$$F_{air} = -\frac{1}{2} C \rho A v^2$$

$$\frac{dp}{dt} = -\frac{1}{2} C \rho A v^2$$

C: unitless constant to account for shape

A: Cross sectional area (not total surface area)

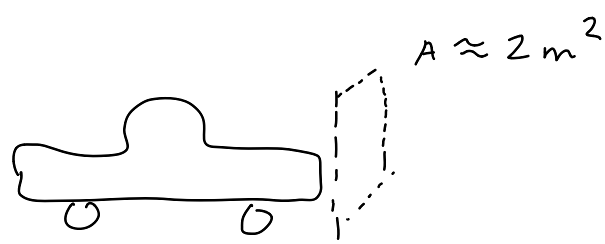
ρ: density $\left(\frac{kg}{m^3}\right)$ of material

Coasting car:

$C = 1$

$A = 2 m^2$

$\rho = 1.2 \frac{kg}{m^3}$



```
import numpy as np
import matplotlib.pyplot as plt

G = 9.8
rho = 1.2
A = 2
C = 1
mass = 1100

def dvdt(v,C,A,rho,mass):
    return -C * A * rho * np.abs(v) * v /mass

vi = 10
yi = 0
dt = 0.01
ti = 0
tf = 100
t = np.arange(ti,tf,dt)
v = np.zeros_like(t)
y = np.zeros_like(t)
v[0] = vi
y[0] = yi
for i in range(1,t.size):
    vprime = dvdt(v[i-1],C,A,rho,mass)
    v[i] = v[i-1] + vprime * dt
    y[i] = y[i-1] + v[i] * dt
```

How do we choose Δt?

Our approximation is valid if the derivative does not change much over Δt

We are approximating $\int v'(t)\,dt$ to be a rectangle

Approximation is perfect if $\frac{dv}{dt}$ is a flat line (if $\frac{d}{dt}\frac{dv}{dt}$ is 0)

The error in our approx is a triangle w/ height $\frac{d^2}{dt^2}v \cdot \Delta t^2$ + base Δt

$$\text{Error} \approx \frac{1}{2} v''(t)\,\Delta t^2$$

$$v''(t) \approx \frac{v'(t+\Delta t) - v'(t)}{\Delta t}$$

Δt should be small enough that $\frac{dv}{dt}$ doesn't change much over Δt

if $\frac{dv}{dt} = -g$, this is always true!

if $\frac{dv}{dt} = -\frac{1}{2m} C A \rho |v| \cdot v$

How much time for $\frac{dv}{dt}$ to change significantly?

To get an underestimate, assume that

$$\frac{dv}{dt} = -\frac{1}{2m} C A \rho v_i^2 \quad \text{is constant}$$

Then $v = v_i - \frac{1}{2m} C A \rho v_i^2 t$

$v = 0$ after $t = \frac{v_i}{\frac{1}{2m} C A \rho v_i^2}$

So it takes at least $t = \frac{v_i}{v'(t_i)}$

for v to go to 0

So over intervals $\Delta t << \frac{v_i}{v'(t_i)}$

$v'(t_i)$ will be approximately flat

$$\Delta t \approx 0.01 \times \frac{v_i}{v'(t_i)}$$

```
import numpy as np
import matplotlib.pyplot as plt

G = 9.8
rho = 1.2
A = 2
C = 1
mass = 75

def dvdt(v,C,A,rho,mass):
    return -C * A * rho * v * np.abs(v) / mass  - G

vi = 0
yi = 10
dt = 0.01
ti = 0
tf = 5
t = np.arange(ti,tf,dt)
v = np.zeros_like(t)
y = np.zeros_like(t)
v[0] = vi
y[0] = yi
for i in range(1,t.size):
    vprime = dvdt(v[i-1],C,A,rho,mass)
    v[i] = v[i-1] + vprime * dt
    y[i] = y[i-1] + v[i] * dt
```

2D:

$$\frac{dV_x}{dt} = 0$$

$$\frac{dV_y}{dt} = -g$$

$$V_{x,n+1} = V_{x,n} + \frac{dV_{x,n}}{dt} \Delta t$$

$$V_{y,n+1} = V_{y,n} + \frac{dV_{y,n}}{dt} \Delta t$$

$$X_{n+1} = X_n + V_{x,n+1} \Delta t$$

$$y_{n+1} = y_n + V_{y,n+1} \Delta t$$

```
import numpy as np
import matplotlib.pyplot as plt

G = 9.8

def dvxdt():
    return 0
def dvydt():
    return -G

vi = 45
theta = 30
yi = 10
dt = 0.01
ti = 0
tf = 5
t = np.arange(ti,tf,dt)
vx = np.zeros_like(t)
vy = np.zeros_like(t)
x = np.zeros_like(t)
y = np.zeros_like(t)
vx[0] = vi * np.cos( np.deg2rad(theta) )
vy[0] = vi * np.sin( np.deg2rad(theta) )
y[0] = yi
for i in range(1,t.size):
    vxprime = dvxdt()
    vyprime = dvydt()
    vx[i] = vx[i-1] + vxprime * dt
    vy[i] = vy[i-1] + vyprime * dt
    x[i] = x[i-1] + vx[i] * dt
    y[i] = y[i-1] + vy[i] * dt
```

In 2D

$$\vec{F} = -\frac{1}{2} C A \rho v^2 \hat{v}$$

$$\hat{v} = \frac{1}{v} \langle v_x , v_y \rangle$$

$$F_x = -\frac{1}{2} C A \rho v^2 \frac{v_x}{v}$$

$$F_x = -\frac{1}{2} C A \rho v v_x$$

$$F_y = -\frac{1}{2} C A \rho v v_y$$

$$F_{x,net} = -\frac{1}{2} C A \rho v v_x$$

$$F_{y,net} = -mg - \frac{1}{2} C A \rho v v_y$$

$$\frac{dv_x}{dt} = -\frac{1}{2m} C A \rho v v_x$$

$$\frac{dv_y}{dt} = -g - \frac{1}{2m} C A \rho v v_x$$

$$t_x \approx \frac{V_{xi}}{\frac{1}{2m} C A \rho V^2}$$

$$t_y \approx \frac{V_{yi}}{\frac{1}{2m} C A \rho V^2 + g}$$

$$t_0 = \min(t_x, t_y)$$

$$\Delta t \approx \frac{t_0}{100}$$