Lab 11 pandemic modelling

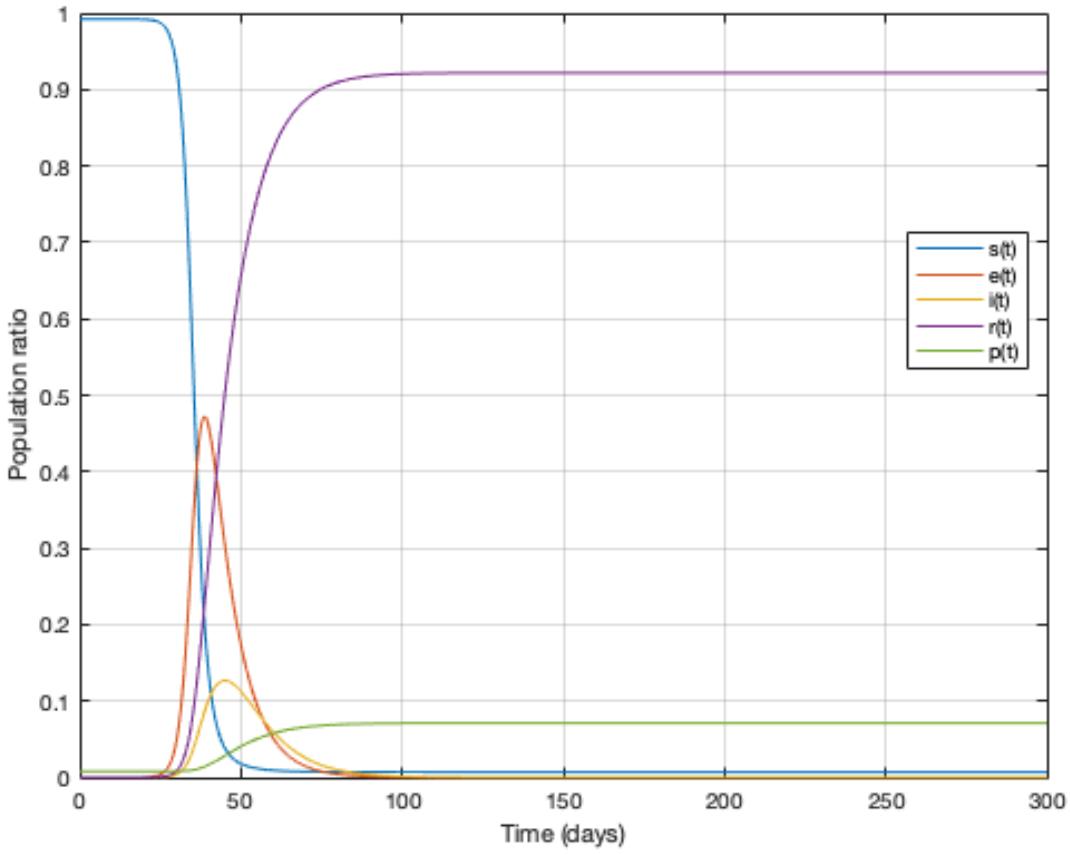
Elizabeth Nemeti, Nov 4 2022

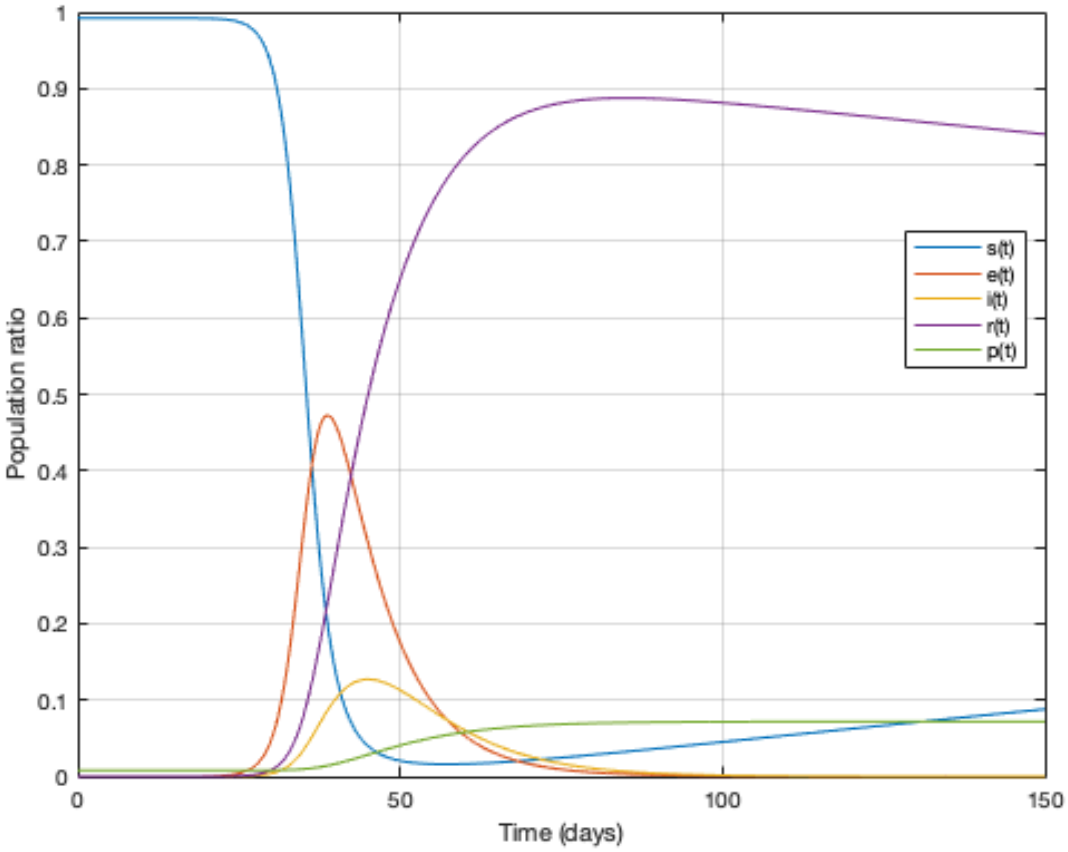
```
clc
close all
clear

constants_1 = [0.65, 0.005, 0.05, 0.08, 0.1, 0.02, 0]; % constants for test
 case 1
constants_2 = [0.65, 0.005, 0.05, 0.08, 0.1, 0.02, 0.001]; % constants for
 test case 2
constants_3 = [0.65, 0.005, 0.05, 0.08, 0.1, 0.02, 0.001]; % constants for
 test case 3

test_1 = pandemic_modelling(constants_1, 300, false)
% saveas(figure(1), "test1.png")
test_2 = pandemic_modelling(constants_2, 150, false)
% saveas(figure(2), "test2.png")
test_3 = pandemic_modelling(constants_3, 4000, false)
% saveas(figure(3), "test3.png")
test_4 = pandemic_modelling(constants_3, 4000, true)
% saveas(figure(4), "test4.png")


function out_plot = pandemic_modelling(constants, final_time, I_E_choice)

    p0 = 8E-3;
    e0=1E-8;
    init_conditions = [1-p0-e0;e0;0;0;p0];
    t = linspace(0, final_time, 10e2);
    options = odeset('RelTol',1e-12,'AbsTol',1e-12);
    [t_out,SIR_out] =
 ode45(@(t,x)model1(t,x,constants),t,init_conditions,options);

    if I_E_choice == true
        out_plot = figure;
        hold on
        plot(t_out, SIR_out(:,2))
        plot(t_out, SIR_out(:,3))
        legend("e(t)", "i(t)", location = "best")
        xlabel("Time (days)")
        ylabel("Population ratio")
        grid on
    else
        out_plot = figure;
        plot(t_out, SIR_out(:,1))
        hold on
        plot(t_out, SIR_out(:,2))
        plot(t_out, SIR_out(:,3))
        plot(t_out, SIR_out(:,4))
        plot(t_out, SIR_out(:,5))
```

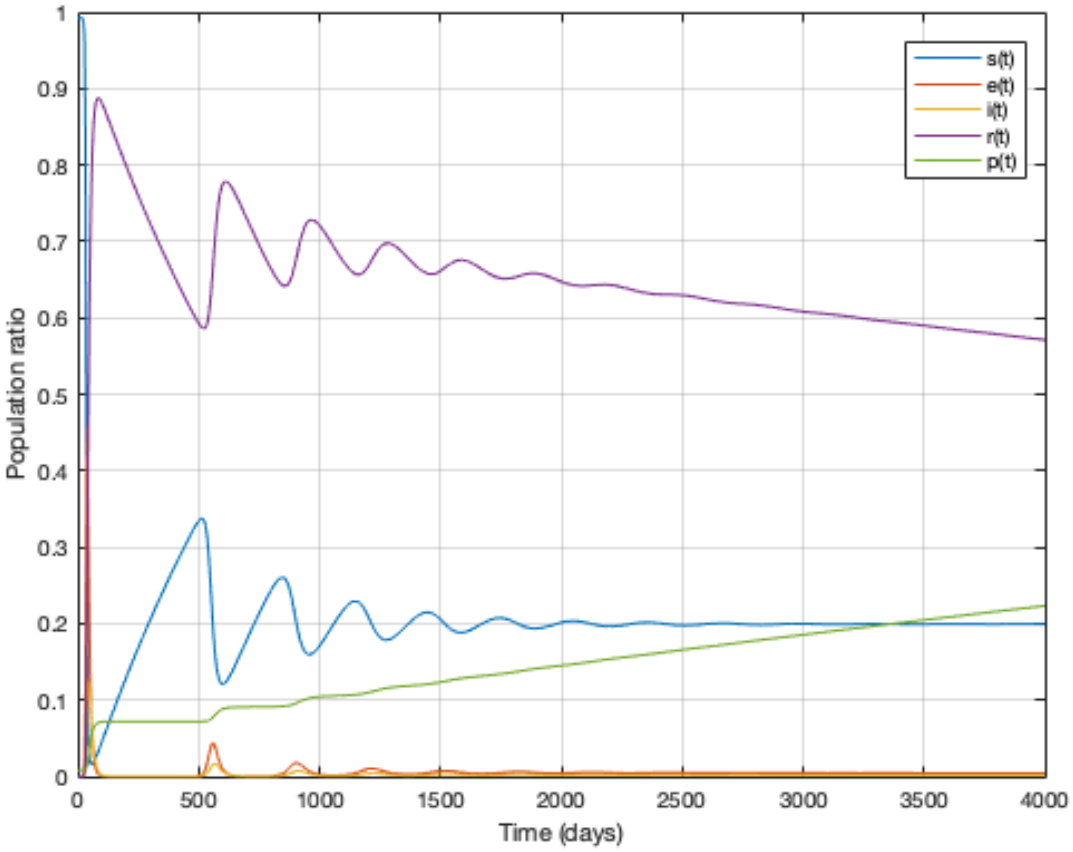
```
        legend("s(t)", "e(t)", "i(t)", "r(t)", "p(t)", location = "best")
        xlabel("Time (days)")
        ylabel("Population ratio")
        grid on
    end

    function dSIR = model1(t,x, const)

        alpha_e = const(1);
        alpha_i = const(2);
        kappa = const(3);
        rho = const(4);
        beta = const(5);
        mu = const(6);
        gamma = const(7);

        dSIR(1) = -alpha_e*x(1)*x(2) -alpha_i*x(1)*x(3)+gamma*x(4);
        dSIR(2) = alpha_e*x(1)*x(2) +alpha_i*x(1)*x(3)- kappa*x(2) -rho*x(2);
        dSIR(3) = kappa*x(2)-beta*x(3)-mu*x(3);
        dSIR(4) = beta*x(3)+rho *x(2) -gamma*x(4);
        dSIR(5) = mu * x(3);
        dSIR = dSIR(:);
    end
end
```

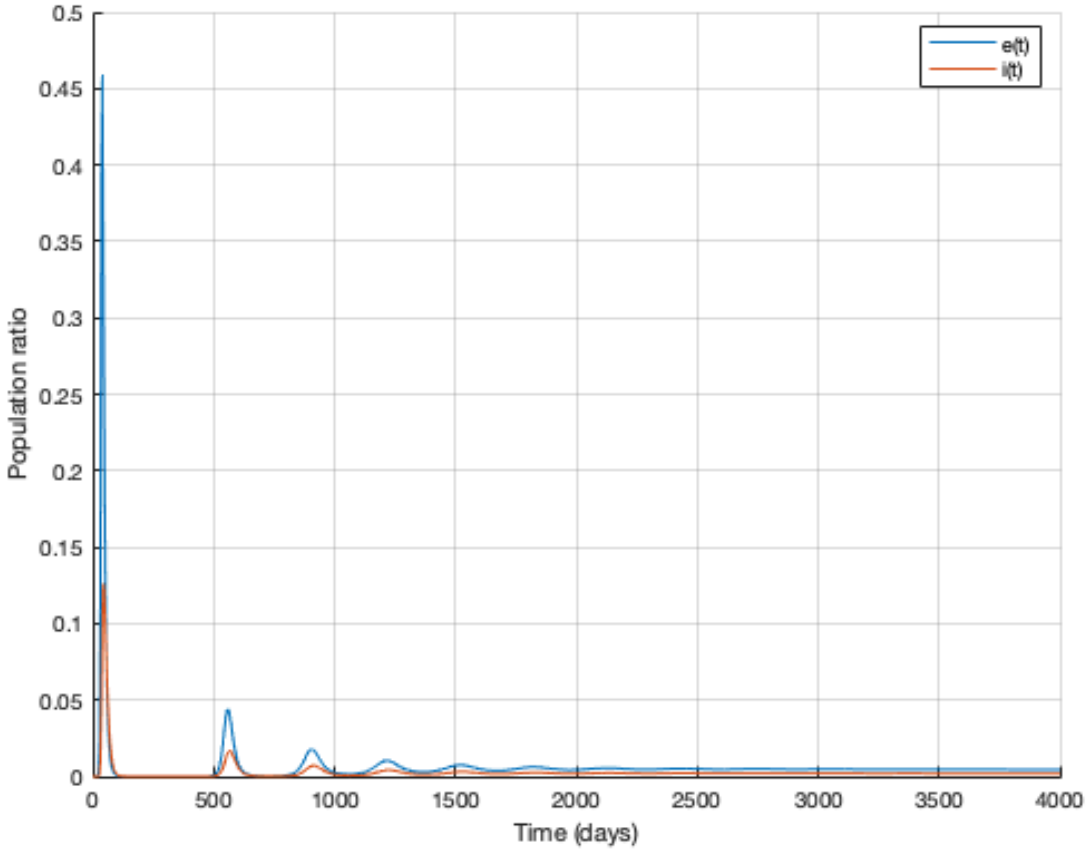
```
test_1 =

  Figure (1) with properties:

      Number: 1
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 278 560 420]
       Units: 'pixels'

  Use GET to show all properties


test_2 =

  Figure (2) with properties:

      Number: 2
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 278 560 420]
       Units: 'pixels'

  Use GET to show all properties


test_3 =
```

```
  Figure (3) with properties:

      Number: 3
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 278 560 420]
       Units: 'pixels'

  Use GET to show all properties


test_4 =

  Figure (4) with properties:

      Number: 4
        Name: ''
       Color: [0.9400 0.9400 0.9400]
    Position: [360 278 560 420]
       Units: 'pixels'

  Use GET to show all properties
```




Population ratio
Time (days)
s(t)
e(t)
i(t)
r(t)
p(t)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
50
100
150


s(t)
e(t)
i(t)
r(t)
p(t)
Population ratio
Time (days)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
500
1000
1500
2000
2500
3000
3500
4000



Published with MATLAB® R2022a