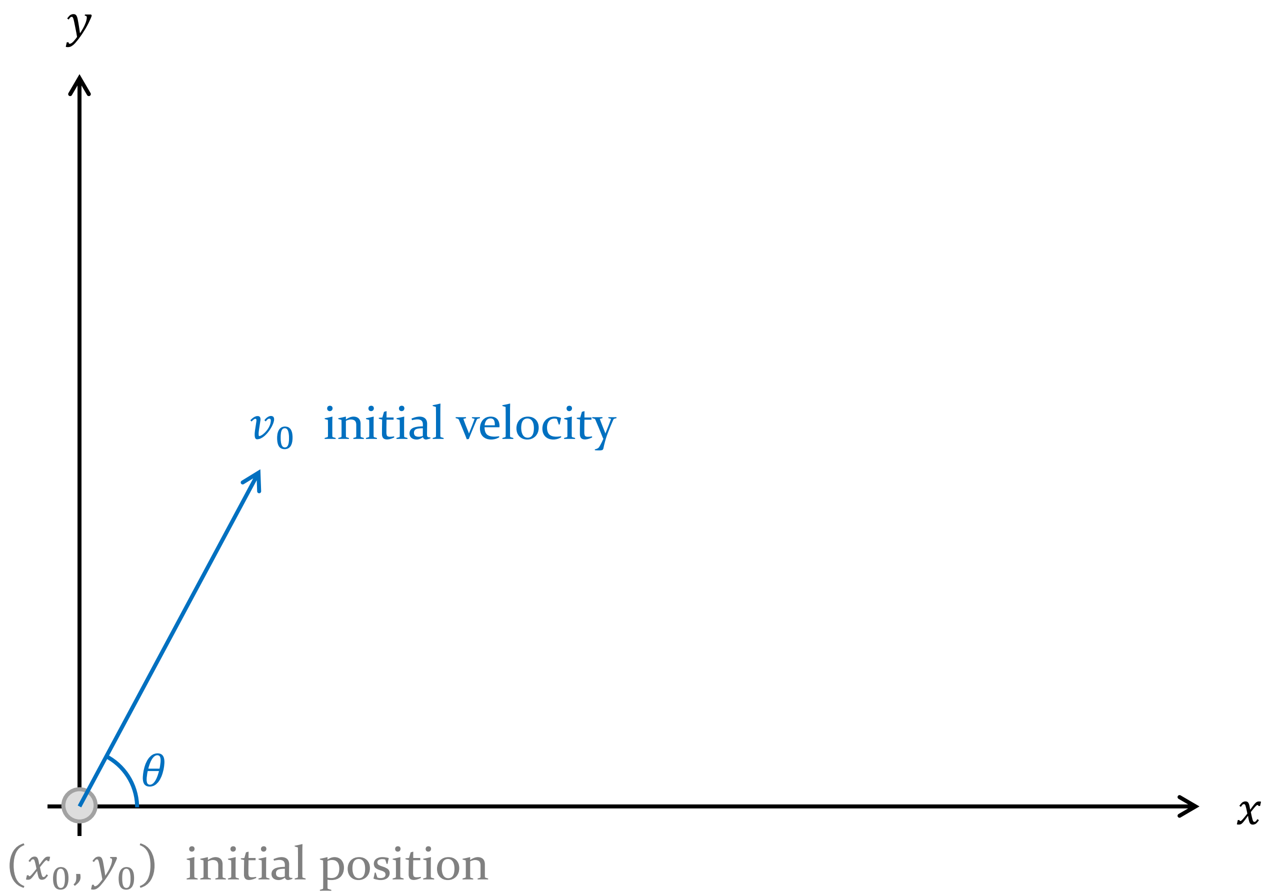
Ordinary differential equations

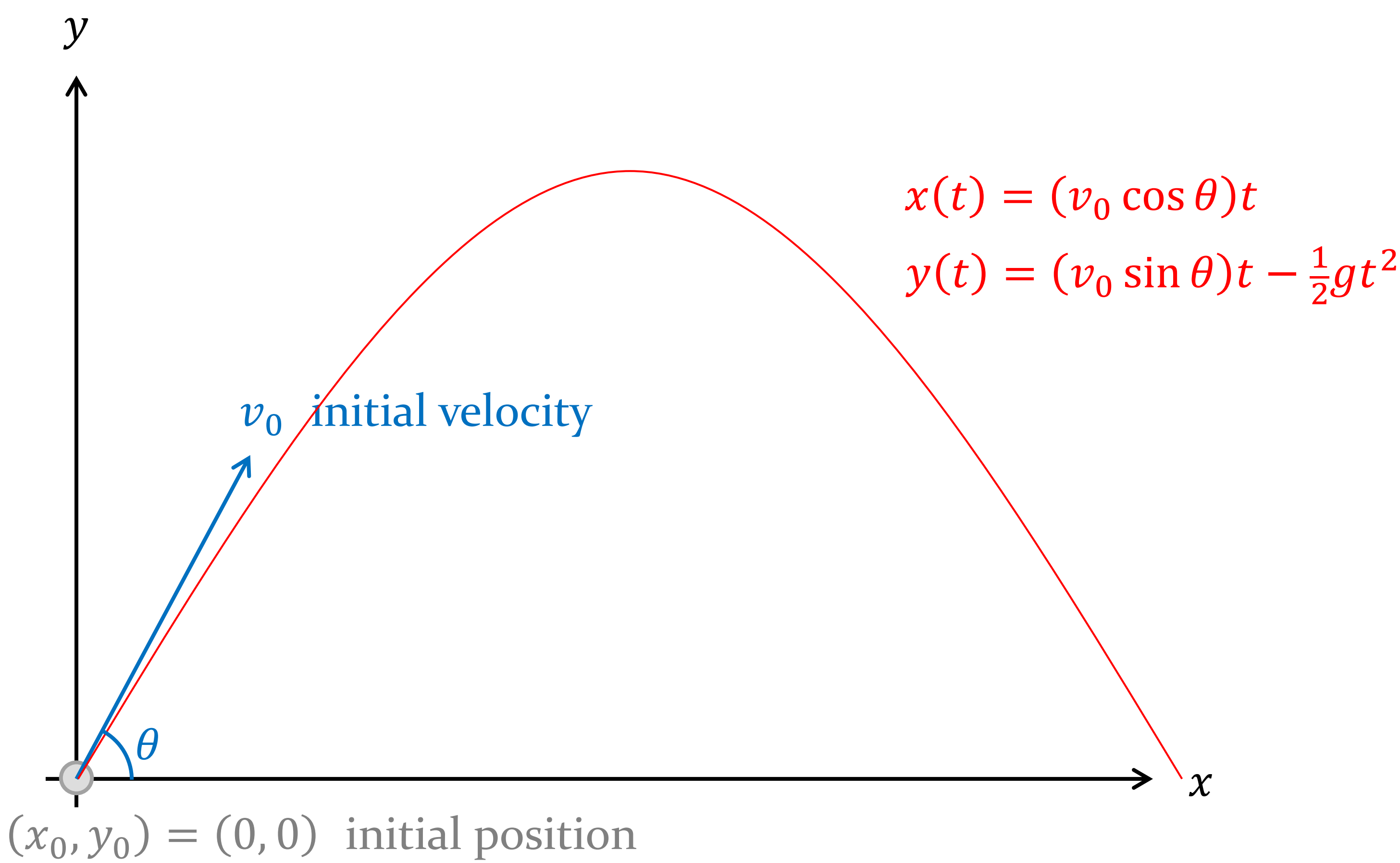
Example problem

- consider the no-drag projectile motion problem



Example problem

- the solution is known to all physics students



Example problem

- if you did not know the solution, could you find an approximate solution using computation?
- we know that velocity is the first derivative of position with respect to time
  - it is tangent to the position curve
  - if we use the tangent to approximate the actual position curve at time $t_0 = 0$ we can estimate the position at time $t_1 = t_0 + \Delta t$ for some small value of Δt

$$x_1 = x_0 + (v_0 \cos\theta)\Delta t = x_0 + v_{x,0}\Delta t$$

$$y_1 = y_0 + (v_0 \sin\theta)\Delta t = y_0 + v_{y,0}\Delta t$$

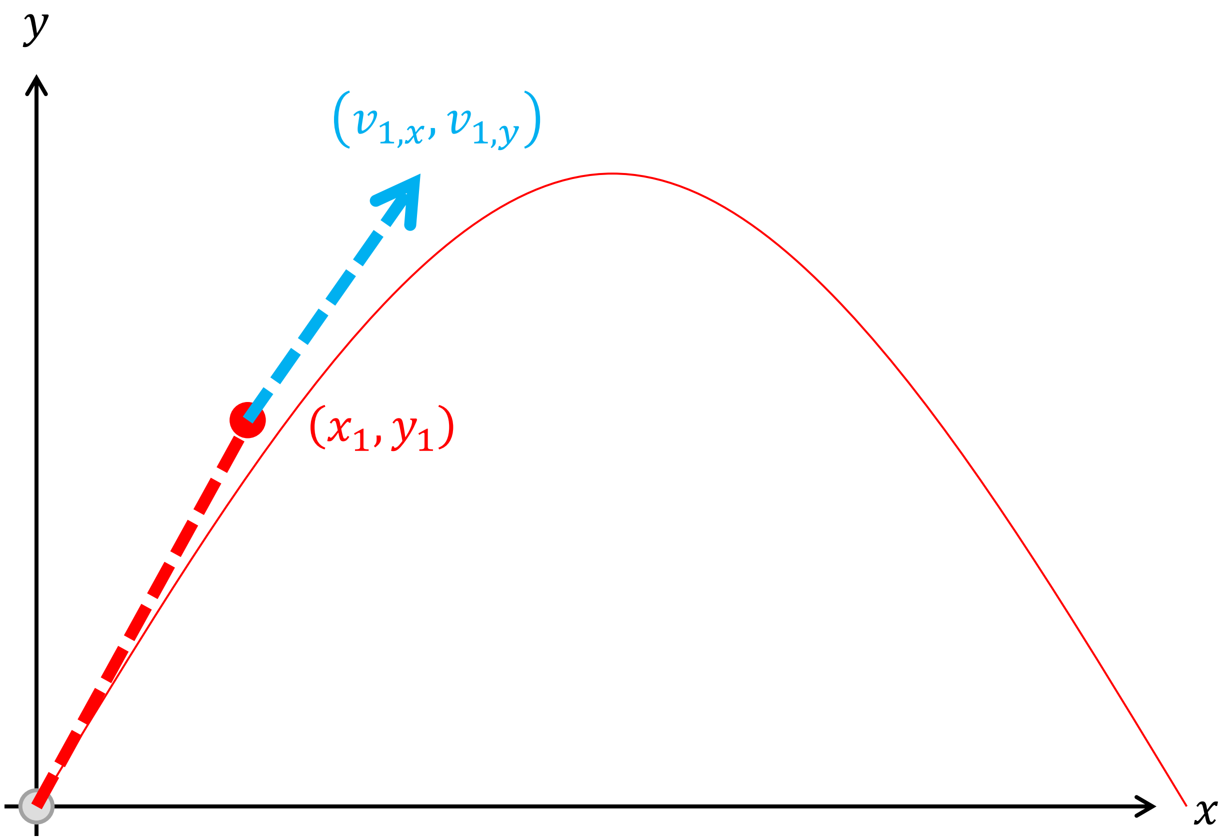
Example problem

- the projectile experiences a force downwards equal to its mass times the constant acceleration of gravity
- acceleration is the first derivative of velocity with respect to time
  - it is tangent to the velocity curve
  - if we use the tangent to approximate the actual velocity curve at time $t_0 = 0$ we can estimate the velocity at time $t_1 = t_0 + \Delta t$ for some small value of Δt

$$v_{x,1} = v_{x,0}$$

$$v_{y,1} = v_{y,0} - g\Delta t$$

Example problem



Example problem

- if we assume that (x_1, y_1) is close to the actual position curve, we can repeat our reasoning to compute a new position (x_2, y_2) and new velocity $\left(v_{x,2}, v_{y,2}\right)$
  - i.e., use the previously estimated position and velocity to estimate the new position and velocity

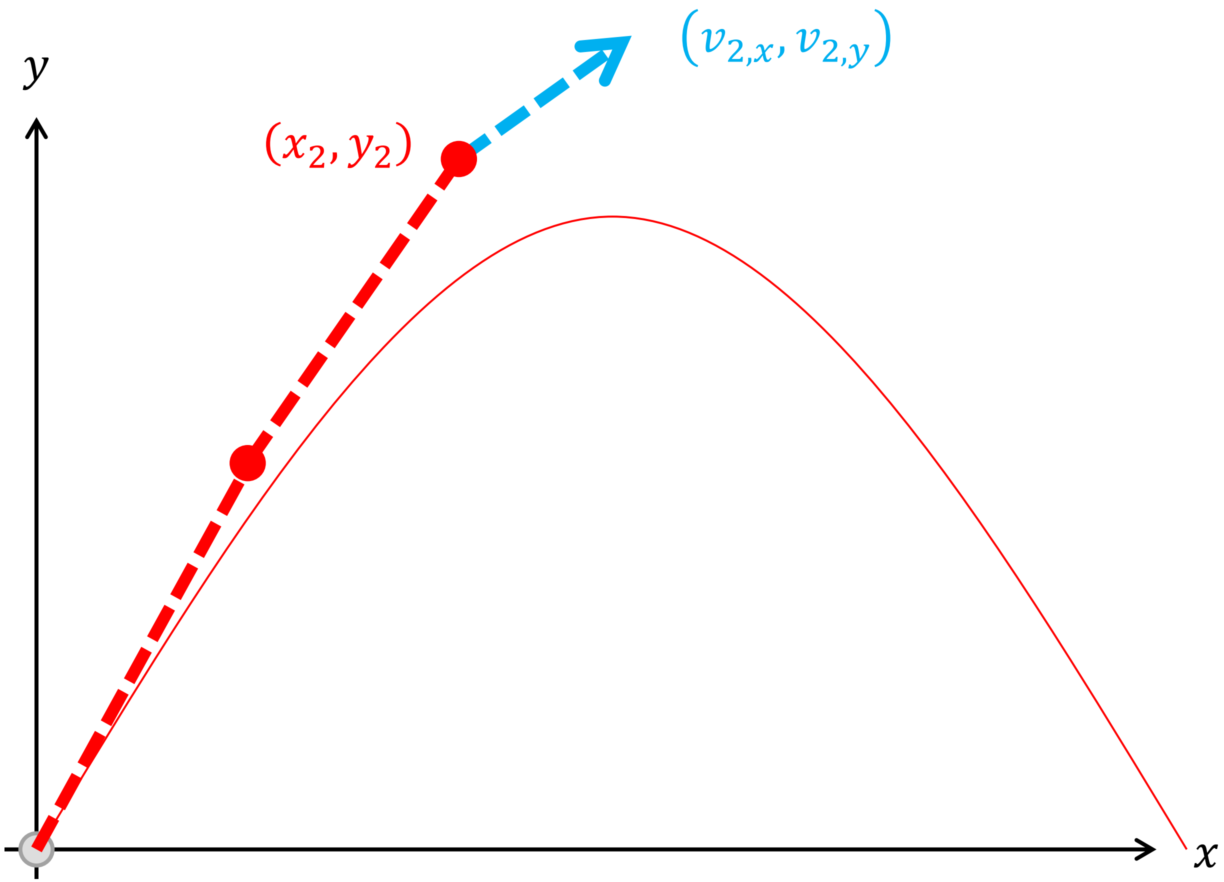
$$x_2 = x_1 + v_{x,1}\Delta t$$

$$y_2 = y_1 + v_{y,1}\Delta t$$

$$v_{x,2} = v_{x,1}$$

$$v_{y,2} = v_{y,1} - g\Delta t$$

Example problem



Example problem

- repeating the process yields the position (x_i, y_i) and velocity $\left(v_{x,i}, v_{y,i}\right)$ at time $t = t_i = t_0 + i\Delta t$

$$x_i = x_{i-1} + v_{x,i-1}\Delta t$$

$$y_i = y_{i-1} + v_{y,i-1}\Delta t$$

$$v_{x,i} = v_{x,i-1}$$

$$v_{y,i} = v_{y,i-1} - g\Delta t$$

- we can implement the above equations in a function

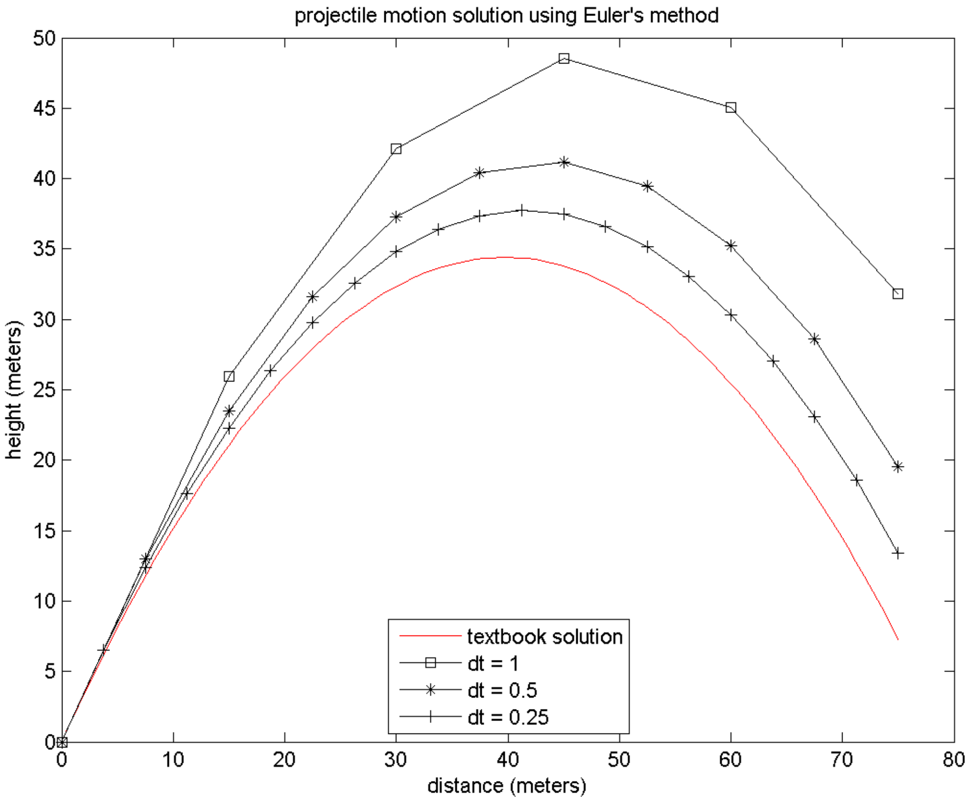
```matlab
function [x, y, vx, vy, t] = solveproj(tf, dt, x0, y0, vx0, vy0)
%SOLVEPROJ Numerical solution for projectile motion
%   [X, Y, VX, VY, T] = SOLVEPROJ(TF, DT, X0, Y0, VX0, VY0)
%   solves the projectile motion problem for a projectile
%   having initial location (X0, Y0) moving with initial
%   velocity (VX0, VY0) over the time period [0, TF] in
%   time steps of DT.
%
%   The position (X, Y) and velocity (VX, VY) of the
%   projectile is evaluated at times T.

t = 0:dt:tf;
% make sure tf is at the end of t
if t(end) ~= tf
    t = [t tf];
end
n = length(t);
```

continued on next slide

```
% estimated position and velocity
x = zeros(size(t));
y = zeros(size(t));
vx = zeros(size(t));
vy = zeros(size(t));
x(1) = x0;
y(1) = y0;
vx(1) = vx0;
vy(1) = vy0;
for i = 2:n
    x(i) = x(i - 1) + vx(i - 1) * dt;
    y(i) = y(i - 1) + vy(i - 1) * dt;
    vx(i) = vx(i - 1);
    vy(i) = vy(i - 1) - 9.81 * dt;
end
```

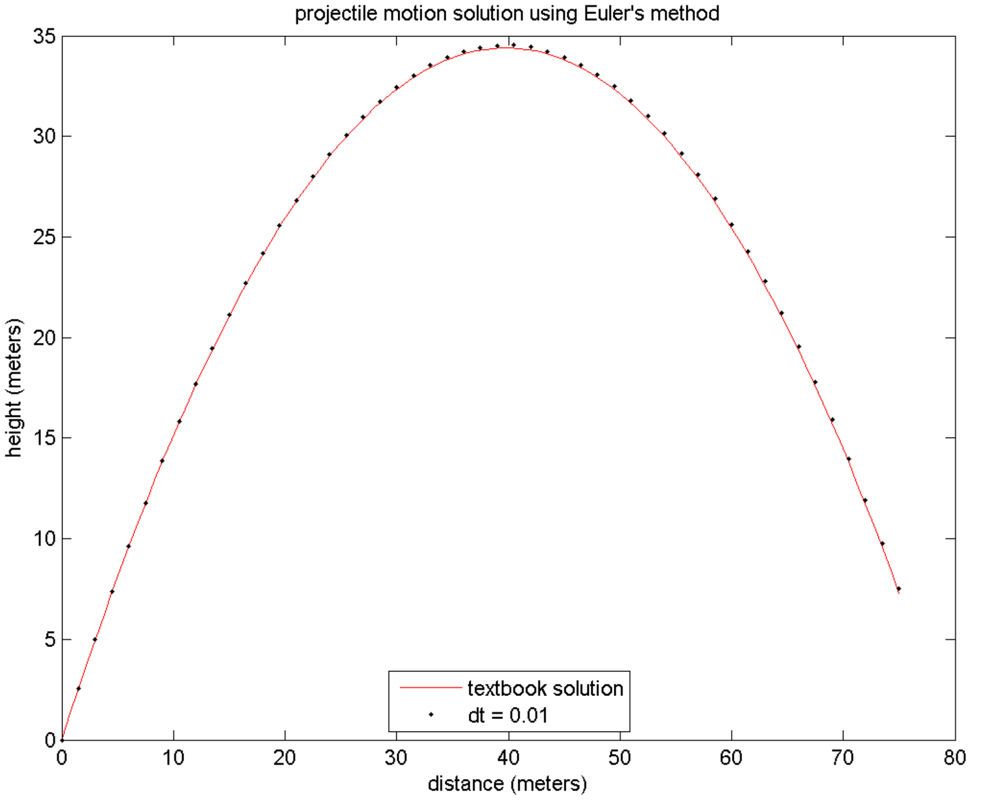
Example problem

- see **`day23.m`**



Example problem

- making Δt small enough yields a solution that closely matches the textbook solution



Example problem

- our numerical solution is actually performing numerical integration using a variation of the rectangle rule

$$x_i = x_{i-1} + v_{x,i-1}\Delta t \approx x_{i-1} + \int_{t_{i-1}}^{t_i} v_{x,i-1}\, dt$$

$$y_i = y_{i-1} + v_{y,i-1}\Delta t \approx y_{i-1} + \int_{t_{i-1}}^{t_i} v_{y,i-1}\, dt$$

$$v_{x,i} = v_{x,i-1} \approx v_{x,i-1} + \int_{t_{i-1}}^{t_i} 0\, dt$$

$$v_{y,i} = v_{y,i-1} - g\Delta t \approx v_{y,i-1} - \int_{t_{i-1}}^{t_i} g\, dt$$

Example problem

- another way to view the problem is that we are trying to solve the following set of equations

$$\frac{dx}{dt} = v_x$$

$$\frac{dy}{dt} = v_y$$

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

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

$$x_0 = 0$$

$$y_0 = 0$$

$$v_{x,0} = v_0 \cos\theta$$

$$v_{y,0} = v_0 \sin\theta$$

initial conditions

Example problem

- replacing the derivatives with a finite forward difference yields

$$\frac{dx}{dt} \approx \frac{x_i - x_{i-1}}{\Delta t} = v_x \implies x_i = x_{i-1} + v_x \Delta t$$

$$\frac{dy}{dt} \approx \frac{y_i - y_{i-1}}{\Delta t} = v_y \implies y_i = y_{i-1} + v_y \Delta t$$

$$\frac{dv_x}{dt} \approx \frac{v_{x,i} - v_{x,i-1}}{\Delta t} = 0 \implies v_{x,i} = v_{x,i-1}$$

$$\frac{dv_y}{dt} \approx \frac{v_{y,i} - v_{y,i-1}}{\Delta t} = -g \implies v_{y,i} = v_{y,i-1} - g\Delta t$$

Ordinary differential equations

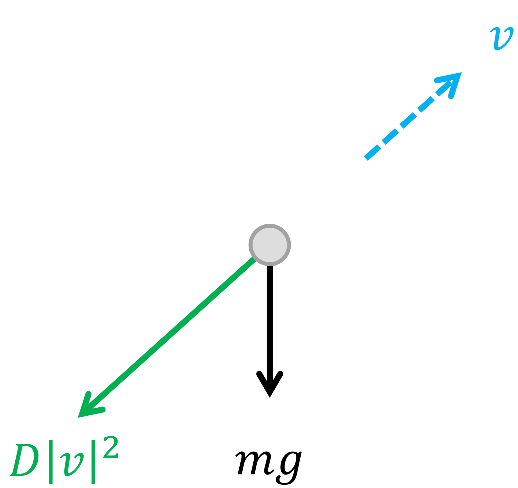
- equations written in terms of ordinary (not partial) derivatives are called *ordinary differential equations*

$$\frac{dx}{dt} = v_x \qquad \frac{dy}{dt} = v_y \qquad \frac{dv_x}{dt} = 0 \qquad \frac{dv_y}{dt} = -g$$

- solving an ordinary differential equation by replacing the derivative with a forward finite difference is called the *Euler method*

Projectile motion with drag

- one model of air resistance (drag) for ball-like objects states that the magnitude f of the air drag force is approximately proportional to the square of the projectile speed, and the direction of the air drag force is opposite to the instantaneous velocity of the projectile

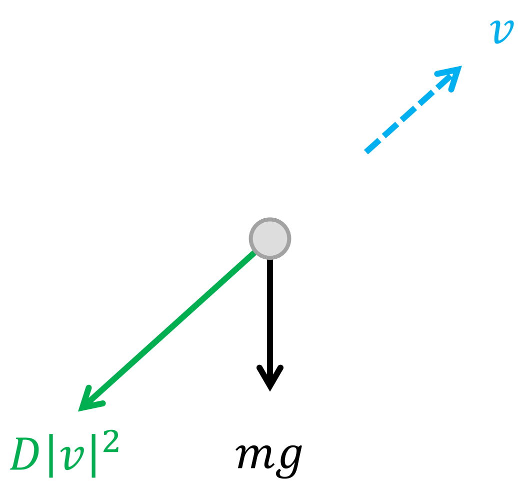


$$\vec{F}_{\text{drag}} = -D|v|\vec{v}$$

$$F_{x,\text{drag}} = -D|v|v_x$$

$$F_{y,\text{drag}} = -D|v|v_y$$

Projectile motion with drag



$$\vec{F}_{\text{drag}} = -D|v|\vec{v}$$

$$F_{x,\text{drag}} = -D|v|v_x$$

$$F_{y,\text{drag}} = -D|v|v_y$$

$$\sum F_x = ma_x = -D|v|v_x \implies a_x = \frac{-D|v|v_x}{m}$$

$$\sum F_y = ma_y = -D|v|v_y - mg \implies a_y = \frac{-D|v|v_y}{m} - g$$

Projectile motion with drag

- our system of differential equations now becomes

$$\frac{dx}{dt} = v_x$$

$$\frac{dy}{dt} = v_y$$

$$\frac{dv_x}{dt} = -\frac{D|v|v_x}{m}$$

$$\frac{dv_y}{dt} = -\frac{D|v|v_y}{m} - g$$

Projectile motion with drag

- replacing the derivatives with a finite forward difference yields

$$\frac{dx}{dt} \approx \frac{x_i - x_{i-1}}{\Delta t} = v_x \quad \Longrightarrow \quad x_i = x_{i-1} + v_x \Delta t$$

$$\frac{dy}{dt} \approx \frac{y_i - y_{i-1}}{\Delta t} = v_y \quad \Longrightarrow \quad y_i = y_{i-1} + v_y \Delta t$$

$$\frac{dv_x}{dt} \approx \frac{v_{x,i} - v_{x,i-1}}{\Delta t} = -\frac{D|v|v_x}{m} \Longrightarrow \quad v_{x,i} = v_{x,i-1} - \frac{D|v_{i-1}|v_{x,i-1}}{m}\Delta t$$

$$\frac{dv_y}{dt} \approx \frac{v_{y,i} - v_{y,i-1}}{\Delta t} = -\frac{D|v|v_y}{m} - g \quad \Longrightarrow \quad v_{y,i} = v_{y,i-1} - \frac{D|v_{i-1}|v_{y,i-1}}{m}\Delta t - g\Delta t$$

Projectile motion with drag

- for a round projectile of radius 5cm an approximate value of D is $D = 0.002$
- the following example uses mass $m = 0.2\text{kg}$

```matlab
function [x, y, vx, vy, t] = solveproj2(tf, dt, x0, y0, vx0, vy0)
%SOLVEPROJ2 Numerical solution for projectile motion with drag
%   [X, Y, VX, VY, T] = SOLVEPROJ2(TF, DT, X0, Y0, VX0, VY0)
%   solves the projectile motion problem with drag for a projectile
%   having initial location (X0, Y0) moving with initial
%   velocity (VX0, VY0) over the time period [0, TF] in
%   time steps of DT.
%
%   The position (X, Y) and velocity (VX, VY) of the
%   projectile is evaluated at times T.

t = 0:dt:tf;
% make sure tf is at the end of t
if t(end) ~= tf
    t = [t tf];
end
n = length(t);
```

```
% estimated position and velocity
x = zeros(size(t));
y = zeros(size(t));
vx = zeros(size(t));
vy = zeros(size(t));
x(1) = x0;
y(1) = y0;
vx(1) = vx0;
vy(1) = vy0;
D = 0.002;
m = 0.2;
for i = 2:n
    % velocity magnitude from t(i - 1)
    vmag = norm([vx(i - 1) vy(i - 1)]);
    x(i) = x(i - 1) + vx(i - 1) * dt;
    y(i) = y(i - 1) + vy(i - 1) * dt;
    vx(i) = vx(i - 1) - D * vmag * vx(i - 1) * dt / m;
    vy(i) = vy(i - 1) - D * vmag * vy(i - 1) * dt / m - 9.81 * dt;
end
```

Projectile motion with drag

- see **`day23.m`**