Formation of spatial patterns in an epidemic model with constant removal rate of the infectives

CL 716 – Course Project

Kaushik Nagaraj
140020116

April 15, 2018

1 Problem definition

In the study of epidemics, a key area of interest has been to predict how diseases transmit in space. A model which has been suggested is the SIR model, which describes the infection and recovery process in terms of three differential equations for susceptibles (S), infected (I) and recovered (R). These give rise to a dynamic patterns in a two-dimensional space. In this project, an attempt has been made to perform a theoretical analysis of the model reported in a study by Liu and Jin and reproduce the patterns generated [1].

1.1 Kinetic model

The model is given by

$$\frac{\partial S}{\partial t} = A - dS - \lambda SI + D_s \nabla^2 S \tag{1}$$

$$\frac{\partial I}{\partial t} = \lambda SI - (d + \gamma)I - h(I) + D_i \nabla^2 I \tag{2}$$

$$\frac{\partial R}{\partial t} = \gamma I + h(I) - dR + D_r \nabla^2 R \tag{3}$$

$$h(I) = \begin{cases} r; & I > 0 \\ 0; & I = 0 \end{cases} \tag{4}$$

where S(t), I (t), and R(t) denote the numbers of susceptible, infective, and recovered individuals at time t, respectively, A is the recruitment rate of the population, d is the natural death rate of the population, γ is the natural recovery rate of the infective individuals. h(I) denotes the removal rate of infectives due to the treatment of infectives. $r > 0$ is a constant and represents the capacity of treatment for infectives. Since the first two equations are independent of the third one, and since the concern is only about the infectives and the susceptibles, it is enough if we consider only the first two equations.

The reduced model thus becomes

$$\frac{\partial S}{\partial t} = A - dS - \lambda SI + D_s \nabla^2 S \tag{5}$$

$$\frac{\partial I}{\partial t} = \lambda SI - (d+\gamma)I - r + D_i \nabla^2 I \tag{6}$$

2 Theoretical analysis

2.1 Fixed points and temporal stability

In order to find the fixed points, the nullclines are obtained, and then the intersection of the nullclines gives the fixed points. The nullclines are given by

$$S = \frac{A}{d+\lambda I} \tag{7}$$

$$\lambda SI - (d+\gamma)I - r = 0 \tag{8}$$

Equating the nullclines,

$$-\lambda(d+\gamma)I^2 + (\lambda A - r\lambda - \gamma d - d^2)I - rd = 0 \tag{9}$$

Setting

$$R = \frac{\lambda A}{d(d+\gamma)}; \qquad H = \frac{\lambda r}{d(d+\gamma)} \tag{10}$$

The quadratic equation in I is rewritten as

$$\frac{\lambda}{d}I^2 - (R-1-H)I + \frac{r}{d+\gamma} = 0 \tag{11}$$

The roots are

$$I_1 = \frac{d}{2\lambda}(R-1-H-\sqrt{(R-1-H)^2-4H}) \tag{12}$$

$$I_2 = \frac{d}{2\lambda}(R-1-H+\sqrt{(R-1-H)^2-4H}) \tag{13}$$

The Jacobian matrix is given by

$$J = \begin{pmatrix} -d-\lambda I & -\lambda S \\ \lambda I & \lambda S - d - \gamma \end{pmatrix} \tag{14}$$

The determinant of the Jacobian matrix is given by

$$det(J) = (-d - \lambda I)(\lambda S - d - \gamma) + \lambda^2 SI \tag{15}$$

$$S = \frac{A - (d+\gamma)I - r}{d} \tag{16}$$

Substituting Eq. 16 in Eq. 15,

$$\begin{aligned} det(J) &= -\lambda A + 2\lambda(d+\gamma)I + \lambda r + d(d+\gamma) \\ &= d(d+\gamma)[-R + \frac{2\lambda}{d}I + H + 1] \end{aligned} \tag{17}$$

The trace of the Jacobian matrix is given by

$$tr(J) = -d - \lambda I + \lambda S - (d+\gamma) \tag{18}$$

Substituting Eq. 16 in Eq. 18,

$$\begin{aligned} tr(J) &= -d - \lambda I + \frac{\lambda A}{d} - \frac{\lambda r}{d} - (d+\gamma) \\ &= -\left(\frac{2d\lambda + \gamma\lambda}{d}\right)I - \left(\frac{2d^2 - \lambda A + r\lambda + \gamma d}{d}\right) \end{aligned} \tag{19}$$

Now,

$$det(J(I_1)) = -d(d+\gamma)\sqrt{(R-1-H)^2 - 4H} < 0 \tag{20}$$

Therefore, I_1 gives rise to a saddle point, and therefore will not be able to give rise to spatio-temporal patterns.

$$det(J(I_2)) = d(d+\gamma)\sqrt{(R_0-1-H)^2 - 4H} > 0 \tag{21}$$

Therefore, I_2 would be stable if $tr(J(I_2)) < 0$.
The condition for temporal stability is given by:

$$tr(J(I_2)) < 0 \tag{22}$$

Substituting Eq.18 in Eq. 22,

$$-\left(\frac{2d\lambda + \gamma\lambda}{d}\right)I_2 - \left(\frac{2d^2 - \lambda A + r\lambda + \gamma d}{d}\right) < 0 \tag{23}$$

$$I_2 < -\frac{2d^2 - \lambda A + r\lambda + \gamma d}{2d\lambda + \gamma\lambda} \tag{24}$$

Now,

$$I_2 = \frac{A}{2(d+\gamma)} - \frac{d}{2\lambda} - \frac{r}{2(d+\gamma)} + \frac{d}{2\lambda}\left[\left(\frac{\lambda A - r\lambda}{d(d+\gamma)} - 1\right)^2 - \frac{4\lambda r}{d(d+\gamma)}\right]^{\frac{1}{2}} \tag{25}$$

Substituting Eq. 25 in Eq. 24, the inequality that we get is

$$\frac{A}{2(d+\gamma)} - \frac{d}{2\lambda} - \frac{r}{2(d+\gamma)} + \frac{d}{2\lambda}\left[\left(\frac{\lambda A - r\lambda}{d(d+\gamma)} - 1\right)^2 - \frac{4\lambda r}{d(d+\gamma)}\right]^{\frac{1}{2}} < -\frac{2d^2 - \lambda A + r\lambda + \gamma d}{2d\lambda + \gamma\lambda} \tag{26}$$

The above inequality was solved for r using Wolfram Mathematica, and the expression obtained is as follows

$$r > \frac{1}{2\lambda}\left[2A\lambda + 2d^2 + 3d\gamma + \gamma^2 - (2d+\gamma)\sqrt{4A\lambda + d^2 + 2d\gamma + \gamma^2}\right] \tag{27}$$

$$r\lambda > \frac{1}{2}\left[2A\lambda + (2d+\gamma)\left(d + \gamma - \sqrt{4A\lambda + (d+\gamma)^2}\right)\right] \tag{28}$$

Now, the feasible parameter space for temporal stability can be determined. For our case, λ and r are chosen as the variable parameters. The other parameters are fixed at values that consistent with Liu and Jin [1]. They are

$$A = 3, d = 0.3, \gamma = 0.8, D_s = 0.02, D_i = 0.05 \tag{29}$$

The feasible parameter space is now determined numerically and is shown in Figure 1.

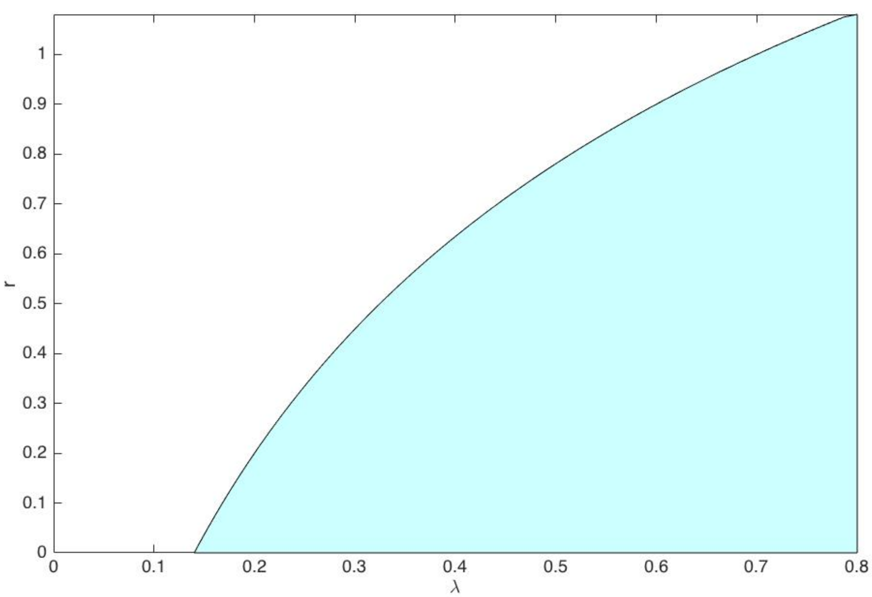

Figure 1: Temporal Stability Region of I_2 (A = 3, d = 0.3, γ = 0.8, D_s = 0.02, D_i = 0.05)

2.2 Diffusion-driven instability

To find the Turing instabilities, we assume perturbations about the fixed point (I_2, S_2). Let ŝ and î be the perturbations about S_2 and I_2 respectively.

Then,

$$
\begin{aligned}
I - I_2 = \hat{i} = exp((k_x x + k_y y)i + \delta_k t) \\
S - S_2 = \hat{s} = exp((k_x x + k_y y)i + \delta_k t)
\end{aligned}
\tag{30}
$$

Substituting Eq. 30 in our model (Eq. 5 and Eq. 6), we get

Now, the model is given by

$$
\begin{aligned}
\frac{\partial S}{\partial t} &= A - dS - \lambda SI + D_s \nabla^2 S = f(S, I) + D_s \nabla^2 S \\
\frac{\partial I}{\partial t} &= \lambda SI - (d + \gamma)I - r + D_i \nabla^2 I = g(S, I) + D_i \nabla^2 I
\end{aligned}
\tag{31}
$$

Now, using a Taylor series expansion, and neglecting second and higher order terms,

$$
\begin{aligned}
f(S, I) = f(S_2, I_2) + \frac{\partial f}{\partial S}(S - S_2) + \frac{\partial f}{\partial I}(I - I_2) \\
g(S, I) = g(S_2, I_2) + \frac{\partial g}{\partial S}(S - S_2) + \frac{\partial g}{\partial I}(I - I_2)
\end{aligned}
\tag{32}
$$

$$
f(S_2, I_2) = g(S_2, I_2) = 0 \tag{33}
$$

Therefore, substituting Eq. 30 and Eq. 33 in Eq. 32,

$$
\begin{aligned}
f(S, I) = \frac{\partial f}{\partial S}\hat{s} + \frac{\partial f}{\partial I}\hat{i} \\
g(S, I) = \frac{\partial g}{\partial S}\hat{s} + \frac{\partial g}{\partial I}\hat{i}
\end{aligned}
\tag{34}
$$

Substituting Eq. 34 in Eq. 31,

$$
\begin{aligned}
\frac{\partial S}{\partial t} &= \frac{\partial f}{\partial S}\hat{s} + \frac{\partial f}{\partial I}\hat{i} + D_s \nabla^2 S \\
\frac{\partial I}{\partial t} &= \frac{\partial g}{\partial S}\hat{s} + \frac{\partial g}{\partial I}\hat{i} + D_i \nabla^2 I
\end{aligned}
\tag{35}
$$

Now, Eq. 35 can be rewritten as

$$
(J(I_2) - \delta_k I - k^2 D).\begin{pmatrix} \hat{s} \\ \hat{i} \end{pmatrix} = 0 \tag{36}
$$

where k represents the wave numbers, δ_k represents the eigenvalues, I is the identity matrix and

$$D = \begin{pmatrix} D_s & 0 \\ 0 & D_i \end{pmatrix} \tag{37}$$

$$J(I_2) = \begin{pmatrix} j_{11} & j_{12} \\ j_{21} & j_{22} \end{pmatrix} = \begin{pmatrix} -d - \lambda I_2 & -\lambda S_2 \\ \lambda I_2 & \lambda S_2 - d - \gamma \end{pmatrix} \tag{38}$$

Now,

$$\begin{aligned} det(J(I_2) - \delta_k I - k^2 D) &= det\begin{pmatrix} j_{11} - k^2 D_s - \delta_k & j_{12} \\ j_{21} & j_{22} - k^2 D_i - \delta_k \end{pmatrix} \\ &= \delta_k^2 - [j_{11} + j_{22} - k^2(D_s + D_i)]\delta_k + h(k^2) \\ &= 0 \end{aligned} \tag{39}$$

where

$$h(k^2) = k^4 D_i D_s - (j_{11} D_i + j_{22} D_s)k^2 + (j_{11} j_{22} - j_{12} j_{21}) \tag{40}$$

Instability is possible either when:

- Coefficient of δ_{k} in Eq. 39 is negative, or,
- $\mathrm{h}(\mathrm{k}^2) < 0$

It can be shown that the coefficient of δ_{k} is always greater than 0. Therefore, the necessary condition for instability is:

$$h(k^2) < 0 \tag{41}$$

However, a sufficient condition for instability is

$$h(k^2) < h(k_c^2) \tag{42}$$

At $\mathrm{k} = \mathrm{k_c}$,

$$\frac{dh(k^2)}{dk^2} = 2k^2 D_i D_s - (j_{11} D_i + j_{22} D_s) = 0 \tag{43}$$

This gives us

$$k_c^2 = \frac{j_{11} D_i + j_{22} D_s}{2 D_i D_s} \tag{44}$$

Substituting Eq. 44 in Eq. 40,

$$h(k_c^2) = \frac{(j_{11} D_i + j_{22} D_s)^2}{4 D_i D_s} - \frac{(j_{11} D_i + j_{22} D_s)^2}{2 D_i D_s} + (j_{11} j_{22} - j_{12} j_{21}) < 0 \tag{45}$$

$$\frac{j_{22}D_s + j_{11}D_i}{4D_iD_s} > j_{11}j_{22} - j_{12}j_{21} \tag{46}$$

We now choose λ as our Turing bifurcation parameter. Taking other parameter values of A = 3, d = 0.3, r = 0.5, γ = 0.8, D_s = 0.02, and D_i = 0.0005, we can numerically solve the above inequality, as we know j_{pq} in terms of the above parameters.

We then get the critical value of the Turing bifurcation parameter λ_c as

$$\lambda_c = 0.547 \tag{47}$$

Substituting the above value in Eq. 44, we get

$$k_c = 17.05 \tag{48}$$

To depict the Turing space, Eq. 40 was used to plot $h(k^2)$ versus k^2 for different values of the turing bifurcation parameter λ, and is shown in Figure 2.

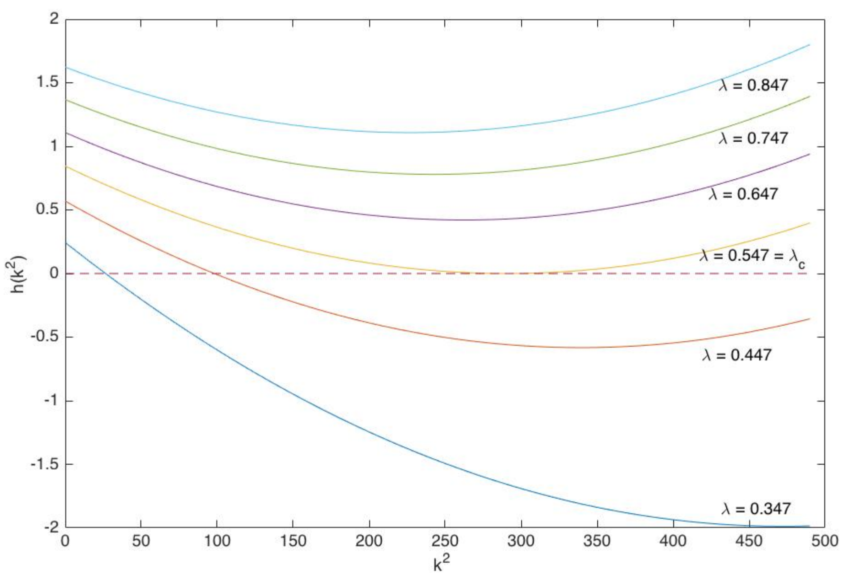

Figure 2: $h(k^2)$ versus k^2 for different values of k^2

The real part of the eigenvalue δ_k has also been plotted as a function of k^2 for different values of the Turing bifurcation parameter λ using Eq. 39. The same is depicted in Figure 3.

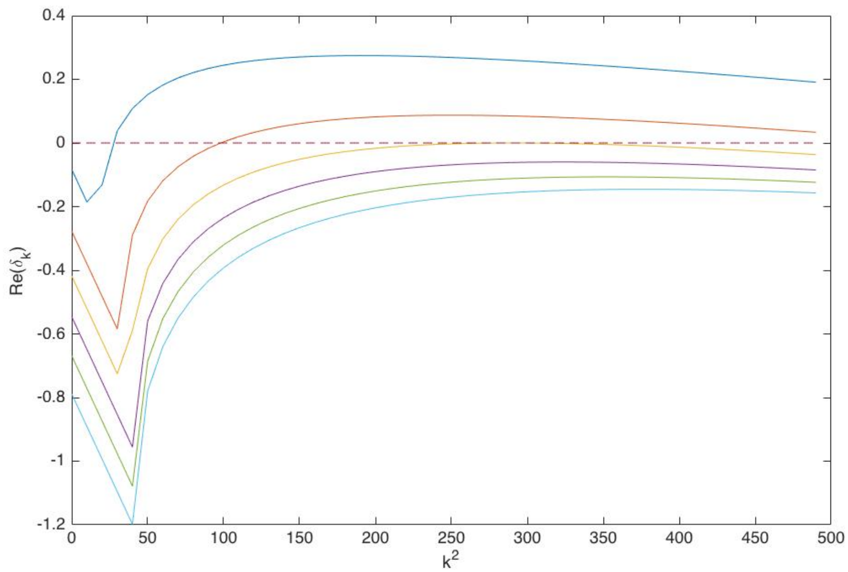

Figure 3: $\mathrm{Re}(\delta_\mathrm{k})$ versus k^2

By equating the $\mathrm{h}(\mathrm{k}^2) = 0$ for different values of λ, and plotting k versus λ, we can get the value of $\mathrm{k_c}$ and λ_c graphically, as shown in Figure 4.

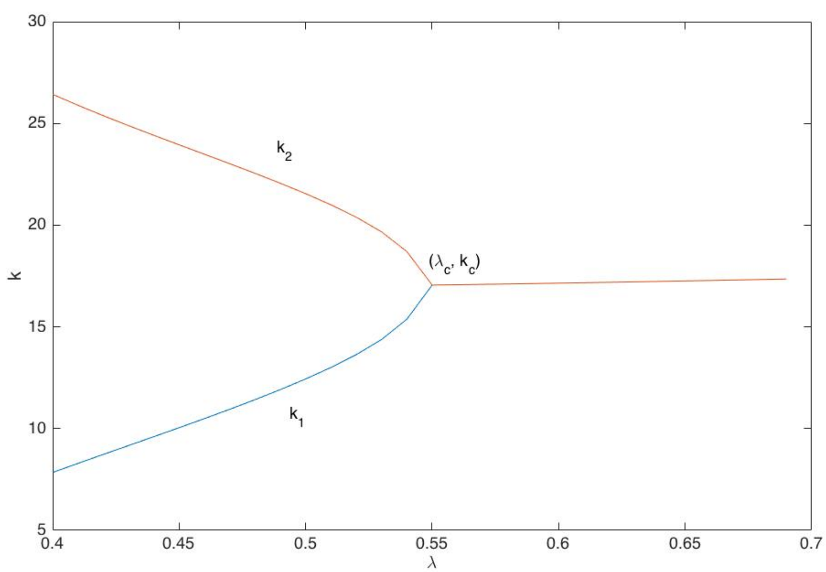

Figure 4: k versus λ

3 Computer simulations

It is observed that for values of the Turing bifurcation parameter λ below λ_c, we get an instability, as evident from Figure 2 and 3, and a stable solution otherwise. So, in order to obtain Turing patterns, we need to choose a value of the parameter $\lambda < \lambda_c = 0.547$. Furthermore, for pattern formation to occur, the value of λ must also fall under the region of temporal stability, governed by Eq. 28 and depicted in Figure 1. The above insights are used while choosing parameters for computer simulations leading to

- No pattern formation
- Pattern formation

3.1 No pattern formation

The value of λ chosen is 0.65. The values of the other parameters are A = 3, d = 0.3, r = 0.5, γ = 0.8, D_s = 0.02, and D_i = 0.0005. It can be verified that for the parameter values chosen above, the system is temporally stable and also spatially stable.

As expected, it is found that the system goes to the equilibrium value of I_2 = 1.687, as shown in Figure 5.

Figure 5: Snapshot of the contour picture of I. The parameter values are A = 3, d = 0.3, r = 0.5, γ = 0.8, λ = 0.65, D_s = 0.02, D_i = 0.0005

3.2 Pattern formation

The value of λ chosen is 0.5. The values of the other parameters are A = 3, d = 0.3, r = 0.5, $\gamma = 0.8$, $D_s = 0.02$, and $D_i = 0.0005$. It can be verified that for the parameter values chosen above, the system is temporally stable and spatially unstable. Hence, it gives rise to spatio-temporal patterns. This is depicted is Figure 6.

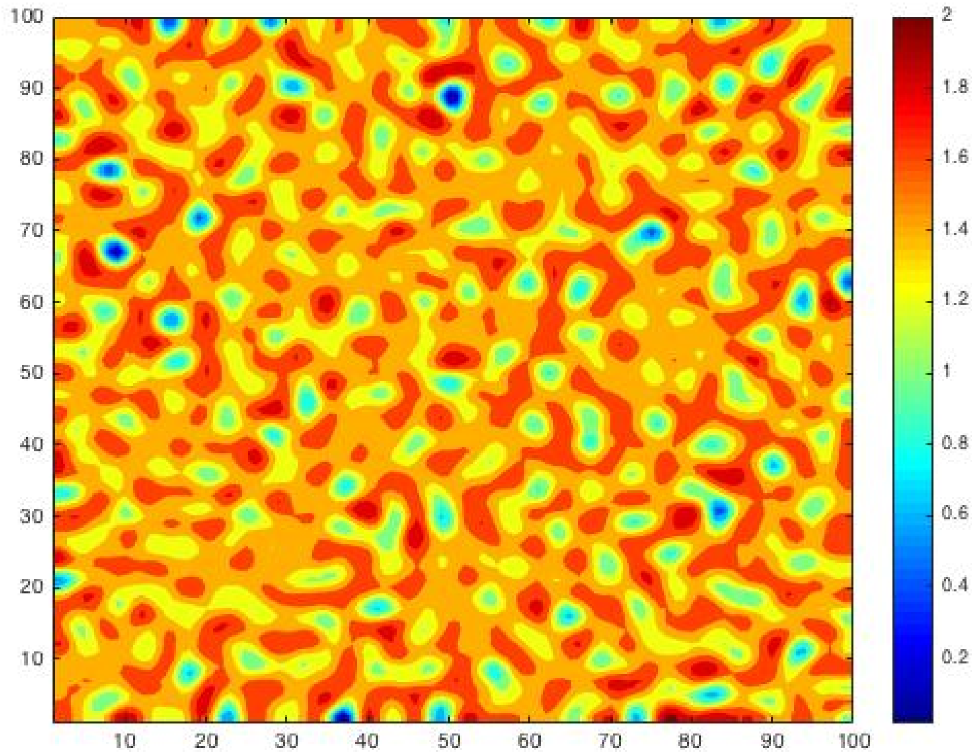

Figure 6: Snapshot of the contour picture of I after 1,000 iterations. The parameter values are A = 3, d = 0.3, r = 0.5, $\gamma = 0.8$, $\lambda = 0.5$, $D_s = 0.02$, $D_i = 0.0005$

In order to obtain the results, the model equation given by Eq. 5 and Eq. 6 are integrated using a simple Euler method. Forward differences are used for the time domain, and central differences are used for the spatial domains. Step sizes used are: $\Delta t = 0.01$, and $\Delta x = \Delta y = 0.05$. The numerical results are obtained in a spatial domain of size 100 x 100. The initial values used for the simulations are random perturbations about the fixed points. The simulations are carried out using the commercial software package MATLAB, by MathWorks.

References

1. Liu, Q.-X. & Jin, Z. Formation of spatial patterns in an epidemic model with constant removal rate of the infectives. *Journal of Statistical Mechanics: Theory and Experiment* **2007,** P05002 (2007).

A MATLAB code for temporal stability region generation

```
clearvars;
clc;
A = 3;
d = 0.3;
g = 0.8;
n = 100;
dl = 0.66/n;
l(1) = 0.14;
for i = 1:n
    r1(i) = (2*A*l(i) + (d+g)*(2*d+g) -(2*d+g)*sqrt(4*A*l(i)+(d+g)^2))/(2*l(i));
    l(i+1) = l(i) + dl;
end
l(i) = [];
area(l,r1, 'FaceColor',[.8 1 1]);
xlabel('\lambda');
ylabel('r');
xlim([0 0.8]);
ylim([0 1.0803]);
```

B MATLAB code for Turing bifurcation diagram generation

```
clearvars;
clc;

% Determination of lambda_c and kc
A = 3;
d = 0.3;
r = 0.5;
g = 0.8;
ds = 0.02;
di = 0.0005;
lc = 0.8;
lc = fsolve(@lambda_c, lc);
R = lc*A/(d*(d+g));
H = lc*r/(d*(d+g));
i = (d/(2*lc))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
s = A/(d + lc*i);
j11 = -d - lc*i;
j12 = -lc*s;
j21 = lc*i;
j22 = lc*s - d - g;
kc = sqrt((j11*di + j22*ds)/(2*di*ds));

% Bifurcation diagram 1
n = 1;
l(n) = 0.4;
dl = 0.01;
while l(n) < 0.7
    R = l(n)*A/(d*(d+g));
    H = l(n)*r/(d*(d+g));
    i = (d/(2*l(n)))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
    s = A/(d + l(n)*i);
    j11 = -d - l(n)*i;
    j12 = -l(n)*s;
    j21 = l(n)*i;
    j22 = l(n)*s - d - g;
    J = [j11 j12; j21 j22];
    k1(n) = sqrt((j11*di + j22*ds - sqrt((j11*di+j22*ds)^2 - 4*ds*di*det(J)))/(2*ds*di));
    k2(n) = sqrt((j11*di + j22*ds + sqrt((j11*di+j22*ds)^2 - 4*ds*di*det(J)))/(2*ds*di));
```

```
    n = n+1;
    l(n) = l(n-1) + dl;
end
l(n) = [];
plot(l,k1);
hold on;
plot(l,k2);

figure;

% Bifurcation diagram 2 & 3
dl = 0.1;
n = 1;
l1(n) = 0.3467;
dk_2 = 10;
while l1(n) < 0.9
    R = l1(n)*A/(d*(d+g));
    H = l1(n)*r/(d*(d+g));
    i = (d/(2*l1(n)))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
    s = A/(d + l1(n)*i);
    j11 = -d - l1(n)*i;
    j12 = -l1(n)*s;
    j21 = l1(n)*i;
    j22 = l1(n)*s - d - g;
    J = [j11 j12; j21 j22];
    q = 1;
    k_2(q) = 0;
    while k_2(q) < 500
        h(n,q) = k_2(q)*k_2(q)*di*ds - k_2(q)*(j22*ds+j11*di) +
           det(J);
        f = [1, -(j11 - k_2(q)*(ds+di) + j22), h(n,q)];
        delta = roots(f);
        re_delta(n,q) = real(delta(2));
        q = q + 1;
        k_2(q) = k_2(q-1) + dk_2;
    end
    n = n+1;
    l1(n) = l1(n-1) + dl;
end
k_2(q) = [];
l1(n) = [];
q = 1;
```

```
k_2(q) = 0;
while k_2(q) < 500
    h(n,q) = 0;
    re_delta(n,q) = 0;
    q = q + 1;
    k_2(q) = k_2(q-1) + dk_2;
end
k_2(q) = [];
for p = 1:n-1
    plot(k_2,h(p,:));
    hold on;
end
p = p+1;
plot(k_2,h(p,:),'--');
figure;

for p=1:n-1
    plot(k_2,re_delta(p,:));
    hold on;
end
p = p +1;
plot(k_2,re_delta(p,:),'--');
```

```
function [ err ] = lambda_c(l)
A = 3;
d = 0.3;
r = 0.5;
g = 0.8;
ds = 0.02;
di = 0.0005;
R = l*A/(d*(d+g));
H = l*r/(d*(d+g));
i = (d/(2*l))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
s = A/(d + l*i);
j11 = -d - l*i;
j12 = -l*s;
j21 = l*i;
j22 = l*s - d - g;
lhs = ((j11*di + j22*ds)^2)/(4*di*ds);
rhs = j11*j22 - j12*j21;
err = abs(lhs-rhs);
end
```

C MATLAB code for no pattern formation

```
clearvars;
clc;

% No Pattern formation
A = 3;
d = 0.3;
r = 0.5;
g = 0.8;
ds = 0.02;
di = 0.0005;
l = 0.65;
dx = 0.05;
dy = 0.05;
dt = 0.01;
R = l*A/(d*(d+g));
H = l*r/(d*(d+g));
i0 = (d/(2*l))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
s0 = A/(d + l*i0);
nx = 5/dx;
ny = 5/dy;
nt = 20/dt;
i(nt,nx,ny) = 0;
s(nt,nx,ny) = 0;
for p=1:nx
    for q=1:ny
        i(1,p,q) = i0*(1+(-1)^p*(-1)^q*rand);
        s(1,p,q) = s0*(1+(-1)^p*(-1)^q*rand);
    end
end
for m=1:nt-1
    for p=2:nx-1
        for q=2:ny-1
            i(m+1,p,q) = i(m,p,q) + dt*(l*s(m,p,q)*i(m,p,q) - (d+g)*i(m,p,q) - r + di*((i(m,p,q+1)+i(m,p,q-1)-2*i(m,p,q))/(dy*dy) + (i(m,p+1,q)+i(m,p-1,q)-2*i(m,p,q))/(dx*dx)));
            s(m+1,p,q) = s(m,p,q) + dt*(-l*s(m,p,q)*i(m,p,q) - d*s(m,p,q) + A + ds*((s(m,p,q+1)+s(m,p,q-1)-2*s(m,p,q))/(dy*dy) + (s(m,p+1,q)+s(m,p-1,q)-2*s(m,p,q))/(dx*dx)));
```

```
            end
            i(m+1,p,1) = i(m+1,p,2);
            i(m+1,p,ny) = i(m+1,p,ny-1);
            s(m+1,p,1) = s(m+1,p,2);
            s(m+1,p,ny) = s(m+1,p,ny-1);
        end
        for q=1:ny
            i(m+1,1,q) = i(m+1,2,q);
            i(m+1,nx,q) = i(m+1,nx-1,q);
            s(m+1,1,q) = s(m+1,2,q);
            s(m+1,nx,q) = s(m+1,nx-1,q);
        end
        disp(m);
end
iplot(nx,ny) = 0;
for p=1:nx
        for q=1:ny
            iplot(p,q) = i(m,p,q);
        end
end
HeatMap(iplot);
```

D MATLAB code for pattern formation

```
clearvars;
clc;

% Pattern formation
A = 3;
d = 0.3;
r = 0.5;
g = 0.8;
ds = 0.02;
di = 0.0005;
l = 0.5;
dx = 0.05;
dy = 0.05;
dt = 0.01;
R = l*A/(d*(d+g));
H = l*r/(d*(d+g));
i0 = (d/(2*l))*(R - 1 - H + sqrt((R-1-H)^2 - 4*H));
s0 = A/(d + l*i0);
nx = 5/dx;
ny = 5/dy;
nt = 10/dt;
i(nt,nx,ny) = 0;
s(nt,nx,ny) = 0;
for p=1:nx
    for q=1:ny
        i(1,p,q) = i0*(1+(-1)^p*(-1)^q*rand);
        s(1,p,q) = s0*(1+(-1)^p*(-1)^q*rand);
        %         i(1,p,q) = i0+1;
        %         s(1,p,q) = s0+1;
    end
end
for m=1:nt-1
    for p=2:nx-1
        for q=2:ny-1
            i(m+1,p,q) = i(m,p,q) + dt*(l*s(m,p,q)*i(m,p,q) - (d+
                g)*i(m,p,q) - r + di*((i(m,p,q+1)+i(m,p,q-1)-2*i(m
                ,p,q))/(dy*dy) + (i(m,p+1,q)+i(m,p-1,q)-2*i(m,p,q)
                )/(dx*dx)));
            s(m+1,p,q) = s(m,p,q) + dt*(-l*s(m,p,q)*i(m,p,q) - d*
                s(m,p,q) + A + ds*((s(m,p,q+1)+s(m,p,q-1)-2*s(m,p,
```

```
                  q))/(dy*dy) + (s(m,p+1,q)+s(m,p-1,q)-2*s(m,p,q))/(
                  dx*dx)));
        end
        i(m+1,p,1) = i(m+1,p,2);
        i(m+1,p,ny) = i(m+1,p,ny-1);
        s(m+1,p,1) = s(m+1,p,2);
        s(m+1,p,ny) = s(m+1,p,ny-1);
    end
    for q=1:ny
        i(m+1,1,q) = i(m+1,2,q);
        i(m+1,nx,q) = i(m+1,nx-1,q);
        s(m+1,1,q) = s(m+1,2,q);
        s(m+1,nx,q) = s(m+1,nx-1,q);
    end
    disp(m);
end
iplot(nx,ny) = 0;
for p=1:nx
    for q=1:ny
        iplot(p,q) = i(m,p,q);
    end
end
contourf(iplot,'LineColor','none');
colormap(jet);
```