Presented by Louis Bouchard

SIMULATING PHYSICS WITH NUMERICAL INTEGRATION

WHY?

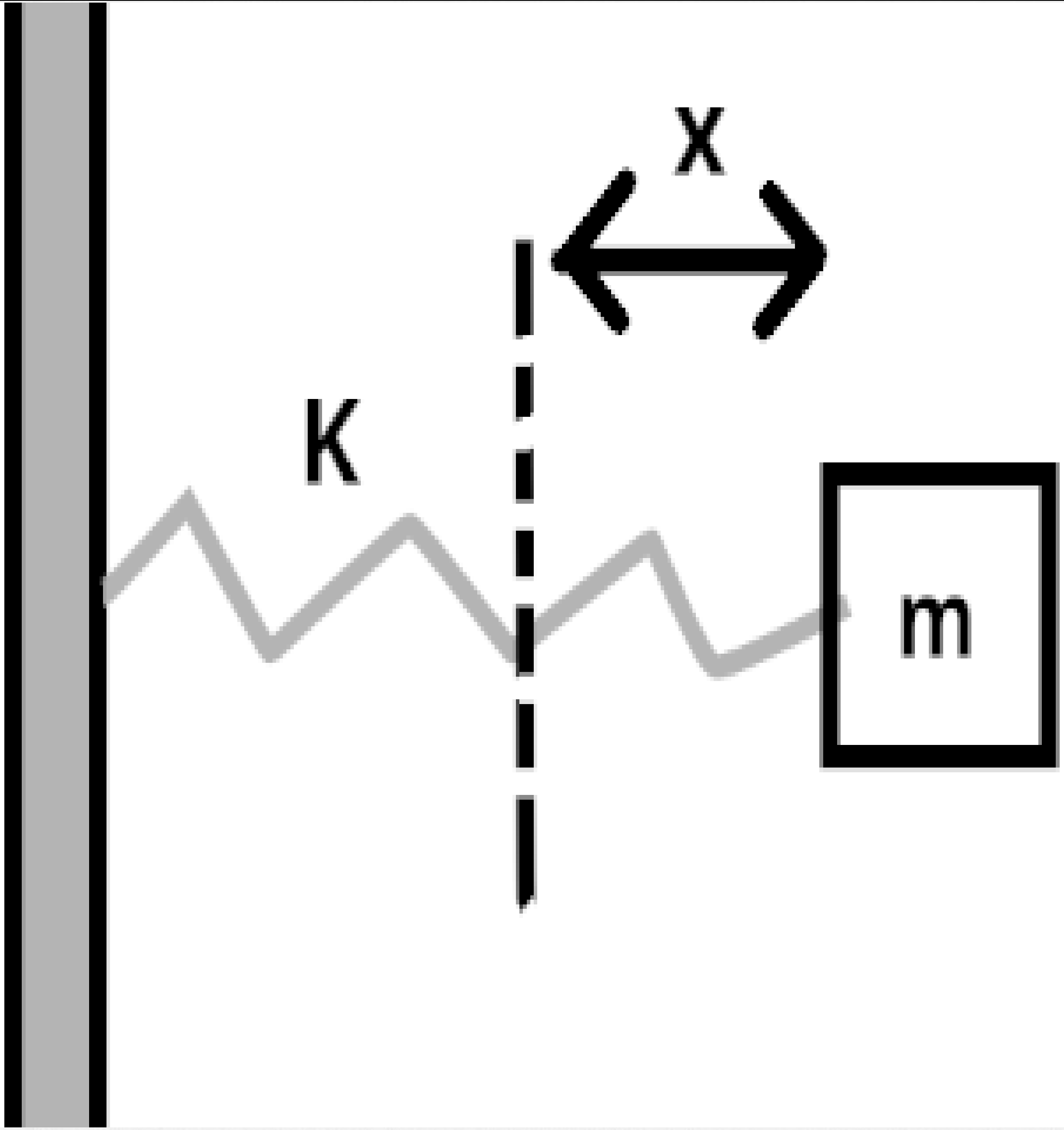
AN EASY PROBLEM

The classic mass on a spring

$$ma = -kx$$

$$\ddot{x} = -\omega^2 x, \;\; \omega = \sqrt{\frac{k}{m}}$$

$$x(t) = A\sin(\omega t + \phi)$$



A SMALL ADJUSTMENT

Adding a term to make it more realistic

$$ma = -kx - \alpha x^3$$

$$\ddot{x} = \frac{-kx}{m} + \frac{-\alpha x^3}{m}$$

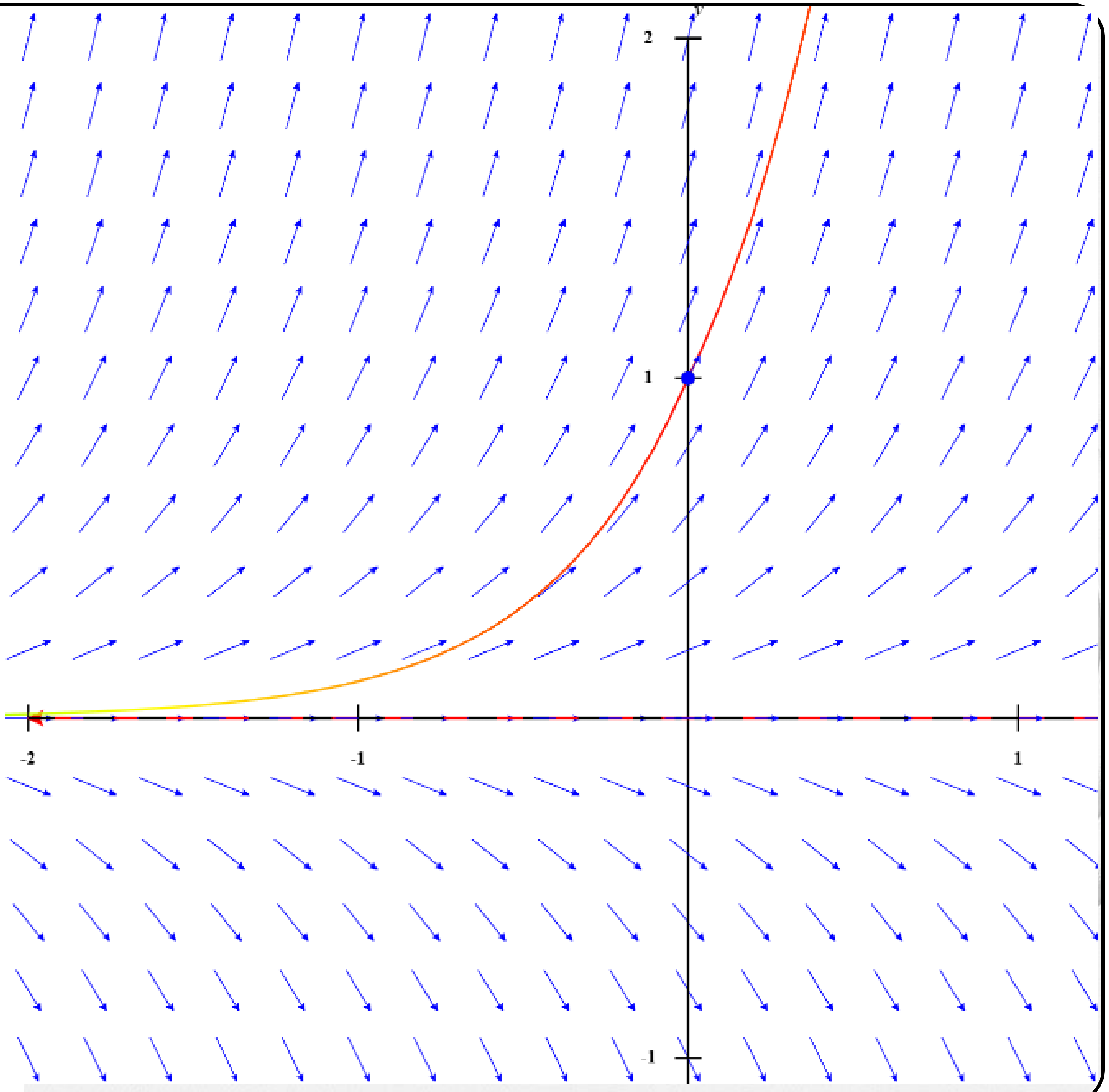
Ain't no easy way to solve it now

WHAT?

VECTOR FIELDS

Given the following ODE

$$\frac{dx}{dt} = 2x$$



Just follow the arrows!

(Needs a starting point)

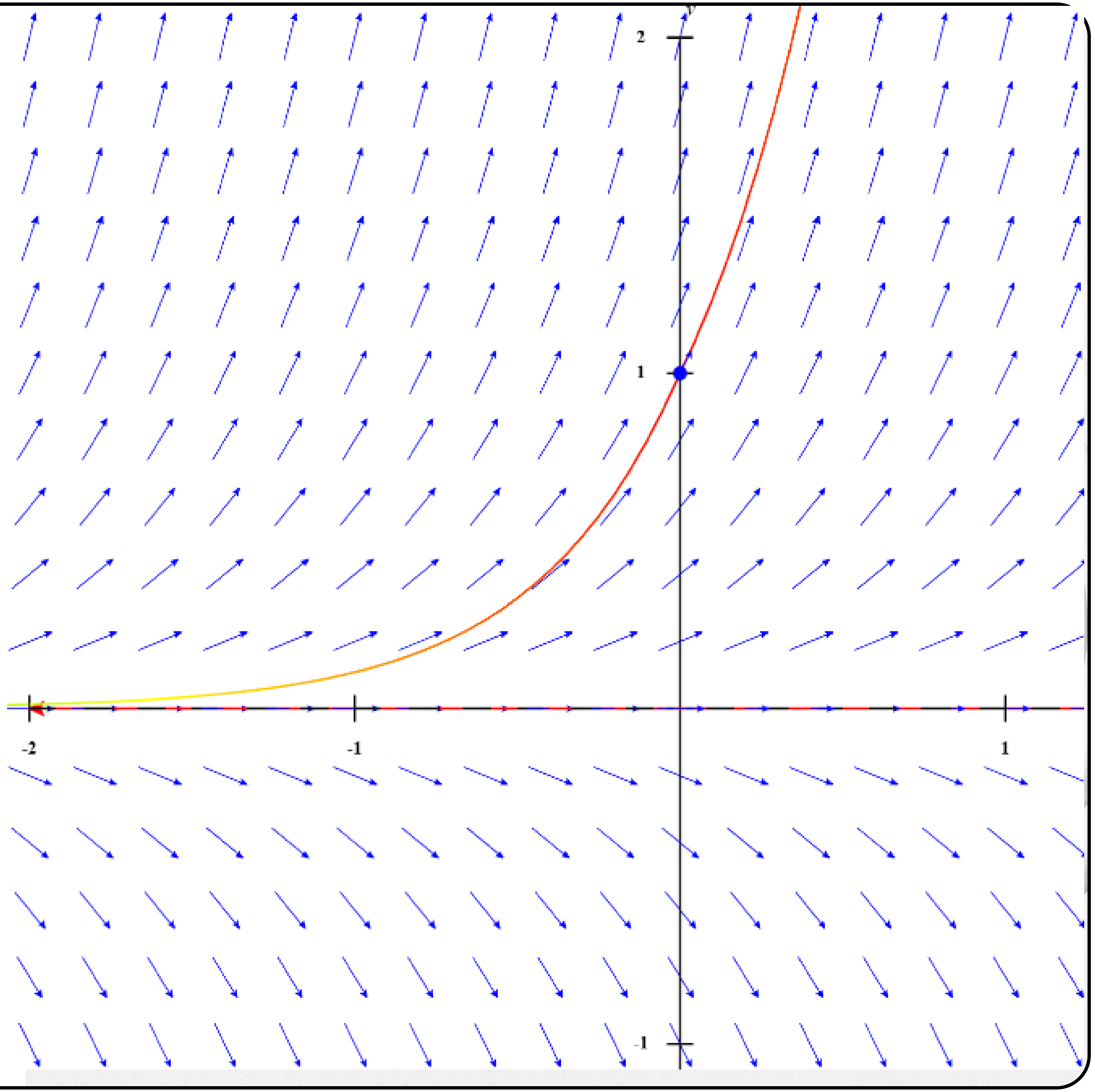
HOW?!

APPROXIMATE IT

Given the same ODE

$$\frac{dx}{dt} = 2x$$

Approximate x incrementally



FORWARD EULER

$$x_{n+1} = x_n + hf(x_n,\ t_n)$$

h: Step Size

x_n: Current value

x_{n+1} : Next value

$t_n = t_0 + nh$: Current time

Initial conditions

x_0 : Starting x value

t_0 : Starting time

FORWARD EULER

$$\frac{dx}{dt} = 2x \qquad h = 0.1$$

$$(x_0 = 1,\ t_0 = 0) : \ x_1 = 1 + 0.1 \cdot 2$$

$$(x_1 = 1.2,\ t_1 = 0.1) : \ x_2 = 1.2 + 0.1 \cdot 2.4$$

$$(x_2 = 1.44,\ t_2 = 0.2)$$

$$\ldots$$

FORWARD EULER

Where from?

Taylor Series!

$$x(t) = x(a) + \dot{x}(a)(t-a) + \ddot{x}(a)\frac{(t-a)^2}{2} + O(t^3)$$

$$x(a+h) = x(a) + h\dot{x}(a) + O(h^2)$$

$$x(t_{n+1}) = x(t_n) + h\dot{x}(t_n) + O(h^2)$$

BACKWARD EULER

$$x_{n+1} = x_n + hf(x_{n+1},\ t_{n+1})$$

But it uses the next value that I
don't have?

Just use forward Euler to
approximate
x_{n+1}

RUNGE-KUTTA (RK)

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$k_1 = hf(t_n, x_n),$$
$$k_2 = hf(t_n + \frac{h}{2}, x_n + \frac{k_1}{2}),$$
$$k_3 = hf(t_n + \frac{h}{2}, x_n + \frac{k_2}{2}),$$
$$k_4 = hf(t_n + h, x_n + k_3).$$

A family of methods of any order.

More accurate but more expensive.

On the left is RK4.

WHAT ABOUT SECOND ORDER ODES?

Most real life systems have acceleration, which is a second order term.

You can use the previous methods by simply converting it into a systems of first order ODEs, which you can then solve normally!

Starting ODE

$$\ddot{x} = -x$$

New variables

$$x_1 = x$$

$$x_2 = \dot{x}$$

ODE transformed into system

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x$$

SYMPLECTIC METHODS

Nice methods that have the sweet property of bounding the error on the total energy.

Forward Euler: Will drift overtime and explode
Backward Euler: Will drift overtime and die out
Runge-Kutta: Very nice, but will still drift and lose/gain energy over time.

SEMI IMPLICIT EULER

Update velocity first, then position.
Uses a simple explicit method (Similar to Forward Euler)

$$\ddot{x} = f(t,\ x,\ \dot{x})$$

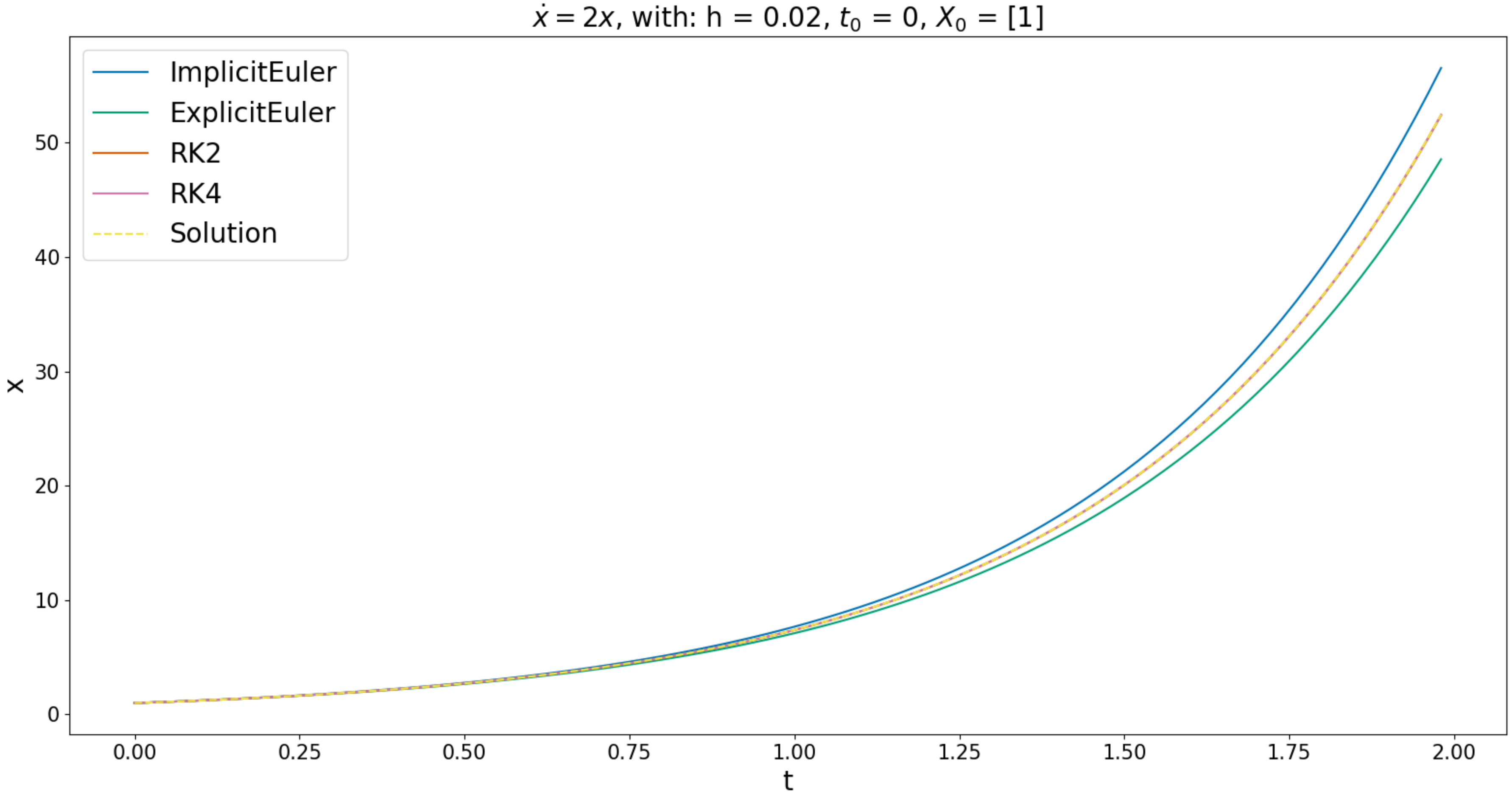
$$v_{n+1} = v_n + hf(t_n,\ x_n)$$

$$x_{n+1} = x_n + h \cdot v_{n+1}$$

EXAMPLES!

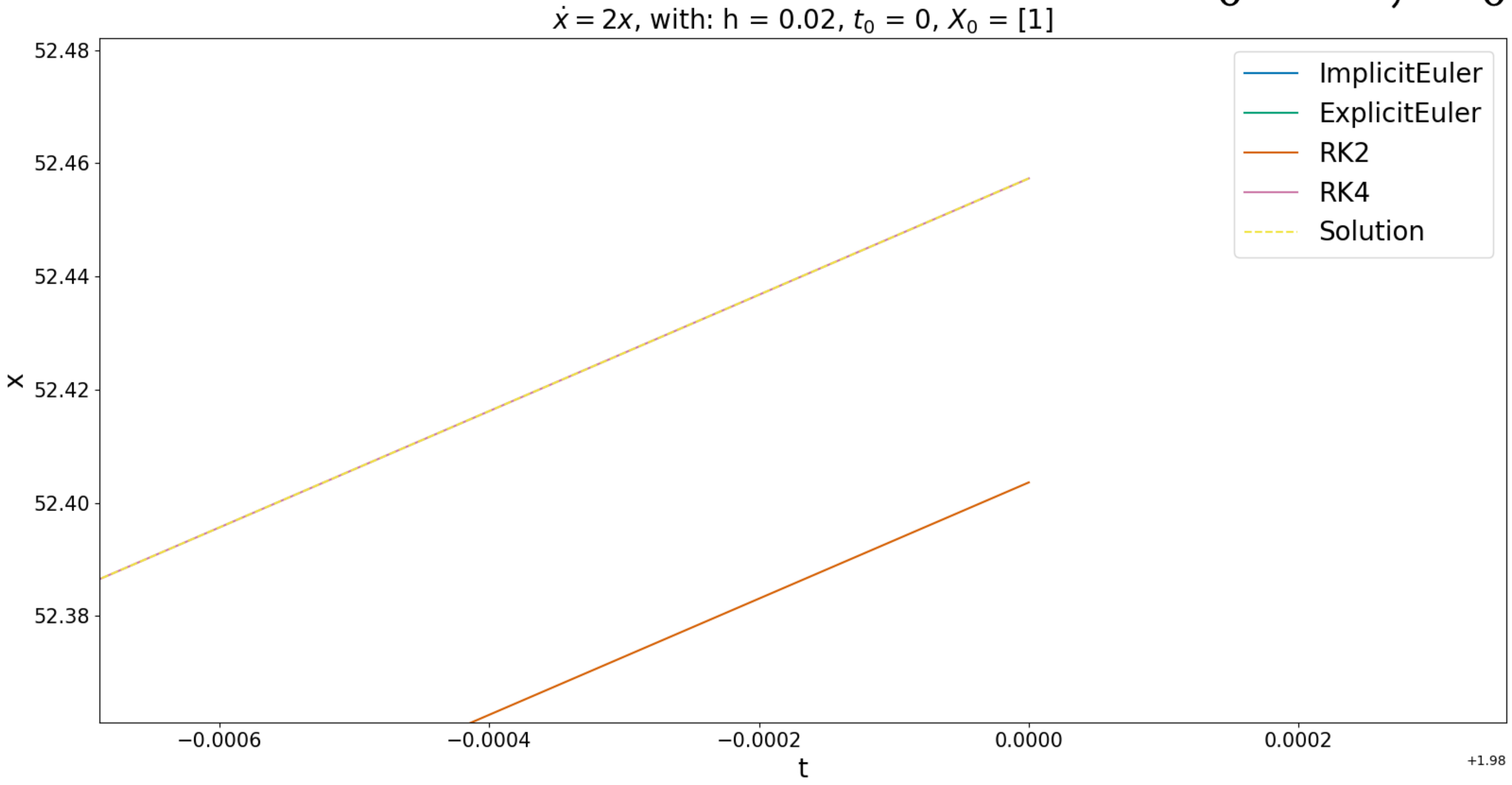
EXPONENTIAL $\dot{x} = 2x$

$$t_0 = 0,\ x_0 = 1$$



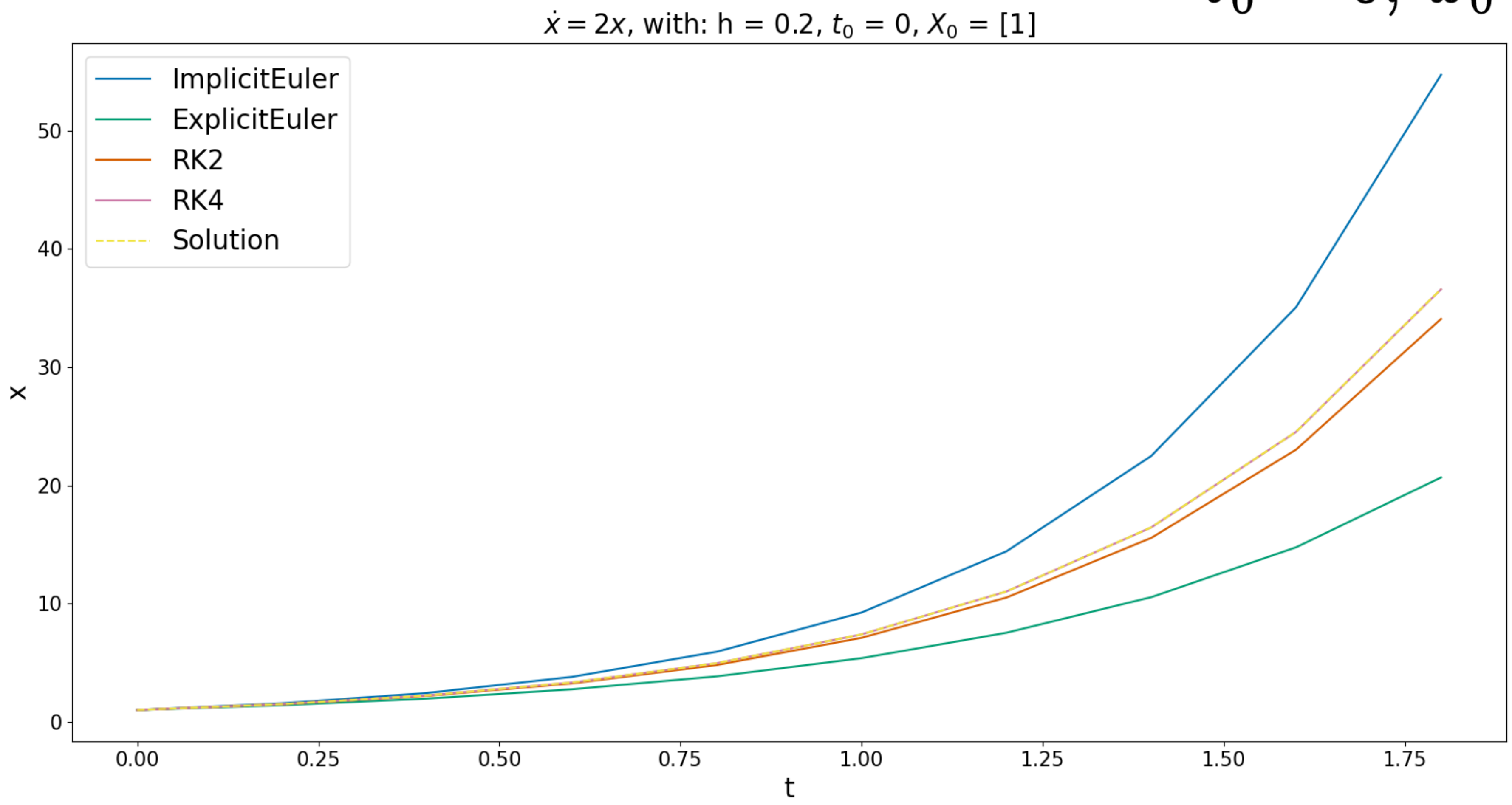
EXPONENTIAL $\dot{x} = 2x$

$t_0 = 0,\ x_0 = 1$



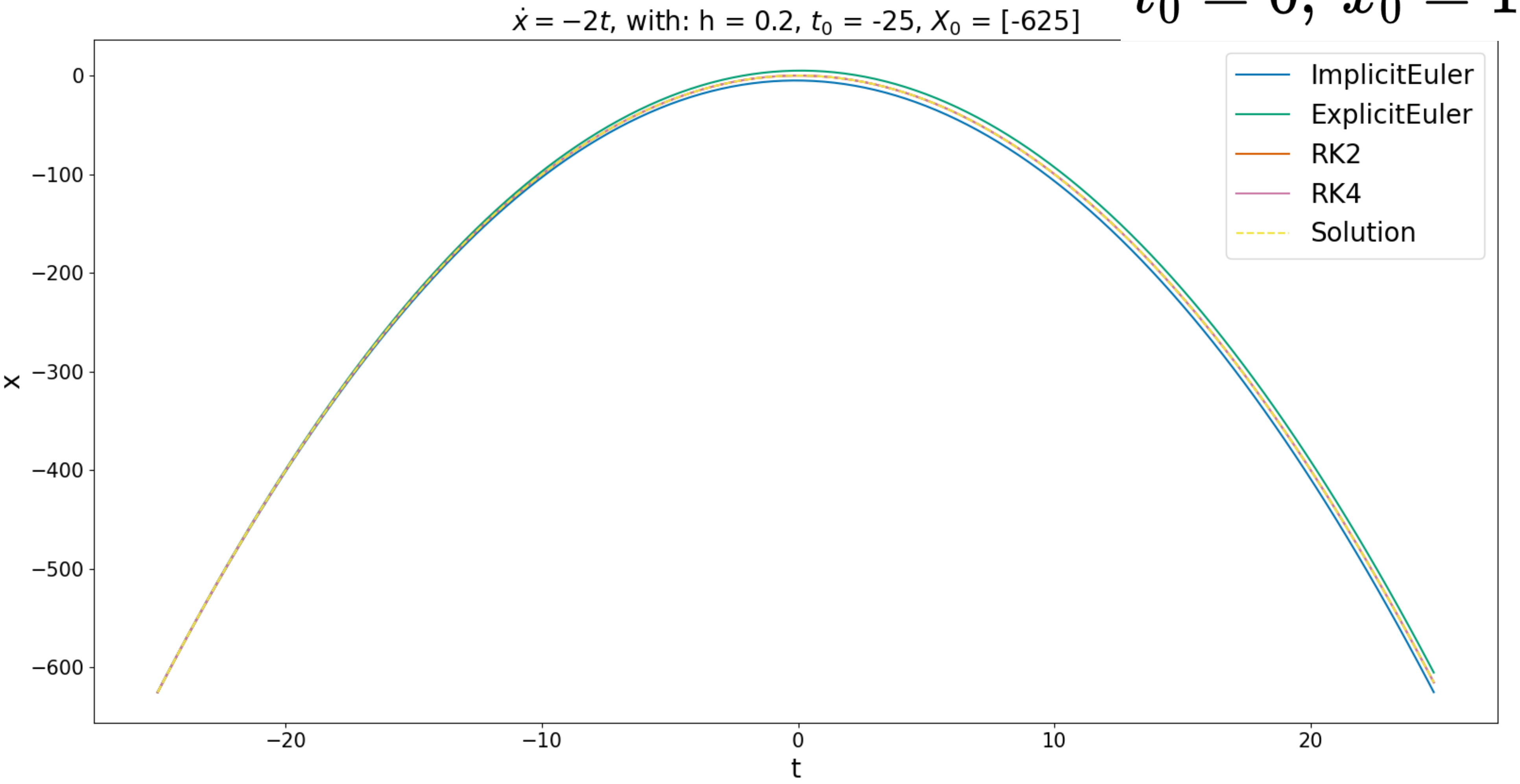
EXPONENTIAL $\dot{x} = 2x$

$$t_0 = 0,\ x_0 = 1$$



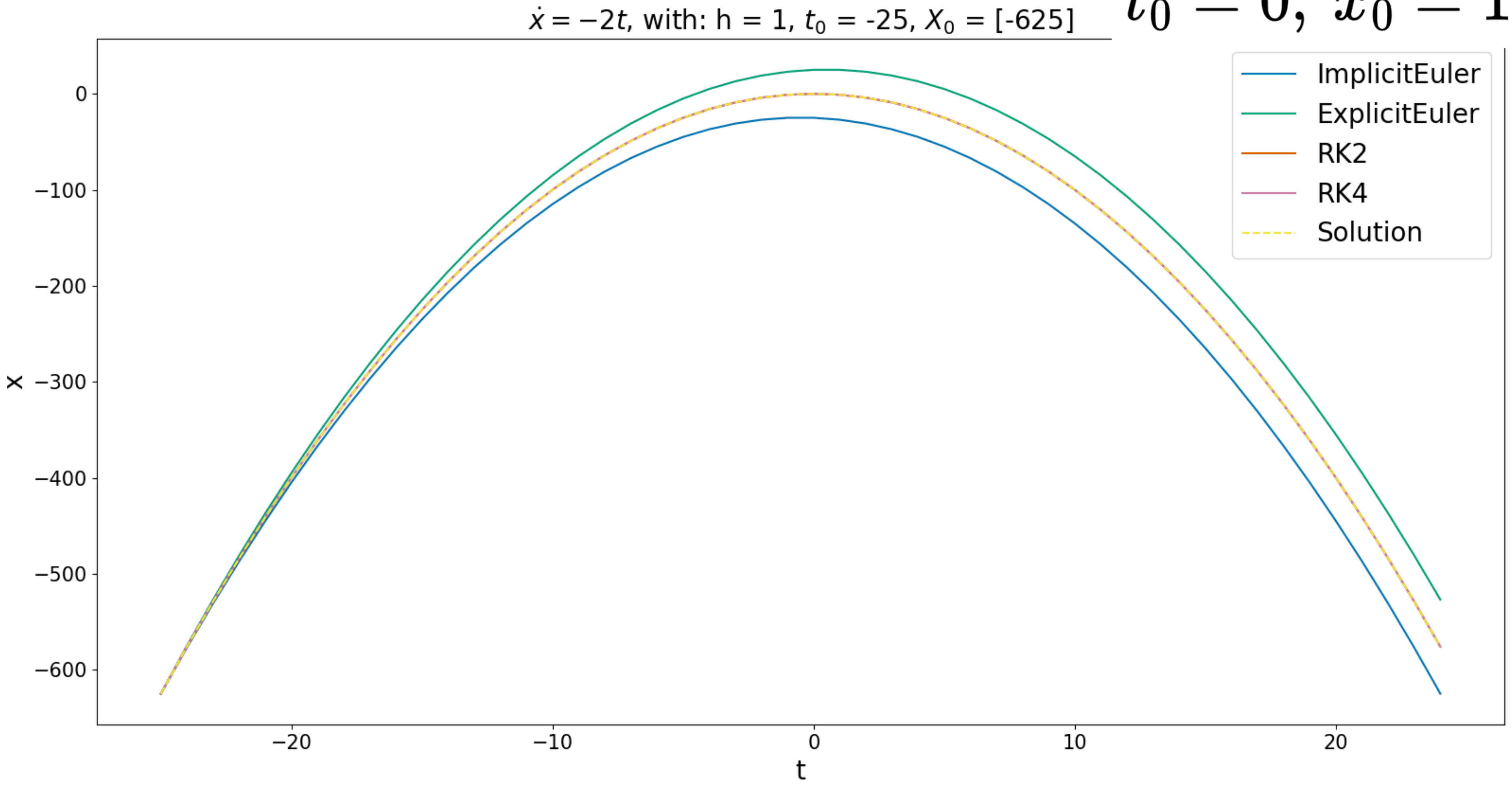
PARABOLIC $\dot{x} = -2t$

$t_0 = 0,\ x_0 = 1$



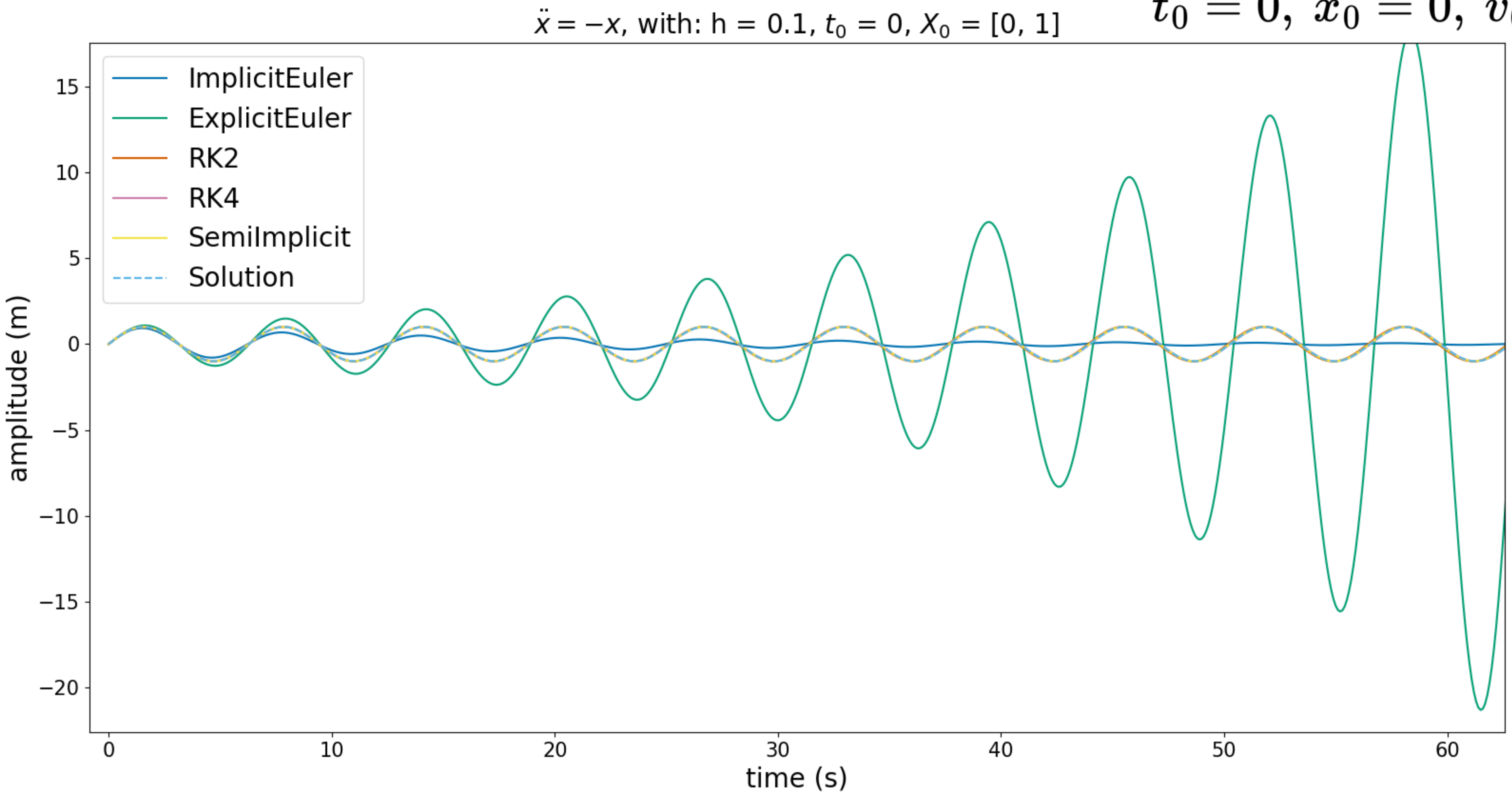
PARABOLIC $\dot{x} = -2t$

$t_0 = 0,\ x_0 = 1$



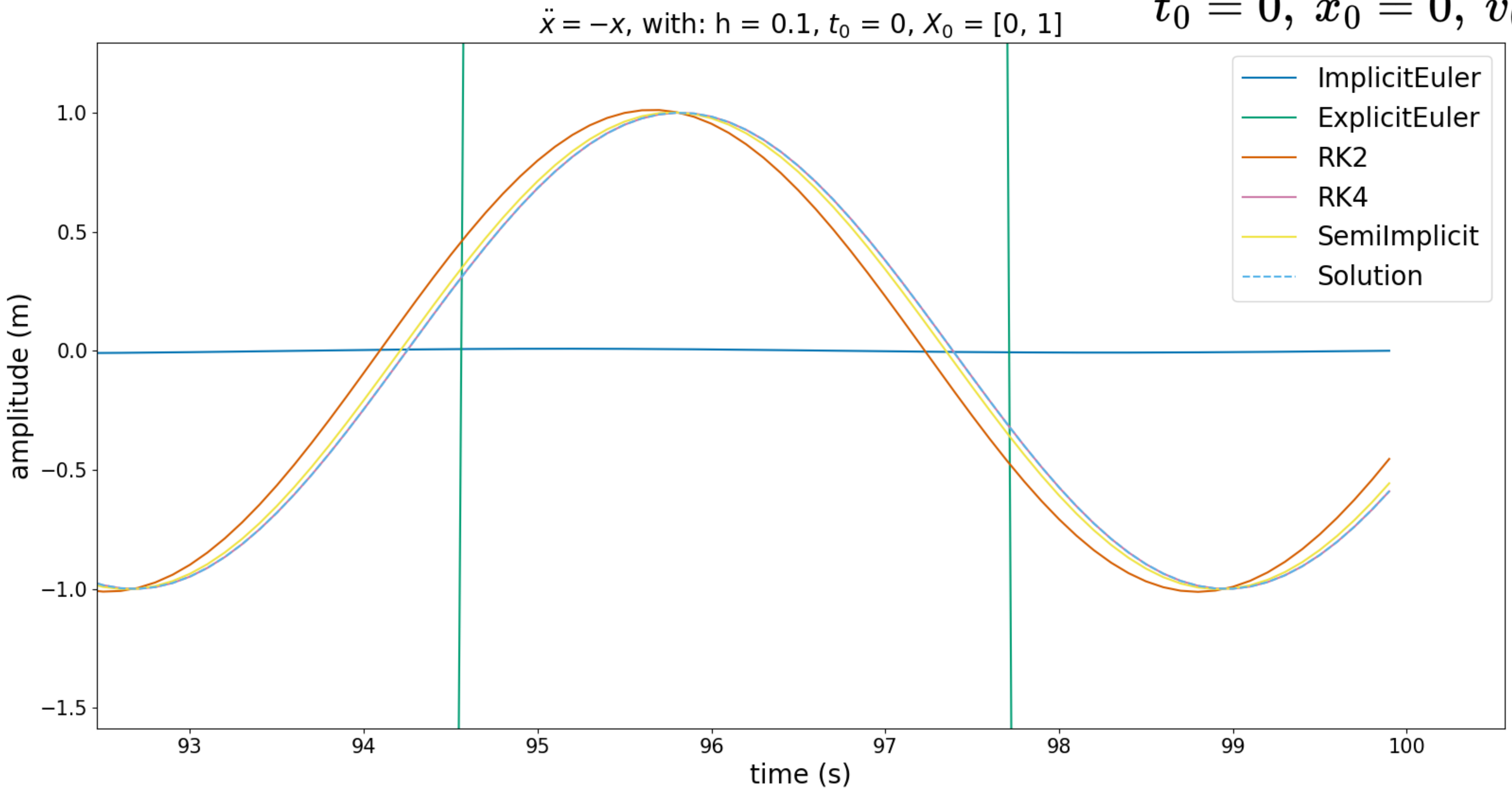
SPRING MASS $\ddot{x} = -x$

$$t_0 = 0,\ x_0 = 0,\ v_0 = 1m/s$$



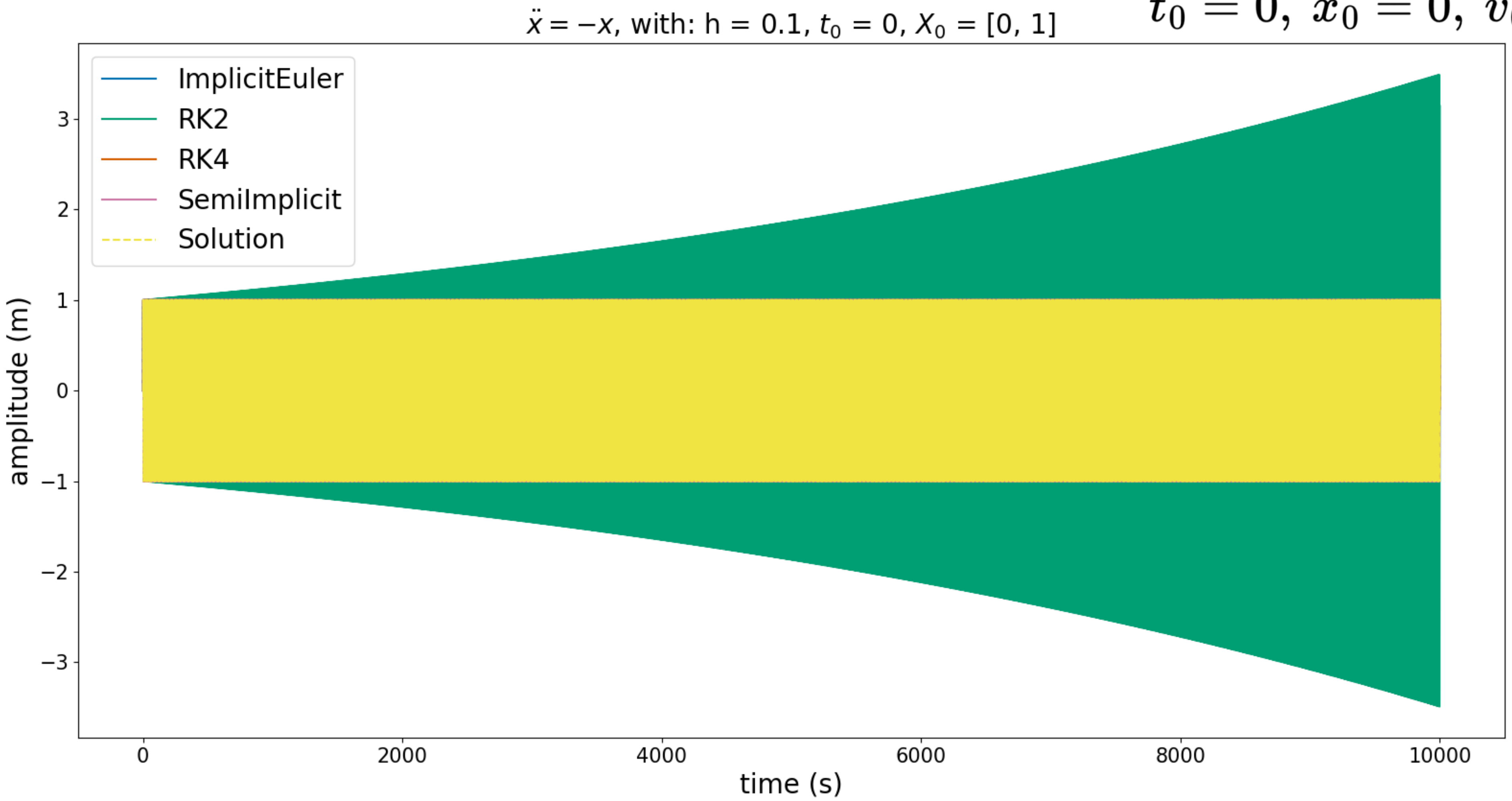
SPRING MASS $\ddot{x} = -x$

$$t_0 = 0,\ x_0 = 0,\ v_0 = 1m/s$$



SPRING MASS $\ddot{x} = -x$

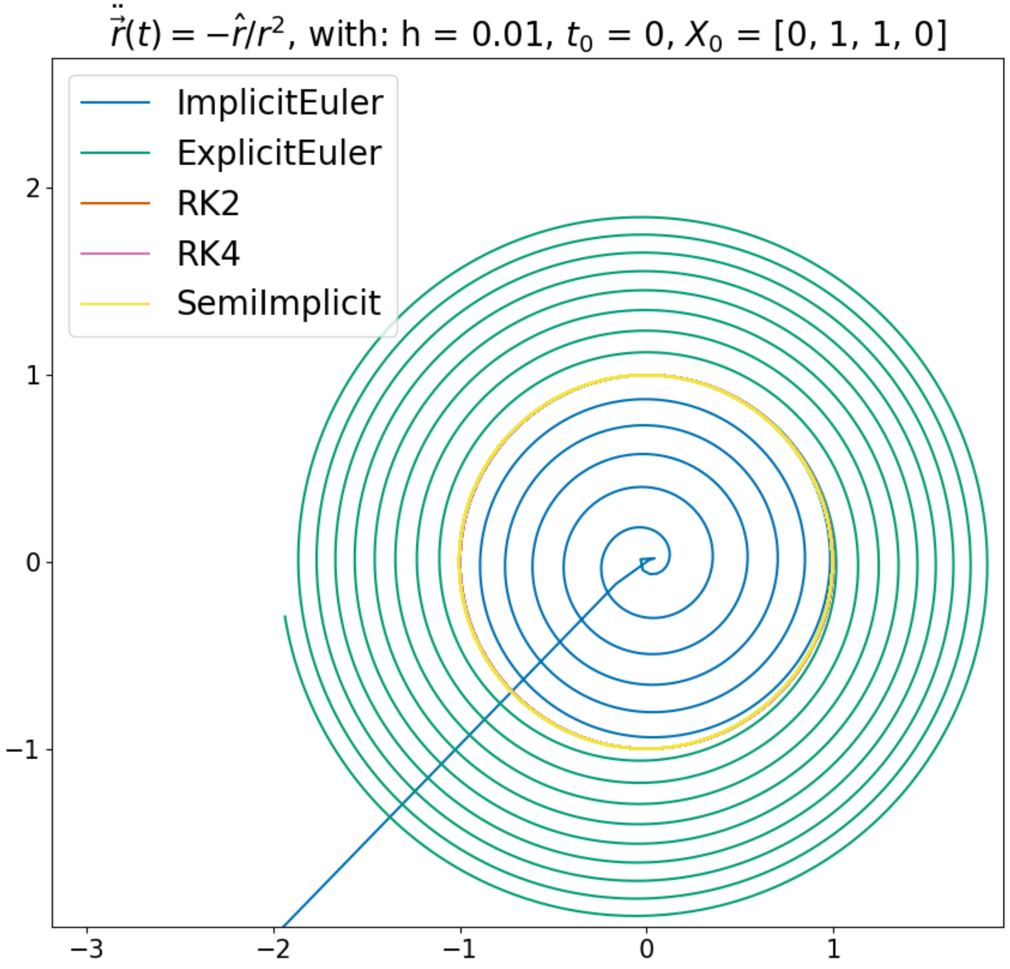
$t_0 = 0, \ x_0 = 0, \ v_0 = 1m/s$



$$\ddot{\vec{r}}(t) = \frac{-\hat{r}}{r^2}$$

ORBIT

$$\vec{r}(0) = \langle 0,\ 1 \rangle$$
$$\dot{\vec{r}}(0) = \langle 1,\ 0 \rangle$$

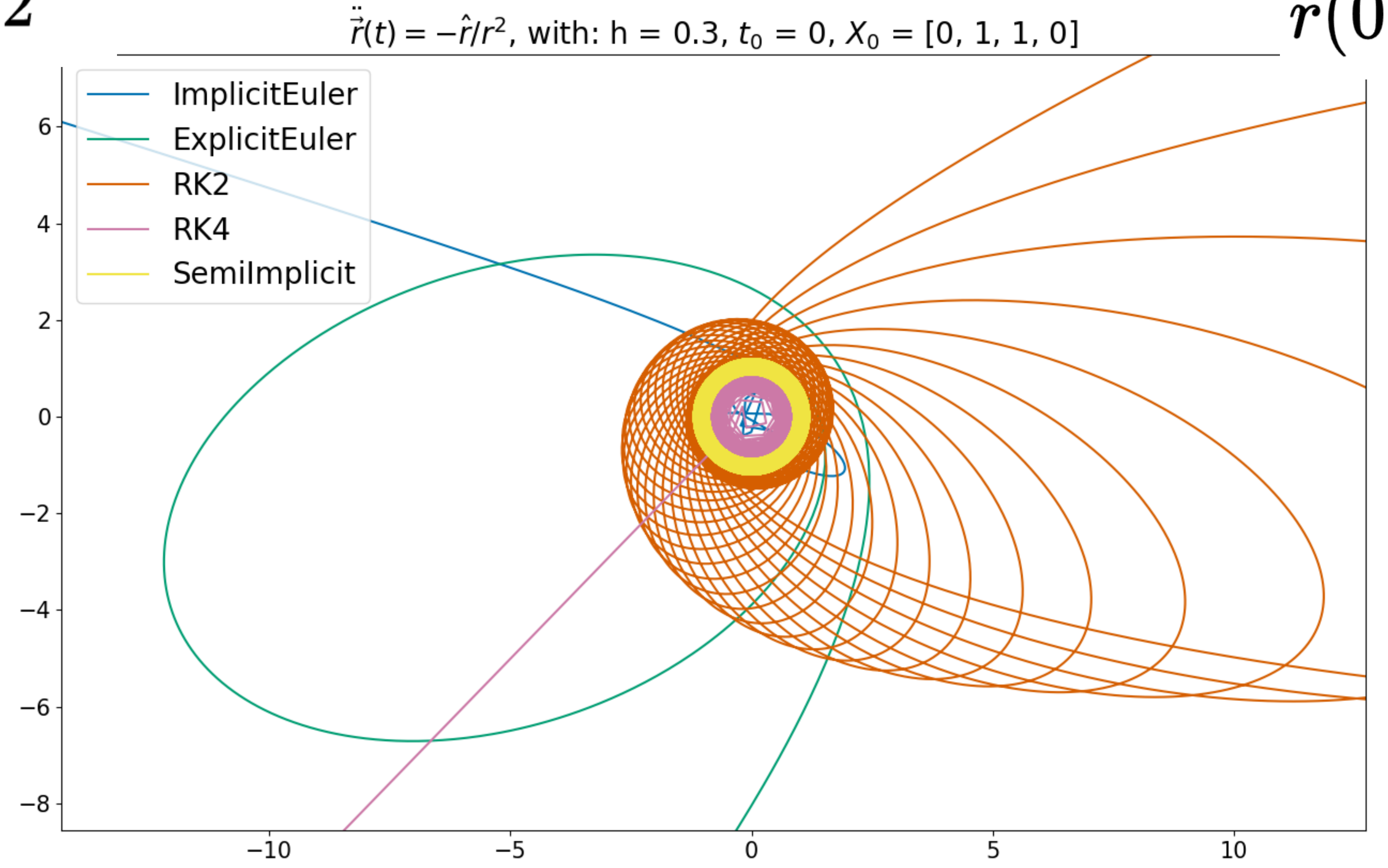


$$\ddot{\vec{r}}(t) = \frac{-\hat{r}}{r^2}$$

ORBIT

$$\vec{r}(0) = \langle 0,\ 1 \rangle$$

$$\dot{\vec{r}}(0) = \langle 1,\ 0 \rangle$$

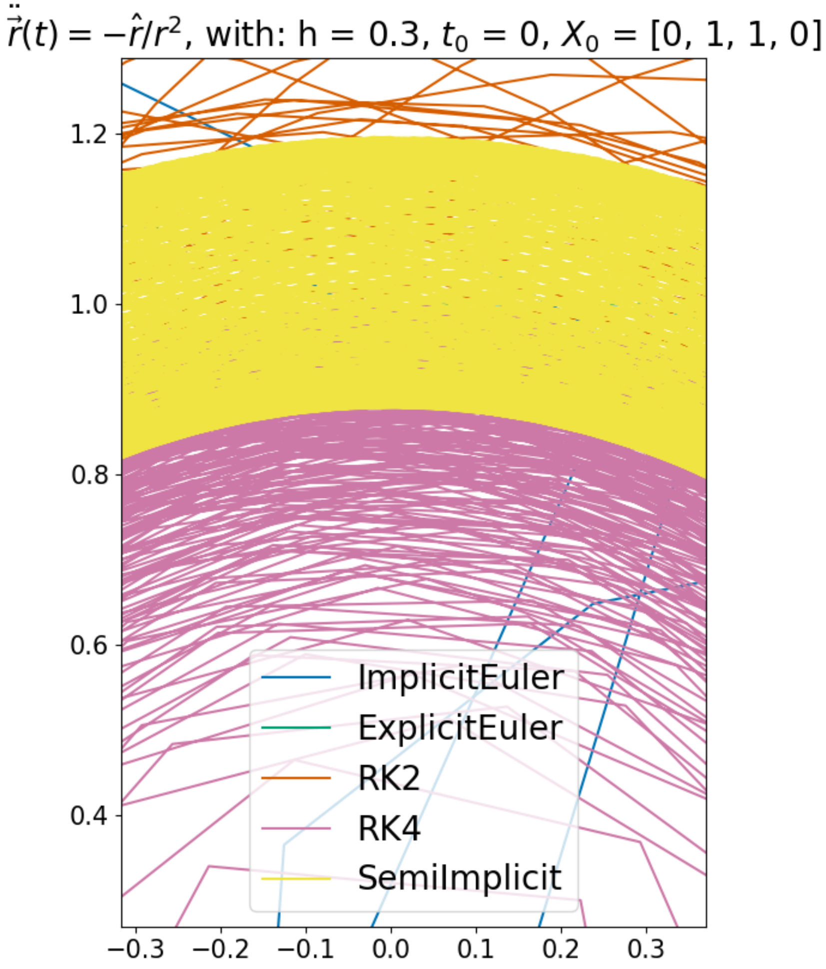


$$\ddot{\vec{r}}(t) = \frac{-\hat{r}}{r^2}$$

ORBIT

$$\vec{r}(0) = \langle 0,\ 1 \rangle$$

$$\dot{\vec{r}}(0) = \langle 1,\ 0 \rangle$$



Ok, so what should I use?

Semi Implicit Euler is probably good enough.

If you want more symplectic methods,
search up Verlet integration.