Updates and MATLAB files for Modeling and Simulation in Medicine and the Life Sciences, Second Edition

Contents

1 Updates to Modeling and Simulation...., 2003

Notation: [page:line] action

Chapter 7:

217:-2 of cells taken from (take → taken)

220:16 If $r = 1.9$, we would... (delete ‘Since...from Table 7.1)

220:-10 ...(1798), equation (; → ,)

222:4 also as being random... (insert being)

223:2 has the dimension of... (dimensions → dimension)

224:-7 ideas and notations will.. (notation → notations)

225:-7 Table 7.2. (delete rest of sentence ‘which ... population.)

225:table 7.2 Row should be

$n \qquad B_n^0 \qquad B_n^1 \qquad B_n^2$

$(B_n \to B_n^0)$

225:-3 the sequence of clean births (insert clean)

230:3 element limiting that growth (insert that)

234:-8 flow (dimensions are volume...) (delete ‘of nutrient’)

238:-3 auxotrophs; that is, (, → .)

239:-16 determined by straightforward (insert by)

245:last The histogram describes (delete where and)

248:-12 can lead to some (have → lead to)

251:Fig7.19 Left plate relabel vertical axis Y_n and horizontal axis A_n

251:2 the next point (A_2, Y_2), etc. (delete as shown there. insert ,)

251:7 near (A^*, Y^*) $((Y^*, A^*) \to (A^*, Y^*))$

253:1 among ages. (delete and their sizes)

258:Table 7.5 The table is misaligned. The three central columns should be:

0	2104	2104
1	2352	2441
2	2573	2831
3	3123	3284
4	3824	3810
5	4572	
6	5117	
7	5876	
8	6356	
9	7480	

	Note that the last line begins with 9.
258:-12	in pairs $(\log P_n, \log P_{n+1})$. (insert log in two places)
258:-2	for mitotic phase $p(t)$: (insert :)
259:7	the cohorts as in section 7.5.1. What... (insert as in section 7.5.1.)
259:-10	a solution for part 2. can be (insert part 2.)
261:-15	of the model (See Ex. 7.16). (insert (see ex. 7.16))
262:-21	delete: ‘show that ... 1. insert: Choose data so emergences are balanced for $L \leq 7$ and periodic for $L \geq 10$.
262:-18	That is, show (insert: ,)
264:5	should be $\sum^N$ ... $(M \to N)$
264:6	should be $\sum^M$ ... $(N \to N)$
264:17	insert , between i and j in each subscript
264:19	same

Chapter 8:

271:-2 In case b (change c to b)

272:7 case b, (change c to b)

273:-7 delete **aa**

273:-6 by the equations (here $q = 1 - p$) (insert: (here $q = 1 - p$))

273:-5 to -3 This system must be replaced: The three equations should be

$$\begin{aligned} \dot{x} &= pbx + pby - dx \\ 2\dot{y} &= qbx + (p+q)by + pbz - 2dy \\ \dot{z} &= qbz + qby - dz \end{aligned}$$

274:-11 $D + H = c$, a constant... (insert: c, a)

276:-5 put comma in subscript between i and j

281:-7 delete this line: $A + B \to C$,

287:-2 a and b, respectively (change c to b)

287:last P_N is c. (change: b to c)

288:16 $+N(cP_N$.. (change: b to c)

288:19 $N(c+$ (change: b to c)

288:20 $N(c + ax^N$... (change: b to c)

288:20 Nax^N (change: exponent from 2 to N))

288:-9 $NabKx^N$ (change: c to b)

288:-6 $(NbK -$ .. (change: c to b)

288:-3 .. $- NbK)$ (change(c to b)

293:12 should be states at each time step. add: step

Chapter 10 and Appendixes

314:4 by the number $U_{m,n}$ (numbers to number)

337:-8 x and y are row vectors (insert: row)

338:17 Matrix division: (change . to :)

343:8 change 2^N to many

344:7 A discrete random variable (insert: discrete)

344:-5 define the cumulative distribution. (insert: cumulative)

345:6 $A < f(\omega) \leq B$ (change: $\leq$ to $<$)

345:9 f is normal (delete: a)

346:14 $= P[y_n$ (change Pr to P)

2 Chapter 6: MATLAB programs

```
%pendulum: This demonstrates the coexistence of stable
%static and running periodic solutions for
%tau z'=-z+cos x, x'=w+z

global w tau

tau=1.5; w=.9; tt=0:0.01:100; ltt=length(tt);

th0=zeros(2,1);th=th0; th0=[0;pi];
[t th]=ode15s('pendrhs',tt,th0);

th0l=zeros(2,1); thl=th0l; th0l=[pi;-pi];
[t thl]=ode15s('pendrhs',tt,th0l);

clf
[x y]=meshgrid(0:0.5:4*pi,-pi:0.5:pi); hold on
figure(1), quiver(x,y,w+y,cos(x)-y)%, axis image
u=0:.01:4*pi;
plot(u,zeros(length(u),1),'-',u,-w*ones(length(u),1),x,cos(x),'*')
plot(mod(th(:,1),4*pi),th(:,2),'.'),plot(thl(:,1),thl(:,2),'r.'),hold off

%%%%%%
function thp=pendrhs(t,th)
global w tau

thp=th;
thp(1)=w+th(2);
thp(2)=(cos(th(1))-th(2))/tau;
```

```
%pll1.m
%This plots the vector field defined
%by the first order phase-locked loop

[x,z]=meshgrid(-4*pi:.628:4*pi,-2*pi:.628:2*pi);
dx=0.2+z; dz=-z+cos(x);
quiver(x,z,dx,dz)
xlabel('\theta'),ylabel('z')
grid on
```

```
%atoll
%This solves the atoll model and describes
%parabolic bursting

tt=0:.1:300; x0=zeros(2,1); x0(2,1)=pi; x=x0;
options=odeset('AbsTol',1e-8,'RelTol',1e-6);
tic
[t x] = ode15s('atorhs',tt,x0,options);
toc
[m n]=size(x);
subplot(2,1,1),plot(t,cos(x(:,1))),xlabel('t'),ylabel('cos x')
subplot(2,1,2),plot(t,cos(x(:,2))), xlabel('t'),ylabel('cos y')

%%%%%%
function xp=atorhs(t,x)

xp=x;
xp(1)=5*(1+0.3+cos(x(1))-cos(x(2)));
xp(2)=0.04*(1+cos(x(2))+10*cos(x(1)));
```

```
%atn2: Thalamus-reticular formation model

global N N2 N3 input s A

N=15;N2=2*N;N3=3*N; input=zeros(N,1); s=input; input(1)=0.01;

A=zeros(N); A=diag(ones(N-1,1),1)+diag(ones(N-1,1),-1);
A(1,2)=2;A(N-1,N)=2;

x0=0.5*pi*ones(N3,1); x=x0; tf=5000; Fs=32; tt=0:1/Fs:tf;
options=odeset('AbsTol',1e-8,'RelTol',1e-6);

tic
[t x]=ode15s('atn2rhs',tt,x0,options);
toc

figure(1),
clf
hold on
for j=1:N3
    %plot(t,max(cos(x(:,j)),0)/2+j);
    plot(t,cos(x(:,j))/2+j);
end
xlabel('t'),ylabel('0.5*cos x_j(t)')

%%%%%%%%
function xp=atn2rhs(t,x)
global N N2 N3 input s A

z=cos(x);zp=max(z,0.0);
nnn=2*floor(t/500);
if(nnn<=12)
    input(min(nnn+1,N),1)=(0.1+nnn*0.1);
else
    input(min(nnn+1,N),1)=0.0;
end
if(nnn>=8)input(3,1)=0; end
if(nnn>=10)input(9,1)=0; end
```

```
v=ones(1,N);%/N;
xp=x;
xp(1:N)=5*(1+z(1:N)-zp(N+1:N2)+input(1:N));
xp(N+1:N2)=0.04*(1+z(N+1:N2)+tanh(2*zp(1:N)-10*v*zp(N+1:N2)));
xp(N2+1:N3)=10*(0.1+z(N2+1:N3)+zp(N+1:N2));
```

```
%knot.m. Solve torus knot problem in presence of parametric noise

global nnoiz dt tf eps lam alpha beta z tstop tt neqn

tf=50;eps=0.5;neqn=2; dt=eps/50.0;lam=1.0;alpha=0.0;beta=1.0;
nnoiz=2;nsamp=10; tt=0:dt:tf;

beta=0.0;alpha=-beta/2; x0=zeros(neqn,1); x=x0;
tic
sdestop;
sdepath;
toc

[t x]=ode15s('knotrhs',tt,x0);
[m n]=size(x);
figure(4),plot(t,x(:,1),'r'),hold on,drawnow
figure(5), plot(t,z,'r')
u=(2+1*cos(2*pi*x(:,1))).*cos(2*pi*x(:,2));
v=(2+1*cos(2*pi*x(:,1))).*sin(2*pi*x(:,2));
w=2*sin(2*pi*x(:,1));
figure(6), plot(u,v,'r'),drawnow
rho=x(m,1)/x(m,2)

beta=5;alpha=-beta/2; rhoden=zeros(nsamp,1);
for k=1:nsamp
    tic
    sdestop; sdepath;

    x0=zeros(neqn,1);x=x0;
    [t x]=ode15s('knotrhs',tt,x0);
    [m n]=size(x);
    rho=x(m,1)/x(m,2);
    [k rho]
    rhoden(k,1)=rho;
figure(1),plot(t,x(:,1)),hold on,drawnow
figure(2), plot(t,z)
u=(2+1*cos(2*pi*x(:,1))).*cos(2*pi*x(:,2));
v=(2+1*cos(2*pi*x(:,1))).*sin(2*pi*x(:,2));
```

```
w=2*sin(2*pi*x(:,1));
figure(3), plot3(u,v,w),drawnow
toc
end

figure(7),hist(rhoden)

%%%%%%%%%%%%%%%
function xp=knotrhs(t,x)  %RHS of knot.m

global nnoiz dt tf eps lam alpha beta z tstop tt neqn

xp=x;

xp(1)=1+z(floor(t/dt)+1,1)+sin(x(1)-x(2));
xp(2)=1+z(floor(t/dt)+1,2)-sin(x(1)-x(2));

%%%%%%%%%%
function tstop=sdestop %Function to create stopping times
global nnoiz dt tf eps lam alpha beta z tstop tt neqn

nstop=floor(tf/eps);%number of stop times
exinc=-lam*log(rand(nstop,1));%exponential distribution of stop time increments
tstop=zeros(nstop,1);
for k=2:nstop
    tstop(k)=tstop(k-1)+exinc(k);
end
ltstop=length(tstop);
tstop=tstop*tf/tstop(ltstop);%scale stop times to fit [0,tf]

%%%%%%%%%%%
function sdepath %Function to create a sample path
global nnoiz dt tf eps lam alpha beta z tstop tt

nstop=length(tstop);ltt=length(tt);
y=alpha+beta*rand(nstop,nnoiz);%generate diffusion sample path at stop times
%generate noise perturbation
z=zeros(ltt,nnoiz);
```

```
for kk=1:nnoiz
   for m=1:ltt
      if((m*dt)<tstop(1))z(m,kk)=y(1,kk);end
      for k=1:(nstop-1)
         if((tstop(k)<=(m*dt))&((m*dt)<tstop(k+1)))z(m,kk)=y(k,kk);break; end
      end
      if((m*dt)>=tstop(nstop))z(m,kk)=y(nstop,kk);end
   end
end
```

```
%Time crystal with rotation

set(gcf,'Renderer','zbuffer')

x=0:.03:(2*pi);a=0:.03:2.1; [X A]=meshgrid(x,a); phase=-pi/2;
f=atan(sin(X+phase)./(A+cos(X+phase)));

figure(1), contour(X,A,f,100)
xlabel('\phi'),ylabel('A')
colormap bone

figure(2), plot3(X,A,f,'.')
axis vis3d
for i=1:36
    camorbit(10,0,'data',[0 0 1])
    xlabel('\phi'),ylabel('A'),zlabel('\psi')
    drawnow
end
```

```
%hopf.m Describe Andronov-Hopf bifurcation: ex 6.8 p213
global lam

tf=100; tt=0:0.1:tf; lt=length(tt);

hold on

grid on

figure(1)
axis([-1 1 -2 2 -2 2])
xlabel('\lambda'),
ylabel('r cos \theta'), zlabel('r sin \theta')

llam=50; u=zeros(2,lt);

for j=1:llam
   %solve for r,th
   z=ones(2,1);
   z0=z;
   lam=-1+j*2/llam;
   z0(1)=0.01+sqrt((lam+abs(lam))/2);
   z0(2)=0.0;
   [t z]=ode15s('hopfrhs',tt,z0);
   [m n]=size(z);
   lam2=lam*ones(lt,1);
   plot3(lam2,z(:,1).*cos(z(:,2)),z(:,1).*sin(z(:,2)))
   drawnow
   hold on
end
rotate3d on

%%%%%%%%%%%
function zp=hopfrhs(t,z,lam)  % hopf rhs
global lam
zp=[z(1)*(lam-z(1)^2);1];
```

3 Chapter 7: MATLAB programs

```
%Fibonacci: Solve Fibonacci's equation and plot stable age distribution

N=100; u=zeros(N,1); u(1)=1;u(2)=1;
a=0.5;b=0.6;

%Solve Fibonacci's equation
for n=3:N
    u(n)=a*u(n-1)+b*u(n-2);
end
figure(1), plot(1:N,u)
title('Birth Rate')

%Find the characteristic roots
p=[1 -a -b]; r=roots(p)

%Find the stable age birth rate
u(N)*r(1)^(-N)

%Calculate and Plot stable age distribution
Ncensus=11; %Number of census classes
A=zeros(Ncensus,1); %Age distribution
lambda=zeros(Ncensus,1); lambda(1)=.95;
for k=2:7
    lambda(k)=(.99^(k-2));
end
for k=7:Ncensus
    lambda(k)=(.95)^(k-7);
end

A=zeros(Ncensus);
for k=1:Ncensus
    A(k)=(r(1))^(-(N-k))*u(N-k)*lambda(k);
end
figure(2), bar(1:Ncensus,1000*A(1:Ncensus))
title('Stable Age Distribution')
```

```
%Effort-yield diagram

global br K

br=2;K=100;
Npop=1000; q = 2;
Y=zeros(Npop,1); E=Y;
for B=1:Npop
    new=effrhs(B);
    if (new >= 1)
        Y(B)=new*B-B;
        E(B)=log(effrhs(B))/q;
    end
end

plot(E,Y),xlabel('Effort'),ylabel('Yield')

%%%%%%%%%%%%%
function rr=effrhs(B)

global br K
%rr=br/(1+B/K);  % Beverton-Holt
rr=br*exp(-B/K); % Ricker's Model
%rr=br*B*exp(-B/K); % Predator Satiation Model
```

```
%Two-age iteration

sigma=.9; br = 3; K = 50;
Ntime=100;
A=zeros(Ntime,1); Y=A;
A(1)= 9*K; Y(1)=1;
figure(1), clf
hold on
for n=2:Ntime
    A(n)=sigma*Y(n-1);
    Y(n)=tworhs(A(n-1));
    plot(A(n),Y(n),’*’), drawnow
    axis([0 10*K 0 10*K])
end

%%%%%%%%
function y=tworhs(a)

global br K

y=br*a^2/(1+a^2/K);
```

```
%Iteration of Ricker's Mapping
%3d riter

set(gcf,'Renderer','zbuffer')

nr=200;
ncells=200;
niter=10000;
d=zeros(nr,ncells);
tic
for ir=1:nr
    x=1;
    r=0+20*ir/nr;
    for k=1:100
        x=r*x*exp(-x);
    end
    for k=1:(niter-100)
        x=r*x*exp(-x);
        ix=floor(ncells*x/8)+1;
        d(ir,ix)=1+d(ir,ix);
    end
end
toc
xi=(1:200)/10; ri=xi;
[Xi Ri]=meshgrid(xi,ri);
colormap(bone)
figure(1), contour(Xi,Ri,min(d,500))
xlabel('x'),ylabel('r')
```

```
%SPQ: Biofilms

global V K Y

Y=1; V=1; K=1;
tf=10; Fs=32; tt=0:(1/Fs):tf;
x0=zeros(3,1);x=x0;
x0(1,1)=100;
x0(2,1)=5;

[t x]=ode15s('spqrhs',tt,x0);

plot(x)

%%%%%%%%%%%%
function xp=spqrhs(t,x)

global V K Y

xp=x;

xp(1)=-(V*x(1)*x(2)/(Y*(K+x(1))));
xp(2)=(V*x(1)*x(2)/(K+x(1))-x(2)/(1+x(1))+x(1)*x(3)/(1+x(1)));
xp(3)=x(2)/(1+x(1))-x(1)*x(3)/(1+x(1));
```

```
%% The Hoppensteadt-Keller Model of Cicada Synchronization
% cicada.m

clear x; clear t; clear P;
%KK=maximum carrying capacity k

%initialize
nmax=100; L=13; KK=10000; alpha=0.95; A=0.042; R=0.95; f=10.0;
t=(1:1:nmax); %time
x=zeros(1,nmax);
P=zeros(1,nmax);
for i=1:L
x(i)=100; %start with population 100
P(i)=0; %no predation for the first L generations
end;
t=(1:1:nmax); %time

%main loop
for n=L+1:nmax-1
Ksum=KK; %compute the current carrying capacity
for j=1:L-1
Ksum=Ksum-x(n-j)*(alpha^j);
end
K(n)=max(Ksum,0); %carrying capacity can't be negative

%update the predator population
P(n+1)=R*P(n)+A*x(n-L)*alpha^L;
J=max(f*(x(n-L)*alpha^L-P(n)),0);
x(n)=min(K(n),J);
end

P=P(:,1:nmax); %decrease the size of P by one element for plotting
%plot results
stairs(t,x);
hold on; %use to plot x and p on the same graph
stairs(t,P,'r'); %the 'r' makes the predator curve red
hold off;
title('Number of nymphs and predators per year');
```

```
% Result for L=7, KK=10,000, f=10, alpha=R=0.95, A=.042, nmax=150:
%approach the balanced solution.
%
% Result using parameters as above, but change L to 13:
%approach the synchronized solution.
```

4 Chapter 8: MATLAB programs

```
%Fisher Wright Model with noise
function gene

global r s t eps rbar sbar tbar

sigamp=3;rhoamp=2;tauamp=1.5;
eps=0.1;nf=1000;
rho=rhoamp*rand(nf,1);
r=ones(nf,1)+eps*rho;
rbar=rhoamp*(1+eps/2);
sig=sigamp*rand(nf,1);
s=ones(nf,1)+eps*sig;
sbar=sigamp*(1+eps/2);
tau=tauamp*rand(nf,1);t=ones(nf,1)+eps*tau;tbar=tauamp*(1+eps/2);

g=zeros(nf,1);g(1,1)=0.2;
for n=2:nf
   g(n)=genf(g(n-1),n);
end

tf=eps*nf;
tt=0:.01:tf;
x0=g(1,1);x=x0;
[t x]=ode15s('genrhs',tt,x0);

plot((1:nf)*eps,g, tt,x,'r')

%%%%%%%
function xp=genrhs(t1,x)
global r s t eps rbar sbar tbar
xp=x;
xp=(rbar.*x.^2+sbar.*x.*(1-x))./(rbar.*x.^2+2*sbar.*x.*(1-x)+tbar.*(1-x).^2)-x;

%%%%%%%
function y=genf(x,m)
global r s t eps rbar sbar tbar
```

```
y=(r(m)*x^2+s(m)*(1-x)*x)/(r(m)*x^2+2*s(m)*x*(1-x)+t(m)*(1-x)^2);
```

```
%Definetti Diagram
% triangular coordinates

L=2/sqrt(3);
a=zeros(4,1);b=a;c=a;
a(1,1)=0;a(2,1)=L;a(3,1)=L/2;a(4,1)=0;
b(1,1)=0;b(2,1)=0;b(3,1)=1;b(4,1)=0;
c=b;
figure(1)
plot3(a,b,c,'-')
figure(2)
plot(a,b,'-')
axis([-.1 1.2 -.1 1.15])
hold on

%Draw Hardy-Weinberg curve
Npts=10;Nfreqs=25;
x=zeros(Npts,1);y=x;z=x;
for j=1:Nfreqs
   p=j/(Nfreqs+1);q=1-p;
   a(j)=p^2;
   b(j)=2*p*q;
   c(j)=q^2;
   x(j)=L*c(j)+L*b(j)/2;
   y(j)=b(j);z(j)=p;
end
figure(1)
plot3(x,y,z)
figure(2)
plot(x,y,'-')
drawnow
```

```
%Fisher-Wright-Haldane Markov Chain
%one locus- two alleles - Binomial Chain

N=10; %population size
Nt=100; %number of time steps
g=zeros(N,1); %the number of a genes
g(1,1)=5;
for nt=2:Nt
    p=g(nt-1,1)/(2*N);
    r=rand
    q=0;
    for j=0:(2*N)
        q=q+factorial(2*N)*p^j*(1-p)^(2*N-j)/(factorial(j)*factorial(2*N-j));
        if(q>r) break; end
    end
    g(nt)=j
end

plot(g)
axis([0 Nt 0 2*N])
```

```
%Plasmid Hypergeometric Markov Chain

global num den1 den2
N=5; %copy number
Nt=100; %number of time steps
g=zeros(Nt,1); %the number of a genes
g(1,1)=1;
denom=plabin(2*N,N);
N2=2*N;
for nt=2:Nt
    i=g(nt-1,1);
    r=rand;
    q=0;
    for j=0:N
        inc=plabin(2*i,j)*plabin(2*(N-i),N-j)/denom;
        q=q+inc;
        if(q >= r) break; end
    end
    g(nt)=j;
end
plot(g)
axis([0 Nt 0 2*N])

%%%%%%%%%%%%%
function b=plabin(m,n)
global num den1 den2
num=m;
den1=n;
den2=m-n;
if(m<n)
    b=0;
elseif (m<(m-n))
    b=0
else
    b=factorial(num)/(factorial(den1)*factorial(den2));
end
```

```
%Markov Chain Simulation

M=10;  % number of states
N=1000; % number of time steps
P=rand(M); %transition matrix
Psum=P*ones(M,1); P=diag(1./Psum)*P; %Normalize P

%Generate sample path of length N
z=ones(N,1); for n=2:N
    P1=tril(ones(M))*P(z(n,1),:)'
    x=rand;
    for k=1:M
        if(x<P1(k,1))break;end
    end
    z(n,1)=k;
end

figure(1), hist(z)
figure(2), plot(z)
figure(3),
bar(1:M,ones(1,M)*P^1000/M), axis([0 M 0 0.2])
```

```
%cusp catastrophe surface

a=0:.1:15;x=-3:.1:3;
[A X]=meshgrid(a,x);
b=(-X.^3+A.*X)/2;
plot3(b,A,X)
rotate3d on
```

5 Chapter 9: MATLAB programs

```
%Epidemics Reed Frost

N=5; %Family size
Nt=10; %final time
p=0.5; %contact probability
S=zeros(Nt,1); I=S;
I(1,1)=1; %Initial number of infectives
S(1,1)=N-I(1,1);

for nt=2:Nt
    q1=(1-p)^(I(nt-1,1));
    r=rand;
    q=0;
    for k=0:S(nt-1,1)
        q=q+reebin(S(nt-1,1),k)*q1^k*(1-q1)^(S(nt-1,1)-k);
        if(q>r) break; end
    end
    S(nt,1)=k;
    I(nt,1)=S(nt-1,1)-k;
end

plot(1:Nt,S,1:Nt,I)

%%%
function b=reebin(m,n)
global num den1 den2
num=m;
den1=n;
den2=m-n;
if(m<n)
    b=0;
elseif (m<(m-n))
    b=0
else
    b=factorial(num)/(factorial(den1)*factorial(den2));
end
```

```
%Kermack-McKendrick SIR model

global r sigma

r=.5; sigma= 1;
N=10; %population size
tf=10; Fs=32; tt=0:(1/Fs):tf;

x0=zeros(3,1);x=x0;

x0(2,1)= 3; %Initial infective population size
x0(1,1)=N-x0(2,1); %Initial susceptible population size
[t x]=ode15s('kerrhs',tt,x0)

figure(1), plot(t,x(:,1),t,x(:,2),t,x(:,3))

figure(2), plot(x(:,1),x(:,2))
axis([0 N 0 N])

%%%%%%%
function xp=kerrhs(t,x)
global r sigma
xp=x;
xp(1)=-r*x(1)*x(2);
xp(2)=r*x(1)*x(2)-sigma*x(2);
xp(3)=sigma*x(2);
```

```
%schistosomiasis

N=100;
g=zeros(N,2);h=g;f=zeros(N,1);
for j=1:N
   c=[1 j/10-.5 j/10];
   g(j,:)=roots(c)’;
end
for k=1:N
   if(imag(g(k,1)))
      h(k,1)=0;
   else
      h(k,1)=-g(k,1);
   end
   if(imag(g(k,2)))
      h(k,2)=0;
   else
      h(k,2)=-g(k,2);
   end
end
del=(1:N)/10;
plot(del,h(:,1),’.’,del,h(:,2),’.’,del,f(:),’.’)
ylabel(’Helminths’),xlabel(’\delta’)
```

6 Chapter 10: MATLAB programs

```
%Patterns

%Sample path 1-D rw with no for loops
N=1000;
y=zeros(N,1);

y=randn(N,1);
z=tril(ones(N))*y;
plot(1:N,z)
```

```
%spg2dif.m
%Formation of ring pattern due to interaction of nutrient, buffer and cell growth

global M D D1

M=20; tt=0:0.1:20;
del=0.1;D1=0.05;D=D1;%orig D1=D=0.1
s0=zeros(M);s=s0;
for k=1:M,
    for j=1:M
      s0(k,j)=50*exp(-del*((k-M/2)^2+(j-M/2)^2));%orig 50->10
  end
end

p0=ones(M); g0=2*ones(M);%orign 2->1

figure(1),
subplot(3,1,1), mesh(s0);
subplot(3,1,2), mesh(p0);
subplot(3,1,3), mesh(g0);

S0=reshape(s0,M^2,1);P0=reshape(p0,M^2,1);G0=reshape(g0,M^2,1);
V0=[S0;P0;G0]; V=V0;

options=odeset('AbsTol',1e-8,'RelTol',1e-6);
%tic
%[t V]=ode15s('spgrhs',tt,V0,options);
%toc
tic
[t V]=ode45('spgrhs',tt,V0,options);
toc

[m,n]=size(V);
s=reshape(V(m,1:M^2),M,M);p=reshape(V(m,M^2+1:2*M^2),M,M);
g=reshape(V(m,2*M^2+1:3*M^2),M,M);
figure(2)
subplot(4,1,1),mesh(s);subplot(4,1,3),mesh(p);
subplot(4,1,4),plot(1:M,p(M/2,:)); subplot(4,1,2),mesh(g);
```

```
%%%%%%%%%%%%%
function vp=spgrhs(t,v)
%spgrhs.m
global M D D1

vp=v; S=v(1:M^2,1);P=v(M^2+1:2*M^2,1);G=v(2*M^2+1:3*M^2,1);
s=reshape(S,M,M);p=reshape(P,M,M);g=reshape(G,M,M);
sp=s;pp=p;gp=g;

for k=1:M
   for j=1:M
      sp(k,j)=-s(k,j)*p(k,j)*g(k,j)/(1+s(k,j)*g(k,j));
      pp(k,j)=-sp(k,j);
      gp(k,j)=sp(k,j);
      sp(k,j)=sp(k,j)+D*(s(spgind(k+1),j)+s(spgind(k-1),j)+..
       s(k,spgind(j+1))+s(k,spgind(j-1))-4*s(k,j));
      gp(k,j)=gp(k,j)+D1*(g(spgind(k+1),j)+g(spgind(k-1),j)+..
       g(k,spgind(j+1))+g(k,spgind(j-1))-4*g(k,j));
   end
end
Sp=reshape(sp,M^2,1);Pp=reshape(pp,M^2,1);Gp=reshape(gp,M^2,1);
vp=[Sp;Pp;Gp];

%%%%
function ii=spgind(k)
global M D
ii=k;
if(k<1)ii=2;
end
if(k>M)ii=M-1;
end
```