MATLAB Codes Set

Note:

- This is a set of our project codes
- Our codes contain 5 models and 1 simulation blocks

1. model_with_phi.m

```
function dx = model_with_phi(t, x)
% This is the model with only phi(I)
% List the parameters and equations here
d = 1;
r = 1;
beta = 1;
N = 1000;
dx(1) = d * N - beta * x(1) * x(2) / (1 + x(2)) - d * x(1);% x(1) is S
dx(2) = beta * x(1) * x(2) / (1 + x(2)) - d * x(2) - r * x(2);% x(2) is I
dx(3) = r * x(2) - d * x(3);% x(3) is R
dx = dx(:);
end
```

2. model_with_phi2.m

```
function dx = model_with_phi2(t, x)
% This is the model with only phi(I)
% List the parameters and equations here
d = 1;
r = 1;
beta = 1;
N = 1000;
dx(1) = d * N - beta * x(1) * x(2) / (1 + 0.001 * x(2) + 0.001 * x(2)^2) - d * x(1);% x(1) is S
dx(2) = beta * x(1) * x(2) / (1 + 0.001 * x(2) + 0.001 * x(2)^2) - d * x(2) - r * x(2);% x(2) is I
dx(3) = r * x(2) - d * x(3);% x(3) is R
dx = dx(:);
end
```

3. model_with_phi_exp.m

```
function dx = model_with_phi_exp(t, x)
% This is the model with only phi(I) = 100 + e^{-I}
% List the parameters and equations here
d = 1;
r = 1;
beta = 1;
N = 1000;
dx(1) = d * N - beta * x(1) * x(2) / (100 + exp(1)^(-x(2))) - d * x(1);% x(1) is S%
(100 + exp(1)^(-x(2)))
dx(2) = beta * x(1) * x(2) / (100 + exp(1)^(-x(2))) - d * x(2) - r * x(2);% x(2) is I
dx(3) = r * x(2) - d * x(3);% x(3) is R
dx = dx(:);
end
```

4. model_with_r.m

```
function dx = model_with_r(t, x)
% This is the model with only r(I)
% List the parameters and equations here
d = 0.01;
% r = 1;
beta = 0.01;
N = 1000;
dx(1) = d * N - beta * x(1) * x(2) - d * x(1);% x(1) is S
dx(2) = beta * x(1) * x(2) - d * x(2) - (0.5 + (1 - 0.5) * 1 / (1 + x(2))) * x(2);%
x(2) is I
dx(3) = (0.5 + (1 - 0.5) * 1 / (1 + x(2))) * x(2) - d * x(3);% x(3) is R
dx = dx(:);
end
```

5. model_with_both.m

```
function dx = model_with_both(t, x)
% This is the model with r(I) & \phi(I)
% List the parameters and equations here
d = 0.01;
% r = 1;
beta = 0.01;
N = 1000;
dx(1) = d * N - beta * x(1) * x(2) / (1 + 0.001 * x(2) + 0.001 * x(2)^2) - d * x(1);%
x(1) is S
dx(2) = beta * x(1) * x(2) / (1 + 0.001 * x(2) + 0.001 * x(2)^2) - d * x(2) - (0.5 +
(1 - 0.5) * 1 / (1 + x(2))) * x(2);% x(2) is I
dx(3) = (0.5 + (1 - 0.5) * 1 / (1 + x(2))) * x(2) - d * x(3);% x(3) is R
dx = dx(:);
end`
```

6.simulation.m

```
 %%
 % The model with only phi(I) = 1 + I
[t, x] = ode45(@model_with_phi, 0:0.001:6, [999, 1, 0]);
% subplot(1, 3, 1)
plot(t,x)
title('S,I,R vs Time, When \phi(I) = 1 + I, \beta = 1, d = 1, N = 1000')
xlabel('time t');
ylabel('# of population');
legend('S', 'I', 'R');
%%
% The model with only phi(I) = 1 + 0.001 * I + 0.001 * I^2
[t, x] = ode45(@model_with_phi2, 0:0.001:6, [999, 1, 0]);
% subplot(1, 3, 1)
plot(t,x)
title('S,I,R vs Time, When \phi(I) = 1 + 0.001I + 0.001I^2, \beta = 1, d = 1, N =
1000')
xlabel('time t');
ylabel('# of population');
legend('S', 'I', 'R');
%%
% The model with only phi(I) exp
[t, x] = ode45(@model_with_phi_exp, 0:0.001:10, [999, 1, 0]);
% subplot(1, 3, 1)
plot(t,x)
title('S,I,R vs Time, When \phi(I) = 100 + e^{-I}, \beta = 1, d = 1, N = 1000')
xlabel('time t');
ylabel('# of population');
legend('S', 'I', 'R');
%%
% The model with only r(I)
[t, x] = ode23(@model_with_r, 0:0.001:50, [999, 1, 0]);
% subplot(1, 3, 1)
plot(t,x)
title('S,I,R vs Time, When r(I) = 0.5 + 0.5 / (I+1), \beta = 0.01, d = 1, N = 1000')
xlabel('time t');
ylabel('# of population');
legend('S', 'I', 'R');
%%
% The model with only phi(I) & r(I)
[t, x] = ode45(@model_with_both, 0:0.001:80, [999, 1, 0]);
% subplot(1, 3, 1)
plot(t,x)
title('S,I,R vs Time with Both r(I) & \Phi(I), \beta = 0.01, d = 1, N = 1000')
xlabel('time t');
ylabel('# of population');
legend('S', 'I', 'R');
```

Edited by Max Liu on 2019/04/24