Game Physics Tutorial 2

Integration

Daniel Hook, Robert Brand
(daniel.hook@tum.de, robert.brand@tum.de)

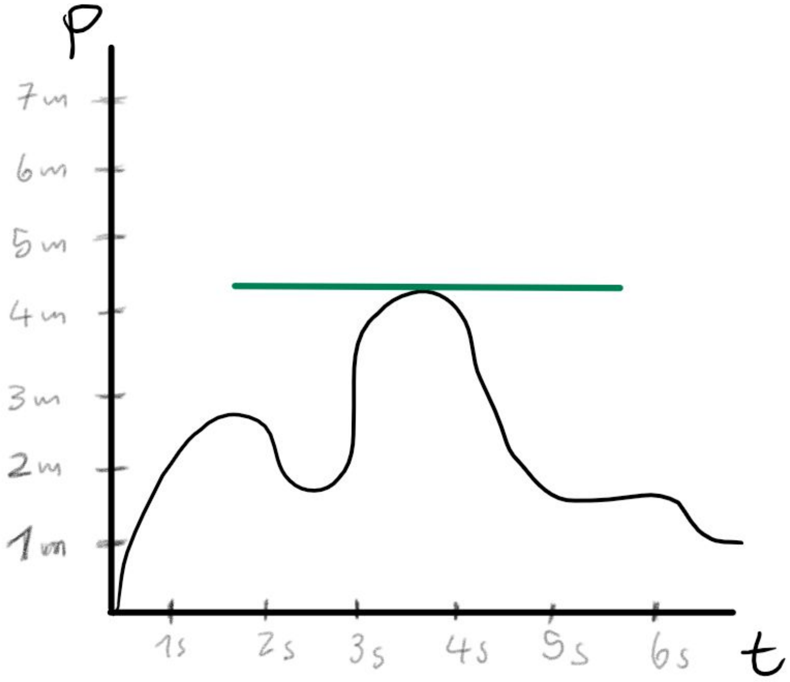
Revision

Movement Basics & Mass Spring Systems

- We simulate the position of massPoints over time, influenced by springs.
- The **position** is influenced by the **velocity** (“speed and direction”).
- The **velocity** is influenced by the **acceleration** (“= Beschleunig.”).
- The **acceleration** is influenced by **forces**.

- Derivatives (= “gradient” = how does something change, e.g. pos over time)
- Hooke’s Law (= “Spring forces” = *If too short → push, if too long → pull*)

Revision



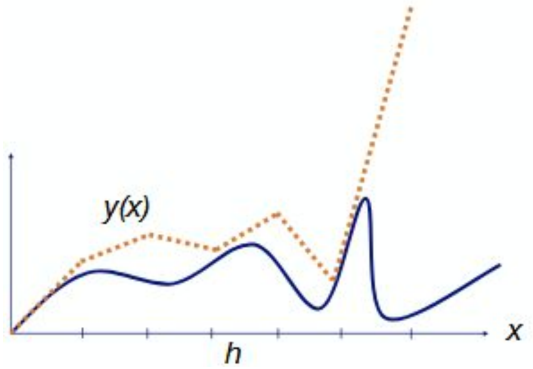
$p(t)$ - position at time t

$\frac{\delta p(t)}{\delta t} = v(t)$ - velocity

$\frac{\delta v(t)}{\delta t} = a(t)$ - acceleration

Today: Integration Methods

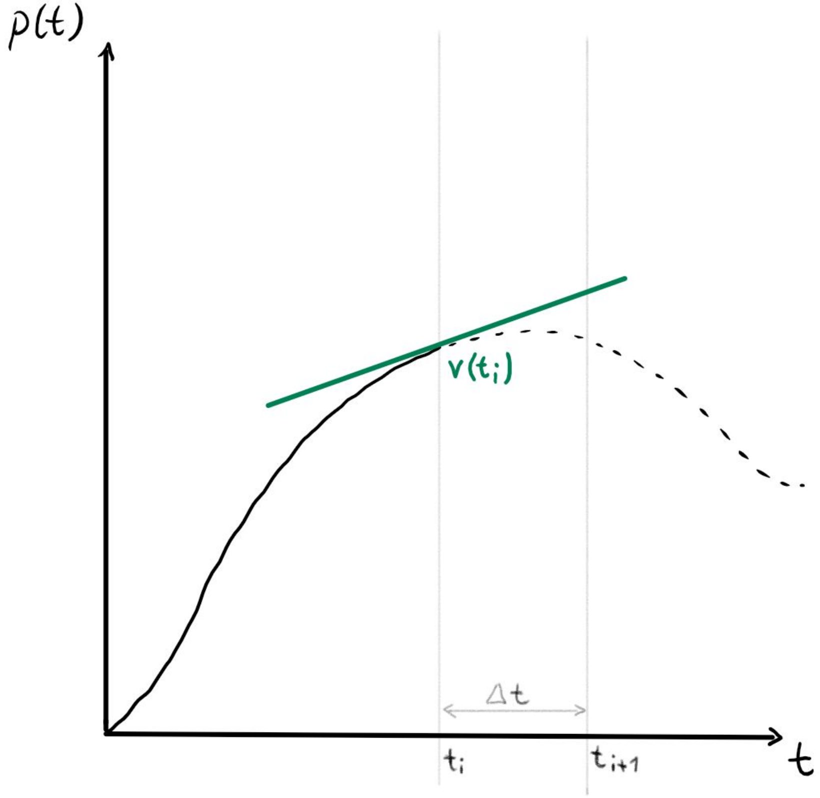
Problem with Explicit Euler integration: Unstable.



What other (more stable) integration methods are there?

Or: In what order do we update the position & velocity, so the simulation doesn't explode?

Explicit Euler



Explicit Euler Integration:

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

approximation!

Explicit Euler (Info Sheet)

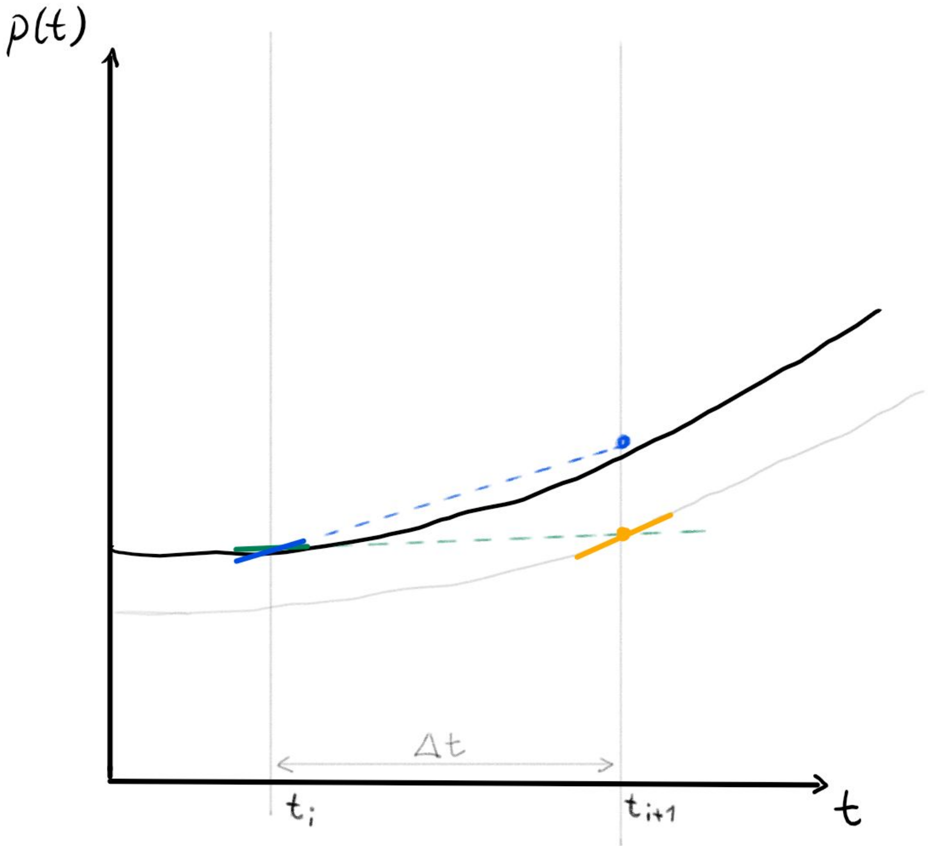
- "Step-by-Step Method: Update position, update velocity"
- First order method (fast to compute), but not accurate (simulation "explodes")

Explicit Euler (Algorithm)

- **newPos** = oldPos + timestep × oldVel
- **newVel** = oldVel + timestep × [Acc at oldPos]

$$y_{n+1} = y_n + hf(x_n, y_n)$$

Heun



Heun's Method:

1. Perform Explicit Euler Step

2. Calculate Velocity at approximated position

3. Use average of velocities for Heun step

$$p(t_{i+1}) = p(t_i) + \Delta t \cdot \frac{1}{2}\left(\tilde{v}(t_{i+1}) + v(t_i)\right)$$

Heun (Info Sheet)

> “Average Method: Make Eulerstep as help, calculate velocity there. Update the position with the average of the current velocity and the velocity at the euler step.”

What does the algorithm look like?

Heun (Algorithm)

- [Pos of Eulerstep] = oldPos + timestep × oldVel
- [Vel at Eulerstep] = oldVel + timestep × [Acc at oldPos]
- **newPos** = oldPos + timestep × ½ × ([Vel at Eulerstep] + oldVel)
- **newVel** = oldVel + timestep × ½ × ([Acc at oldPos] + [Acc at Pos of Eulerstep])

$$\tilde{y} = y_0 + h f(x_0, y_0)$$

$$y_1 = y_0 + \frac{h}{2}\left(f(x_0, y_0) + f(x_1, \tilde{y})\right)$$

We have to calculate acceleration twice!
→ calculate forces twice
→ second order (double workload!)

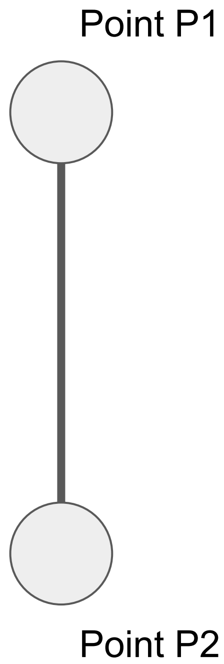
Heun - Exercise 1

- <u>Calculate the positions and velocities of P1 & P2 after one time step!</u>

Initial positions	**P1.x0 = (-2, 2)**	**P2.x0 = (3, 2)**.
Initial velocities	**P1.v0 = (-1, -1.5)**	**P2.v0 = (1, -1.5)**
Masses	**P1.m = 2**	**P2.m = 3**

The points are affected by **gravity (0, -10)**.

P1 and P2 are connected by a spring with **rest length 6** and **stiffness 4**. The **timestep is 0.5**.



Heun - Exercise 1 (Solution)

1. Perform Euler Step as Help:

$\tilde{v}_1(t_{i+1}) = \begin{pmatrix} -1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} -2 \\ -10 \end{pmatrix} = \begin{pmatrix} -2 \\ -6.5 \end{pmatrix}$

$\tilde{v}_2(t_{i+1}) = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 4/3 \\ -10 \end{pmatrix} = \begin{pmatrix} 5/3 \\ -6.5 \end{pmatrix}$

$\tilde{p}_1(t_{i+1}) = \begin{pmatrix} -2 \\ 2 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} -1 \\ -1.5 \end{pmatrix} = \begin{pmatrix} -2.5 \\ 1.25 \end{pmatrix}$

$\tilde{p}_2(t_{i+1}) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 1 \\ -1.5 \end{pmatrix} = \begin{pmatrix} 3.5 \\ 1.25 \end{pmatrix}$

2. Calculate new positions:

$p_1(t_{i+1}) = \begin{pmatrix} -2 \\ 2 \end{pmatrix} + 0.5 \cdot \frac{1}{2}\left(\begin{pmatrix} -2 \\ -6.5 \end{pmatrix} + \begin{pmatrix} -1 \\ -1.5 \end{pmatrix}\right) = \begin{pmatrix} -2.75 \\ 0 \end{pmatrix}$

$p_2(t_{i+1}) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} + 0.5 \cdot \frac{1}{2}\left(\begin{pmatrix} 5/3 \\ -6.5 \end{pmatrix} + \begin{pmatrix} 1 \\ -1.5 \end{pmatrix}\right) = \begin{pmatrix} 11/3 \\ 0 \end{pmatrix}$

3. Calculate acceleration at $\tilde{p}$:

$\tilde{d} = 6 \quad \rightsquigarrow \tilde{f}_1(t_{i+1}) = -4 \cdot (6-6) \cdot \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

$\tilde{f}_2(t_{i+1}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

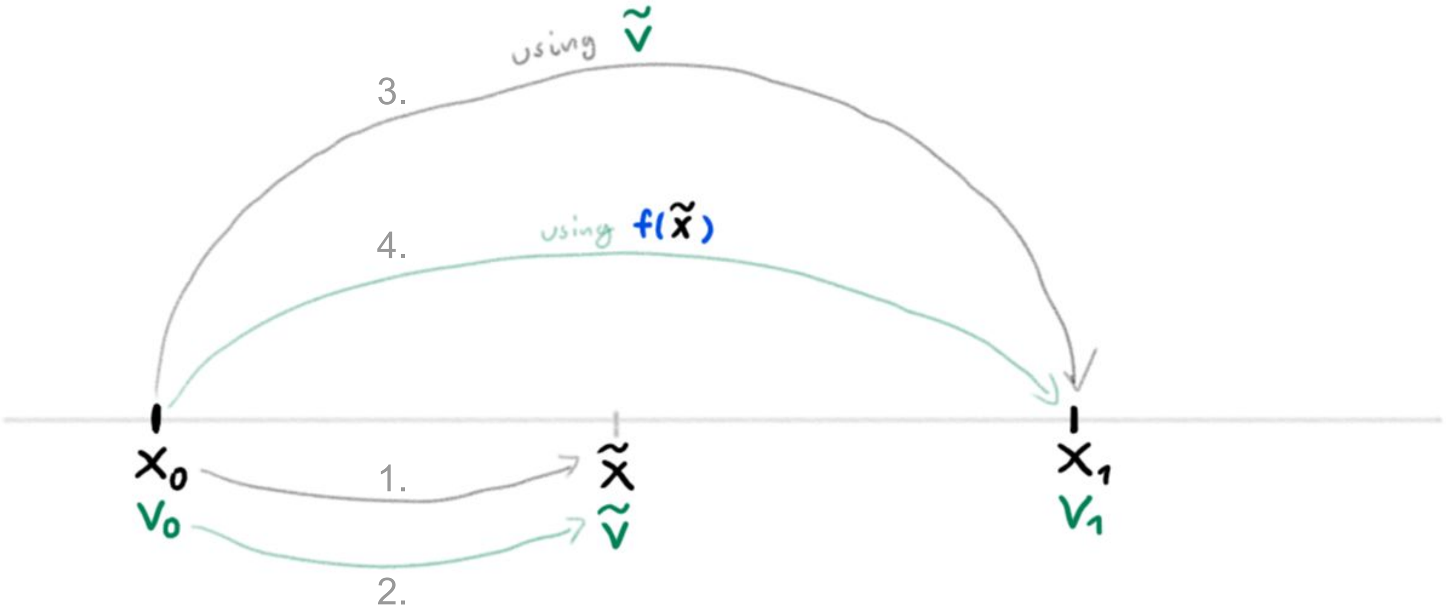
$\rightsquigarrow \tilde{a}_1(t_{i+1}) = \begin{pmatrix} 0 \\ -10 \end{pmatrix} \quad \tilde{a}_2(t_{i+1}) = \begin{pmatrix} 0 \\ -10 \end{pmatrix}$

4. Calculate new velocities:

$v_1(t_{i+1}) = \begin{pmatrix} -1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \frac{1}{2}\left(\begin{pmatrix} -2 \\ -10 \end{pmatrix} + \begin{pmatrix} 0 \\ -10 \end{pmatrix}\right) = \begin{pmatrix} -1.5 \\ -6.5 \end{pmatrix}$

$v_2(t_{i+1}) = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \frac{1}{2}\left(\begin{pmatrix} 4/3 \\ -10 \end{pmatrix} + \begin{pmatrix} 0 \\ -10 \end{pmatrix}\right) = \begin{pmatrix} 4/3 \\ -6.5 \end{pmatrix}$

Midpoint



Midpoint (Info Sheet)

- “Make a half step, update the point with the velocity of this old step”
- Second order! (also called Runge-Kutta 2nd order)

What does the algorithm look like?

Midpoint (Algorithm)

- [Pos of Midstep] = oldPos + ½ × timestep × oldVel
- [Vel at Midstep] = oldVel + ½ × timestep × [Acc at oldPos]
- **newPos** = oldPos + timestep × [Vel at Midstep]
- **newVel** = oldVel + timestep × [Acc at Midstep]

$$\tilde{\mathbf{x}} = \mathbf{x}(t) + h/2\mathbf{v}(t, \mathbf{x}(t))$$
$$\mathbf{x}(t+h) = \mathbf{x}(t) + h\tilde{\mathbf{v}}$$

$$\tilde{\mathbf{v}} = \mathbf{v}(t) + h/2\, \mathbf{a}(t, \mathbf{x}(t), \mathbf{v}(t))$$
$$\mathbf{v}(t+h) = \mathbf{v}(t) + h\, \mathbf{a}(t + h/2, \tilde{\mathbf{x}}, \tilde{\mathbf{v}})$$

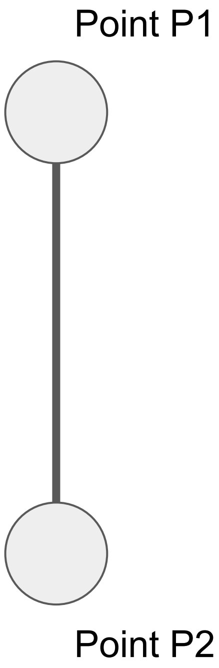
Midpoint - Exercise 2

- <u>Calculate the positions and velocities of P1 & P2 after one time step!</u>

Initial positions	**P1.x0 = (-2, 2)**	**P2.x0 = (3, 2)**.
Initial velocities	**P1.v0 = (-1, -1.5)**	**P2.v0 = (1, -1.5)**
Masses	**P1.m = 2**	**P2.m = 3**

The points are affected by **gravity (0, -10)**.

P1 and P2 are connected by a spring with **rest length 6** and **stiffness 4**. The **timestep is 0.5**.



Midpoint - Exercise 2 (Solution)

1. Perform Midpoint Step:

$$\tilde{p}_1(t_{i+0.5}) = \begin{pmatrix}-2\\2\end{pmatrix} + \frac{1}{2}\cdot 0.5 \cdot \begin{pmatrix}-1\\-1.5\end{pmatrix} = \begin{pmatrix}-2.25\\1.625\end{pmatrix}$$

$$\tilde{p}_2(t_{i+0.5}) = \begin{pmatrix}3\\2\end{pmatrix} + \frac{1}{2}\cdot 0.5 \cdot \begin{pmatrix}1\\-1.5\end{pmatrix} = \begin{pmatrix}3.25\\1.625\end{pmatrix}$$

$$\tilde{v}_1(t_{i+0.5}) = \begin{pmatrix}-1\\-1.5\end{pmatrix} + \frac{1}{2}\cdot 0.5 \cdot \begin{pmatrix}-2\\-10\end{pmatrix} = \begin{pmatrix}-1.5\\-4\end{pmatrix}$$

$$\tilde{v}_2(t_{i+0.5}) = \begin{pmatrix}1\\-1.5\end{pmatrix} + \frac{1}{2}\cdot 0.5 \cdot \begin{pmatrix}4/3\\-10\end{pmatrix} = \begin{pmatrix}4/3\\-4\end{pmatrix}$$

2. Calculate acceleration at Midpoint:

$d = 5.5$

$$\rightsquigarrow \tilde{f}_1(t_{i+0.5}) = -4\cdot(5.5-6)\cdot\begin{pmatrix}-1\\0\end{pmatrix} = \begin{pmatrix}-2\\0\end{pmatrix}$$

$$\tilde{f}_2(t_{i+0.5}) = -4\cdot(5.5-6)\cdot\begin{pmatrix}1\\0\end{pmatrix} = \begin{pmatrix}2\\0\end{pmatrix}$$

$$\rightsquigarrow \tilde{a}_1(t_{i+0.5}) = \begin{pmatrix}-2\\0\end{pmatrix}\cdot\frac{1}{2} + \begin{pmatrix}0\\-10\end{pmatrix} = \begin{pmatrix}-1\\-10\end{pmatrix}$$

$$\tilde{a}_2(t_{i+0.5}) = \begin{pmatrix}2\\0\end{pmatrix}\cdot\frac{1}{3} + \begin{pmatrix}0\\-10\end{pmatrix} = \begin{pmatrix}2/3\\-10\end{pmatrix}$$

3. Use Midstep values to perform full step:

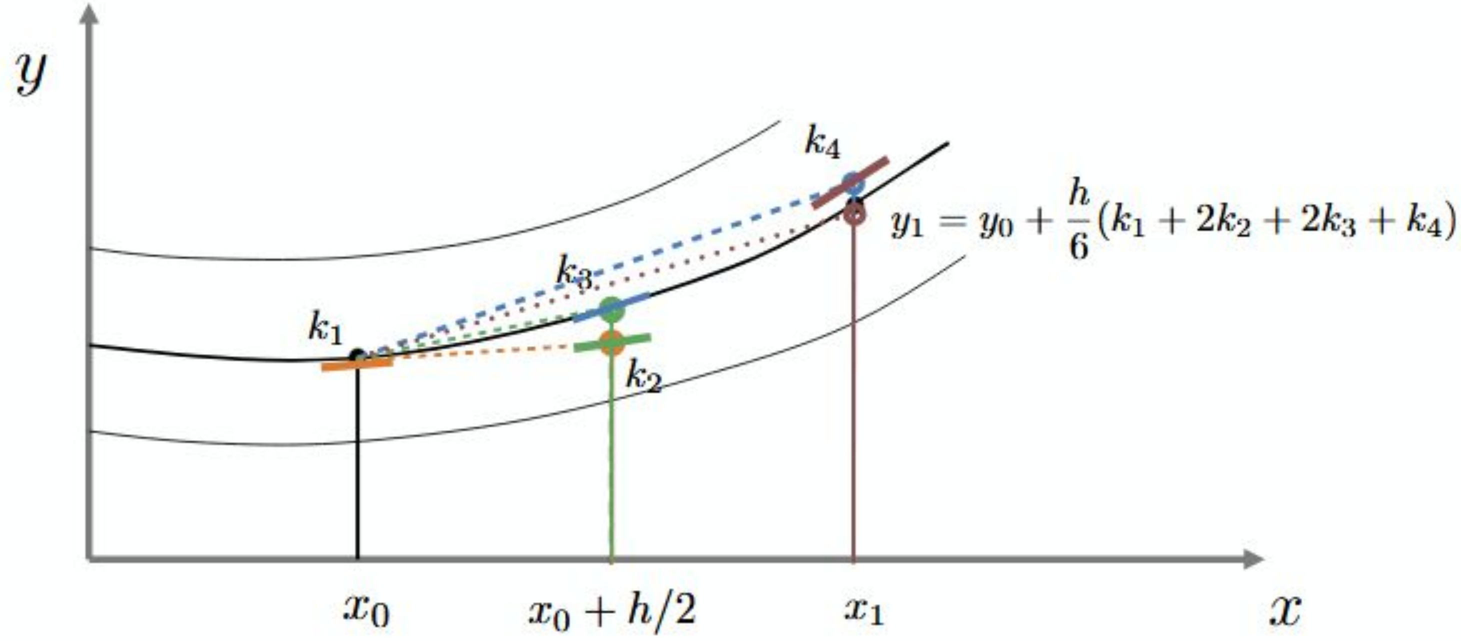
$$p_1(t_{i+1}) = \begin{pmatrix}-2\\2\end{pmatrix} + 0.5\cdot\begin{pmatrix}-1.5\\-4\end{pmatrix} = \begin{pmatrix}-2.75\\0\end{pmatrix}$$

$$p_2(t_{i+1}) = \begin{pmatrix}3\\2\end{pmatrix} + 0.5\cdot\begin{pmatrix}4/3\\-4\end{pmatrix} = \begin{pmatrix}11/3\\0\end{pmatrix}$$

$$v_1(t_{i+1}) = \begin{pmatrix}-1\\-1.5\end{pmatrix} + 0.5\cdot\begin{pmatrix}-1\\-10\end{pmatrix} = \begin{pmatrix}-1.5\\-6.5\end{pmatrix}$$

$$v_2(t_{i+1}) = \begin{pmatrix}1\\-1.5\end{pmatrix} + 0.5\cdot\begin{pmatrix}2/3\\-10\end{pmatrix} = \begin{pmatrix}4/3\\-6.5\end{pmatrix}$$

Runge-Kutta

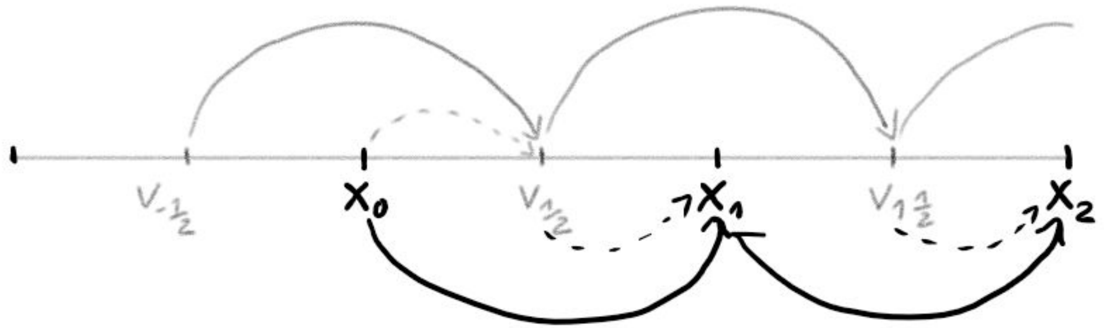


Runge-Kutta (Info Sheet)

- “Midpoint + weighted average”
- Fourth order method!
- → Can be expanded to use any order.
- Better than four (small) Euler steps!

This is too much to calculate, so we won’t get into more details...

Leap-Frog



Leap-Frog (Info Sheet)

- “Inverted Euler”
- In practice, it’s just Explicit Euler, but small change: update the velocity before the position
- Good: Only one force evaluation (“first order”), but second order accuracy

Leap-Frog (Algorithm)

- **newVel** = oldVel + timestep × [Acc at oldPos]
- **newPos** = oldPos + timestep × newVel

$$\begin{aligned} \mathbf{v}(t+h/2) &= \mathbf{v}(t-h/2) + h\ \mathbf{a}(t) \\ \mathbf{x}(t+h) &= \mathbf{x}(t) + h\ \mathbf{v}(t+h/2) \end{aligned}$$

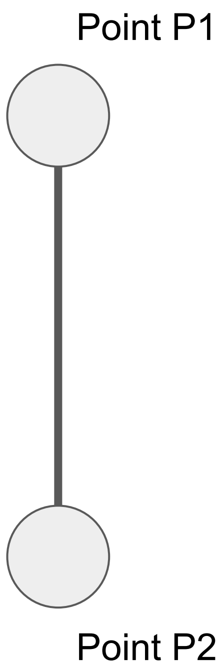
Leap-Frog - Exercise 3

- <u>Calculate the positions of P1 & P2 after one time step!</u>

Initial positions (t = 0)	**P1.x0 = (-2, 2)**	**P2.x0 = (3, 2)**
Initial velocities (t = -0.25)	**P1.v0 = (-1, -1.5)**	**P2.v0 = (1, -1.5)**
Masses	**P1.m = 2**	**P2.m = 3**

The points are affected by **gravity (0, -10)**.

P1 and P2 are connected by a spring with **rest length 6** and **stiffness 4**. The **timestep is 0.5**.



Leap-Frog - Exercise 3 (Solution)

Use $a(t=0)$ to update $v(t=-0.25)$:

$$v_1(t=0.25) = \begin{pmatrix} -1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} -2 \\ -10 \end{pmatrix} = \begin{pmatrix} -2 \\ -6.5 \end{pmatrix}$$

$$v_2(t=0.25) = \begin{pmatrix} 1 \\ -1.5 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 4/3 \\ -10 \end{pmatrix} = \begin{pmatrix} 5/3 \\ -6.5 \end{pmatrix}$$

Use $v(t=0.25)$ to update $p(t=0)$:

$$p_1(t=0.5) = \begin{pmatrix} -2 \\ 2 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} -2 \\ -6.5 \end{pmatrix} = \begin{pmatrix} -3 \\ -1.25 \end{pmatrix}$$

$$p_2(t=0.5) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} + 0.5 \cdot \begin{pmatrix} 5/3 \\ -6.5 \end{pmatrix} = \begin{pmatrix} 23/6 \\ -1.25 \end{pmatrix}$$

Calculate **Acceleration** (from Explicit Euler Exercise):

$$P_1: a(t=0) = \begin{pmatrix} -4 \\ 0 \end{pmatrix} \cdot \frac{1}{2} + \begin{pmatrix} 0 \\ -10 \end{pmatrix} = \begin{pmatrix} -2 \\ -10 \end{pmatrix}$$

$$P_2: a(t=0) = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \cdot \frac{1}{3} + \begin{pmatrix} 0 \\ -10 \end{pmatrix} = \begin{pmatrix} 4/3 \\ -10 \end{pmatrix}$$

Summary

- Explicit Euler: *“Step by Step”* (1st order)
- Leapfrog: *“Inverted Euler”* (1st order, but 2nd order accuracy)
- Heun: *“Average”* (2nd order)
- Midpoint: *“Midpoint”* (2nd order Runge-Kutta)
- Runge-Kutta: *“Midpoint + Weighted Average”* (any order, e.g. 4th)
- Implicit Methods: “System of Equations” (not really used)
- Higher order and/or smaller timesteps → better stability