Appendix

Ryan Ly

December 18, 2018

1 Matlab Code

1.1 Continuous SIR Model

```
%continuous sir model ode
function dydt = sir(t,y,beta,gamma,b,N)
ds = -beta*y(1)*y(2)/N + b*(y(2)+y(3));
di = beta*y(1)*y(2)/N - (b+gamma)*y(2);
dr = gamma*y(2)-b*y(3);
dydt=[ds;di;dr];
end

%least squares best fit of SIR model
function J = lst_sir(p,xdata)
N = 106487;
i0=161;
s0=N-i0;
r0=N-i0-s0;
time=[1:1:52];

[t,y] = ode23(@sir,time,[s0,i0,r0],[],p(1),p(2),0,N)
errx=y(:,2)-xdata(:,1);
J = errx'*errx;
end

%plot model vs data infected individuals
xdata=[161;188;201;236;325;394;558;608;997;1281;1912;3125;3801;4327;
4538;3954;4334;4278;3990;3327;2649;1837;1486;1275;1012;982;821;596;
436;329;247;118;69;56;28;25;22;14;9;21;23;29;18;21;20;23;45;18;36;
48;46;36];
[x,fval]=fminsearch(@lst_sir,[3.9928,3.5170],[],xdata);

N = 106487;
```

```
i0=161;
s0=N-i0;
r0=N-i0-s0;
time=[1:1:52];
[t,y] = ode23(@sir,time,[s0,i0,r0],[],x(1),x(2),0,N);

plot(t,y(:,2),'r','Linewidth',2,'DisplayName','Continuous SIR');
hold on
plot(t,xdata(:),'o','Linewidth',2,'DisplayName','CDC Data');
xlabel('Time');
ylabel('Number of Infected Individuals');
%title('Continuous SIR Best Fit')
legend(gca,'show','location','northeastoutside');
```

1.2 Continuous Time Markov Chain Model

```
% Continuous Time Markov Chain
% SIR Epidemic Model
clear
beta=1.1837;
b=0;
gamma=0.8691;
N=106487;
i0=161;
s0=N-i0;
time=52;
sim=3;
for k=1:sim
    clear t s i r
    t(1)=0;
    i(1)=i0;
    s(1)=s0;
    r(1)=N-i0-s0;
    j=1;
    while i(j)>0 && t(j)<time
        u1=rand; % uniform random number
        u2=rand; % uniform random number
        a=(beta/N)*i(j)*s(j)+(gamma)*i(j)+b*(N-s(j));
        prob0=(beta*i(j)*s(j)/N)/a;
        prob1=(gamma*i(j))/a;
        prob2=(b*i(j))/a;
        prob3=(b*r(j))/a;
        t(j+1)=t(j)-log(u1)/a;
        if u2 ≤ prob0;
        i(j+1)=i(j)+1;
        s(j+1)=s(j)-1;
        r(j+1)=r(j);
        elseif prob0 < u2 ≤ prob1+prob0;
        i(j+1)=i(j)-1;
        s(j+1)=s(j);
        r(j+1)=r(j)+1;
        elseif prob1+prob0 < u2 ≤ prob2+prob1+prob0;
```

```
        i(j+1)=i(j)-1;
        s(j+1)=s(j)+1;
        r(j+1)=r(j);
        else prob2+prob1+prob0 < u2 ≤ prob3+prob2+prob1+prob0;
        i(j+1)=i(j);
        s(j+1)=s(j)+1;
        r(j+1)=r(j);
        end
        j=j+1;
    end
    plot(t,s,'r','LineWidth',2)
    hold on
    plot(t,i,'b','LineWidth',2)
    plot(t,r,'y','LineWidth',2)
    title('CTMC');
    xlabel('Time');
    ylabel('Number of Individuals');
    legend('Susceptible','Infected','Recovered','location','northeastoutside');
end
```

1.3 Stochastic Differential Equation Model

```
% Stochastic Differential Equation
% SIR Epidemic Model
clear
beta=1.1837;
b=0;
gamma=0.8691;
N=106487;
i0=161;
s0=N-i0;
dt=0.01;
time=52;
sim=3;
for k=1:sim
    clear i t s r
    j=1;
    i(j)=i0;
    s(j)=s0;
    r(j)=N-i0-s0;
    t(j)=dt;
    while i(j)>0 && t(j)<time
        mu_s=-1*beta*s(j)*i(j)/N;
        V=[beta*s(j)*i(j)/N, -1*beta*s(j)*i(j)/N; ...
            -1*beta*s(j)*i(j)/N, beta*s(j)*i(j)/N+gamma*i(j)];
        B=sqrt(V);
        mu_i=beta*s(j)*i(j)/N-(gamma)*i(j);
        rn1=randn; % standard normal random number
        rn2=randn;
        s(j+1)=s(j)+mu_s*dt+B(1)*sqrt(dt)*rn1+B(3)*sqrt(dt)*rn2;
        i(j+1)=i(j)+mu_i*dt+B(2)*sqrt(dt)*rn1+B(4)*sqrt(dt)*rn2;
        r(j+1)=N-s(j+1)-i(j+1);
```

```
30          t(j+1)=t(j)+dt;
31          j=j+1;
32      end
33      plot(t,s,'r','Linewidth',2);
34      hold on
35      plot(t,i,'b','Linewidth',2);
36      plot(t,r,'y','Linewidth',2);
37      title('SDE');
38      xlabel('Time');
39      ylabel('Number of Individuals');
40      legend('Susceptible','Infected','Recovered','location','northeastoutside');
41  end
```