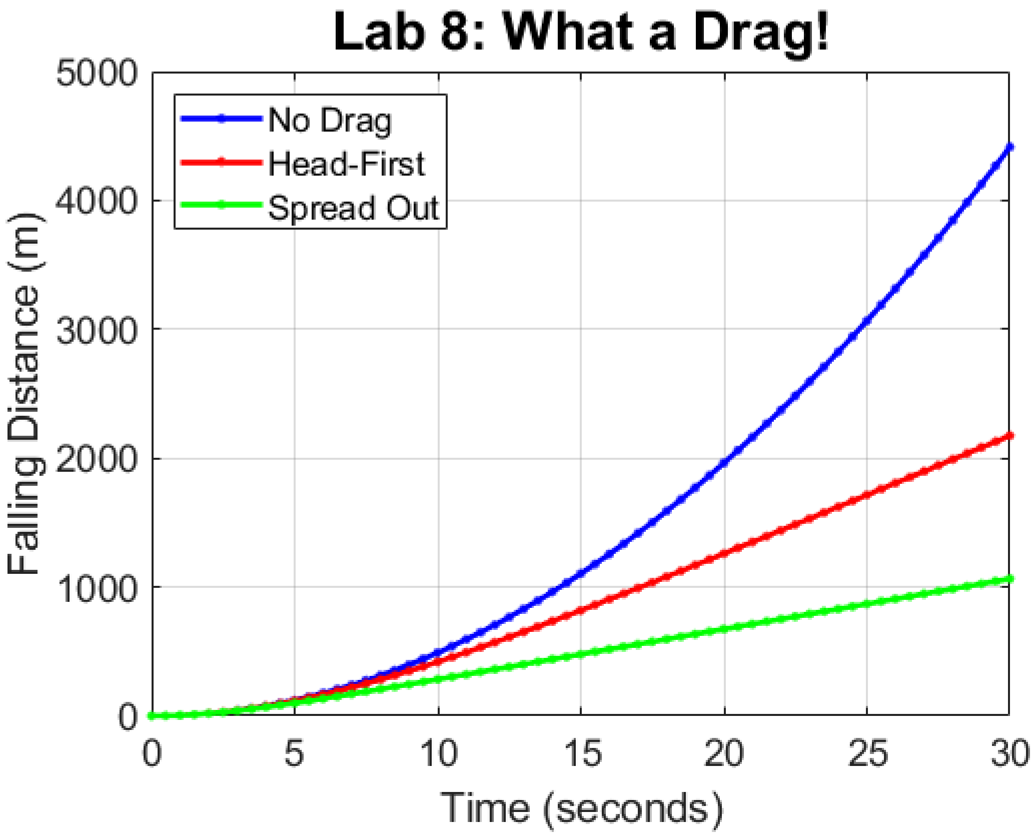
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MATH_151_Lab8
%--------------------------------------------------------------------------
% C Rocheleau, Colorado State University
% 10/25/2023
%--------------------------------------------------------------------------
% Answer key for MATH-151 Lab 8 for the Fall 2023 semester
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

close all; clear all; clc;
```

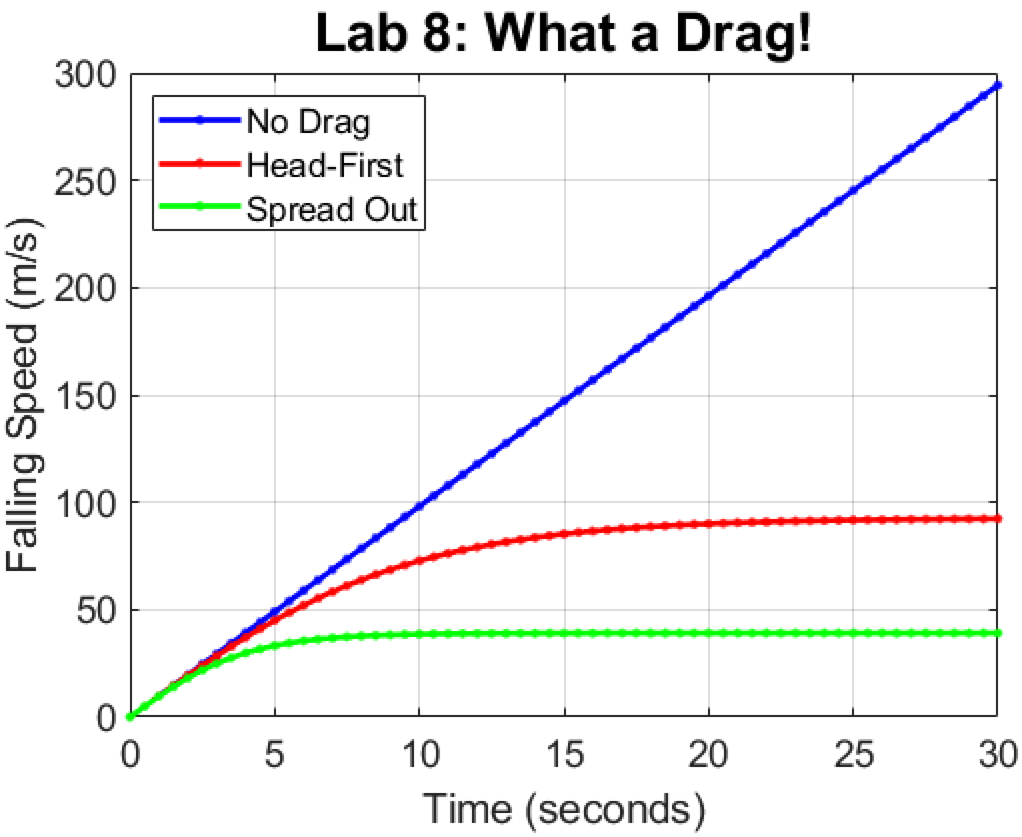
Task 1: What a Drag!

```
% Defining Constants
g = 9.81;       % gravity, m/s^2
m = 70;         % mass, kg
dt = 0.5;       % time step, s
t  = 0:dt:30;   % time vector

% a) Modling a skydiver falling with no drag

H      = zeros(size(t));
Hprime = 0;       v = H;

% Solving second-order ODE with constant acceleration using Runge-Kutta 4
for iTime = 2:length(t)
    haccel = m*g/m;
    k1 = haccel;    % Acceleraiton is constant, so it is simple
    l1 = Hprime;

    k2 = haccel;
    l2 = Hprime + (dt/2)*k1;

    k3 = haccel;
    l3 = Hprime + (dt/2)*k2;

    k4 = haccel;
    l4 = Hprime + dt*k3;


    Hprime = Hprime + (dt/6)*(k1 + 2*k2 + 2*k3 + k4);
    v(iTime) = Hprime;  % Store time-series of velocity for plotting later
    H(iTime) = H(iTime-1) + (dt/6)*(l1 + 2*l2 + 2*l3 + l4);
end

figure();
h_grav = plot(t,H,'b.-','linewidth',2,'markersize',10); hold on;
grid on; set(gca,'fontsize',14);
title('Lab 8: What a Drag!','fontsize',20,'fontweight','b');
xlabel('Time (seconds)'); ylabel('Falling Distance (m)');
```

```
% b) Adding drag to the equation (Skydiver falling head-first)

H_headfirst = zeros(size(t));
Hprime_headfirst = 0;  v_headfirst = zeros(size(t));
headfirst_Cd = 0.08;
for iTime = 2:length(t)
    kd1 = (m*g - headfirst_Cd*Hprime_headfirst^2)/m;
    ld1 = Hprime_headfirst;

    % Now that our acceleration is based on velocity we need to update our
    % calculations at each step!
    kd2 = (m*g - headfirst_Cd*(Hprime_headfirst + (dt/2)*kd1)^2)/m;
    ld2 = Hprime_headfirst + (dt/2)*kd1;

    kd3 = (m*g - headfirst_Cd*(Hprime_headfirst + (dt/2)*kd2)^2)/m;
    ld3 = Hprime_headfirst + (dt/2)*kd2;

    kd4 = (m*g - headfirst_Cd*(Hprime_headfirst + (dt)*kd3)^2)/m;
    ld4 = Hprime_headfirst + (dt)*kd3;

    Hprime_headfirst = Hprime_headfirst + (dt/6)*(kd1 + 2*kd2 + 2*kd3 + kd4);
    v_headfirst(iTime) = Hprime_headfirst;
    H_headfirst(iTime) = H_headfirst(iTime-1) + (dt/6)*(ld1 + 2*ld2 + 2*ld3 +
 ld4);
end
h_hf = plot(t,H_headfirst,'r.-','linewidth',2,'markersize',10);

% c) Changing drag to spread out skydiver

H_spread = zeros(size(t));
Hprime_spread = 0;  v_hdrag = zeros(size(t));
spread_Cd = 0.45;
for iTime = 2:length(t)

    khd1 = (m*g - spread_Cd*Hprime_spread^2)/m;
    lhd1 = Hprime_spread;

    khd2 = (m*g - spread_Cd*(Hprime_spread + (dt/2)*khd1)^2)/m;
    lhd2 = Hprime_spread + (dt/2)*khd1;

    khd3 = (m*g - spread_Cd*(Hprime_spread + (dt/2)*khd2)^2)/m;
    lhd3 = Hprime_spread + (dt/2)*khd2;

    khd4 = (m*g - spread_Cd*(Hprime_spread + (dt)*khd3)^2)/m;
    lhd4 = Hprime_spread + (dt)*khd3;

    Hprime_spread = Hprime_spread + (dt/6)*(khd1 + 2*khd2 + 2*khd3 + khd4);
    v_hdrag(iTime) = Hprime_spread;
    H_spread(iTime) = H_spread(iTime-1) + (dt/6)*(lhd1 + 2*lhd2 + 2*lhd3 +
 lhd4);
end
h_sp = plot(t,H_spread,'g.-','linewidth',2,'markersize',10);
legend([h_grav, h_hf, h_sp],{'No Drag','Head-First','Spread
 Out'},'location','NorthWest');
```

```
% d) We see that by adding drag we basically cap out falling speed at some
% terminal velocity. This means the more drag we can create, the slower we
% can fall!

% Create a Plot of Speed vs Time (for fun)
figure()
plot(t,v,'b.-','linewidth',2,'markersize',10); hold on
plot(t,v_headfirst,'r.-','linewidth',2,'markersize',10);
plot(t,v_hdrag,'g.-','linewidth',2,'markersize',10);
grid on; set(gca,'fontsize',14);
title('Lab 8: What a Drag!','fontsize',20,'fontweight','b');
xlabel('Time (seconds)'); ylabel('Falling Speed (m/s)');
legend('No Drag','Head-First','Spread Out','location','NorthWest');
```





Published with MATLAB® R2022b