MATLAB, Lab 7 – Individual work

1. Change the function both **`Dog_Euler`** in order to make an animation of the dog and master motion (hint: use the **`drawnow`** or **`pasue(dt)`** function)

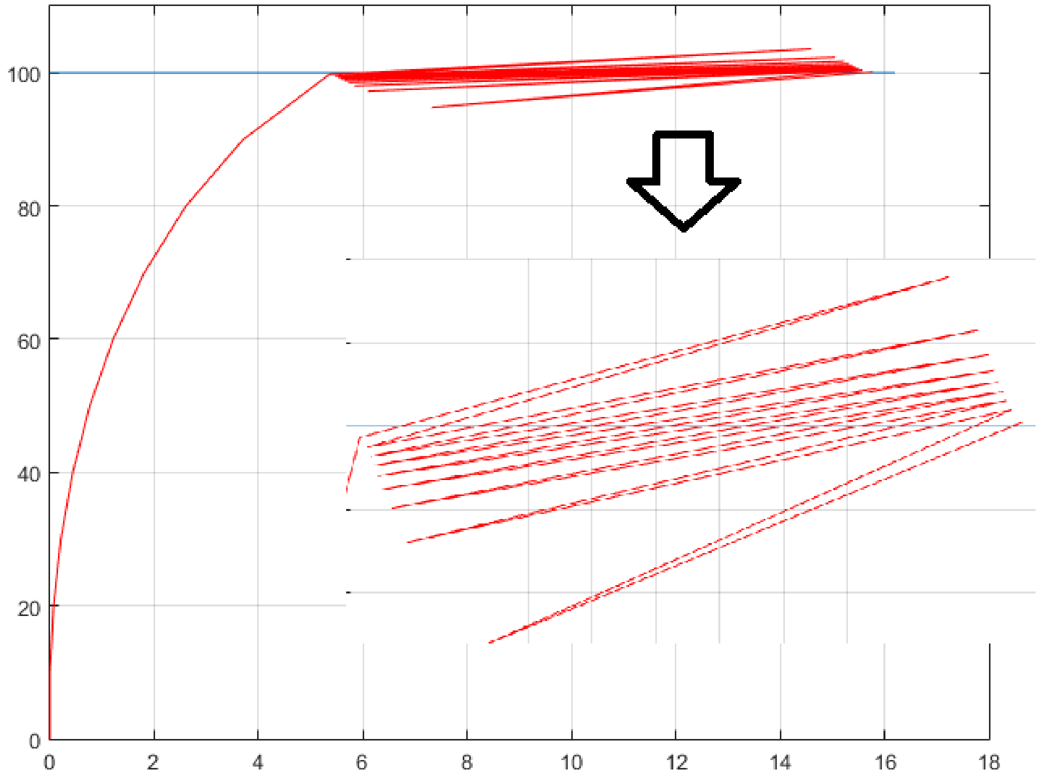
Code:

```
function Dog_Euler

    % Euler's method of solving the Dog-Master problem
         % Constant parameters:
    yM = 100;       % Master's y position [m]
    vM = 3;         % Master's velocity [m/s]
    vD = 5;         % Dog's velocity [m/s]
    n = 500;        % Maximum number of iterations
    dt = 0.2;       % Time step [s]
         % Memory reservation for arrays:
    xM = zeros (1, n);      % Master's x position
    xD = zeros (1, n);      % Dog's x position
    yD = zeros (1, n);      % Dog's y position
         % Initial conditions:
    xM(1) = 0;
    xD(1) = 0;
    yD(1) = 0;
    vDx = 0;        % x component of dog's velocity
    vDy = vD;       % y component of dog's velocity
    DM = yM;        % distance between the dog and the master
    DMmin = 0.5;    % DM that satisfies the solution
    % Iterations:
ii = 2;         % counter of the iterations
while (DM > DMmin & ii <= n)
    xM(ii) = xM(ii-1) + vM*dt;
    xD(ii) = xD(ii-1) + vDx*dt;
    yD(ii) = yD(ii-1) + vDy*dt;
    delta_x = xM(ii) - xD(ii);
    delta_y = yM - yD(ii);
    DM = sqrt(delta_x^2 + delta_y^2);
    if DM > DMmin
       vDx = vD * delta_x/DM;
       vDy = vD * delta_y/DM;
  else
       vDx = vM;
       vDy = 0;
  end
  ii = ii + 1;
end;
ii = ii-1;
for t =1 : ii    % Drawing the solution:
     plot(xM(1:t),zeros(1,t)+yM);
     hold on;
     plot(xD(1:t), yD(1:t));
     grid on;
   % drawnow;     %drawnow function also can be used
   pause(dt)
end

ylim (gca, [0 1.1*yM]);
```

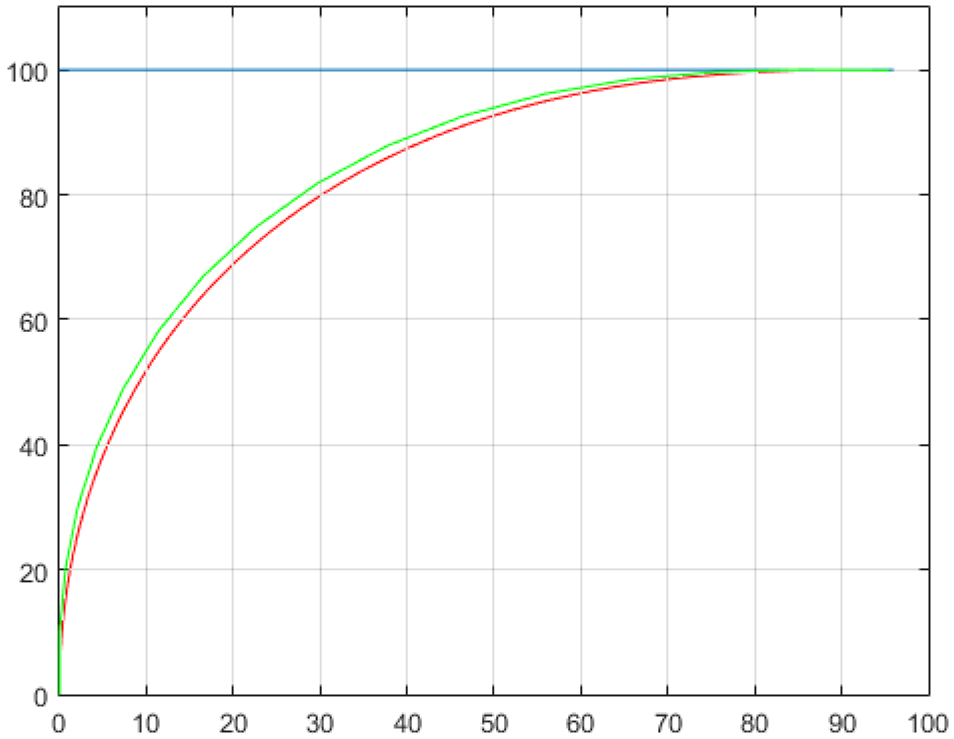
2. Increase the velocity of the dog, try to manipulate another parameters in the program **`Dog_Euler`** in order to obtain a non-stable solution. Provide the screenshot with example of a non-stable solution. In the case of instability, decrease the time step (probably the maximum number of iterations should be changed). Give some comments about stability of the solution and proper choice of the timestep.

Screenshot:


Comments:
The smaller the timestep, the easier it is to find the moment in time when the distance between the dog and the master is smaller than Dmin (afterwards, calculations are being terminated).

3. Repeat the simulation with two values of the timestep Δt_1 and $\Delta t_2 = 10\Delta t_1$. Create a plot that shows both cases and discuss the importance of choosing appropriate timestep. Is it possible to avoid this problem by introducing variable timestep? If so, the explain how this can be done (just in words, do not code it).

Screenshot:



Comments:

Smaller timestep rewards with more fitting, precise and elegant results. This precision can even lead to different time period of obtaining all satisfactory conditions – while zoomed, it’s clearly seen that timestep equal to 2 seconds “has trouble” satisfying the DM < DMmin condition comparing to dt = 0.2 sec.

Variable timestep could be implemented in such a way: the closer the plot is to be ended, the smaller timestep is implemented. At the beginning high precision is not very necessary – the results are unarguably different, but in the end it doesn’t even matter.