Mathematical Modelling in the Natural Sciences SS21
Solutions to Exercises on Sheet 3

Exercises und Lecture Notes

Contents

• Exercise 1: Expected Values, SIR Model

◦ Task

Suppose there exist probabilities $s_n(t) = P(X(t) = S)$, $i_n(t) = P(X(t) = I)$ and $r_n(t) = P(X(t) = R)$ for the stochastic process $X(t) : [0, \infty) \to \{S, I, R\}$,

$$\{(s_n(t), i_n(t), r_n(t) : t \geq 0, n \in \mathbb{N}_0)\} : [0, \infty) \to \mathcal{C}^1([0, \infty))^\infty$$

solving the infinite system of differential equations,

$$\begin{cases} s_n'(t) &= \beta s_{n-1}(t) - \{\beta + n[\mu + \epsilon + \lambda \bar{I}(t)]\} s_n(t) + (n+1)[\mu + \epsilon + \lambda \bar{I}(t)] s_{n+1}(t) \\ i_n'(t) &= [(n-1)\lambda + \epsilon] \bar{S}(t) i_{n-1}(t) - [(n\lambda + \epsilon)\bar{S}(t) + n(\mu + \gamma)] i_n(t) + (n+1)(\mu + \gamma) i_{n+1}(t) \\ r_n'(t) &= \gamma \bar{I}(t) r_{n-1}(t) - [\gamma \bar{I}(t) + n\mu] r_n(t) + (n+1)\mu r_{n+1}(t) \\ & \bar{S}(t) = \sum_{n=0}^{\infty} n s_n(t), \quad \bar{I}(t) = \sum_{n=0}^{\infty} n i_n(t), \quad \bar{R}(t) = \sum_{n=0}^{\infty} n r_n(t) \end{cases}$$

for which the expected values $\bar{S}(t), \bar{I}(t), \bar{R}(t) : [0,\infty) \to \mathcal{C}^1([0,\infty))$ are well defined and satisfy

$$\bar{S}'(t) = \sum_{n=0}^{\infty} n s_n'(t), \quad \bar{I}'(t) = \sum_{n=0}^{\infty} n i_n'(t), \quad \bar{R}'(t) = \sum_{n=0}^{\infty} n r_n'(t)$$

and additionally

$$\frac{d}{dt}\sum_{n=0}^{\infty} s_n(t) = \sum_{n=0}^{\infty} s_n'(t), \quad \frac{d}{dt}\sum_{n=0}^{\infty} i_n(t) = \sum_{n=0}^{\infty} i_n'(t), \quad \frac{d}{dt}\sum_{n=0}^{\infty} r_n(t) = \sum_{n=0}^{\infty} r_n'(t).$$

Derive a 3×3 system of ODEs for $(\bar{S}, \bar{I}, \bar{R})$. Implement the model for an ever larger number of possible states and compare the expected values $\bar{S}$, $\bar{I}$ and $\bar{R}$ computed from the (stochastic) truncated infinite system with those computed from the (deterministic) 3×3 system.

◦ Solution

For convenience let the explicit dependencies upon t be suppressed in the following. The expected value $\bar{S}$ satisfies:

$$\begin{aligned}
\bar{S}' = \sum_{n=0}^{\infty} n s_n' &= \beta \sum_{n=1}^{\infty} n s_{n-1} - \beta \sum_{n=0}^{\infty} n s_n - [\mu + \epsilon + \lambda \bar{I}] \sum_{n=0}^{\infty} n^2 s_n + [\mu + \epsilon + \lambda \bar{I}] \sum_{n=0}^{\infty} n(n+1) s_{n+1} \\
&= \beta \underbrace{\left\{ \sum_{n=1}^{\infty} s_{n-1} \right\}}_{=1} + \beta \underbrace{\left\{ \sum_{n=1}^{\infty} (n-1) s_{n-1} - \sum_{n=0}^{\infty} n s_n \right\}}_{=0} \\
&+ [\mu + \epsilon + \lambda \bar{I}] \underbrace{\left\{ \sum_{n=0}^{\infty} (n+1)^2 s_{n+1} - \sum_{n=0}^{\infty} n^2 s_n \right\}}_{=0} - [\mu + \epsilon + \lambda \bar{I}] \underbrace{\left\{ \sum_{n=0}^{\infty} (n+1) s_{n+1} \right\}}_{=\bar{S}}
\end{aligned}$$

or

$$\bar{S}' = \beta - [\mu + \epsilon + \lambda \bar{I}] \bar{S}$$

The expected value $\bar{I}$ satisfies:

$$\begin{aligned}
\bar{I}' = \sum_{n=0}^{\infty} n i_n' &= \lambda \bar{S} \sum_{n=1}^{\infty} (n-1) n i_{n-1} + \epsilon \bar{S} \sum_{n=1}^{\infty} n i_{n-1} - \epsilon \bar{S} \sum_{n=0}^{\infty} n i_n \\
&+ (\mu + \gamma) \sum_{n=0}^{\infty} n(n+1) i_{n+1} - [\lambda \bar{S} + (\mu + \gamma)] \sum_{n=0}^{\infty} n^2 i_n \\
&= \lambda \bar{S} \underbrace{\left\{ \sum_{n=1}^{\infty} (n-1) i_{n-1} \right\}}_{=\bar{I}} + \lambda \bar{S} \underbrace{\left\{ \sum_{n=1}^{\infty} (n-1)^2 i_{n-1} - \sum_{n=0}^{\infty} n^2 i_n \right\}}_{=0} \\
&+ \epsilon \bar{S} \underbrace{\left\{ \sum_{n=1}^{\infty} (n-1) i_{n-1} - \sum_{i=0}^{\infty} n i_n \right\}}_{=0} + \epsilon \bar{S} \underbrace{\left\{ \sum_{i=1}^{\infty} i_{n-1} \right\}}_{=1} \\
&+ (\mu + \gamma) \underbrace{\left\{ \sum_{n=0}^{\infty} (n+1)^2 i_{n+1} - \sum_{n=0}^{\infty} n^2 i_n \right\}}_{=0} - (\mu + \gamma) \underbrace{\left\{ \sum_{n=0}^{\infty} (n+1) i_{n+1} \right\}}_{=\bar{I}}
\end{aligned}$$

or

$$\bar{I}' = [\lambda \bar{S} - (\mu + \gamma)] \bar{I} + \epsilon \bar{S}$$

The expected value $\bar{R}$ satisfies:

$$\begin{aligned}\bar{R}' = \sum_{n=0}^{\infty} n r_n' &= \gamma\bar{I}\sum_{n=1}^{\infty} n r_{n-1} - \gamma\bar{I}\sum_{n=0}^{\infty} n r_n - \mu\sum_{n=0}^{\infty} n^2 r_n + \mu\sum_{n=0}^{\infty} n(n+1) r_{n+1} \\ &= \gamma\bar{I}\left\{\sum_{n=1}^{\infty} r_{n-1}\right\}_{=1} + \gamma\bar{I}\left\{\sum_{n=1}^{\infty}(n-1) r_{n-1} - \sum_{n=0}^{\infty} n r_n\right\}_{=0} \\ &+ \mu\left\{\sum_{n=0}^{\infty}(n+1)^2 r_{n+1} - \sum_{n=0}^{\infty} n^2 r_n\right\}_{=0} - \mu\left\{\sum_{n=0}^{\infty}(n+1) r_{n+1}\right\}_{=\bar{R}}\end{aligned}$$

or

$$\bar{R}' = \gamma\bar{I} - \mu\bar{R}$$

The following Matlab code is used to perform simulations for the truncated infinite system and for the 3×3 system.

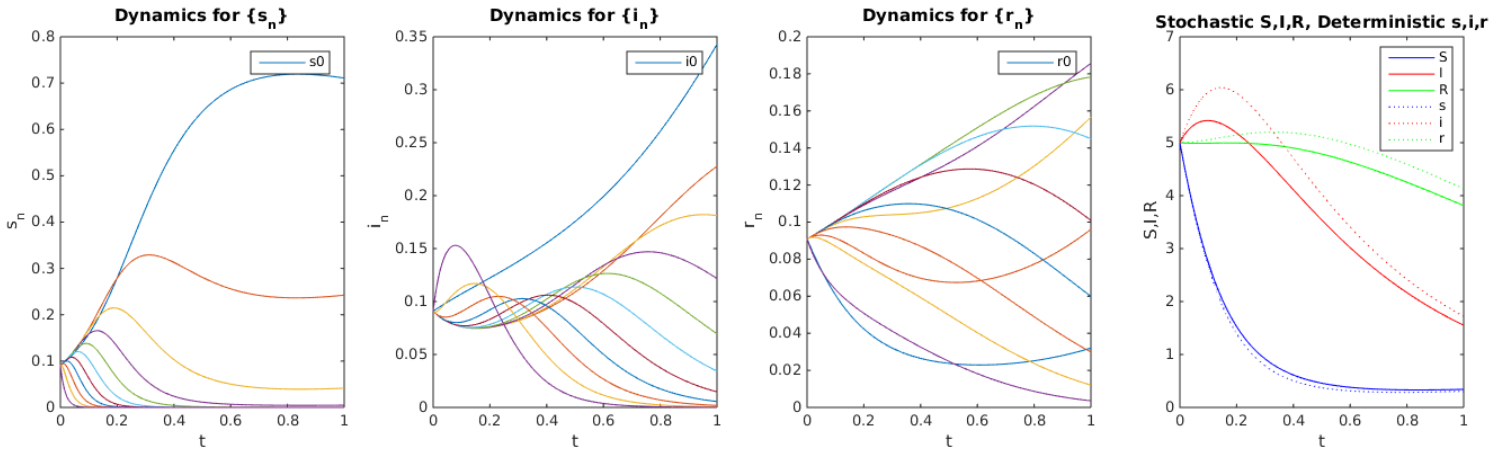
```
% setup figure
    h1 = figure(1);
    close(h1);
    h1 = figure(1);
    set(h1,'Position',[10 10 1600 400]);

% temporal parameters
    T  = 1;
    Nt = 1000;
    dt = T/Nt;
    th = 0.5;

% model parameters
    N  = 50;
    be = 1.0e0;
    mu = 1.0e0;
    ga = 1.0e0;
    la = 1.0e0;
    ep = 0.0e0;

% identity
    I  = speye(N+1);

% initial conditions
    N0 = min(11,N+1);
    s  = zeros(N+1,1); s(1:N0) = 1/N0;
    i  = zeros(N+1,1); i(1:N0) = 1/N0;
    r  = zeros(N+1,1); r(1:N0) = 1/N0;
    sb = dot(s,(0:N));
    ib = dot(i,(0:N));
    rb = dot(r,(0:N));

% starting stochastic values
```

```
    sv = s;
    iv = i;
    rv = r;
    tv = 0;

% starting expected values
    Sb = sb;
    Ib = ib;
    Rb = rb;
    tb = 0;

% starting deterministic values
    Sd = Sb;
    Id = Ib;
    Rd = Rb;
    Sv = Sd;
    Iv = Id;
    Rv = Rd;

% start time stepping
    for k=1:Nt

        A = -spdiags(be + ((mu+ep) + la*ib)*(0:N)', 0,N+1,N+1) ...
          +  spdiags(               be*ones(N+1,1),-1,N+1,N+1) ...
          +  spdiags(     ((mu+ep) + la*ib)*(0:N)', 1,N+1,N+1);
        A(N+1,N+1) = -N*((mu+ep) + la*ib);

        B = -spdiags(ep*sb + (la*sb + mu+ga)*(0:N)', 0,N+1,N+1) ...
          +  spdiags(          ep*sb + la*sb*(0:N)',-1,N+1,N+1) ...
          +  spdiags(                (mu+ga)*(0:N)', 1,N+1,N+1);
        B(N+1,N+1) = -N*(mu+ga);

        C = -spdiags(ga*ib + mu*(0:N)', 0,N+1,N+1) ...
          +  spdiags(ga*ib*ones(N+1,1),-1,N+1,N+1) ...
          +  spdiags(        mu*(0:N)', 1,N+1,N+1);
        C(N+1,N+1) = -N*mu;

% implicit time stepping for probabilities
        s = (I - th*dt*A) \ ((I + (1-th)*dt*A)*s);
        i = (I - th*dt*B) \ ((I + (1-th)*dt*B)*i);
        r = (I - th*dt*C) \ ((I + (1-th)*dt*C)*r);

% keep s,i,r in [0,1] and maintain probability distributions
        s = max(min(s,1),0);
        s = s/sum(s);
        i = max(min(i,1),0);
        i = i/sum(i);
        r = max(min(r,1),0);
```

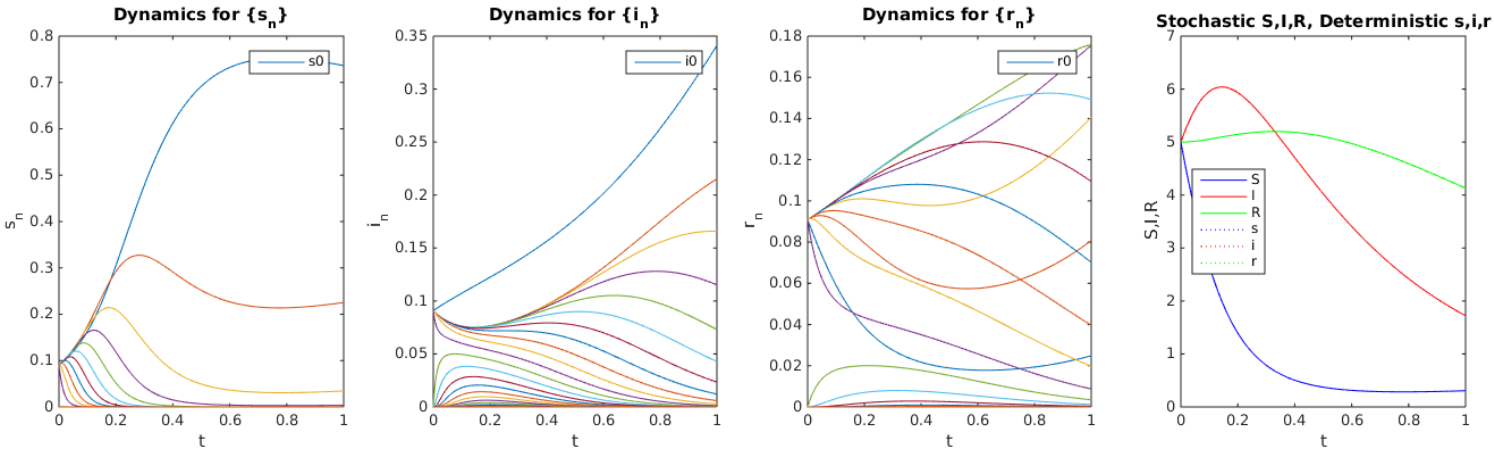
```
        r = r/sum(r);

% save values for plotting
        sv = [sv,s];
        iv = [iv,i];
        rv = [rv,r];
        tv = [tv,k*dt];

% current expected values
        sb = dot(s,(0:N));
        ib = dot(i,(0:N));
        rb = dot(r,(0:N));

        Sb = [Sb,sb];
        Ib = [Ib,ib];
        Rb = [Rb,rb];
        tb = [tb,k*dt];

% implicit time stepping for deterministic states
% S' = be   - (mu + ep + la*I)*S
% I' = ep*S - (mu + ga - la*S)*I
% R' = ga*I - mu*R
        dS = be;     A = mu + ep + la*Id;
        dI = ep*Sd;  B = mu + ga - la*Sd;
        dR = ga*Id;  C = mu;
        Sd = (Sd + dt*dS - (1-th)*dt*A.*Sd)./(1 + th*dt*A);
        Id = (Id + dt*dI - (1-th)*dt*B.*Id)./(1 + th*dt*B);
        Rd = (Rd + dt*dR - (1-th)*dt*C.*Rd)./(1 + th*dt*C);

% store deterministic state for plotting
        Sv = [Sv,Sd];
        Iv = [Iv,Id];
        Rv = [Rv,Rd];

    end

    disp(['relative Sb-Sd= ',num2str(norm(Sv-Sb)/norm(Sb))])
    disp(['relative Ib-Id= ',num2str(norm(Iv-Ib)/norm(Ib))])
    disp(['relative Rb-Rd= ',num2str(norm(Rv-Rb)/norm(Rb))])

% graphical representation

    subplot(1,4,1)
    plot(tv,sv)
    legend('s0')
    xlabel('t')
    ylabel('s_n');
    title('Dynamics for \{s_n\}')
```

```
subplot(1,4,2)
plot(tv,iv)
legend('i0')
xlabel('t')
ylabel('i_n');
title('Dynamics for \{i_n\}')

subplot(1,4,3)
plot(tv,rv)
legend('r0')
xlabel('t')
ylabel('r_n');
title('Dynamics for \{r_n\}')

subplot(1,4,4)
plot(tb,Sb,'b',tb,Ib,'r',tb,Rb,'g',tb,Sv,'b:',tb,Iv,'r:',tb,Rv,'g:')
legend('S','I','R','s','i','r','Location','Best')
xlabel('t')
ylabel('S,I,R');
title('Stochastic S,I,R, Deterministic s,i,r')
```

The following results were obtained with $N = 20$, and the differences between the stochastic and the determinsitic solutions are of the order 10^{-1}.



The following results were obtained with $N = 50$, and the differences between the stochastic and the determinsitic solutions are of the order 10^{-12}.



Note that for both of these cases the same initial conditions were used so that the results could be compared fairly. These computational results are consistent with the theoretical result that the differences should converge to zero for $N \to \infty$.

• Exercise 2: Endemic State Convergence, SIR Model

◦ Task

Suppose there exists a solution $\{(s_n(t), i_n(t), r_n(t) : t \geq 0, n \in \mathbb{N}_0)\} : [0, \infty) \to \mathcal{C}^1([0, \infty))^\infty$ to the infinite system of differential equations shown above for which the expected values $\bar{S}(t), \bar{I}(t), \bar{R}(t) : [0, \infty) \to \mathcal{C}^1([0, \infty))$ are well defined and satisfy the properties stated above. Recall the endemic equilibrium state

$$S_2^\star = \frac{\mu + \gamma}{\lambda}, \quad I_2^\star = \frac{\beta}{\mu + \gamma} - \frac{\mu}{\lambda}, \quad R_2^\star = \frac{\gamma}{\mu}\left[\frac{\beta}{\mu + \gamma} - \frac{\mu}{\lambda}\right]$$

and suppose further with $I_2^\star > 0$ that the following hold:

$$i_n(0) \in (0,1), \quad \forall n \in \mathbb{N}_0, \quad \sum_{n=0}^{\infty} i_n(0) = 1 \quad \text{and} \quad (\bar{S}(0), \bar{I}(0), \bar{R}(0)) = (S_2^*, I_2^*, R_2^*).$$

Show that $i_n(t) \to \delta_{n,0}$, $t \to \infty$. Implement the truncated model for an ever larger number of possible states starting always with an initial condition corresponding to the endemic equilibrium.

◦ Solution

It must first be shown that from $i_n(0) \in (0,1)$ it follows that $i_n(t) \in (0,1)$, $\forall t \geq 0$. Recalling the system of differential equations derived above for the expected values, it follows with the given initial conditions that the endemic equilibrium state is maintained by the expected values,

$$(\bar{S}(t), \bar{I}(t), \bar{R}(t)) = (S_2^*, I_2^*, R_2^*), \quad \forall t \geq 0.$$

With $S_2^\star = (\mu+\gamma)/\lambda$ the differential equation for i_n becomes simplified as

$$i_n'(t) + 2n(\mu+\gamma) i_n(t) = (\mu+\gamma)[(n-1) i_{n-1}(t) + (n+1) i_{n+1}(t)]$$

Using the integration factor $e^{2n(\mu+\gamma)t}$ gives

$$\begin{aligned} [e^{2n(\mu+\gamma)t} i_n(t)]' &= e^{2n(\mu+\gamma)t}(\mu+\gamma)[(n-1) i_{n-1}(t) + (n+1) i_{n+1}(t)] \\ &= (n-1)(\mu+\gamma) e^{2(\mu+\gamma)t} [e^{2(n-1)(\mu+\gamma)t} i_{n-1}(t)] \\ &+ (n+1)(\mu+\gamma) e^{-2(\mu+\gamma)t} [e^{2(n+1)(\mu+\gamma)t} i_{n+1}(t)] \end{aligned}$$

Defining $y_n(t) = e^{2n(\mu+\gamma)t} i_n(t)$ gives

$$y_n'(t) = A_{n-1}(t) y_{n-1}(t) + B_{n+1}(t) y_{n+1}(t)$$

where

$$A_n(t) = n(\mu+\gamma) e^{+2(\mu+\gamma)t} > 0, \quad B_n(t) = n(\mu+\gamma) e^{-2(\mu+\gamma)t} > 0.$$

Since $y_n(0) > 0$ holds $\forall n \in \mathbb{N}_0$, it follows from the differential equation for y_n that $y_n'(0) > 0$ holds $\forall n \in \mathbb{N}_0$. So it must be that $y_n'(t) > 0$ and so $y(t) > 0$ both hold $\forall t \geq 0$. Therefore,

$$i_n(t) = e^{-2n(\mu+\gamma)t} y_n(t) > 0, \quad \forall t \geq 0, \quad \forall n \in \mathbb{N}_0.$$

Summing the differential equations for i_n gives

$$\sum_{n=0}^{\infty} i_n' = \lambda S_2^\star \underbrace{\left\{ \sum_{n=1}^{\infty} (n-1) i_{n-1} - \sum_{n=0}^{\infty} n i_n \right\}}_{=0} + (\mu+\gamma) \underbrace{\left\{ \sum_{n=0}^{\infty} (n+1) i_{n+1} - \sum_{n=0}^{\infty} n i_n \right\}}_{=0} = 0$$

and combining this with the initial condition gives

$$\frac{d}{dt} \sum_{n=0}^{\infty} i_n(t) = \sum_{n=0}^{\infty} i_n'(t) = 0, \quad \sum_{n=0}^{\infty} i_n(0) = 1, \quad \Rightarrow \quad \sum_{n=0}^{\infty} i_n(t) = 1, \quad \forall t \geq 0.$$

It follows that

$$0 < i_m(t) < \sum_{n=0}^{\infty} i_n(t) = 1, \quad \forall m \in \mathbb{N}_0.$$

Returning to the differential equations for i_n and beginning with $n = 0$,

$$2 \geq |i_0(t) - i_0(0)| = \left| \int_0^t i_0'(s)ds \right| = (\mu + \gamma) \left| \int_0^t i_1(s)ds \right| = (\mu + \gamma) \int_0^t i_1(s)ds$$

one obtains

$$0 < \int_0^\infty i_1(t)dt < \infty \quad \Rightarrow \quad i_1(t) \overset{t\to\infty}{\longrightarrow} 0.$$

For $n = 1$,

$$2 + 2(\mu + \gamma) \int_0^\infty i_1(t)dt \geq \left| \int_0^t i_1'(s)ds + 2(\mu + \gamma) \int_0^t i_1(s)ds \right| = 2(\mu + \gamma) \int_0^t i_2(s)ds$$

one obtains

$$0 < \int_0^\infty i_2(t)dt < \infty \quad \Rightarrow \quad i_2(t) \overset{t\to\infty}{\longrightarrow} 0.$$

The cases for $n > 1$ are similar, so it holds that

$$i_n(t) \overset{t\to\infty}{\longrightarrow} 0, \quad \forall n \in \mathbb{N}.$$

For $n = 0$, note that $i_0 = y_0$ is increasing. Suppose there is an $\epsilon \in (0, 1)$ for which i_0 increases and satisfies

$$0 < i_0(t) \to 1 - \epsilon < 1, \quad t \to \infty.$$

Choose N large enough so that $I_2^\star < N\epsilon$. Then

$$\begin{aligned} \frac{1}{N} I_2^\star = \frac{1}{N} \sum_{n=0}^\infty n i_n(t) &\geq \frac{1}{N} \sum_{n=N}^\infty n i_n(t) \geq \sum_{n=N}^\infty i_n(t) \\ &= 1 - i_0(t) - \sum_{i=1}^{N-1} i_n(t) \overset{t\to\infty}{\longrightarrow} \epsilon. \end{aligned}$$

and the contradiction implies that

$$i_0(t) \overset{t\to\infty}{\longrightarrow} 0.$$

The following Matlab code is used to perform simulations for the truncated model.

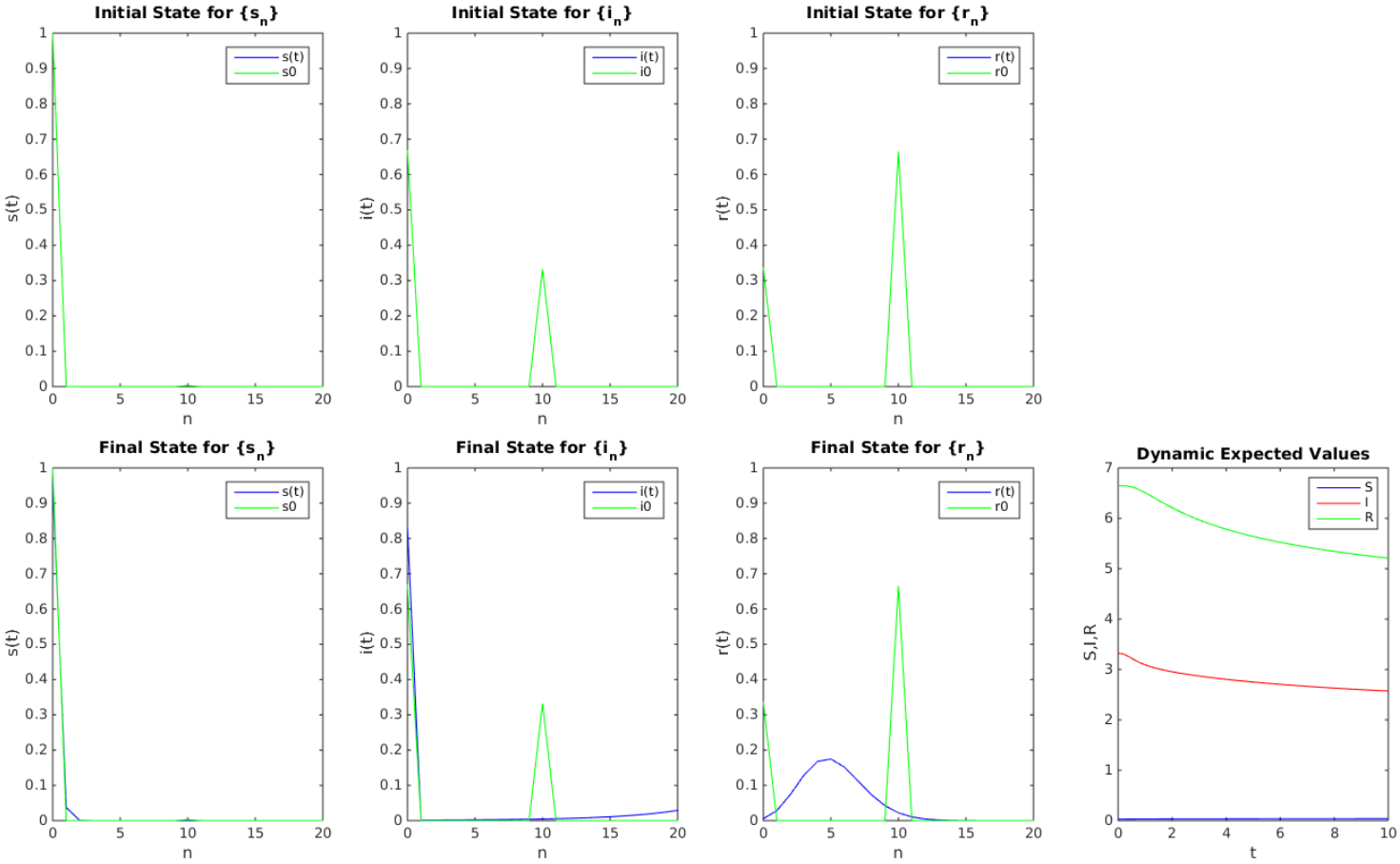
```
% setup figure
    h1 = figure(1);
    close(h1);
    h1 = figure(1);
    set(h1,'Position',[10 10 1600 400]);

% temporal parameters
    T  = 10;
    Nt = 100;
    dt = T/Nt;
    th = 0.5;

% model parameters
    N  = 50;
```

```
    be = 1.0e1;
    mu = 1.0e0;
    ga = 2.0e0;
    la = 1.0e2;

% identity
    I  = speye(N+1);

% endemic equilbrium state
    S2 = (mu+ga)/la;
    I2 = be/(mu+ga) - mu/la;
    R2 = I2*ga/mu;

% initial conditions
    N0 = min(10,N);
    s  = zeros(N+1,1); s(N0+1) = S2/N0; s(1) = 1-sum(s(2:(N+1)));
    i  = zeros(N+1,1); i(N0+1) = I2/N0; i(1) = 1-sum(i(2:(N+1)));
    r  = zeros(N+1,1); r(N0+1) = R2/N0; r(1) = 1-sum(r(2:(N+1)));
    s0 = s;
    i0 = i;
    r0 = r;
    sb = dot(s,(0:N));
    ib = dot(i,(0:N));
    rb = dot(r,(0:N));

% starting expected values
    Sb = sb;
    Ib = ib;
    Rb = rb;
    tb = 0;

% initial graphical representation

    subplot(1,4,1)
    plot(0:N,s,'b',0:N,s0,'g')
    legend('s(t)','s0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('s(t)');
    title('Initial State for \{s_n\}')

    subplot(1,4,2)
    plot(0:N,i,'b',0:N,i0,'g')
    legend('i(t)','i0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('i(t)');
    title('Initial State for \{i_n\}')
```

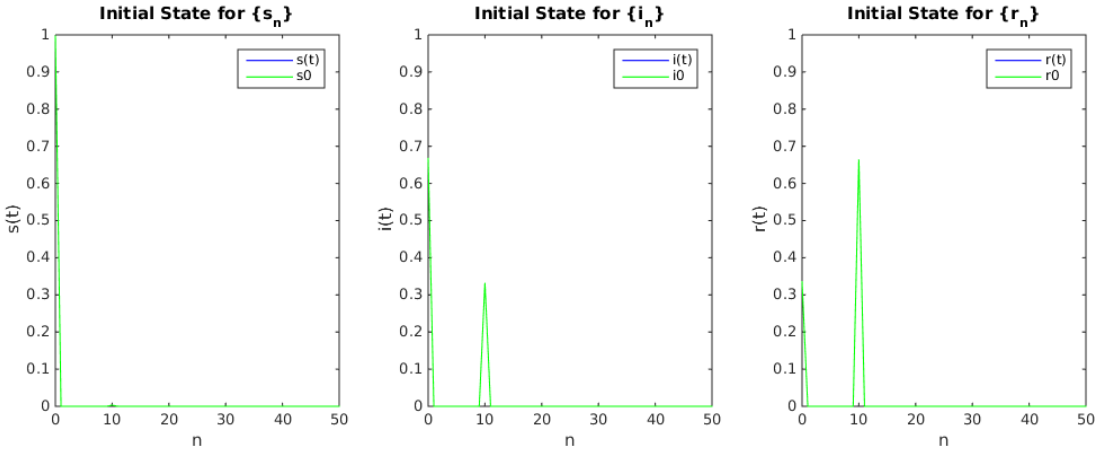
```
    subplot(1,4,3)
    plot(0:N,r,'b',0:N,r0,'g')
    legend('r(t)','r0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('r(t)');
    title('Initial State for \{r_n\}')

    input('?')

% start time stepping
    for k=1:Nt

        A = -spdiags(be + (mu + la*ib)*(0:N)', 0,N+1,N+1) ...
          +  spdiags(          be*ones(N+1,1),-1,N+1,N+1) ...
          +  spdiags(     (mu + la*ib)*(0:N)', 1,N+1,N+1);
        A(N+1,N+1) = -N*(mu + la*ib);

        B = -spdiags((la*sb + mu+ga)*(0:N)', 0,N+1,N+1) ...
          +  spdiags(          la*sb*(0:N)',-1,N+1,N+1) ...
          +  spdiags(        (mu+ga)*(0:N)', 1,N+1,N+1);
        B(N+1,N+1) = -N*(mu+ga);

        C = -spdiags(ga*ib + mu*(0:N)', 0,N+1,N+1) ...
          +  spdiags(ga*ib*ones(N+1,1),-1,N+1,N+1) ...
          +  spdiags(        mu*(0:N)', 1,N+1,N+1);
        C(N+1,N+1) = -N*mu;

% implicit time stepping
        s = (I - th*dt*A) \ ((I + (1-th)*dt*A)*s);
        i = (I - th*dt*B) \ ((I + (1-th)*dt*B)*i);
        r = (I - th*dt*C) \ ((I + (1-th)*dt*C)*r);

% keep s,i,r in [0,1] and maintain probability distributions
        s = max(min(s,1),0);
        s = s/sum(s);
        i = max(min(i,1),0);
        i = i/sum(i);
        r = max(min(r,1),0);
        r = r/sum(r);

% current expected values
        sb = dot(s,(0:N));
        ib = dot(i,(0:N));
        rb = dot(r,(0:N));

        Sb = [Sb,sb];
```

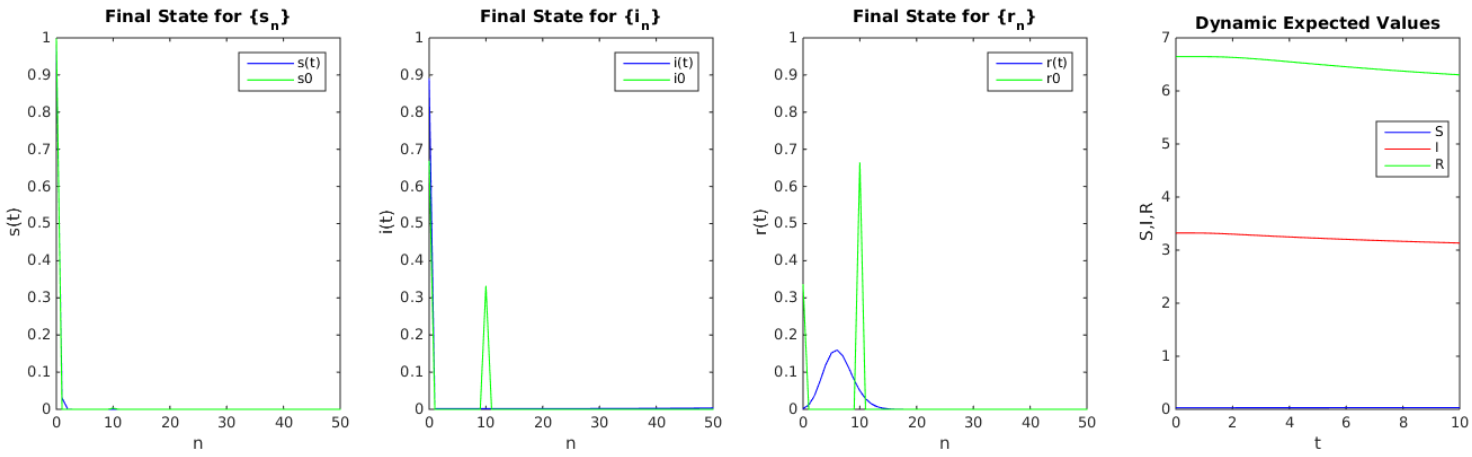
```
        Ib = [Ib,ib];
        Rb = [Rb,rb];
        tb = [tb,k*dt];

    end

% final graphical representation

    subplot(1,4,1)
    plot(0:N,s,'b',0:N,s0,'g')
    legend('s(t)','s0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('s(t)');
    title('Final State for \{s_n\}')

    subplot(1,4,2)
    plot(0:N,i,'b',0:N,i0,'g')
    legend('i(t)','i0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('i(t)');
    title('Final State for \{i_n\}')

    subplot(1,4,3)
    plot(0:N,r,'b',0:N,r0,'g')
    legend('r(t)','r0')
    axis([0 N 0 1])
    xlabel('n')
    ylabel('r(t)');
    title('Final State for \{r_n\}')

    subplot(1,4,4)
    plot(tb,Sb,'b',tb,Ib,'r',tb,Rb,'g')
    xlabel('t')
    ylabel('S,I,R');
    legend('S','I','R','Location','Best')
    title('Dynamic Expected Values')
```

For $N = 20$ the initial and final states are shown graphically as follows.



Note that the mass in the i-distribution moves to larger values of n in the course of the evolution, in this case as well as in the next. For $N = 50$ the initial and final states are shown graphically as follows.





These results show that as N increases, the system resists ever more to depart from the endemic equilibrium state.

• Exercise 3: Markov Chain, SIR Model

◦ Task

For $N = 2$, $M = 1$ and given $\beta, \gamma, \lambda, \mu$ (and possibly ϵ) of the cellular automaton on pages 137 – 142 of the lecture notes, write the transition probabilities of the cellular automaton as a stochastic matrix and find the equilibrium. Bonus: Develop a general method for larger N and M so that results can be compared with the Monte-Carlo simulations from the lecture.

◦ Solution

A Matlab Code for the computation of the stochastic matrix and of the equilibrium for given parameters is given as follows.

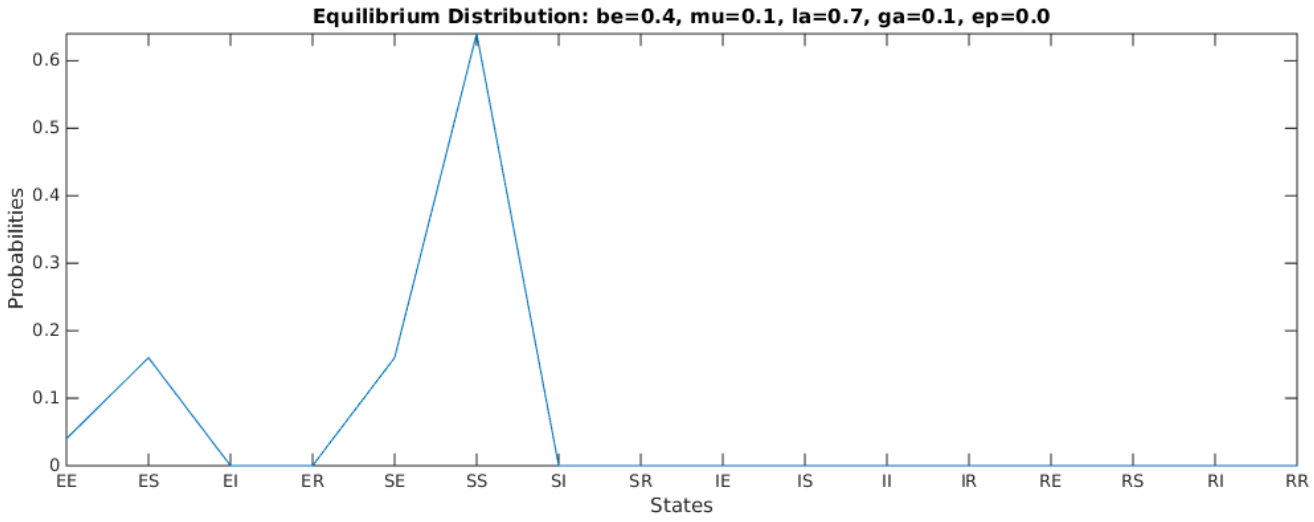
```
h1 = figure(1);
close(h1);
h1 = figure(1);
set(h1,'Position',[10 10 1200 400]);

be = 0.4;
mu = 0.1;
la = 0.7;
ga = 0.1;
ep = 0.1;

%    E       S       I       R
% E  1-be    be      0       0
% S  mu      1-mu-ep ep      0
% I  mu      0       1-mu-ga ga
% R  mu      0       0       1-mu

P0 = [(1-be),     be,      0,   0; ...
```

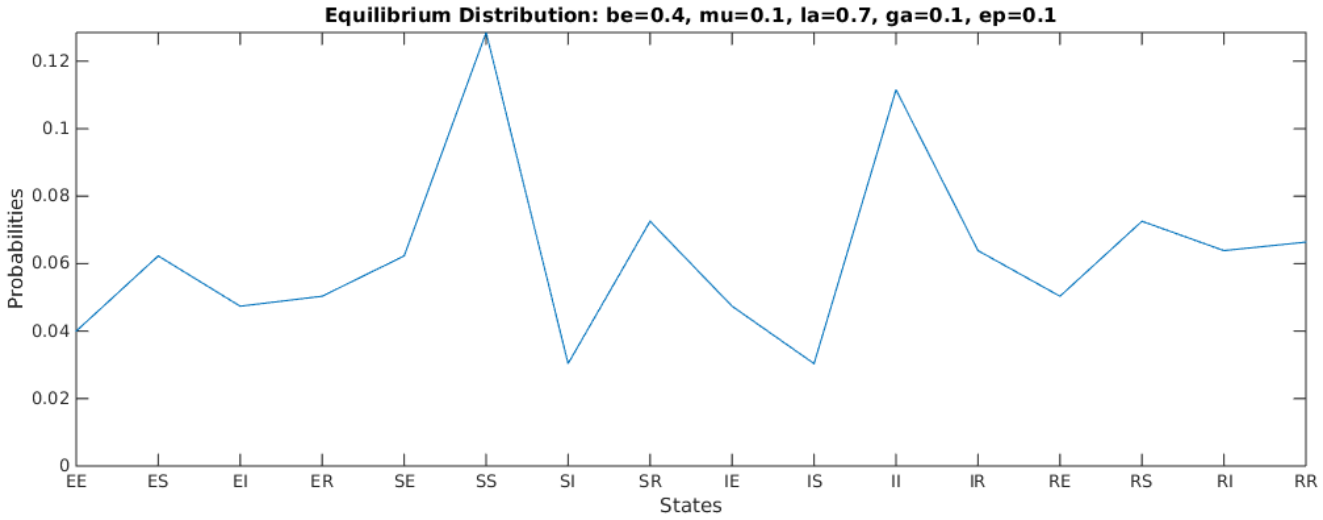
```
          mu,1-mu-ep,     ep,   0; ...
          mu,      0,1-mu-ga,  ga; ...
          mu,      0,      0,1-mu];
P = kron(P0,P0);

% SI -> EE
i = 2; j= 3; k = 1; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*mu;
% SI -> ES
i = 2; j= 3; k = 1; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*0;
% SI -> EI
i = 2; j= 3; k = 1; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*(1-mu-ga);
% SI -> ER
i = 2; j= 3; k = 1; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*ga;

% SI -> SE
i = 2; j= 3; k = 2; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ep-la)*mu;
% SI -> SS
i = 2; j= 3; k = 2; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ep-la)*0;
% SI -> SI
i = 2; j= 3; k = 2; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ep-la)*(1-mu-ga);
% SI -> SR
i = 2; j= 3; k = 2; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ep-la)*ga;

% SI -> IE
i = 2; j= 3; k = 3; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (ep+la)*mu;
% SI -> IS
i = 2; j= 3; k = 3; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (ep+la)*0;
% SI -> II
i = 2; j= 3; k = 3; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (ep+la)*(1-mu-ga);
% SI -> IR
i = 2; j= 3; k = 3; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (ep+la)*ga;

% SI -> RE
i = 2; j= 3; k = 4; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*mu;
% SI -> RS
```

```
i = 2; j= 3; k = 4; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*0;
% SI -> RI
i = 2; j= 3; k = 4; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*(1-mu-ga);
% SI -> RR
i = 2; j= 3; k = 4; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*ga;

% IS -> EE
i = 3; j= 2; k = 1; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*mu;
% IS -> ES
i = 3; j= 2; k = 1; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*(1-mu-ep-la);
% IS -> EI
i = 3; j= 2; k = 1; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*(la+ep);
% IS -> ER
i = 3; j= 2; k = 1; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = mu*0;

% IS -> SE
i = 3; j= 2; k = 2; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*mu;
% IS -> SS
i = 3; j= 2; k = 2; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*(1-mu-ep-la);
% IS -> SI
i = 3; j= 2; k = 2; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*(la+ep);
% IS -> SR
i = 3; j= 2; k = 2; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = 0*0;

% IS -> IE
i = 3; j= 2; k = 3; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ga)*mu;
% IS -> IS
i = 3; j= 2; k = 3; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ga)*(1-mu-ep-la);
% IS -> II
i = 3; j= 2; k = 3; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ga)*(la+ep);
% IS -> IR
i = 3; j= 2; k = 3; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = (1-mu-ga)*0;
```

```
% IS -> RE
i = 3; j= 2; k = 4; l = 1; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = ga*mu;
% IS -> RS
i = 3; j= 2; k = 4; l = 2; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = ga*(1-mu-ep-la);
% IS -> RI
i = 3; j= 2; k = 4; l = 3; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = ga*(ep+la);
% IS -> RR
i = 3; j= 2; k = 4; l = 4; I = (i-1)*4+j; J = (k-1)*4+l;
P(I,J) = ga*0;

[V,D] = eig(P');
v = V(:,1);
v = v/sum(v);
plot([1:16],v)
set(gca,'XTick',1:16,'XTickLabel', ...
   {'EE','ES','EI','ER', ...
    'SE','SS','SI','SR', ...
    'IE','IS','II','IR', ...
    'RE','RS','RI','RR'})
axis([1 16 0 max(v)])
xlabel('States')
ylabel('Probabilities');
title(sprintf('Equilibrium Distribution: ...
    be=%0.1f, mu=%0.1f, la=%0.1f, ga=%0.1f, ep=%0.1f', ...
    be,mu,la,ga,ep))
```

Results for the indicated parameters are displayed graphically as follows.





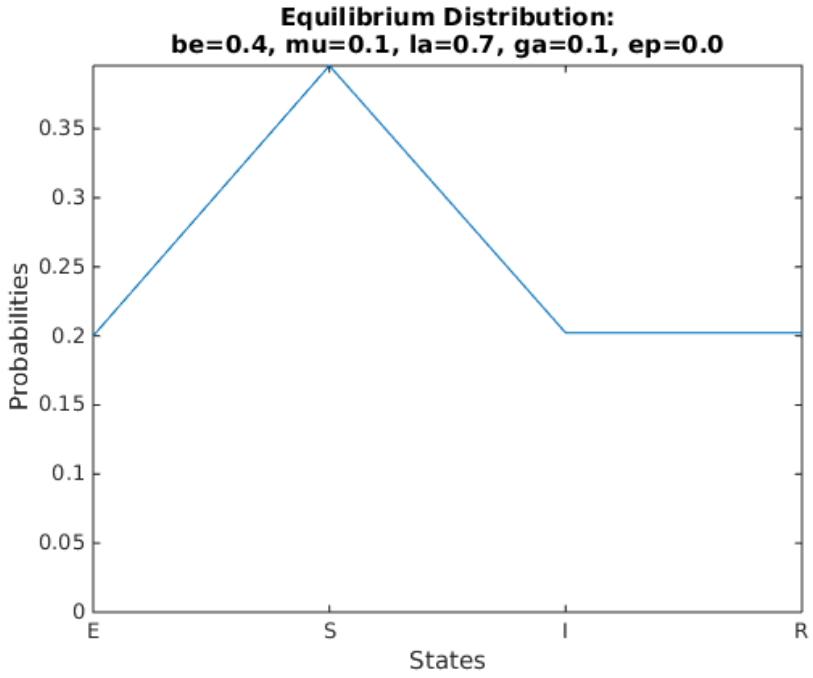
Notice that for $\epsilon = 0$ the infection dies out but not for $\epsilon > 0$.

Bonus: Let `Ib` be estimated as the conditional probability $P(\{C_{i+p,j+q} = 2 : \|(p,q)\|_\infty = 1\}|C_{i,j} = 1)$, which can be obtained empirically by running the Monte-Carlo simulation of problem 4 and computing the proportion of susceptible cells which have an infected neighbor. Based upon this estimate of `Ib` the following Matlab code estimates the equilibrium states using only a 4×4 stochastic matrix.

```
be = 0.4;
mu = 0.1;
la = 0.7;
ga = 0.1;
ep = 0.0;

% estimate Ib = P(neighbor is infected | cell is susceptble)
Ib = 0.146;
% approximate result of Monte-Carlo with parameters above:
% Ebar = 0.2, Sbar = 0.4, Ibar = 0.2, Rbar = 0.2

%    E            S             I            R
% E  1-be         be            0            0
% S  mu           1-mu-ep-la*Ib ep+la*Ib     0
% I  mu           0             1-mu-ga      ga
% R  mu           0             0            1-mu

P = [(1-be),          be,            0,   0; ...
        mu,1-mu-ep-la*Ib,     ep+la*Ib,   0; ...
        mu,            0,      1-mu-ga,  ga; ...
        mu,            0,            0,1-mu];

[V,D] = eig(P');
D = diag(D);
i = find(abs(D - 1) < 1.0e-3);
v = V(:,i)/sum(V(:,i));
```

```
plot([1:4],v)
set(gca,'XTick',1:4,'XTickLabel',{'E','S','I','R'})
axis([1 4 0 max(v)])
xlabel('States')
ylabel('Probabilities')
title(sprintf('Equilibrium Distribution: ...
    be=%0.1f, mu=%0.1f, la=%0.1f, ga=%0.1f, ep=%0.1f', ....
    be,mu,la,ga,ep))
```

The results are shows graphically as follows,



and these agree with the results from the Monte-Carlo simulation!

• Exercise 4: Cellular Automaton, SIR Model

◦ Task

Implement the Monte-Carle simulation of the cellular automaton on pages 137 – 142 of the lecture notes.

◦ Solution

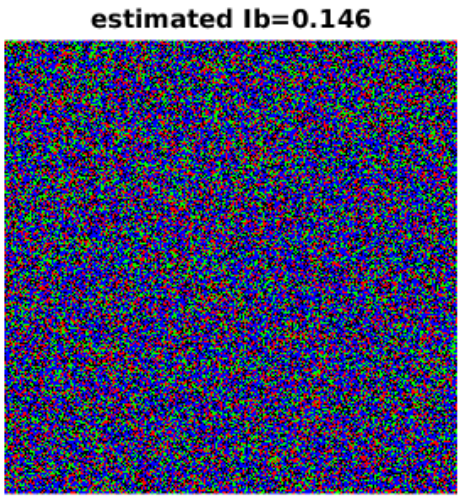
A Matlab code for the Monte-Carlo simulation is given as follows.

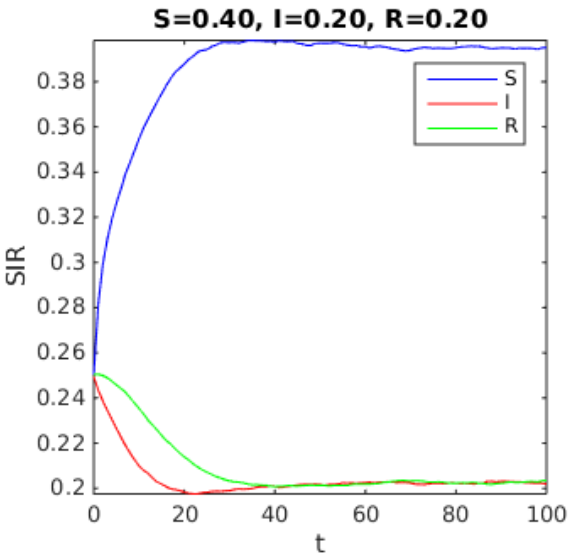
```
% setup figure
h1 = figure(1);
close(h1);
h1 = figure(1);
set(h1,'Position',[10 10 800 400]);
```

```
% model parameters
be = 0.4;
mu = 0.1;
la = 0.7;
ga = 0.1;
ep = 0.0;

% size of grid with torus BCs
N = 1000;
M = 1000;

% number of computed generations
kmax = 100;

% C is state variable
%    C = 0 => empty
%    C = 1 => susceptible
%    C = 2 => infected
%    C = 3 => recovered

example = 1;
switch example
    case 1
        C = randi(4,N,M)-1;
    case 2
        C = ones(N,M);
        C(randi(N),randi(M)) = 2;
    case 3
        C = ones(N,M);
        C(round(3*N/8):round(5*N/8),round(3*M/8):round(5*M/8)) = 2;
end

% save the initial state
C0 = C;

subplot(1,2,1)
imagesc(C,[0,3]);
axis image;
axis off;
colormap([0,0,0;0,0,1;1,0,0;0,1,0]);
drawnow;

% the average number susceptibles, infected and recovered
Sbar = mean(C(:) == 1);
Ibar = mean(C(:) == 2);
Rbar = mean(C(:) == 3);
```

```
Ibhis = [];
for k=1:kmax
    Ib = -double(C==2);
    for i=-1:1; for j=-1:1;
            Ib = Ib + circshift(C==2,[i,j]);
    end; end;
    Ib=Ib/8;
% estimate the conditional probability P(infected neighbor | susceptible)
    i = find(C == 1);
    Ibhis = [Ibhis,mean(Ib(i))];
    Ibest = mean(Ibhis(max(1,k-50):k));

    Z = rand(N,M);

    C = 1*((C == 0) & (Z < be))                              ...
      + 2*((C == 1) & (Z >= mu) & (Z < (mu + ep + la*Ib))) ...
      + 1*((C == 1) & (Z >= (mu + ep + la*Ib)))            ...
      + 3*((C == 2) & (Z >= mu) & (Z < (mu + ga)))         ...
      + 2*((C == 2) & (Z >= (mu + ga)))                    ...
      + 3*((C == 3) & (Z >= mu));

    subplot(1,2,1)
    imagesc(C,[0,3]);
    title(sprintf('estimated Ib=%0.3f',Ibest))
    axis image;
    axis off;
    colormap([0,0,0;0,0,1;1,0,0;0,1,0]);
    drawnow;

    sbar = mean(C(:) == 1);
    ibar = mean(C(:) == 2);
    rbar = mean(C(:) == 3);
    Sbar = [Sbar,sbar];
    Ibar = [Ibar,ibar];
    Rbar = [Rbar,rbar];

    subplot(1,2,2)
    plot(0:k,Sbar,'b',0:k,Ibar,'r',0:k,Rbar,'g')
    title(sprintf('S=%0.2f, I=%0.2f, R=%0.2f',sbar,ibar,rbar))
    xlabel('t')
    ylabel('SIR');
    axis tight; axis square;
    legend('S','I','R')

end
```

Results are shown graphically as follows for the case of random initial conditions.





Note that the conditional probability $P(\{C_{i+p,j+q} = 2 : \|(p,q)\|_\infty = 1\}|C_{i,j} = 1)$ is estimated and used for problem 3.

• Exercise 5: Continous Chaotic SIR Model

◦ Task

Develop a continous time lumped parameter infection model which possesses periodic solutions and perhaps even period doubling transitioning to chaos. Feel free to include any relevant effects, even such as loss of immunity, vaccination, etc. This challenge was met by Elias Windisch who contributed page 126 in the script!

◦ Solution

The model is

$$\left\{\begin{array}{rcl} S'(t) &=& \beta - \lambda(t)SI - \mu S \\ I'(t) &=& \lambda(t)SI - \gamma I - \mu I \\ R'(t) &=& \gamma I - \mu R \end{array}\right.$$

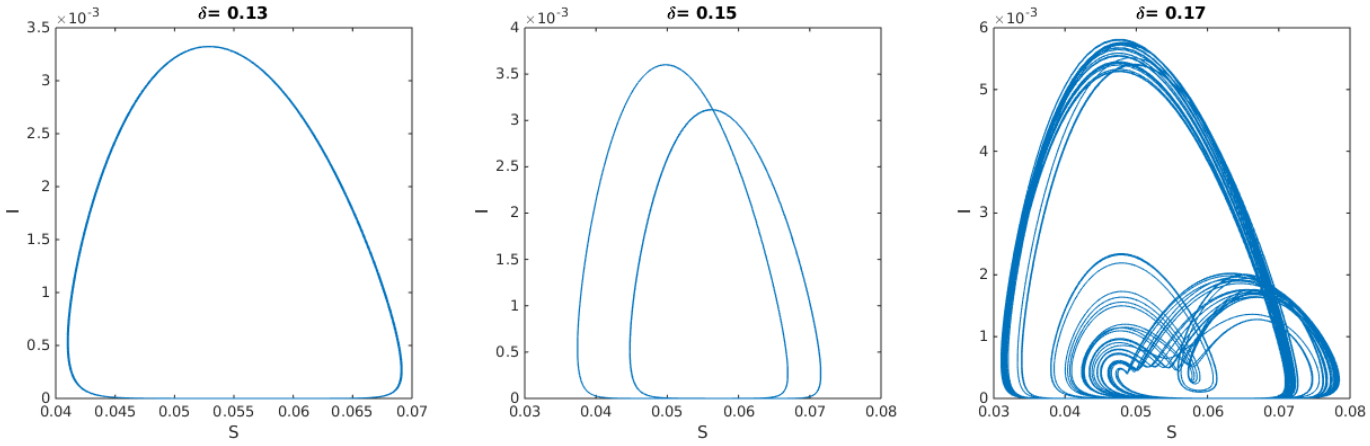
with a time dependent $\lambda(t)$ to model the dependence of infectiousness on season,

$$\lambda(t) = (\delta \sin(2\pi t) + 1)\lambda_0, \quad \lambda_0 \in (0,\infty), \quad \delta \in [0,1].$$

There is no backward coupling with R, so the qualitative properties of the model are determined by the first two equations. This non-autonomous 2×2 system can be rewritten as a 3×3 system in autonomous form, which therefore theoretically permits the emergence of chaos. Simulations are carried out for the (S, I)-system with the parameters:

$$\mu = \beta = 0.04, \quad \lambda_0 = 1800, \quad \gamma = 100.$$

The parameter δ is increased, $\lambda = 0.13, 0.15, 0.17$, and apparent period doubling of limit cycles and transition to chaos are exhibited:



These results are obtained with the following Matlab code.

```
% setup figure
    h1 = figure(1);
    close(h1);
    h1 = figure(1);
    set(h1,'Position',[10 10 400 400]);

% set paramters
    be = 0.04;
    mu = 0.04;
    ga = 100;

    de = 0.13;    % periodic limit cycle
%   de = 0.15;    % two-period limit cycle
%   de = 0.17;    % chaos

    l0 = 1800;

% function definitions
    la = @(t) l0*(1 + de*sin(2*pi*t));
    f  = @(t,y) [be - (mu + la(t)*y(2))*y(1); ...
                 (la(t)*y(1) - mu - ga)*y(2); ...
                          -mu*y(3) + ga*y(2);];
% initial values
    y0=[0.7 0.3 0]';

% compute solution
    opts = odeset('MaxStep',0.01,'RelTol',1.0e-10);
    [t,y]=ode45(f,[0 100],y0,opts);

% graphical display
```

```
kmin = find(t>20,1);
plot(y(kmin:end,1),y(kmin:end,2));
xlabel('S');
ylabel('I');
title(['\delta= ',num2str(de)])
```

• Exercise 6: Discrete Chaotic SIR Model

◦ Task

Develop a discrete time lumped parameter infection model which possesses periodic solutions and perhaps even period doubling transitioning to chaos. Feel free to include any relevant effects, even such as loss of immunity, vaccination, etc.

◦ Solution

The predator-prey model for (S, I) without back-coupling from R,

$$\left\{ \begin{array}{rcl} S' & = & a_1 S(1 - S/a_2) - a_3 SI/(1 + S/a_4) \\ I' & = & a_5 I(1 - a_6 I/S) \end{array} \right.$$

with parameters

$$a_1 = 10, \quad a_2 = 100, \quad a_3 = 10, \quad a_4 = 10, \quad a_5 = 1, \quad a_6 = 10,$$

and the unstable equilibrium

$$S^\star = 5(\sqrt{41} - 1) \approx 27, \quad I^\star = (\sqrt{41} - 1)/2 \approx 2.7$$

is known to exhibit a stable limit cycle. The dynamic motivates the consideration of the discretization

$$\left\{ \begin{array}{rcl} (S^{k+1} - S^k)/\Delta t & = & 10 S^k (1 - S^k/100) - (\lambda/\Delta t) S^k I^k/(1 + S^k/10) \\ (I^{k+1} - I^k)/\Delta t & = & I^k (1 - 10 I^k/S^k) \end{array} \right.$$

for a suitable Δt, where $a_3 = 10$ has been replaced with λ/Δ for an infectiousnes parameter λ, which is suspected to play a role in transition to chaos. With $\Delta t = 1/10$ the model becomes

$$\left\{ \begin{array}{rcl} S^{k+1} & = & S^k + S^k(1 - S^k/100) - \lambda S^k I^k/(1 + S^k/10) \\ I^{k+1} & = & I^k + I^k(1/10 - I^k/S^k) \end{array} \right.$$

The equilibrium states and their stability are determined with the following Mathematica Code.

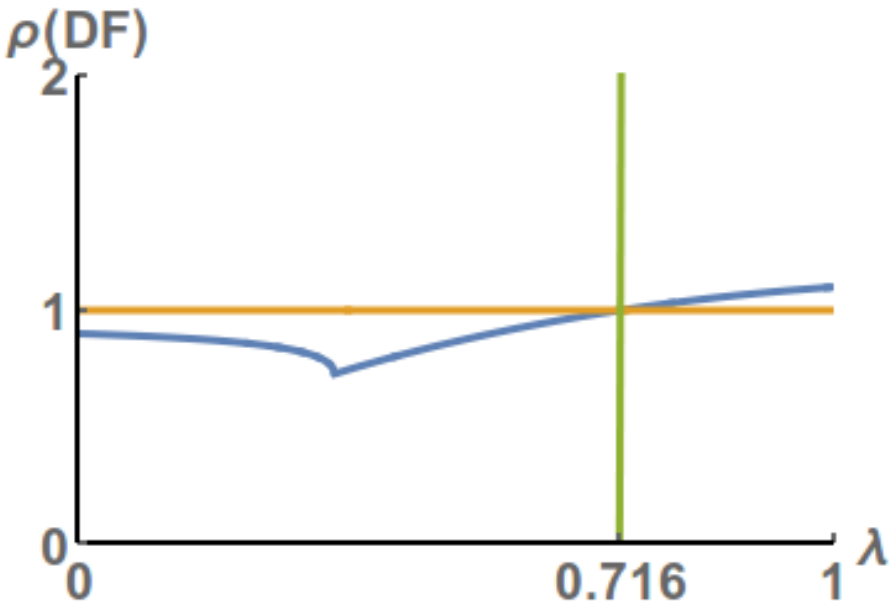
```
G[x_,y_,a1_,a2_,a3_,a4_,a5_,a6_] = {x + a1*x*(1 - x/a2) - a3*x*y/(1 + x/a4),y + a5*y*(1 - a6*y/x)}
Solve[G[x,y,a1,a2,a3,a4,a5,a6] == {x,y},{x,y}]
Xeq1[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],1],1],2]
Yeq1[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],1],2],2]
Xeq2[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],2],1],2]
Yeq2[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],2],2],2]
Xeq3[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],3],1],2]
Yeq3[a1_,a2_,a3_,a4_,a5_,a6_]=Part[Part[Part[Solve[G[x,y,a1,a2,a3,a4,a5,a6]=={x,y},{x,y}],3],2],2]
F[x_, y_, l_] = {x + x (1 - x/100) - l x y/(1 + x/10), y + y (1/10 - y/x)}
x1[l_] = Xeq1[1,100,l,10,1/10,10]
y1[l_] = Yeq1[1,100,l,10,1/10,10]
```

```
x2[l_] = Xeq2[1,100,l,10,1/10,10]
y2[l_] = Yeq2[1,100,l,10,1/10,10]
DF[x_, y_, l_] = D[F[x, y, l], {{x, y}}]
N1[l_] = Abs[Part[Eigenvalues[DF[x1[l], y1[l], l]],1]]
N2[l_] = Abs[Part[Eigenvalues[DF[x1[l], y1[l], l]],2]]
M1[l_] = Abs[Part[Eigenvalues[DF[x2[l], y2[l], l]],1]]
M2[l_] = Abs[Part[Eigenvalues[DF[x2[l], y2[l], l]],2]]
Plot[{Max[M1[l],M2[l]],1,1+1000(l-0.71645)}, {l, 0, 1},
 BaseStyle -> {FontWeight -> Bold, FontSize -> 20},
 PlotRange -> {{0,1},{0,2}},
 Ticks->{{0,0.716,1},{0,1,2}},
 PlotStyle -> {Thickness[0.01]}, AxesLabel -> {\[Lambda], "\[Rho](DF)"}, AxesStyle -> Thick]
```

For instance, the equilibria of interest are `Xeq2` and `Yeq2` from the code given by

$$\begin{aligned} S^\star &= (900 + \sqrt{400000 + (900 - 1000\lambda)^2} - 1000\lambda)/20 \\ I^\star &= (90 + \sqrt{400000 + (900 - 1000\lambda)^2}/10 - 100\lambda)/20 \end{aligned}$$

and the spectral radius of the Jacobian is shown with respect to λ by the following graphic,

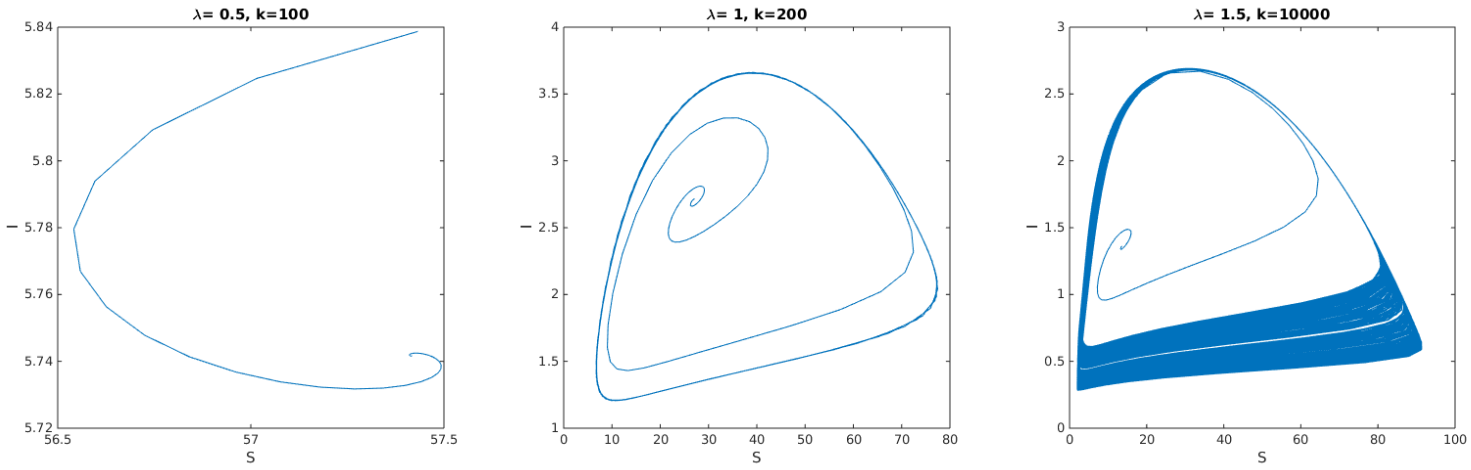


which indicates a stable equilibrium for $\lambda \in (0, 0.716)$. This theoretical result agrees with the simulations shown below, obtained using the following Matlab Code.

```
% setup figure
    h1 = figure(1);
    close(h1);
    h1 = figure(1);
    set(h1,'Position',[10 10 500 500]);

% model parameters
    la = 0.5;    % stable equilibrium
    la = 1.0;    % stable limit cycle
```

```
    la = 1.5;    % chaos

% equilibrium state
    Seq = (900+sqrt(400000+(900-1000*la)^2)-1000*la)/20;
    Ieq = (90+sqrt(400000+(900-1000*la)^2}/10-100*la)/20;

% perturbation of the equilibrium
    ns = 1.0e-10;
    S  = Seq + ns*rand(1);
    I  = Ieq + ns*rand(1);

% plotting vectores
    Sv = S;
    Iv = I;

% number of generations
    kmax = 100000;

for k=1:kmax
    S = S + S.*(1 - S/100) - la*S.*I./(1 + S/10);
    I = I + I.*(1/10 - I./S);

    Sv = [Sv,S];
    Iv = [Iv,I];
    plot(Sv,Iv)
    drawnow;
end
```

Results are shows for the respective values of λ.



In the first case with $\lambda = 0.5$ on the left, the equilibrium ist asymptotically stable. For the second case with $\lambda = 1.0$ in the middle there is apparently a stable periodic orbit. For the thrid case with $\lambda = 1.5$ on the right, chaos has apparently emerged!