Computer Animation Algorithms and Techniques

Integration

Integration

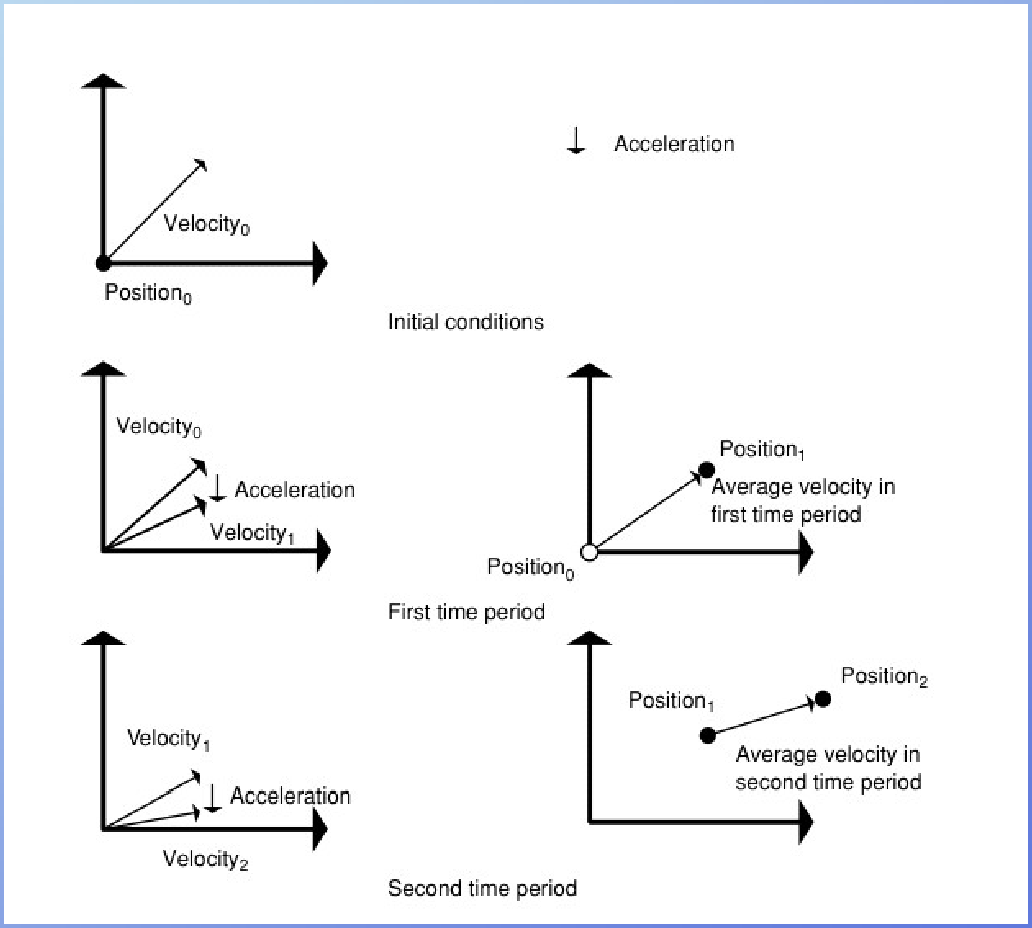
Given acceleration, compute velocity & position by integrating over time

$$f = ma$$

$$a = f / m$$

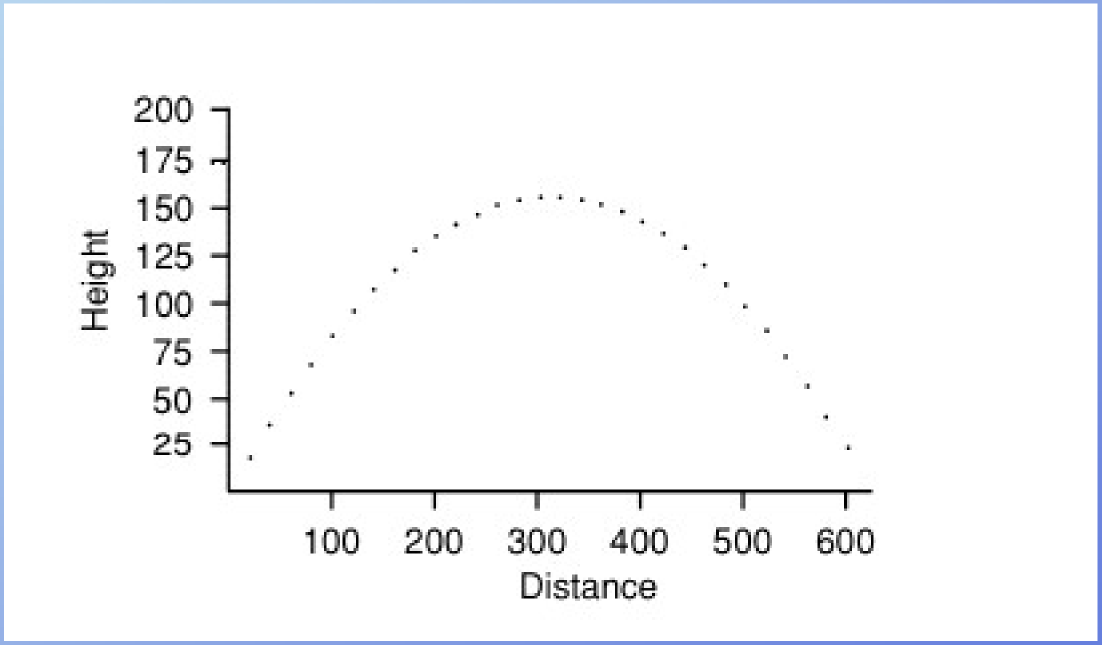
$$v' = v + a \cdot t$$

$$p' = p + vt + \frac{1}{2}at^2$$



Projectile

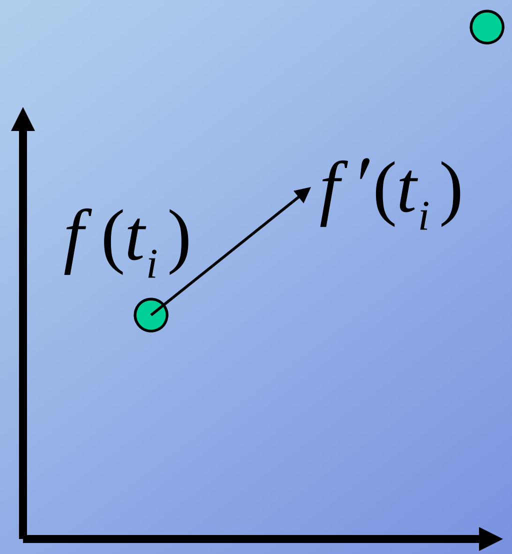
given initial velocity under gravity



Euler integration

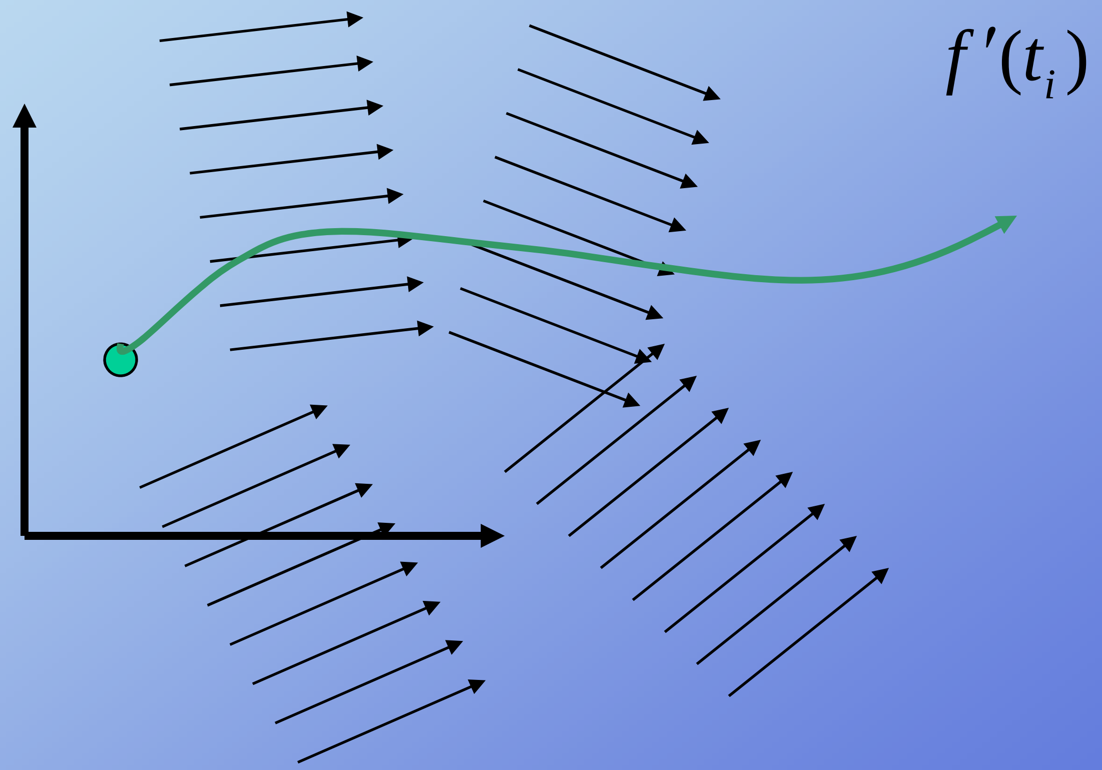
For arbitrary function, f(t)

$$f(t_{i+1}) = f(t_i) + f'(t_i) \cdot \Delta t$$



Integration – derivative field

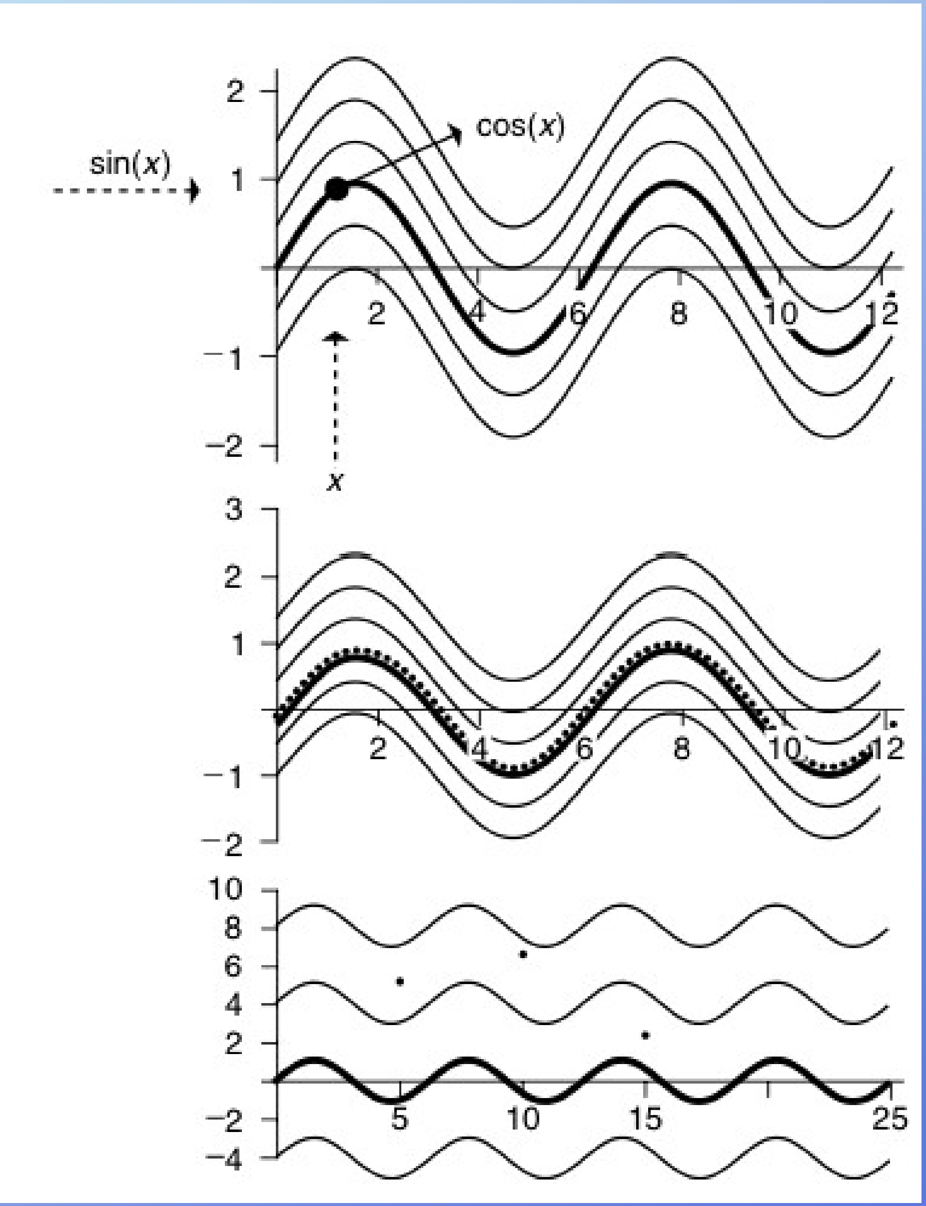
For arbitrary function, f(t)



Step size

$$\Delta x = 0.2$$

$$\Delta x = 5$$

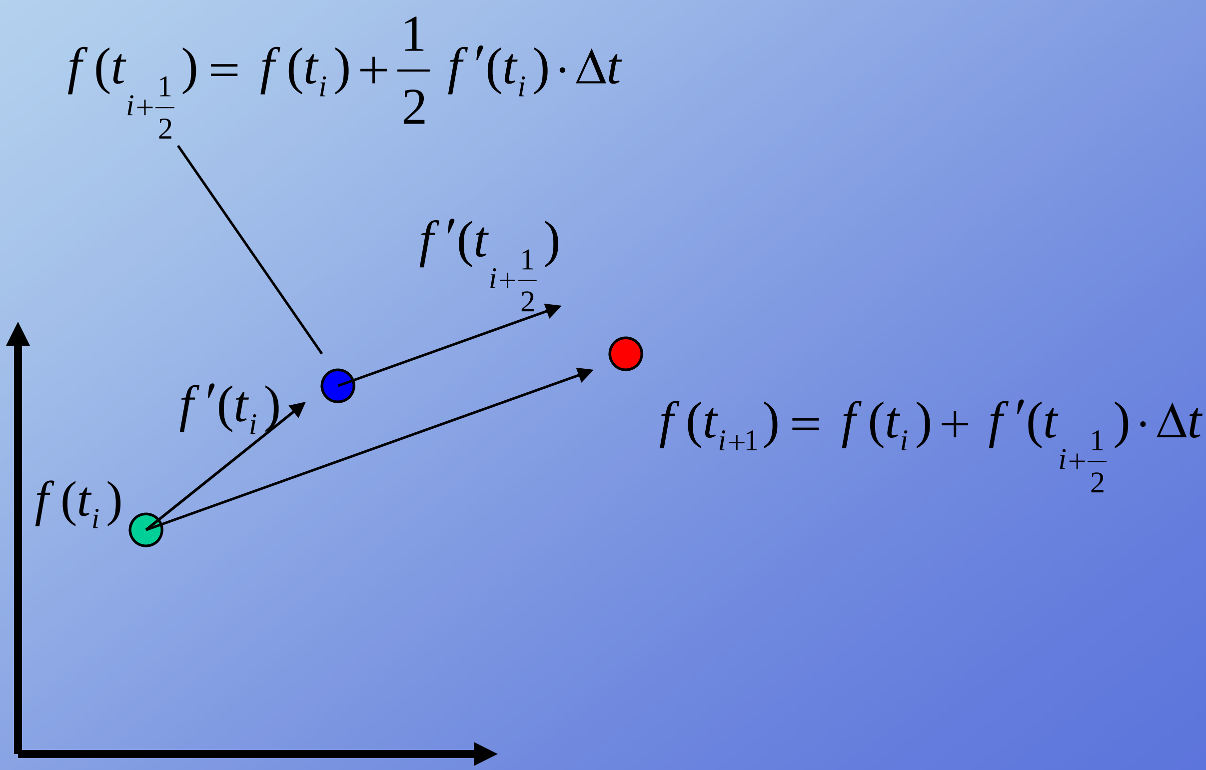


Numeric Integration Methods

(explicit or forward) Euler Integration
2nd order Runga Kutta Integration (Midpoint Method)
4th order Runga Kutta Integration
Implicit (backward) Euler Integration
Semi-implicit Euler Integration

Runge Kutta Integration: 2nd order Aka Midpoint Method

For unknown function, f(t); known f ’(t)

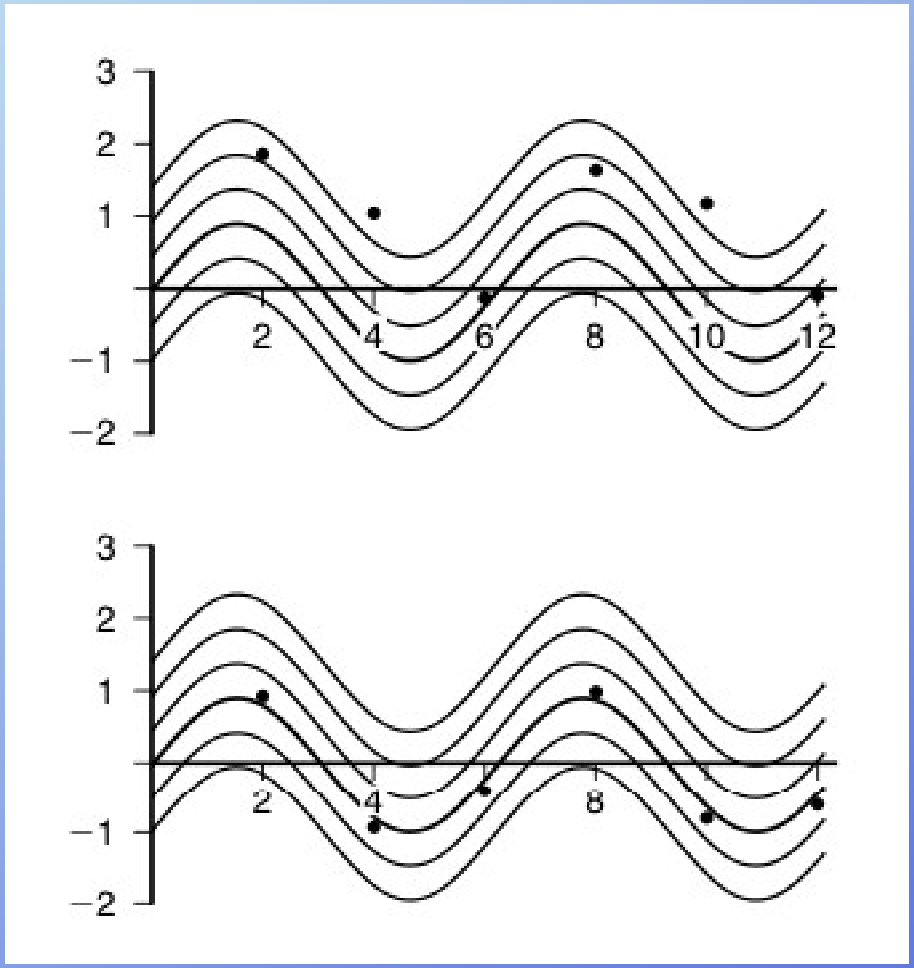


Step size

$$\Delta x = 2$$

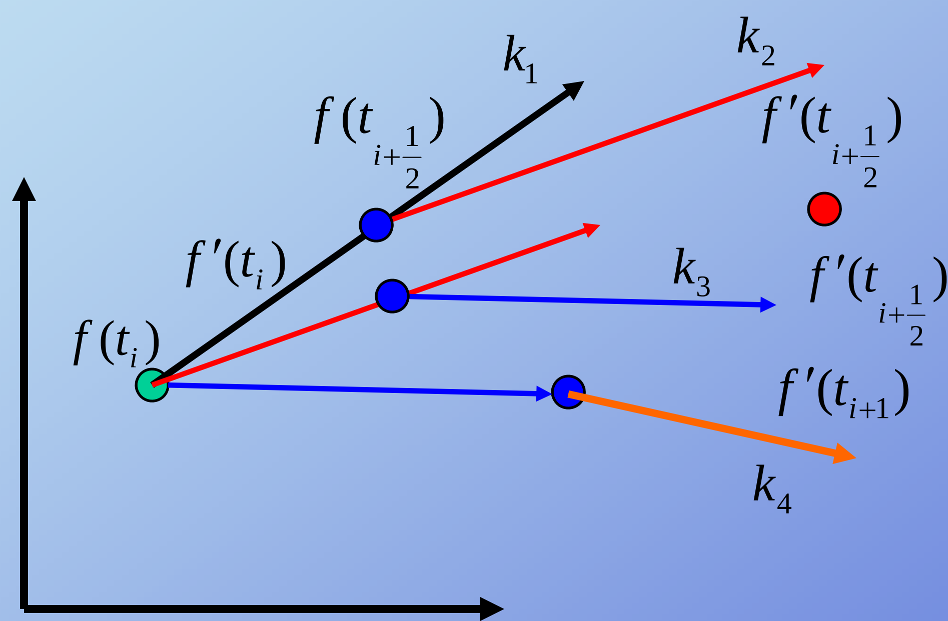
Euler Integration

Midpoint Method



Runge Kutta Integration: 4th order

For unknown function, f(t); known f ’(t)

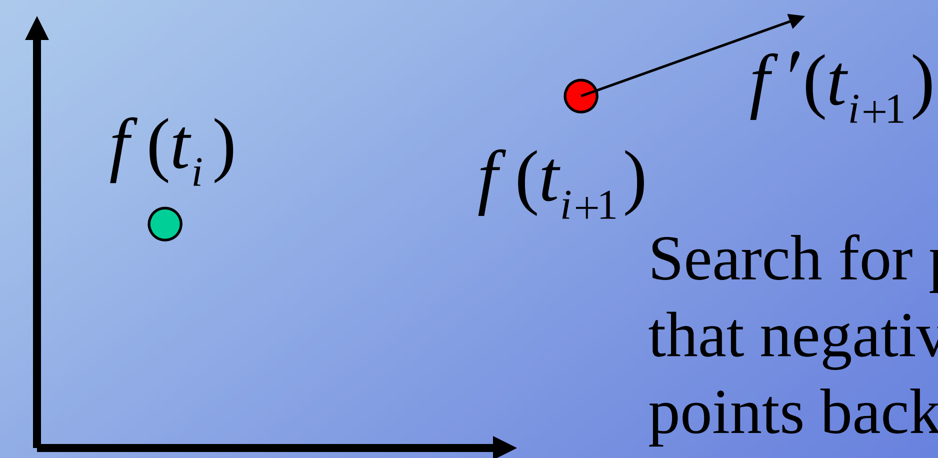


$$f(t_{i+1}) = f(t_i) + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$$

Implicit Euler Integration

For arbitrary function, f(t), find next point whose derivative updates last value to this value: required numeric method (e.g. Newton-Raphson)

$$y_{n+1} = y_n + hf'(t_{i+1})$$



Search for point such that negative of tangent points back at original point

Differential equation, initial boundary problem

$$f(t_0) = y_0$$

(explicit/forward)
Euler method

$$f'(t, y_t) \approx \frac{y_{t+1} - y_t}{h}$$

$$y_{t+1} = y_t + hf'(t, y_t)$$

(implicit/backward)
Euler method

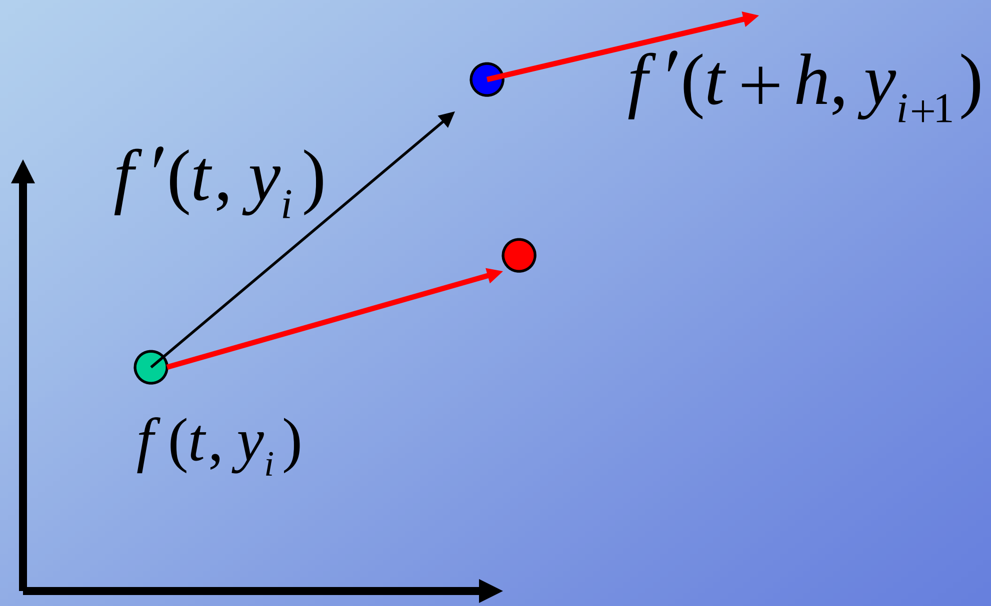
$$f'(t, y_t) \approx \frac{y_t - y_{t-1}}{h}$$

$$y_{t+1} = y_t + hf'(t+h, y_{t+1})$$

e.g. linearize f’ and use Newton-Raphson

Semi-Implicit Euler Integration

$$y_{i+1} = y_i + hf'(t+h, y_i + hf'(t, y_i))$$



Methods specific to update position from acceleration

Heun Method
Verlet Method
Leapfrog Method

Heun Method

$$v(t_{i+1}) = v(t_i) + a(t_i)\Delta t$$

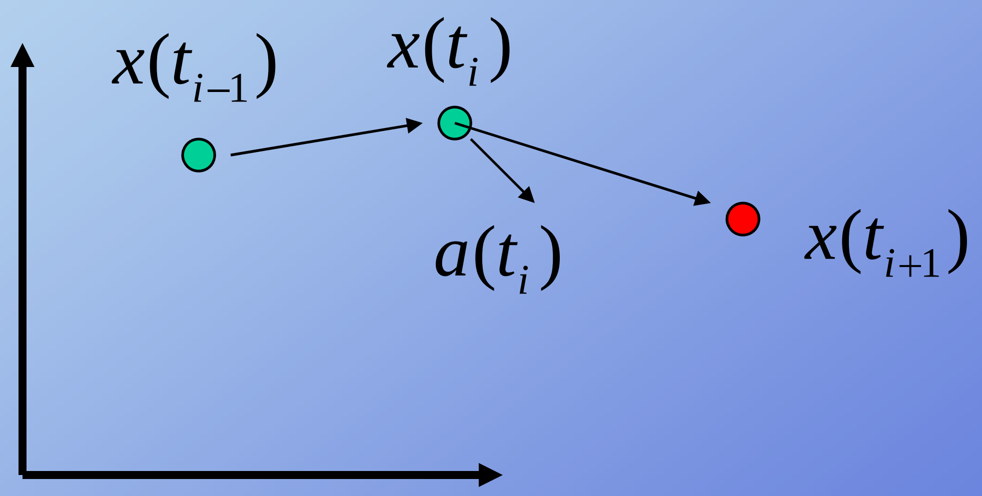
$$x(t_{i+1}) = x(t_i) + \frac{1}{2}(v(t_i) + v(t_{i+1}))\Delta t$$

$$x(t_{i+1}) = x(t_i) + v(t_i)\Delta t + \frac{1}{2}a(t_i)\Delta t^2$$

Verlet Method

$$x(t_{i+1}) = 2x(t_i) - x(t_{i-1}) + a(t_i)\Delta t^2$$

$$x(t_{i+1}) = x(t_i) + (x(t_i) - x(t_{i-1})) + a(t_i)\Delta t^2$$



Leapfrog Method

$$v(t_{i+\frac{1}{2}}) = v(t_{i-\frac{1}{2}}) + a(t_{i-1})\Delta t$$

$$x(t_{i+1}) = x(t_i) + v(t_{i+\frac{1}{2}})\Delta t$$

$$v(t_{i+\frac{3}{2}}) = v(t_{i+\frac{1}{2}}) + a(t_i)\Delta t$$