% It calculates ODE using Runge-Kutta 4th order method

h=1; % step size

x = 1:h:10; % Calculates up to y(10)

y = zeros(1,length(x));

y(1) = 0; % initial condition

fprintf('y(1) = %.3f\n',y(1));

F\_xy = @(t,r) 1/t - r/t - r; % change the function as you desire

for i=1:(length(x)-1) % calculation loop

k\_1 = F\_xy(x(i),y(i));

k\_2 = F\_xy(x(i)+0.5\*h,y(i)+0.5\*h\*k\_1);

k\_3 = F\_xy((x(i)+0.5\*h),(y(i)+0.5\*h \*k\_2));

k\_4 = F\_xy((x(i)+h),(y(i)+h\* k\_3));

y(i+1) = y(i) + (1/6)\*(k\_1+2\*k\_2+2\*k\_3+k\_4)\*h;% main equation

fprintf('y(%.0f) = %.5f\n',x(i + 1), y(i+1))

end

plot(x,y, 'o')