình lan truyền.

3 Mã nguồn

```
function virus_simulation_ui
    % Tao giao dien nguoi dung
    fig = uifigure('Position', [100 100 600 500], 'Name', 'Virus␣
        Simulation␣UI');

    % Them tieu de
    lbl_title = uilabel(fig, 'Position', [200 450 200 30], 'Text', '
        Virus␣Simulation␣Parameters', ...
                        'FontSize', 16, 'FontWeight', 'bold', '
                            HorizontalAlignment', 'center');

    % Hop van ban cho so luong virus toi da
    lbl_n = uilabel(fig, 'Position', [50 380 150 22], 'Text', 'Max␣
        Number␣of␣Viruses:');
    txt_n = uieditfield(fig, 'numeric', 'Position', [220 380 100
        22], 'Value', 1000);

    % Slider cho tham so dieu chinh SOR
    lbl_w = uilabel(fig, 'Position', [50 320 150 22], 'Text', 'SOR␣
        Parameter␣w:');
    sld_w = uislider(fig, 'Position', [220 330 200 3], 'Limits', [1
        2], 'Value', 1.89);
    lbl_w_value = uilabel(fig, 'Position', [430 320 50 22], 'Text',
        '1.89');
    sld_w.ValueChangedFcn = @(sld, event) updateLabelDecimal(sld,
        lbl_w_value);

    % Slider cho tham so xac suat
    lbl_p = uilabel(fig, 'Position', [50 260 150 22], 'Text', '
        Probability␣Parameter␣p:');
```

```
    sld_p = uislider(fig, 'Position', [220 270 200 3], 'Limits', [0
        2], 'Value', 0);
    lbl_p_value = uilabel(fig, 'Position', [430 260 50 22], 'Text',
        '0');
    sld_p.ValueChangedFcn = @(sld, event) updateLabelDecimal(sld,
        lbl_p_value);

    % Hop van ban cho kich thuoc luoi
    lbl_size = uilabel(fig, 'Position', [50 200 150 22], 'Text', '
        Grid Size:');
    txt_size = uieditfield(fig, 'numeric', 'Position', [220 200 100
        22], 'Value', 50);

    % Hop van ban cho vi tri virus ban dau (X)
    lbl_initial_x = uilabel(fig, 'Position', [50 140 150 22], 'Text'
        , 'Initial Virus Position X:');
    txt_initial_x = uieditfield(fig, 'numeric', 'Position', [220 140
         100 22], 'Value', 25);

    % Hop van ban cho vi tri virus ban dau (Y)
    lbl_initial_y = uilabel(fig, 'Position', [50 80 150 22], 'Text',
         'Initial Virus Position Y:');
    txt_initial_y = uieditfield(fig, 'numeric', 'Position', [220 80
        100 22], 'Value', 25);

    % Nut chay mo phong
    btn_run = uibutton(fig, 'Position', [250 30 100 30], 'Text', '
        Run Simulation', ...
                       'ButtonPushedFcn', @(btn, event)
                          run_simulation(txt_n.Value, sld_w.Value,
                          sld_p.Value, txt_size.Value, txt_initial_x
                          .Value, txt_initial_y.Value));
end

function updateLabelDecimal(sld, lbl)
    lbl.Text = num2str(sld.Value, '%.2f');
end

function run_simulation(n, w, p, size, initial_x, initial_y)
    n = round(n);
    size = round(size);
    initial_x = round(initial_x);
    initial_y = round(initial_y);

    % So virus hien co
    nVirus = 1;

    % Mang bieu thi nang luong thuc an, khoi tao tat ca cac gia tri
        bang 0
```

```
C_sor = zeros(size);
C_gauss = zeros(size);

% Mang danh dau vi tri cac virus da xuat hien, khoi tao tat ca
   cac gia tri bang 0
grow_sor = zeros(size);
grow_gauss = zeros(size);

grow_sor(initial_x, initial_y) = 1;
grow_gauss(initial_x, initial_y) = 1;

% Voi phuong phap lap, tinh tai buoc k+1 se co gia tri cua buoc
   k+1
% tai vi tri (i-1,j) va (i,j-1)
for i = 2:(size-1)
    for j = 2:(size-1)
        C_sor(i,j) = 1;  % Dat tat ca cac gia tri ben trong ma
           tran C_sor thanh 1
        C_gauss(i,j) = 1;  % Dat tat ca cac gia tri ben trong ma
            tran C_gauss thanh 1
    end
end

% Dat virus dau tien tai vi tri moi
C_sor(initial_x, initial_y) = 0;
C_gauss(initial_x, initial_y) = 0;

% Luu tru so luong virus truoc khi cap nhat
prevNVirus_sor = nVirus;
prevNVirus_gauss = nVirus;

% Mang tam de danh dau cac vi tri co the phat trien virus
candidate_sor = zeros(size);
candidate_gauss = zeros(size);

% Tao luoi toa do X, Y cho viec ve do thi
x = 1:size;
y = 1:size;
[X,Y] = meshgrid(x,y);

% Ham cap nhat SOR
function C = update_sor(C, w, size)
    for i = 2:(size-1)
        for j = 2:(size-1)
            if C(i,j) ~= 0
                C(i,j) = (w/4) * (C(i+1,j) + C(i-1,j) + C(i,j+1)
                    + C(i,j-1)) + (1-w) * C(i,j);
            end
        end
```

```
    end
end

% Ham cap nhat Gauss-Seidel
function C = update_gauss(C, size)
    for i = 2:(size-1)
        for j = 2:(size-1)
            if C(i,j) ~= 0
                C(i,j) = (1/4) * (C(i+1,j) + C(i-1,j) + C(i,j+1)
                    + C(i,j-1));
            end
        end
    end
end

% Vong lap chinh de mo phong su phat trien cua virus
while 1
    sumOfChance_sor = 0;  % Tong xac suat cua cac vi tri co the
       phat trien cho SOR
    sumOfChance_gauss = 0;  % Tong xac suat cua cac vi tri co
       the phat trien cho Gauss

    % Tinh toan SOR
    C_sor = update_sor(C_sor, w, size);

    % Tinh toan Gauss-Seidel
    C_gauss = update_gauss(C_gauss, size);

    % Tim cac vi tri co the phat trien virus (candidates) cho
       SOR
    for i = 2:(size-1)
        for j = 2:(size-1)
            if grow_sor(i,j) == 1
                C_sor(i,j) = 0;
                if grow_sor(i-1,j) == 0 && candidate_sor(i-1,j)
                   == 0
                    candidate_sor(i-1,j) = 1;
                end
                if grow_sor(i+1,j) == 0 && candidate_sor(i+1,j)
                   == 0
                    candidate_sor(i+1,j) = 1;
                end
                if grow_sor(i,j-1) == 0 && candidate_sor(i,j-1)
                   == 0
                    candidate_sor(i,j-1) = 1;
                end
                if grow_sor(i,j+1) == 0 && candidate_sor(i,j+1)
                   == 0
                    candidate_sor(i,j+1) = 1;
```

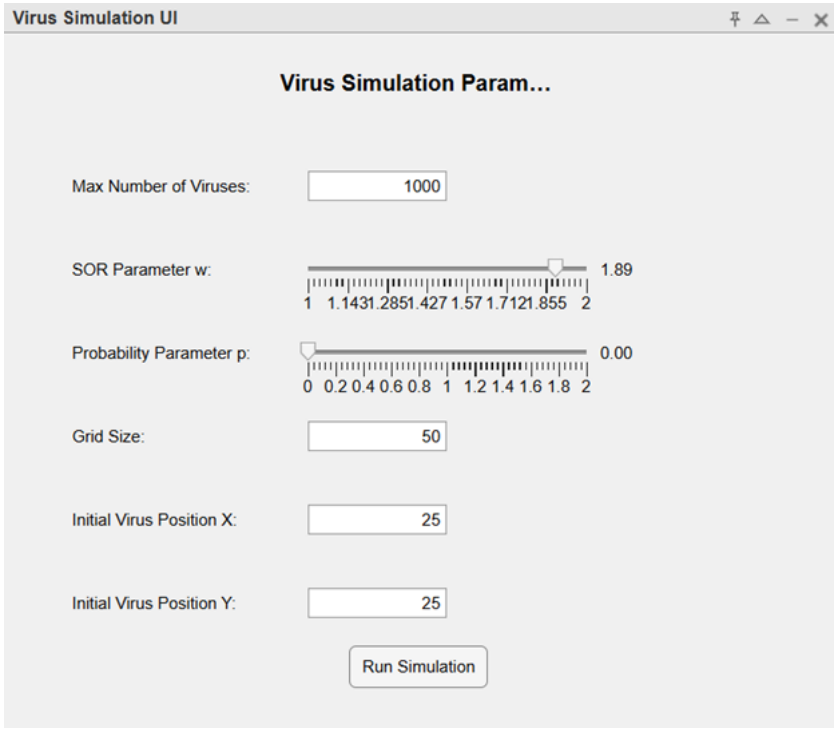
```
            end
        end
    end
end

% Tim cac vi tri co the phat trien virus (candidates) cho
   Gauss-Seidel
for i = 2:(size-1)
    for j = 2:(size-1)
        if grow_gauss(i,j) == 1
            C_gauss(i,j) = 0;
            if grow_gauss(i-1,j) == 0 && candidate_gauss(i
               -1,j) == 0
                candidate_gauss(i-1,j) = 1;
            end
            if grow_gauss(i+1,j) == 0 && candidate_gauss(i
               +1,j) == 0
                candidate_gauss(i+1,j) = 1;
            end
            if grow_gauss(i,j-1) == 0 && candidate_gauss(i,j
               -1) == 0
                candidate_gauss(i,j-1) = 1;
            end
            if grow_gauss(i,j+1) == 0 && candidate_gauss(i,j
               +1) == 0
                candidate_gauss(i,j+1) = 1;
            end
        end
    end
end

% Tinh tong xac suat cho SOR
for i = 1:size
    for j = 1:size
        if candidate_sor(i,j) == 1
            sumOfChance_sor = sumOfChance_sor + p * C_sor(i,
               j);
        end
    end
end

% Tinh tong xac suat cho Gauss-Seidel
for i = 1:size
    for j = 1:size
        if candidate_gauss(i,j) == 1
            sumOfChance_gauss = sumOfChance_gauss + p *
               C_gauss(i,j);
        end
    end
```

```
end

if sumOfChance_sor > 0
    for i = 1:size
        for j = 1:size
            if candidate_sor(i,j) == 1
                if rand <= p * C_sor(i,j) / sumOfChance_sor
                    grow_sor(i,j) = 1;
                    candidate_sor(i,j) = 0;
                    nVirus = nVirus + 1;
                end
            end
        end
    end
end

if sumOfChance_gauss > 0
    for i = 1:size
        for j = 1:size
            if candidate_gauss(i,j) == 1
                if rand <= p * C_gauss(i,j) /
                   sumOfChance_gauss
                    grow_gauss(i,j) = 1;
                    candidate_gauss(i,j) = 0;
                    nVirus = nVirus + 1;
                end
            end
        end
    end
end

% Dung mo phong khi so luong virus dat den so luong toi da
if nVirus >= n
    break;
end

% Ve do thi tai moi buoc
subplot(1,2,1);
mesh(X, Y, C_sor);
title('SOR Method');
xlabel('X'); ylabel('Y'); zlabel('C_sor');
subplot(1,2,2);
mesh(X, Y, C_gauss);
title('Gauss-Seidel Method');
xlabel('X'); ylabel('Y'); zlabel('C_gauss');
drawnow;

% Kiem tra dieu kien dung khac
if nVirus == prevNVirus_sor && nVirus == prevNVirus_gauss
```

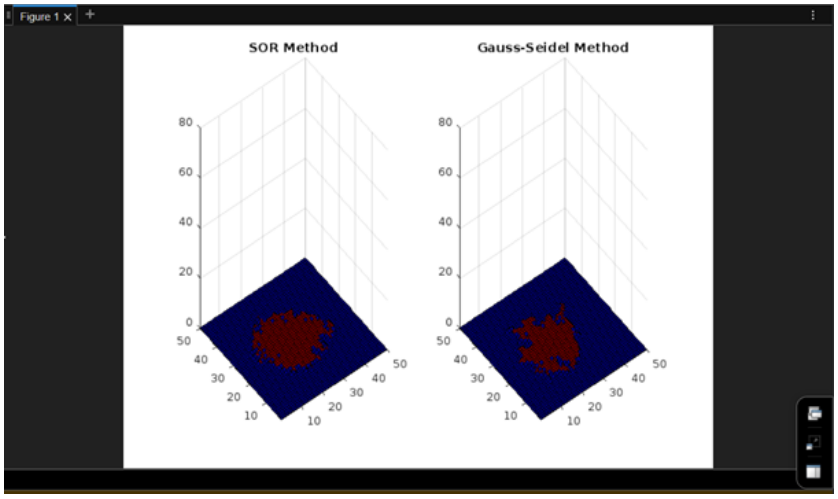
```
            break;
        end
        prevNVirus_sor = nVirus;
        prevNVirus_gauss = nVirus;
    end
end
```

Listing 1: Source Code in Matlab

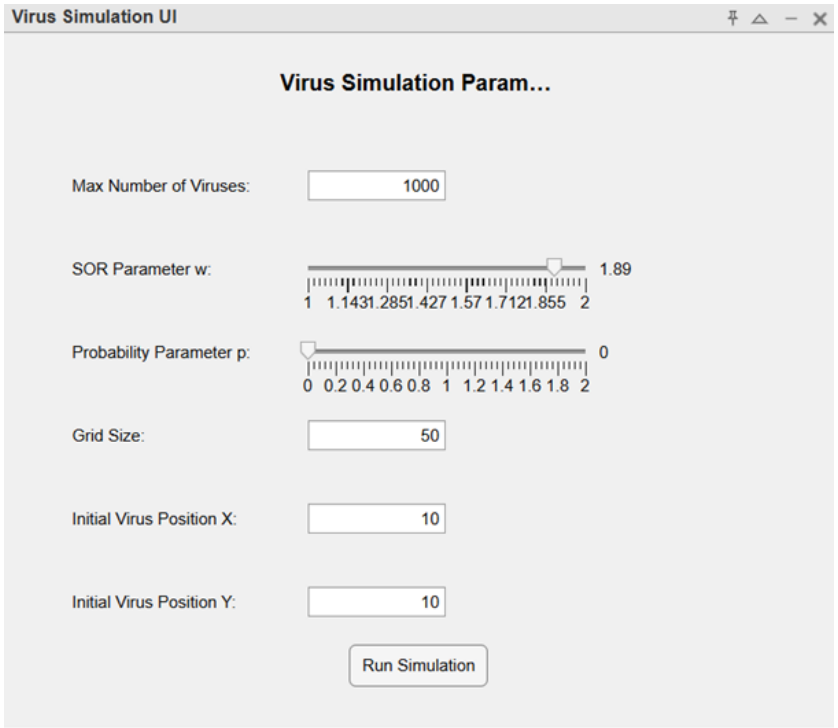
4 Kết quả



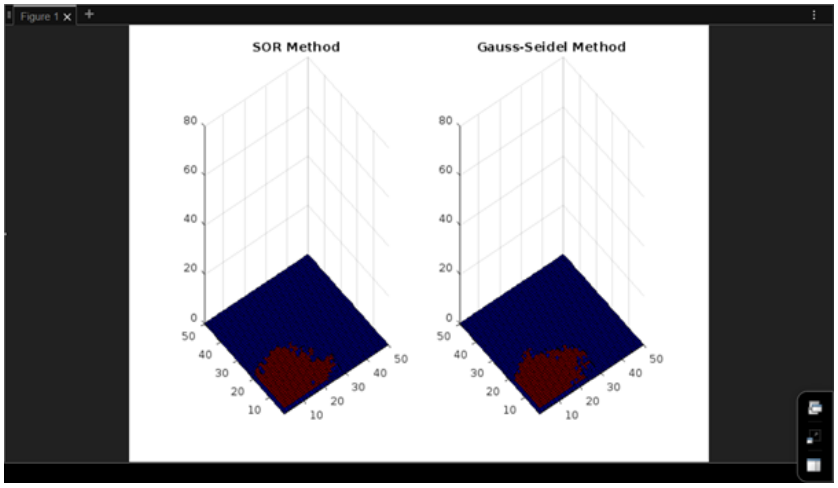
Hình 1: Tham số đầu vào



Hình 2: Kết quả nhận thấy



Hình 3: Tham số đầu vào



Hình 4: Kết quả nhận thấy

5 Kết luận

- Mặc dù SOR có thể làm tăng tốc độ hội tụ so với Gauss-Seidel, việc sử dụng nó cũng có nhược điểm của việc cần phải tinh chỉnh thêm hệ số trọng số ω và đôi khi có thể làm cho phương pháp không hội tụ hoặc hội tụ chậm nếu chọn sai giá trị của ω.
- Tuy nhiên, với sự kết hợp của hiểu biết về tính chất của bài toán cụ thể và các kỹ thuật tối ưu hóa, SOR vẫn là một công cụ quan trọng và mạnh mẽ trong giải quyết các bài toán tuyến tính trong thực tế.