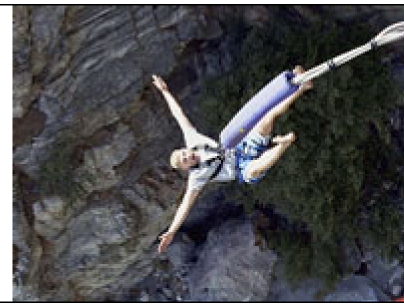
The Math Behind Bungee Jumping

Definition of the Problem

This demonstration shows how to use MATLAB to model a simple physics problem faced by a college student. During spring break, John Smith wants to go bungee jumping. John has to determine which elastic cord is best suited for his weight (90 kg).

Elastic Cord	Spring Constant
A	=5 N/m
B	=40 N/m
C	=500 N/m

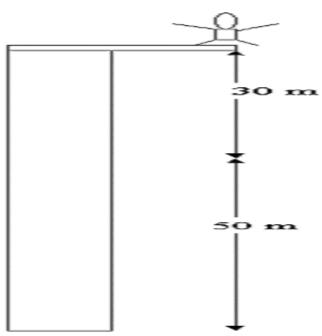


The air resistance that the bungee jumper faces is

$$R = a_1 \times v - a_2 \times |v| \times v$$

where $a_1 = 1$ and $a_2 = 1$

The length of the unstretched cord is 30m. The bungee jumpers is 80m above the ground.



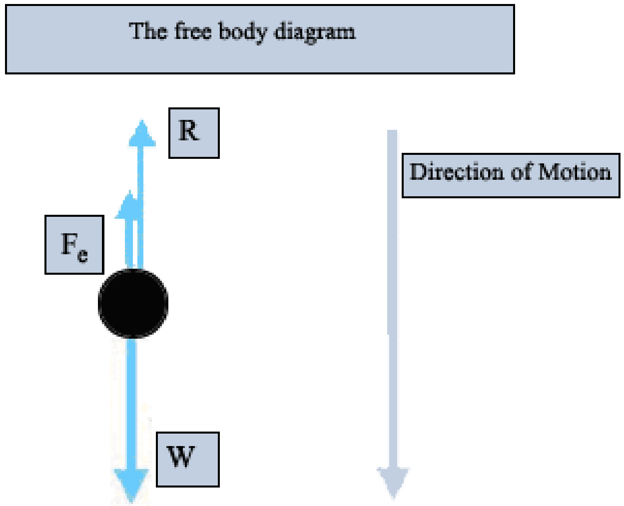
Analysis of the Problem

To solve this physics problem, we need to:

1. Determine the forces acting on the body.

Weight (W):	Air Resistance (R):	Force from the elastic cord (F_e):
$W = m \times a$ m = 90 kg g = 10 m/s^2	$R = a_1 \times v + a_2 \times \|v\| \times v$ $a_1 = 1$ $a_2 = 1$ $v = dx/dt$	$F_e = k \times x$ if $x>0$ 0 if $x<0$

2. Draw the free body diagram. Please note that we have selected **downwards** as the **positive axis**.



3. Apply Newton's second law.

4. $$NetForces = m \times a$$

$$W - R - F_e = m \times a$$

$$mg - F_e - a_1 \times v + a_2 \times |v| \times v = m \times a$$

where $v = dx/dt$ and $a = dv/dt$

$$\frac{d^2x}{dt^2} = g - \frac{F_e}{m} - \frac{a_1}{m} \times \frac{dx}{dt} - \frac{a_2}{m} \times \left|\frac{dx}{dt}\right| \times \frac{dx}{dt}$$

Approach to the Solution

We need to use the ODE solvers. The ODE solvers solve initial value problems for ordinary differential equations (ODEs), in this case ODE45. The syntax for ODE45 is:

[T,Y] = ode45(odefun,tspan,y0,options,p1,p2...)

odefun	A function that evaluates the right-hand side of the differential equations. All solvers solve systems of equations in the form y'= f(t,y)
tspan	A vector specifying the interval of integration, [t0,tf]. To obtain solutions at specific times (all increasing or all decreasing), use tspan = [t0,t1,...,tf].y0A vector of initial conditions.
Options	Optional integration argument created using the odeset function. See odeset for details.
p1,p2..	Optional parameters that the solver passes to odefun and all the functions specified in options

First we need to create the odefun function: We need to rewrite our equation so that it will be in the form of y'=f(t,y)

Assume :

X1= x

X2= dx/dt

Therefore:

X1dot = X2;

X2dot = $g - F_e/m - a_1/m \times X2 - a_2 \times |X2| \times X2$

Now let's express the solution in terms of MATLAB scripting instructions:

K	`function dxdt = bungeeode (t,x,k)`

Constants	`m=90;`
	`g=10;`
	`a1=1;`
	`a2=1;`

Weight	`W=m*g;`
Air Resistance	`R= a1*x(2)+a2*abs(x(2))*x(2);`

Elastic Force	`if x(1)>0`
	`Fe = k*x(1);`
	`else`
	`Fe = 0;`
	`end`

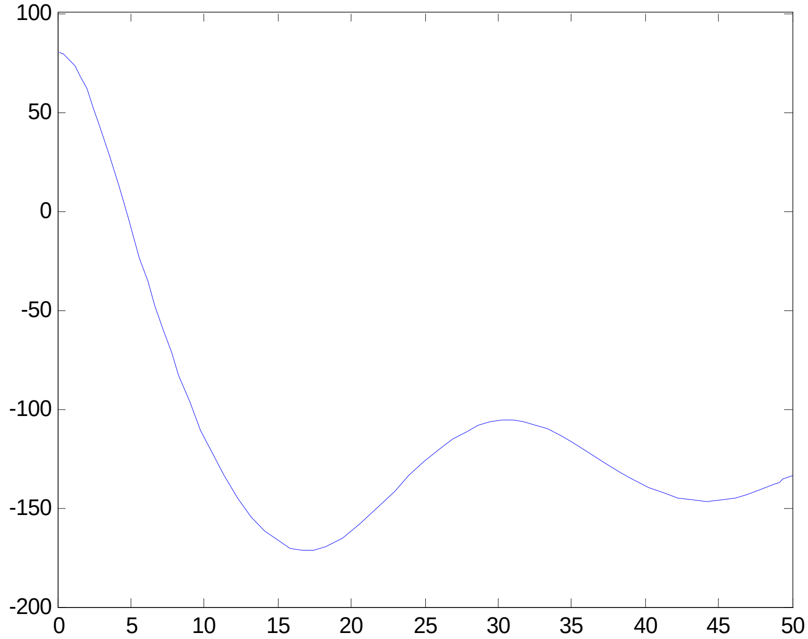
System of ODES	`dxdt= [ x(2) ; (W-Fe-R)/m];`

Execution of the Solution

The time span can be from 0 to 50 s. To use the default options, we assign the option as [ ].

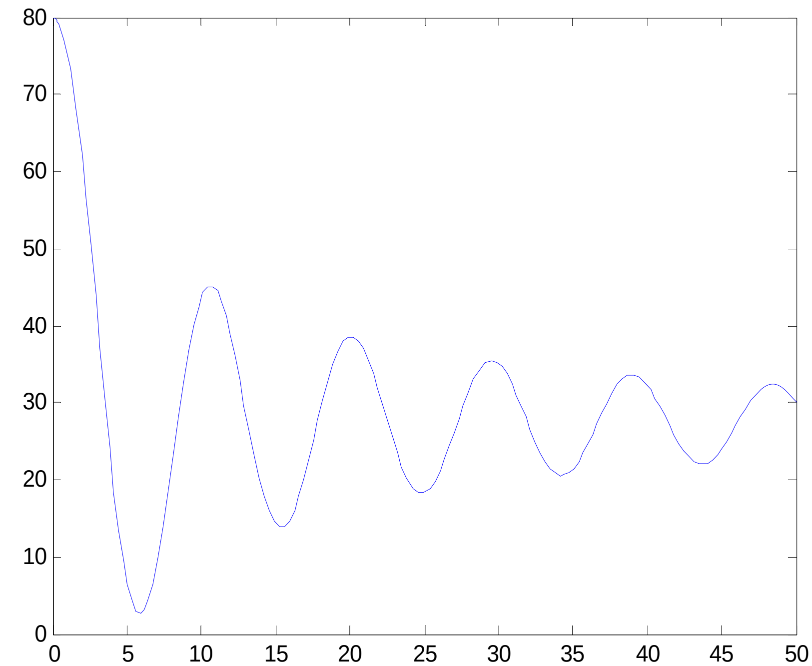
For Elastic Cord A:

```
figure
[t,xsol]=ode45(@bungeeode,[0 50],[-30 0],[],5);
plot(t,50-xsol(:,1))
createInferenceFigure
```



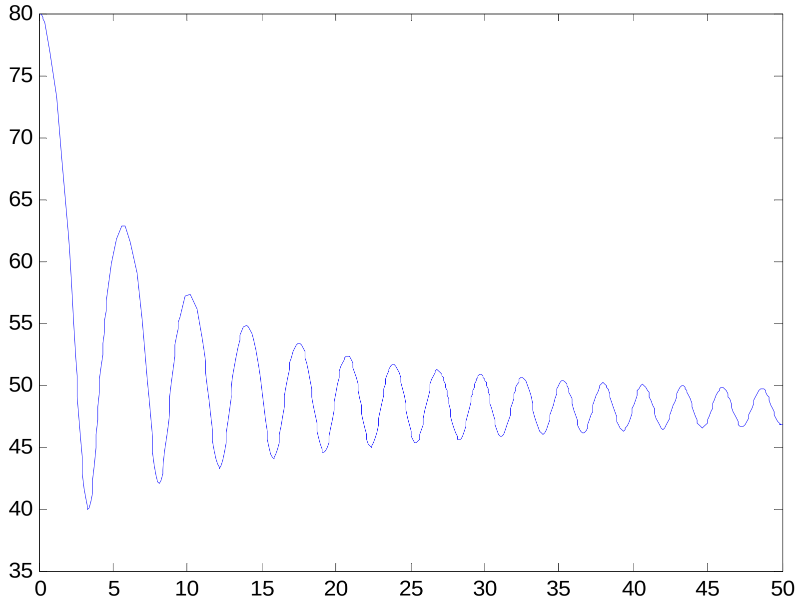
For Elastic Cord B:

```
figure
[t,xsol]=ode45(@bungeeode,[0 50],[-30 0],[],40);
plot(t,50-xsol(:,1))
createInferenceFigure
```



For Elastic Cord C:

```
figure
[t,xsol]=ode45(@bungeeode,[0 50],[-30 0],[],500);
plot(t,50-xsol(:,1))
createInferenceFigure
```



Results and Conclusions

Reviewing the plots of vertical distance versus time reveals the following:

- With elastic cord A, the jumper hits the ground
- With elastic cord B, the jumper gets close to the ground
- With elastic cord C, the jumper never gets close to the ground

Conclusion: **the best choice is ELASTIC CORD B**.