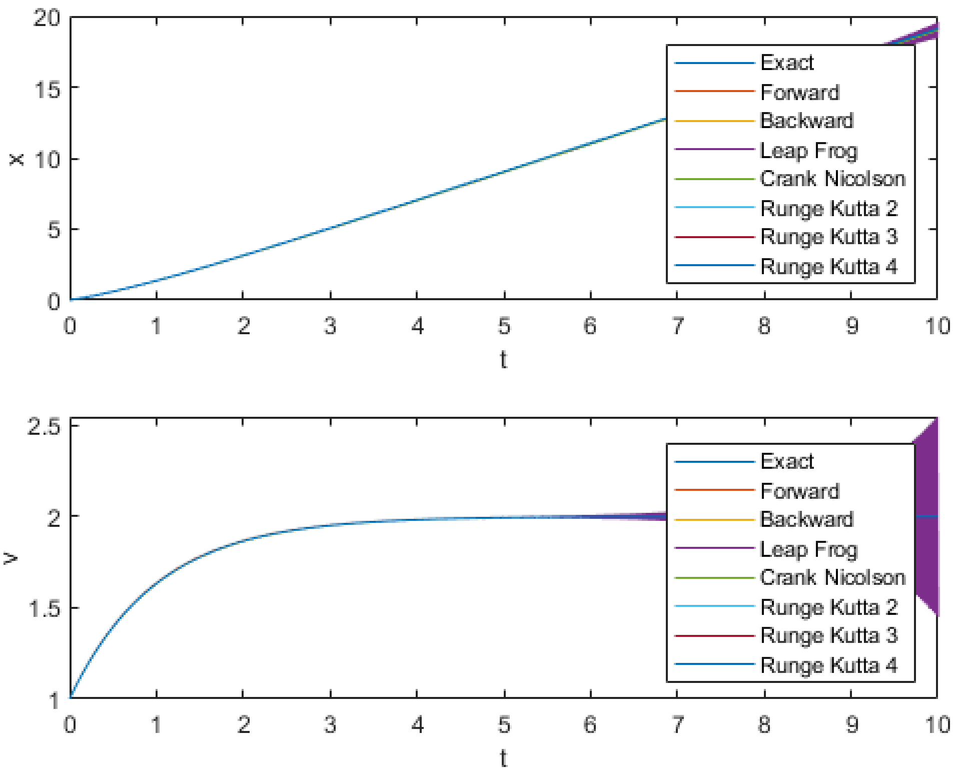
```
clc, clear;
close all;

% Lord fogive me for what im about to do with this code
% make da grid
delta = .01;
t = 0:delta:10;

m = 3; g = 2;
c1 = m*g; c2 = m;
xexact = m/c2*(c1/c2-1)*exp(-c2/m*t)-m/c2*(c1/c2-1)+c1/c2*t;
vexact = -(c1/c2-1)*exp(-c2/m*t)+(c1/c2);

[x,v] = p1a(1,0,delta,'eulerF1a',t);
[xbe,vbe] = p1a(1,0,delta,'eulerB1a',t);
[xlf,vlf] = p1a(1,0,delta,'leapFrog1a',t);
[xck,vck] = p1a(1,0,delta,'crank1a',t);
[xrk2,vrk2] = p1a(1,0,delta,'RK2',t);
[xrk3,vrk3] = p1a(1,0,delta,'RK3',t);
[xrk4,vrk4] = p1a(1,0,delta,'RK4',t);

% Plot area
figure(1)
subplot(2,1,1)
plot(t,xexact,t,x)
hold on;
plot(t,xbe)
hold on;
plot(t,xlf)
hold on;
plot(t,xck)
hold on;
plot(t,xrk2)
hold on;
plot(t,xrk3)
hold on;
plot(t,xrk4)
xlabel('t')
ylabel('x')
legend('Exact','Forward','Backward','Leap Frog','Crank Nicolson',...
        'Runge Kutta 2','Runge Kutta 3','Runge Kutta 4','Location','SE')
subplot(2,1,2)
plot(t,vexact,t,v)
hold on;
plot(t,vbe)
hold on;
plot(t,vlf)
hold on;
plot(t,vck)
hold on;
plot(t,vrk2)
hold on;
```

```
plot(t,vrk3)
hold on;
plot(t,vrk4)
xlabel('t')
ylabel('v')
legend('Exact','Forward','Backward','Leap Frog','Crank Nicolson',...
        'Runge Kutta 2','Runge Kutta 3','Runge Kutta 4','Location','SE')

function [x,v] = p1a(v0,x0,delta,method,t)
m = 3; g = 2;
c1 = m*g; c2 = m;
ydot = @(v) [c1/m-c2/m*v;v];

%Function handler for different methods
fEuler = @(dy,y,h) dy.*h+y;
leapFrog = @(dy,y,h) dy*2.*h+y;

switch method
    case 'eulerF1a'
        yold = [v0;x0];
        for i = 1:length(t)-1
            y = yold(:,i);
            dy = ydot(y(1,1));
            ynew = fEuler(dy,y,delta);
            yold(:,i+1) = ynew;
        end
        y = yold;
        v = y(1,:);
        x = y(2,:);
    case 'eulerB1a'
        bEulerV = @(y,c1,c2,m,h) (y.*m+c1.*h)/(m+c2.*h); %explicit backward
 euler for v
        bEulerX = @(y,v,h) y+v.*h;  %explicit backward euler for x
        gv = v0;
        gx = x0;
        v(1) = v0;
        x(1) = x0;
        % Fixed point iterations
        for i = 1:length(t)-1
            v(i+1) = bEulerV(gv,c1,c2,m,delta);
            x(i+1) = bEulerX(gx,v(i),delta);
            gv = v(i+1);
            gx = x(i+1);
        end
    case 'leapFrog1a'
        % Using forward euler to get the second point (y_n)
        y = [v0;x0];
        dy = ydot(v0);
        y2 = fEuler(dy,y,delta);
        y(:,2) = y2;
        for i = 1:length(t)-2
            dy = ydot(y(1,i+1));
            yi = y(:,i);
            ynew = leapFrog(dy,yi,delta);
            y(:,i+2) = ynew;
        end
        x = y(2,:);
        v = y(1,:);
    case 'crank1a'
        crankV = @(v,c1,c2,m,delta) (2*c1*delta+v*(2*m-c2*delta))/(2*m
+c2*delta);
        crankX = @(v1,v2,x) delta*v1/2+delta*v2/2+x;
        vdot = @(v) c1/m-c2/m*v;
        y = [v0;x0];
        for i=1:length(t)-1
            vi = y(1,i);
            xi = y(2,i);
            dv = vdot(vi);
            vnew = crankV(vi,c1,c2,m,delta);
            xnew = crankX(vi,vnew,xi);
            y(:,i+1) = [vnew;xnew];
        end
        x = y(2,:);
        v = y(1,:);
    case 'RK2'
        y = [v0;x0];
        for i=1:length(t)-1
            vi = y(1,i);
            k1v = delta*(c1/m-c2/m*vi);
            k2v = delta*(c1/m-c2/m*(vi+k1v));
            k1x = delta*vi;
            k2x = delta*(vi+k1x);
            x = y(2,i)+.5*(k1x+k2x);
            v = vi + .5*(k1v+k2v);
            y(:,i+1) = [v;x];
        end
        x = y(2,:);
        v = y(1,:);
    case 'RK3'
        y = [v0;x0];
        for i=1:length(t)-1
            vi = y(1,i);
            k1v = delta*(c1/m-c2/m*vi);
            k2v = delta*(c1/m-c2/m*(vi+k1v/2));
            k3v = delta*(c1/m-c2/m*(vi-k1v+2*k2v));
            k1x = delta*vi;
            k2x = delta*(vi+k1x/2);
            k3x = delta*(vi-k1x+2*k2x);
            x = y(2,i) + 1/6*(k1x+k2x*4+k3x);
            v = vi + 1/6*(k1v+k2v*4+k3v);
            y(:,i+1) = [v;x];
        end
        x = y(2,:);
        v = y(1,:);
    case 'RK4'
        y = [v0;x0];
        for i=1:length(t)-1
            vi = y(1,i);
```

```
      k1v = delta*(c1/m-c2/m*vi);
      k2v = delta*(c1/m-c2/m*(vi+k1v/2));
      k3v = delta*(c1/m-c2/m*(vi+k2v/2));
      k4v = delta*(c1/m-c2/m*(vi+k3v));
      k1x = delta*vi;
      k2x = delta*(vi+k1x/2);
      k3x = delta*(vi+k2x/2);
      k4x = delta*(vi+k3x);
      x = y(2,i) + 1/6*(k1x+k2x*2+k3x*2+k4x);
      v = vi + 1/6*(k1v+k2v*2+k3v*2+k4v);
      y(:,i+1) = [v;x];
    end
    x = y(2,:);
    v = y(1,:);
end

end
```



Published with MATLAB® R2021b