Introdução à Física Computacional II (4300318)

Prof. André Vieira
apvieira@if.usp.br
Sala 3120 – Edifício Principal

Aula 5

Reversão temporal e conservação da energia
Método *leapfrog*
Método de Verlet

Conservação da energia

- Até aqui nossa preocupação foi com melhorar a precisão de cada passo do cálculo, mas não verificamos se os erros ainda cometidos possuem viés.
- Por que temos que nos perguntar isso?

Conservação da energia

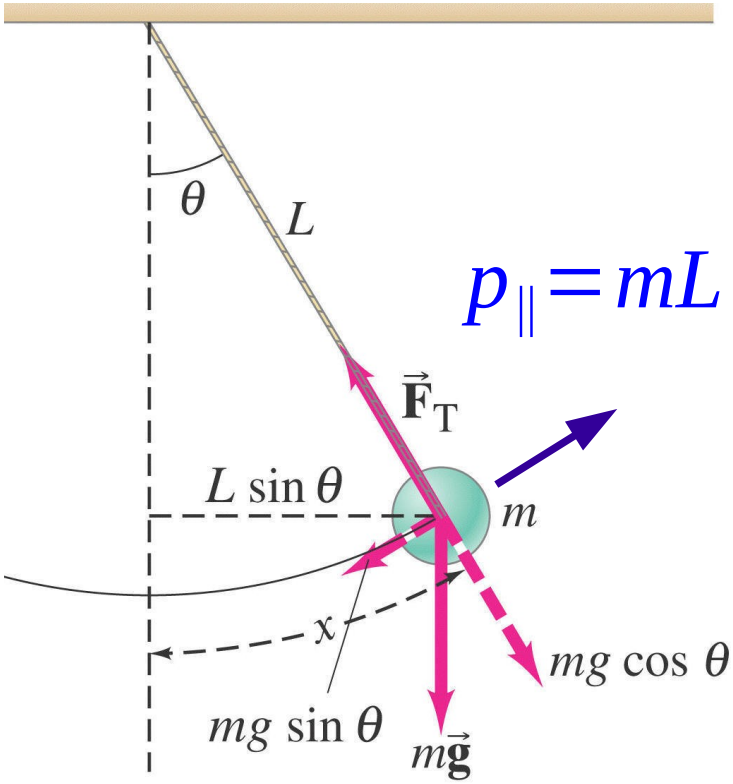
- Até aqui nossa preocupação foi com melhorar a precisão de cada passo do cálculo, mas não verificamos se os erros ainda cometidos possuem viés.
- Por que temos que nos perguntar isso? Porque a existência de viés tem implicações sobre princípios físicos fundamentais, entre eles as leis de conservação.

Exemplo 1: o pêndulo não linear

$$\frac{d\vec{p}_{\parallel}}{dt} = \vec{F}_{\parallel}$$

$$mL\frac{d^2\theta}{dt^2} = -mg\,\mathrm{sen}\,\theta$$

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\,\mathrm{sen}\,\theta$$



$$p_{\parallel} = mL\frac{d\theta}{dt}$$

$$F_{\parallel} = -mg\,\mathrm{sen}\,\theta$$

$$E = \frac{1}{2}m\left(L\frac{d\theta}{dt}\right)^2 + mgL\left(1-\cos\theta\right) = \text{constante}$$

Exemplo 1

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

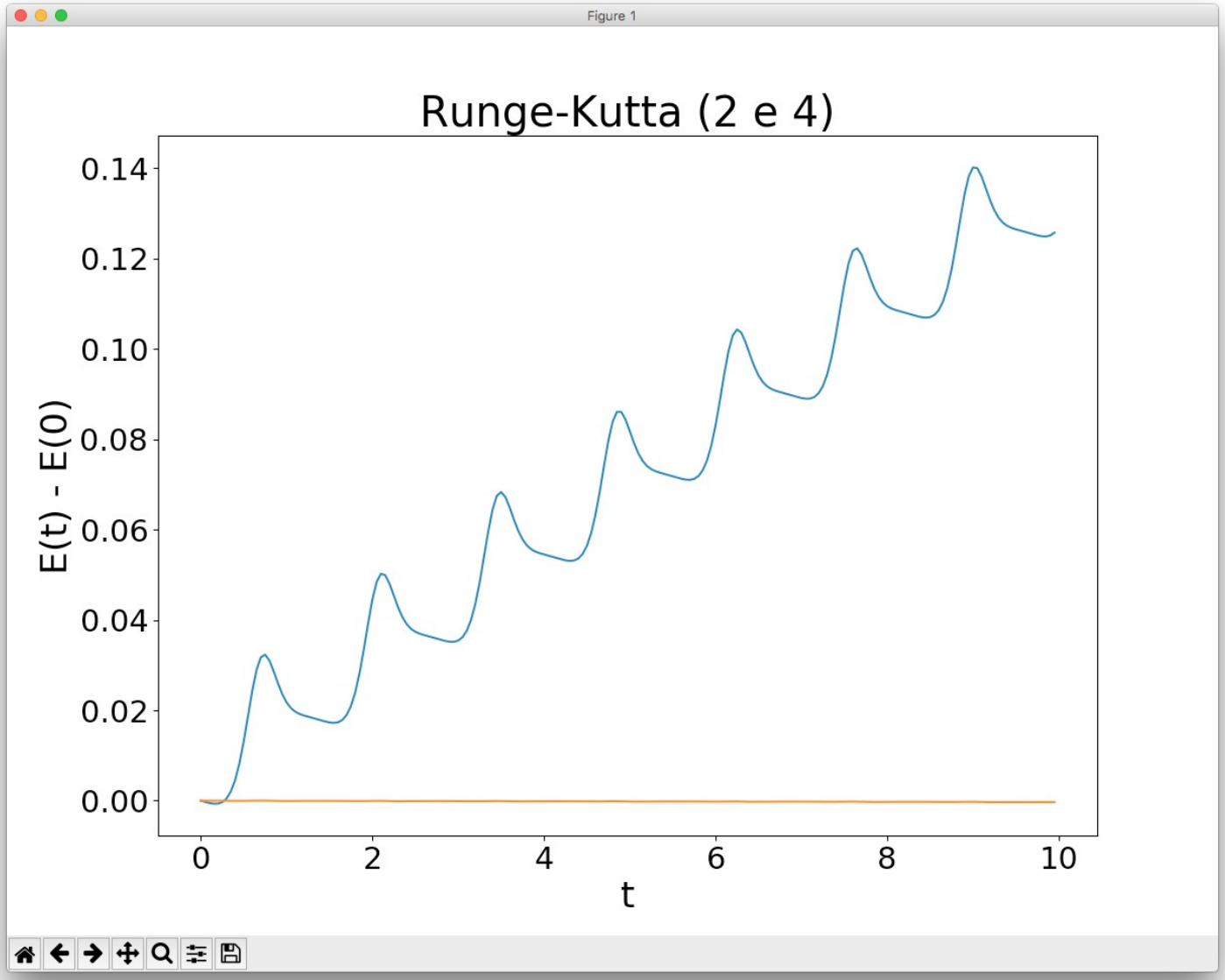
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



As curvas mostram a diferença entre a energia no instante *t* e a energia inicial como calculada pelos métodos de Runge–Kutta de segunda ordem e de quarta ordem.

Exemplo 1

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

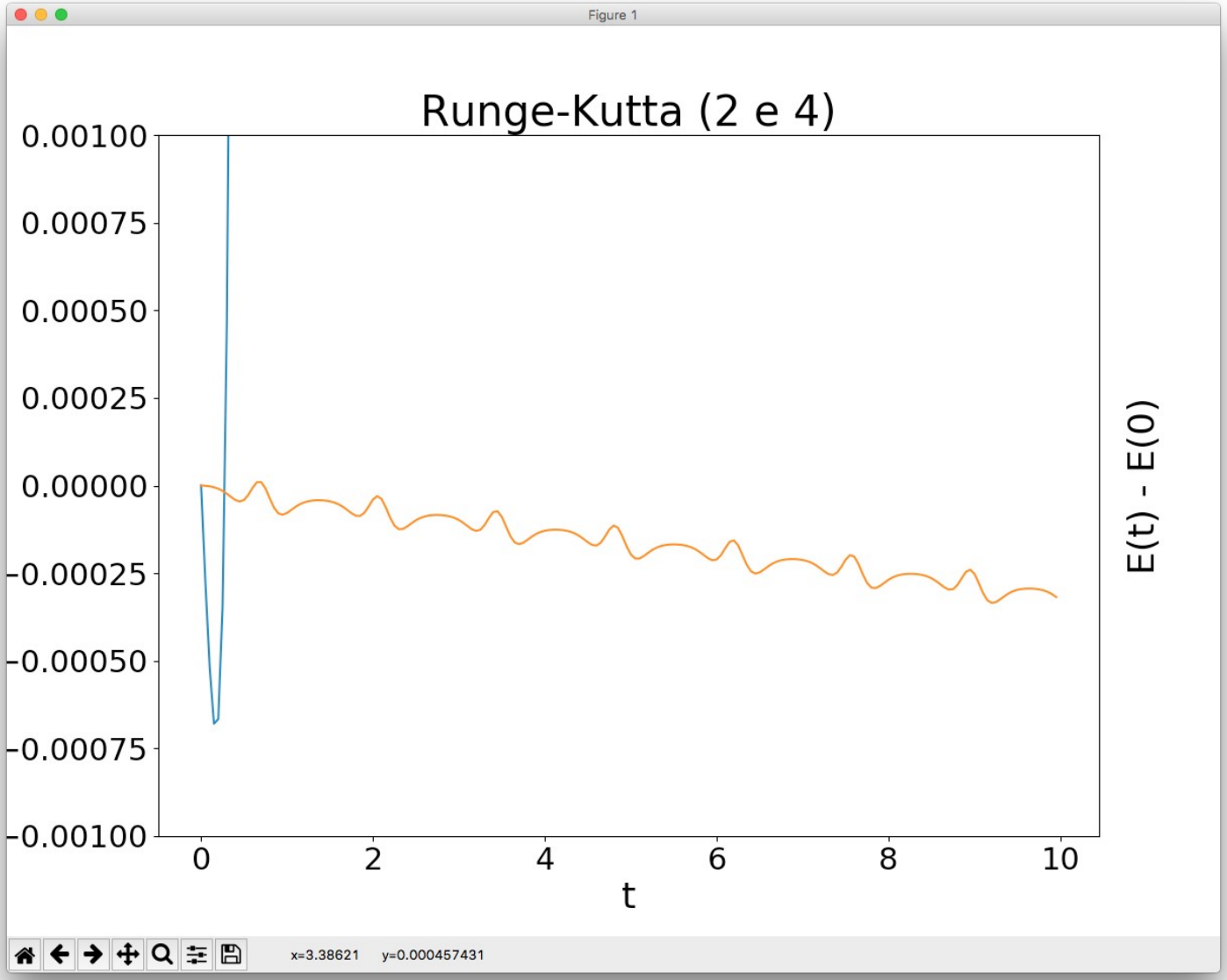
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



As curvas mostram a diferença entre a energia no instante *t* e a energia inicial como calculada pelos métodos de Runge–Kutta de segunda ordem e de quarta ordem.

Exemplo 1

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

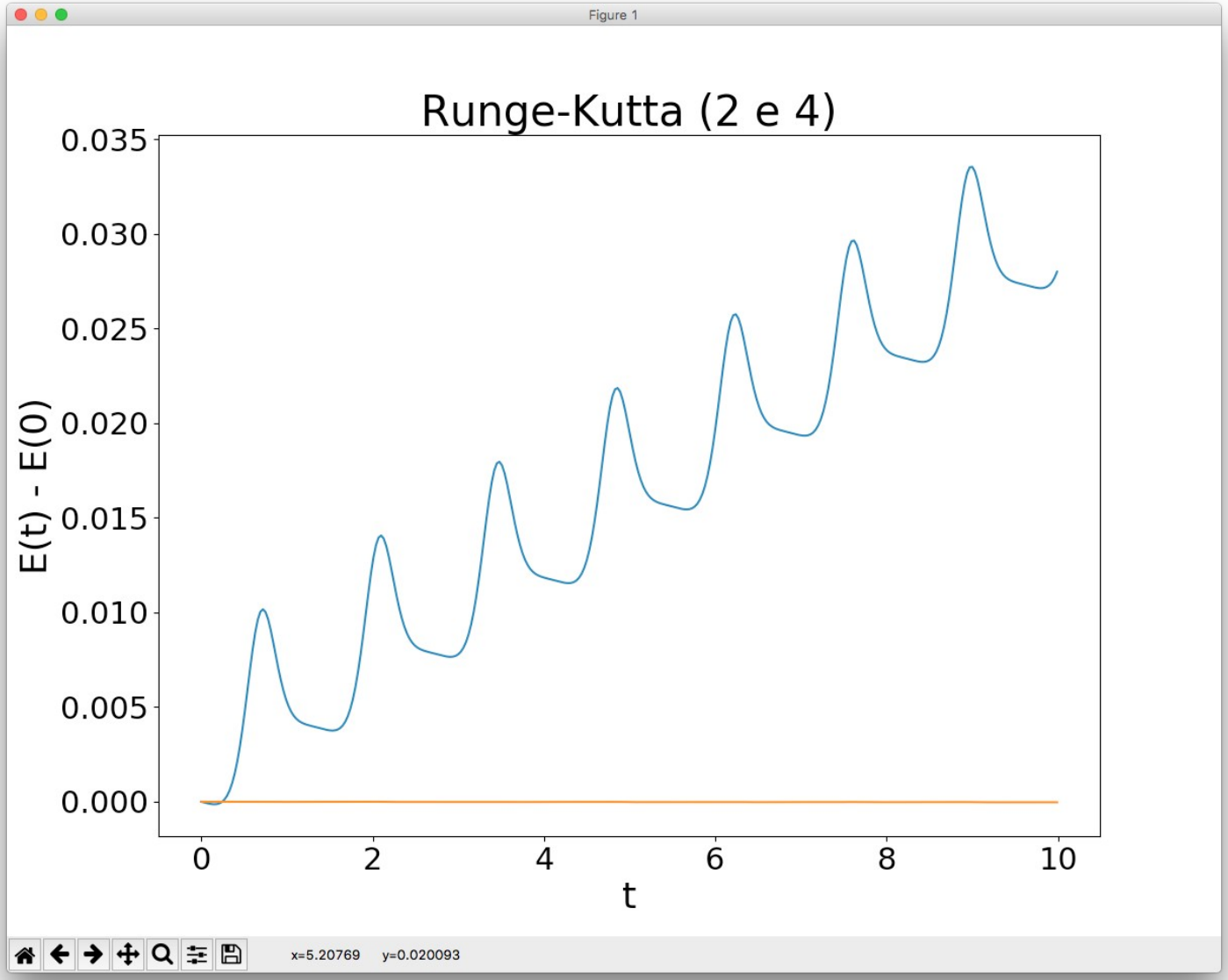
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.03$$



Diminuir o tamanho do passo de integração reduz a taxa de crescimento do erro na energia, mas não elimina esse erro.

Exemplo 1

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

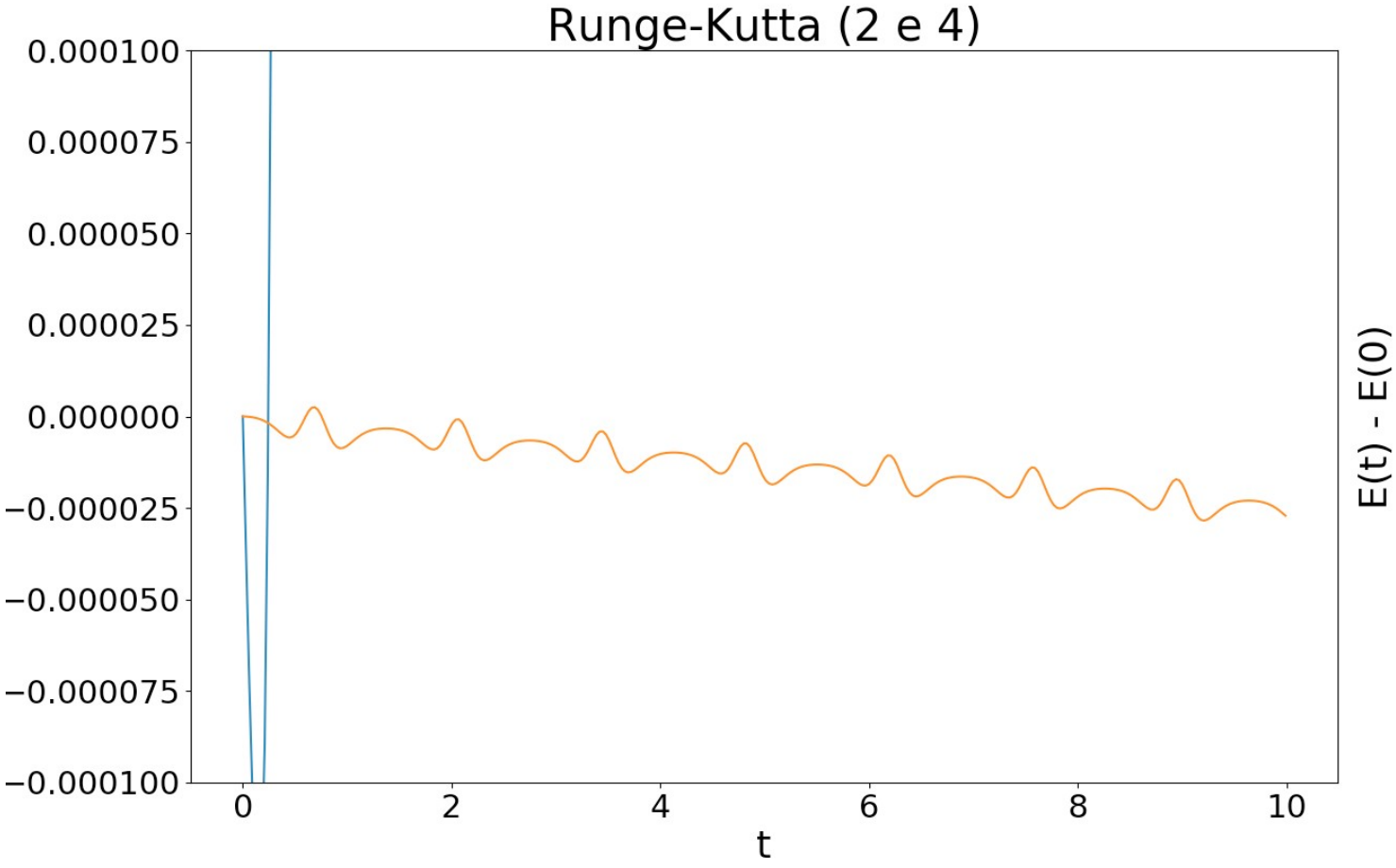
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.03$$



Diminuir o tamanho do passo de integração reduz a taxa de crescimento do erro na energia, mas não elimina esse erro.

O que está por trás do erro sistemático?

- Sistemas conservativos são invariantes por **reversão temporal**: se trocarmos t por $-t$, as equações de movimento permanecem as mesmas.

$$m\frac{d^2\vec{r}}{dt^2}=\vec{F}(\vec{r}) \quad \Rightarrow \quad m\frac{d^2\vec{r}}{d(-t)^2}=\vec{F}(\vec{r})$$

De onde vem o erro sistemático?

- Mas o que ocorre se trocarmos t por $-t$, no método de Runge–Kutta? Vejamos, por simplicidade, o método de segunda ordem.
  - Partindo de $x(t)$ em direção ao "futuro", temos:

$$k_+ = h f\left(x, t\right), \quad x\left(t+h\right) = x\left(t\right) + h f\left(x\left(t\right) + \tfrac{1}{2} k_+, t + \tfrac{1}{2} h\right)$$

$$\Rightarrow \quad x\left(t\right) = x\left(t+h\right) - h f\left(x\left(t\right) + \tfrac{1}{2} k_+, t + \tfrac{1}{2} h\right)$$

  - Partindo de $x(t+h)$ em direção ao "passado", temos:

$$k_- = -h f\left(x\left(t+h\right), t + \tfrac{1}{2} h\right)$$

$$x_-\left(t\right) = x\left(t+h\right) - h f\left(x\left(t+h\right) + \tfrac{1}{2} k_-, t + \tfrac{1}{2} h\right) \neq x\left(t\right)$$

Mesmo que *f* não dependa do tempo, os argumentos em verde e vermelho não são iguais, de modo que $x_-(t)$ e $x(t)$ são diferentes.

De onde vem o erro sistemático?

- Vamos fazer um teste explícito para o caso do pêndulo não linear do exemplo 1.
  - Partindo do repouso em $\theta(0)=1$ com $h=0.1$, temos:

$$\theta(h)=0.958768,\quad \dot{\theta}(h)=-0.824642$$

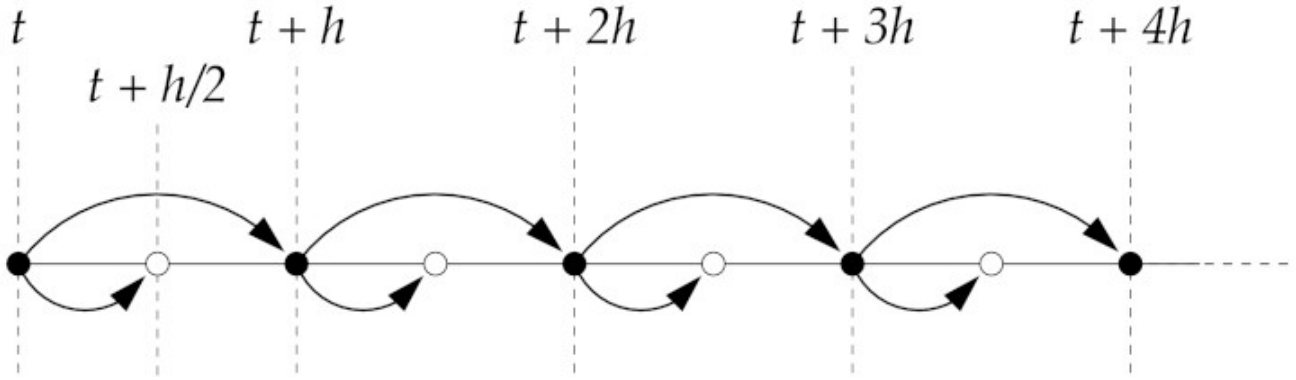
  - Partindo de $\theta(h)$, usando a velocidade angular calculada acima, em direção ao “passado” e com o mesmo passo de integração, temos:

$$\theta(0)=1.00113,\quad \dot{\theta}(0)=0.0$$

O fato de a velocidade angular inicial ser recuperada corretamente quando invertemos o tempo é restrito a termos partido do repouso. Veja o programa `teste_invtemp.py` na pasta de arquivos da aula no moodle. O programa também contém um teste com o algoritmo de quarta ordem.

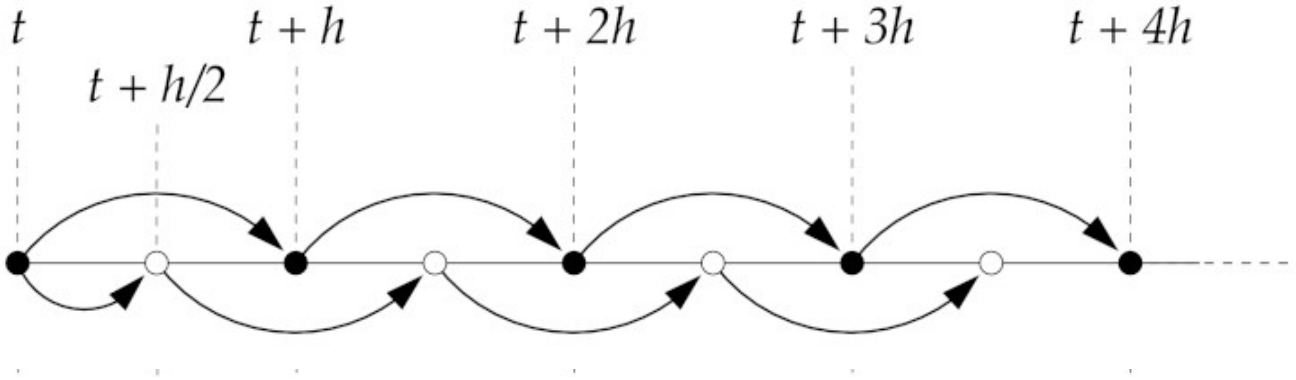
O algoritmo *leapfrog*

- O método de Runge–Kutta de segunda ordem pode ser representado graficamente pelo esquema da figura abaixo. Estimamos a derivada no ponto médio (círculos abertos) apenas a partir do resultado calculado no ponto “inteiro” (círculos fechados), e a utilizamos para estimar o próximo ponto “inteiro”.



O algoritmo *leapfrog*

- O método *leapfrog* é uma variante que pode ser representada graficamente pelo esquema da figura abaixo. Estimamos a derivada no ponto médio (círculos abertos) levando em conta o resultado calculado no ponto médio anterior, e continuamos a utilizá-la para estimar o próximo ponto “inteiro” (círculos fechados).



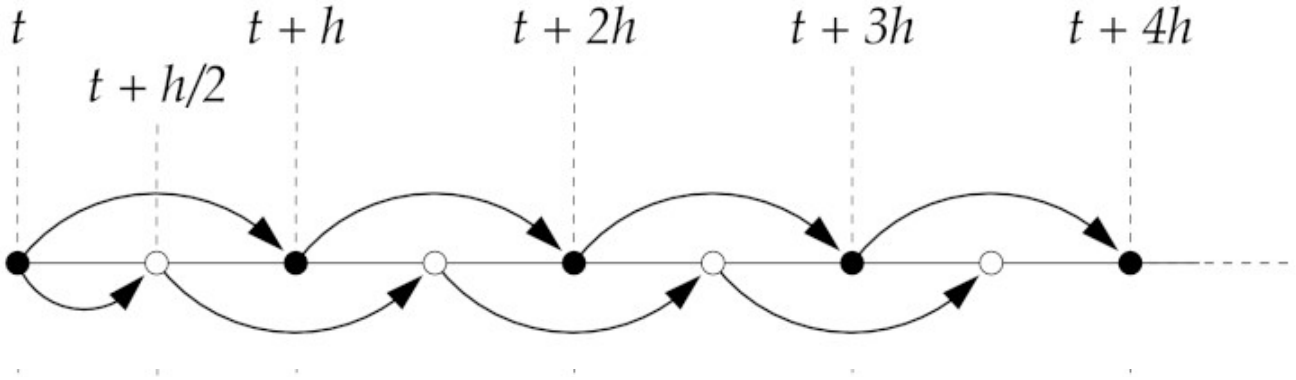
O algoritmo *leapfrog*

- Em termos de equações:

$$\vec{r}\left(t+h\right)=\vec{r}\left(t\right)+h\,\vec{f}\left(\vec{r}\left(t+\tfrac{1}{2}h\right),t+\tfrac{1}{2}h\right)$$

$$\vec{r}\left(t+\tfrac{3}{2}h\right)=\vec{r}\left(t+\tfrac{1}{2}h\right)+h\,\vec{f}\left(\vec{r}\left(t+h\right),t+h\right)$$

- Um passo tem erro cúbico em h, como no método RK2. E quanto à reversão temporal?



O algoritmo *leapfrog*

- Levando h em $-h$ nas equações:

$$\vec{r}\left(t-h\right)=\vec{r}\left(t\right)-h\vec{f}\left(\vec{r}\left(t-\tfrac{1}{2}h\right),t-\tfrac{1}{2}h\right)$$

$$\vec{r}\left(t-\tfrac{3}{2}h\right)=\vec{r}\left(t-\tfrac{1}{2}h\right)-h\vec{f}\left(\vec{r}\left(t-h\right),t-h\right)$$

- Levando agora t em $t+3h/2$, chegamos a

$$\vec{r}\left(t+\tfrac{1}{2}h\right)=\vec{r}\left(t+\tfrac{3}{2}h\right)-h\vec{f}\left(\vec{r}\left(t+h\right),t+h\right)$$

$$\vec{r}\left(t\right)=\vec{r}\left(t+h\right)-h\vec{f}\left(\vec{r}\left(t+\tfrac{1}{2}h\right),t+\tfrac{1}{2}h\right)$$

e rearranjando obtemos as mesmas equações da página anterior. Logo, o algoritmo *leapfrog* é invariante por reversão temporal.

O algoritmo *leapfrog*

- E quanto à conservação da energia? Vamos usar o pêndulo não linear para ilustrar uma propriedade do algoritmo, mas é possível formular um argumento mais geral, com base no caráter “simplético” do algoritmo, que emula uma propriedade da dinâmica newtoniana. Os curiosos podem querer consultar um texto (em inglês) disponível no moodle.

Exemplo 2

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1,\ L = 1$$

$$g = 9.8$$

$$h = 0.05$$

```
def f(r,t):
    theta, omega = r[0], r[1]
    return array([omega,-g/l*sin(theta)],float)

def passo_rk2(f,r,t,h):              # Calcula um passo no método de RK2
    k1 = h*f(r,t)
    k2 = h*f(r+0.5*k1,t+0.5*h)
    return k2                        # Retorna o INCREMENTO em r

def passo_lpf(f,r,rpm,t,h):          # Calcula um passo no método leapfrog
    r += h*f(rpm,t+0.5*h)            # Calculando valores nos pontos "inteiros"
    rpm += h*f(r,t+h)                # Calculando valores nos pontos "médios"
    return r, rpm                    # Retorna os NOVOS VALORES de r e rpm

a = 0.0              # Início do intervalo
b = 10.0             # Final do intervalo

theta_a, omega_a = 2*pi/3, 0.0                    # Condições iniciais
E_a = m*(l*omega_a)**2/2 + m*g*l*(1-cos(theta_a)) # Energia inicial


h = 5e-2       # Tamanho do passo de integração
theta_rk2, theta_lpf = [], []
E_rk2, E_lpf = [], []
t_lista = []
r_rk2 = array([theta_a,omega_a],float)
r_lpf = array([theta_a,omega_a],float)
t = 0
rpm = r_lpf + passo_rk2(f,r_lpf,t,h/2)  # Inicializando valores no ponto "médio"
while t<=b:    # Realizando as integrações numéricas
    t_lista.append(t)
    theta_rk2.append(r_rk2[0])
    theta_lpf.append(r_lpf[0])
    E_rk2.append(m*(l*r_rk2[1])**2/2 + m*g*l*(1-cos(r_rk2[0])) - E_a)
    E_lpf.append(m*(l*r_lpf[1])**2/2 + m*g*l*(1-cos(r_lpf[0])) - E_a)
    r_rk2 += passo_rk2(f,r_rk2,t,h)
    r_lpf, rpm = passo_lpf(f,r_lpf,rpm,t,h)
```

Note que inicializamos o algoritmo com meio passo RK2

Exemplo 2

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

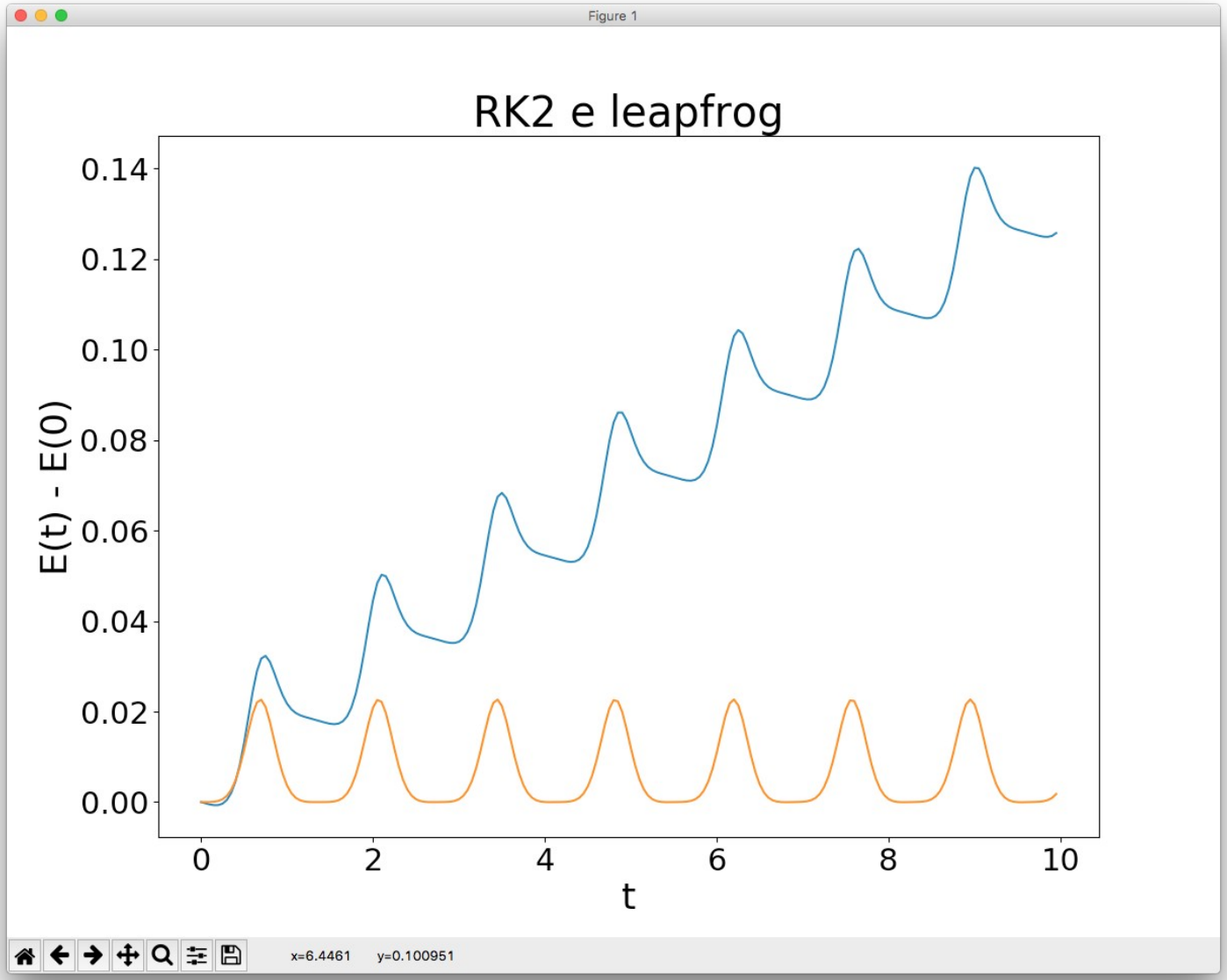
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



A curva laranja mostra a diferença entre a energia no tempo *t* e a energia inicial obtida pelo algoritmo *leapfrog*. A curva oscila entre valores máximo e mínimo, com o valor médio diferente de zero mas consistentemente constante.

Exemplo 2

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1,\ L = 1$$

$$g = 9.8$$

$$h = 0.05$$

E se utilizarmos exatamente o mesmo processo de integração, exceto por inicializarmos o algoritmo com um passo de Euler, em vez de um passo de Runge-Kutta de segunda ordem? Como mostra o próximo slide, a energia ainda é conservada “em média”, mas a amplitude das oscilações não é mais constante!

Exemplo 2

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

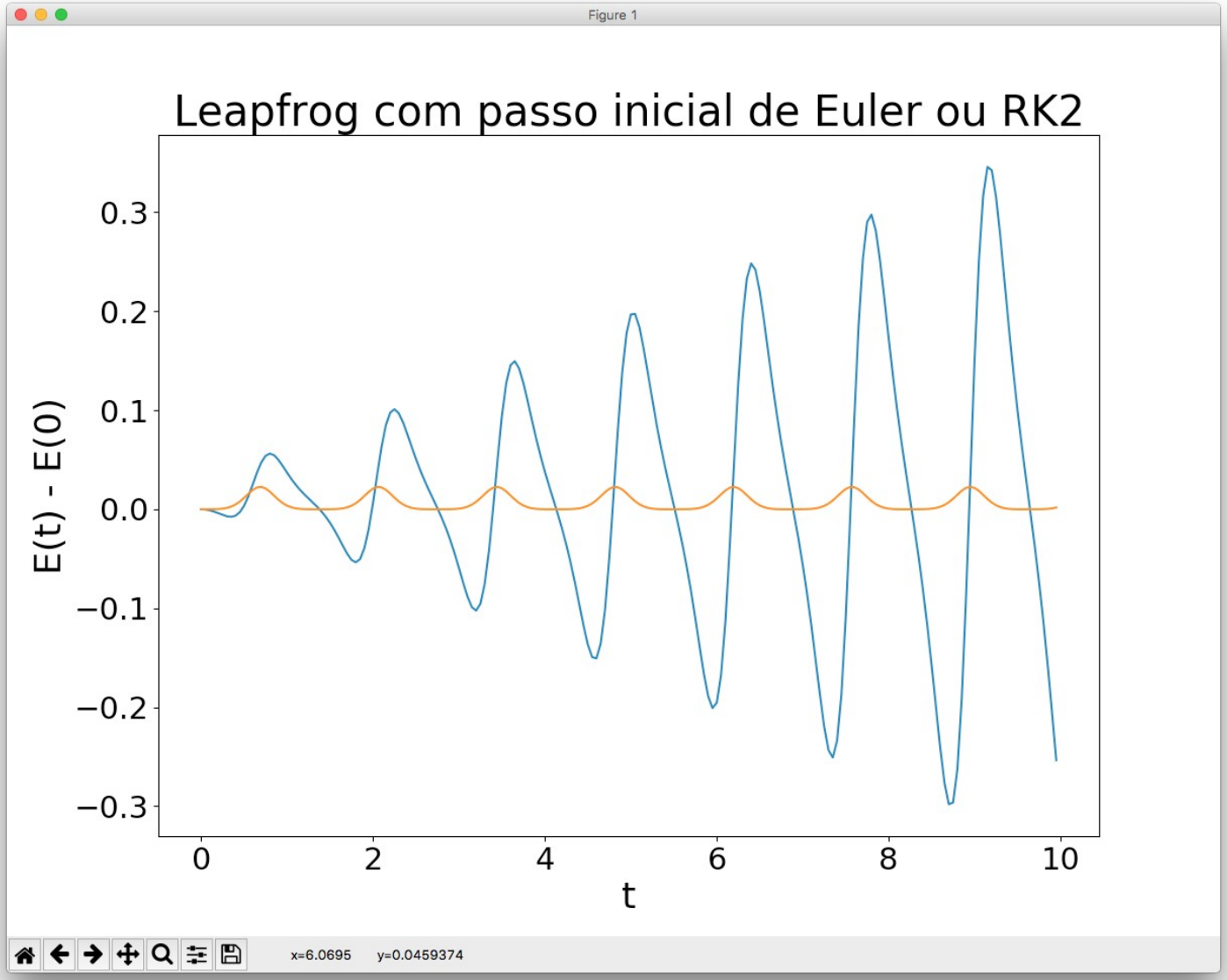
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



Diferença entre a energia no tempo *t* e a energia inicial, obtida pelo algoritmo *leapfrog* com um passo inicial de Euler (curva azul) ou de RK2 (curva laranja). Ambas as curvas oscilam em torno de um valor constante, mas a amplitude das oscilações cresce com o tempo se o passo inicial é de Euler. Vejam `exemplo2_teste.py`

O algoritmo de Verlet

- No caso específico de equações de 2ª ordem, podemos simplificar o algoritmo *leapfrog*.
- Considere um movimento em 1D com força **independente da velocidade**:

$$\frac{d^2x}{dt^2} = f(x,t) \quad \Rightarrow \quad \frac{dx}{dt} = v, \quad \frac{dv}{dt} = f(x,t)$$

- Uma implementação do *leapfrog* pode ser

$$x\left(t+h\right) = x\left(t\right) + h\,v\left(t+\tfrac{1}{2}h\right)$$

$$v\left(t+\tfrac{3}{2}h\right) = v\left(t+\tfrac{1}{2}h\right) + h\,f\left(x\left(t+h\right), t+h\right)$$

Note que a equação para x não depende de x no ponto “médio”, enquanto a equação para v não depende de v no ponto “inteiro”.

O algoritmo de Verlet

- E se precisarmos levar em conta a velocidade nos pontos “inteiros”, para, por exemplo, calcular a energia?
- Modificamos o algoritmo ligeiramente, usando meio passo de Euler (para trás!) para determinar a velocidade em um ponto inteiro. Isso define o **algoritmo de Verlet**:

$$x\left(t+h\right)=x\left(t\right)+h\,v\left(t+\tfrac{1}{2}h\right)$$

$$k=h\,f\left(x\left(t+h\right),t+h\right)$$

$$v\left(t+h\right)=v\left(t+\tfrac{1}{2}h\right)+\tfrac{1}{2}k$$

$$v\left(t+\tfrac{3}{2}h\right)=v\left(t+\tfrac{1}{2}h\right)+k$$

A equação em vermelho segue de rearranjar o passo “para trás” descrito por

$$v\left(t+\tfrac{1}{2}h\right)=v\left(t+h\right)-\tfrac{1}{2}h\,f\left(x\left(t+h\right),t+h\right)$$

O algoritmo de Verlet

- Generalizando para o caso do movimento em mais dimensões, o algoritmo de Verlet se escreve como

$$\frac{d^2\vec{r}}{dt^2}=\vec{f}\left(\vec{r},t\right) \;\Rightarrow\; \frac{d\vec{r}}{dt}=\vec{v},\; \frac{d\vec{v}}{dt}=\vec{f}\left(\vec{r},t\right)$$

$$\vec{r}\left(t+h\right)=\vec{r}\left(t\right)+h\vec{v}\left(t+\tfrac{1}{2}h\right)$$
$$\vec{k}=h\vec{f}\left(\vec{r}\left(t+h\right),t+h\right)$$
$$\vec{v}\left(t+h\right)=\vec{v}\left(t+\tfrac{1}{2}h\right)+\tfrac{1}{2}k$$
$$\vec{v}\left(t+\tfrac{3}{2}h\right)=\vec{v}\left(t+\tfrac{1}{2}h\right)+\vec{k}$$

- Inicializamos com posição e velocidade em t=0 e

$$\vec{v}\left(\tfrac{1}{2}h\right)=\vec{v}\left(0\right)+\tfrac{1}{2}h\vec{f}\left(\vec{r}\left(0\right),0\right)$$

Exemplo 3

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1,\ L = 1$$

$$g = 9.8$$

$$h = 0.05$$

```
def f(theta,t):
    return -g/l*sin(theta)

def passo_vrl(f,r,v,vpm,t,h):    # Calcula um passo no método de Verlet
    r += h*vpm                   # Calculando a posição no próximo t "inteiro"
    k = h*f(r,t+h)
    v = vpm + 0.5*k              # Calculando a velocidade no próximo t "inteiro"
    vpm += k                     # Calculando a velocidade no próximo t "médio"
    return r, v, vpm             # Retorna os NOVOS VALORES de r, v e vpm

a = 0.0              # Início do intervalo
b = 10.0             # Final do intervalo

theta_a, omega_a = 2*pi/3, 0.0                      # Condições iniciais
E_a = m*(l*omega_a)**2/2 + m*g*l*(1-cos(theta_a))   # Energia inicial


h = 5e-2        # Tamanho do passo de integração
theta_vrl, omega_vrl = [], []
E_vrl = []
t_lista = []
r_vrl, v_vrl = theta_a, omega_a
t = 0
vpm = v_vrl + 0.5*h*f(r_vrl,t)  # Inicializando valores no ponto "médio"
while t<= b:    # Realizando a integração numérica
    t_lista.append(t)
    theta_vrl.append(r_vrl)
    omega_vrl.append(v_vrl)
    E_vrl.append(m*(l*v_vrl)**2/2 + m*g*l*(1-cos(r_vrl)) - E_a)
    r_vrl, v_vrl, vpm = passo_vrl(f,r_vrl,v_vrl,vpm,t,h)
    t += h
```

Exemplo 3

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

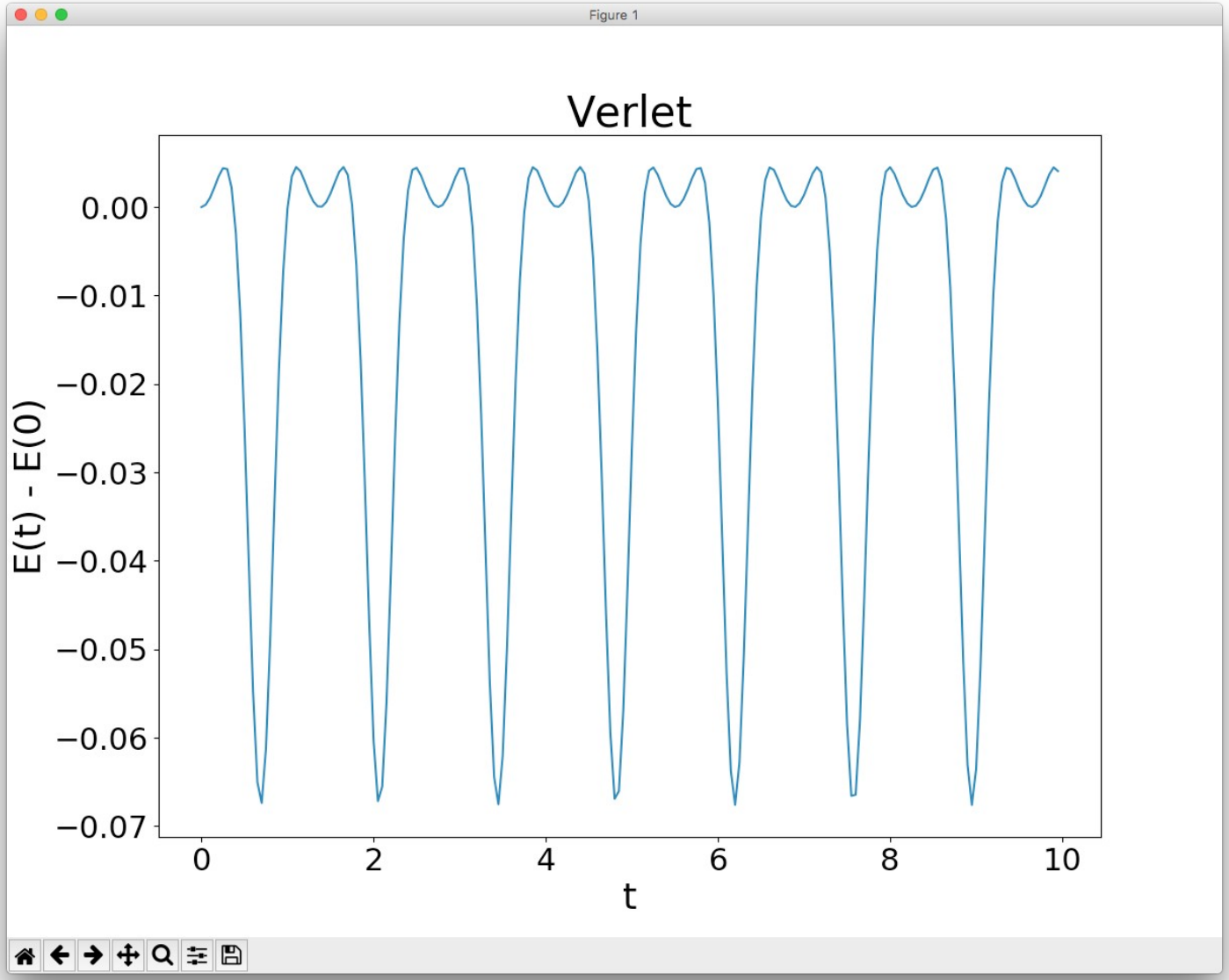
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



Assim como o algoritmo *leapfrog*, o algoritmo de Verlet conserva a energia em média. Isso e seu baixo custo computacional o fazem ser bastante utilizado em cálculos do movimento de muitas partículas em sistemas conservativos.

Exemplo 3

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

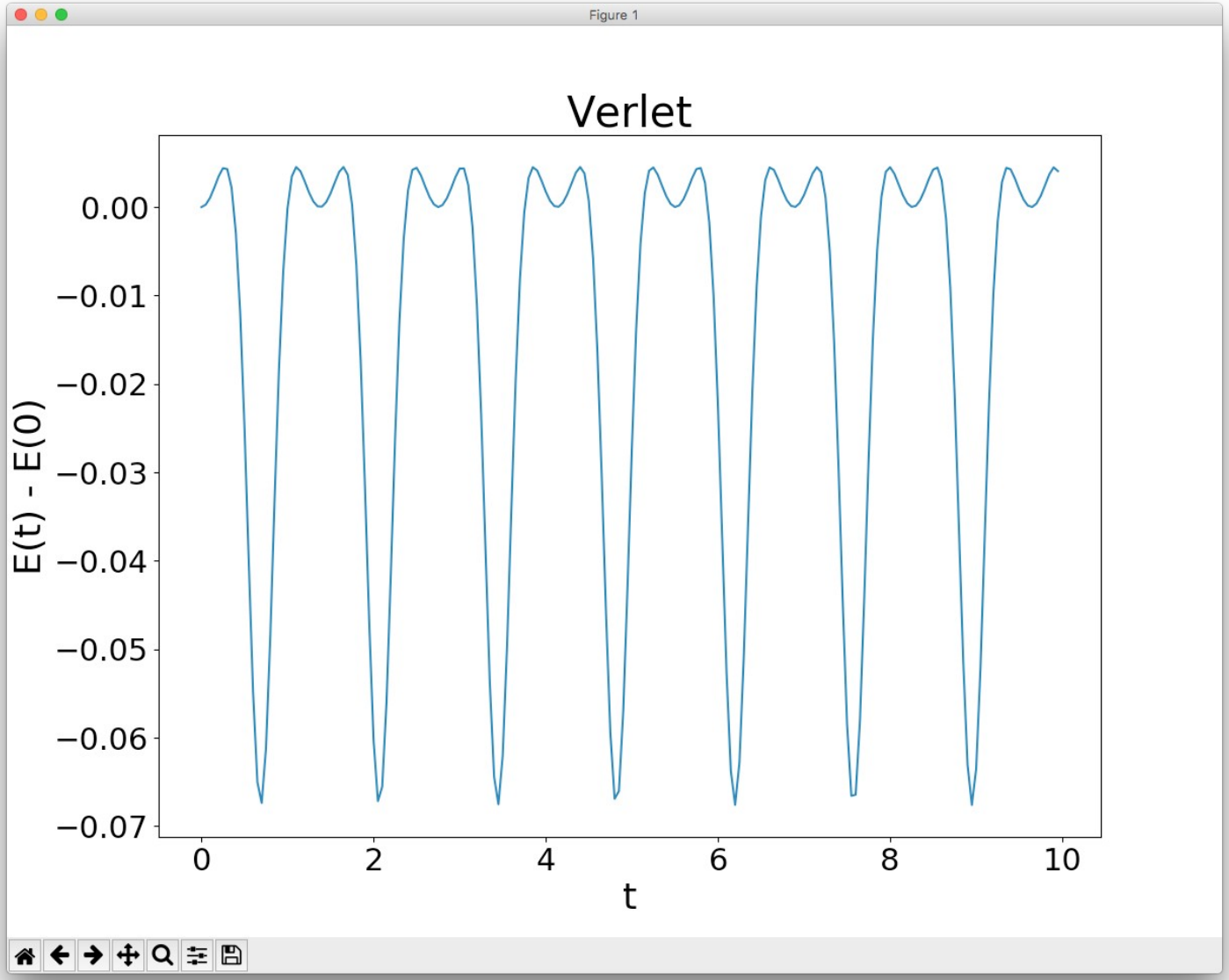
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



Ao contrário do que verificamos com o algoritmo *leapfrog*, a curva de energia prevista pelo algoritmo de Verlet não muda substancialmente se o inicializarmos com um passo do algoritmo de Euler (como nesse exemplo) ou de RK2 (veja `exemplo3_teste.py`)

Exemplo 3

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

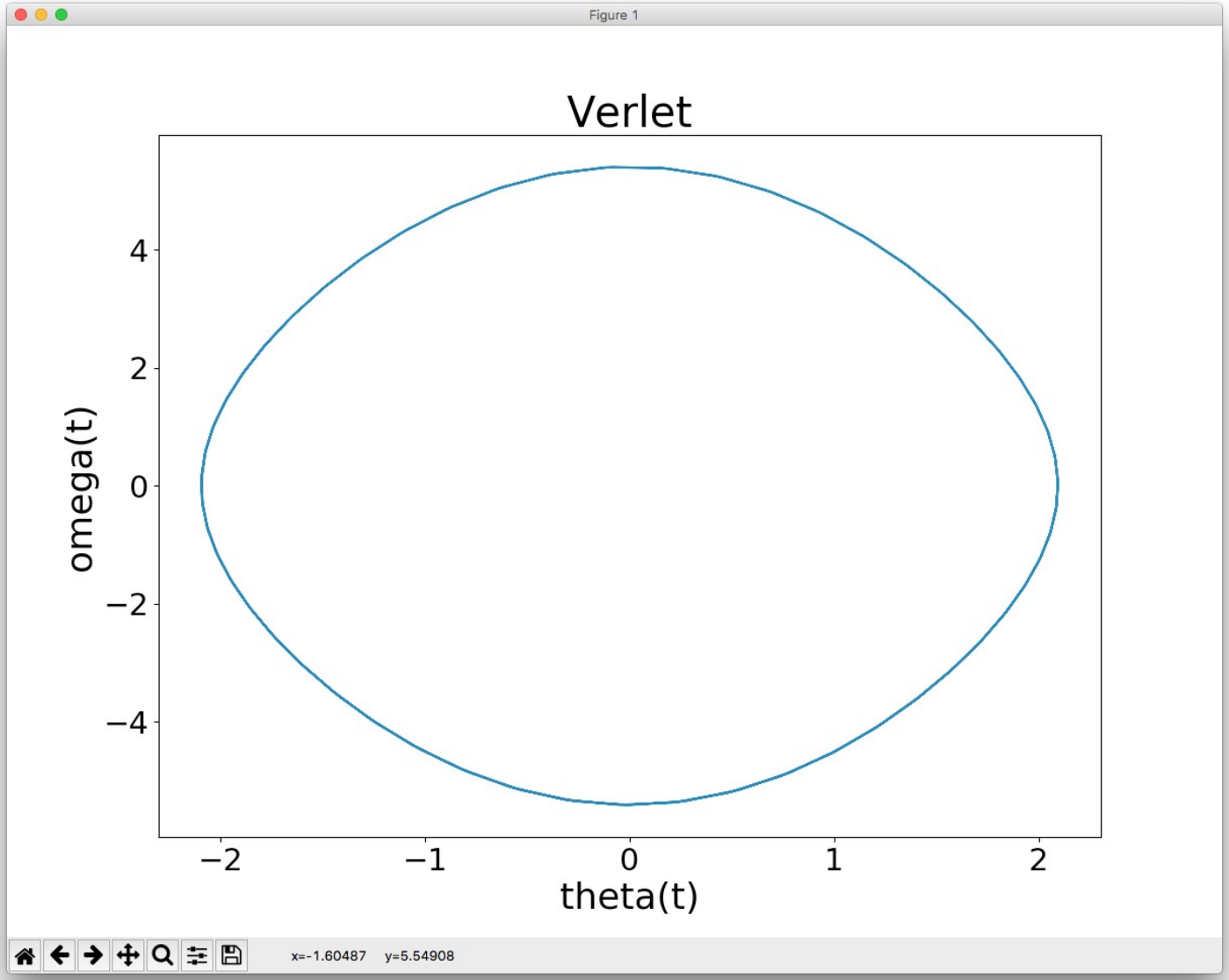
$$\theta(0) = \frac{2\pi}{3},$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

$$h = 0.05$$



Um gráfico do espaço de fase, $\omega(t) \times \theta(t)$, reforça que o algoritmo de Verlet mantém o movimento próximo de uma curva de energia constante. Mas como fica o erro associado ao método?

Exemplo 3

$$\frac{d^2\theta}{dt^2} = -\frac{g}{L}\operatorname{sen}\theta$$

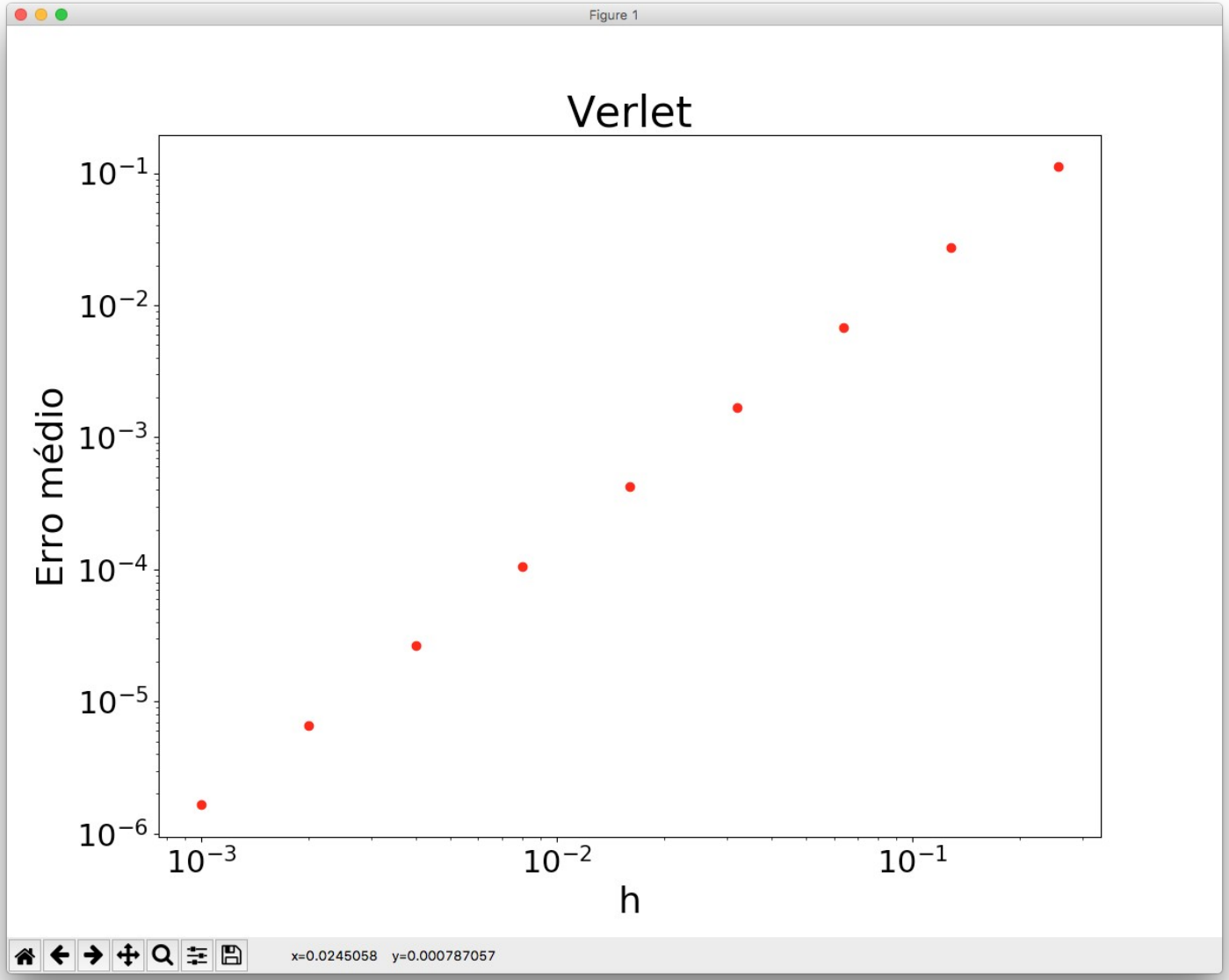
$$\theta(0) = \frac{\pi}{10} \ll 1,$$

$$\dot{\theta}(0) = 0$$

$$m = 1, \quad L = 1$$

$$g = 9.8$$

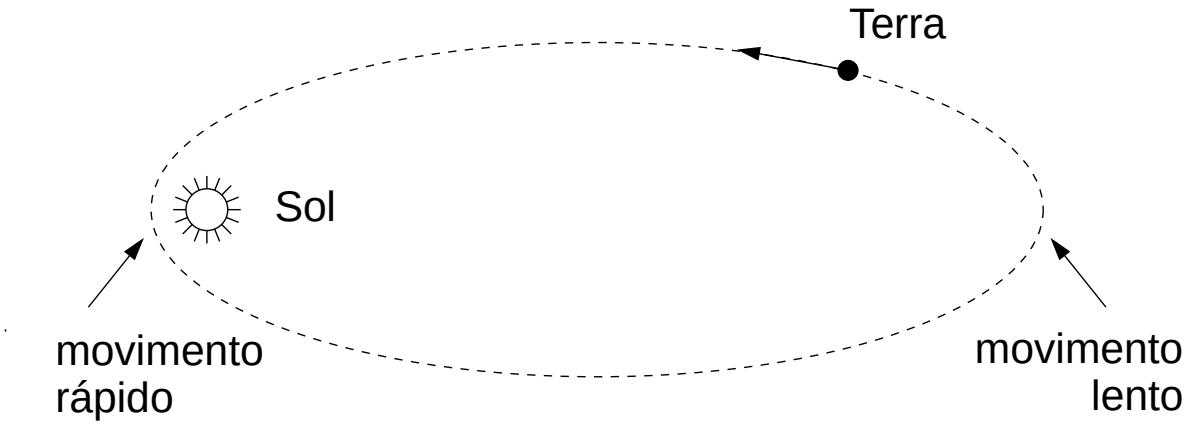
$$h = 0.05$$



Assim como o método de Runge–Kutta de segunda ordem, o erro associado a um passo do algoritmo de Verlet (e do algoritmo leapfrog) é cúbico em h, o que significa que o erro acumulado é quadrático em h.

Exemplo 4: órbita da Terra

Vamos considerar o movimento de translação da Terra, de massa m, em órbita elíptica em torno do Sol, de massa M. Os métodos de passo adaptativo que discutimos na aula anterior não são invariantes por reversão temporal, e portanto não conservam a energia em média. Vamos agora utilizar o algoritmo de Verlet para determinar o movimento. **(A elipse no desenho está grandemente exagerada!)**



$$m\frac{d^2\vec{r}}{dt^2} = -\frac{GmM}{|\vec{r}|^2}\hat{r}$$

Movimento no plano xy:

$$\frac{d^2x}{dt^2} = -GM\frac{x}{r^3}, \quad \frac{d^2y}{dt^2} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

Exemplo 4: órbita da Terra

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

$$\frac{dv_x}{dt} = -GM\frac{x}{r^3}$$

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

$$\frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

$$h = 30\,\text{dias}$$

```
# Limites e passo de integração
a = 0.                      # Instante inicial em segundos
b = 16*365*(60*60*24)       # Instante final: 16 anos (em segundos)
h = 30*24*60*60             # Passo de tempo: 30 dias (em segundos)

# Condições inicias (supondo o Sol na origem)
r_a = array([-1.4710e11,0.,0.]) # Posição da Terra no periélio
v_a = array([0.,-3.0287e4,0.0]) # Velocidade da Terra no periélio

def f(r,t):                     # Força/massa atuando sobre a esfera
    magr = linalg.norm(r)
    return -GM*r/magr**3

def passo_vrl(f,r,v,vpm,t,h):   # Calcula um passo no método de Verlet
    r += h*vpm                  # Calculando a posição no próximo t "inteiro"
    k = h*f(r,t+h)
    v = vpm + 0.5*k             # Calculando a velocidade no próximo t "inteiro"
    vpm += k                    # Calculando a velocidade no próximo t "médio"
    return r, v, vpm            # Retorna os NOVOS VALORES de r, v e vpm

t = 0
r_vrl, v_vrl = r_a, v_a
vpm = v_vrl + 0.5*h*f(r_vrl,t)  # Inicializando valores no ponto "médio"
t_lista, K_lista, Ug_lista, E_lista = [], [], [], []
x_lista, y_lista = [], []
while t<= b:   # Realizando a integração numérica
    K = 0.5*m*linalg.norm(v_vrl)**2
    U_g = -GMm/linalg.norm(r_vrl)
    t_lista.append(t)
    K_lista.append(K)
    Ug_lista.append(U_g)
    E_lista.append(K+U_g)
    x_lista.append(r_vrl[0])
    y_lista.append(r_vrl[1])
    r_vrl, v_vrl, vpm = passo_vrl(f,r_vrl,v_vrl,vpm,t,h)
    t += h
```

Exemplo 4: órbita da Terra

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

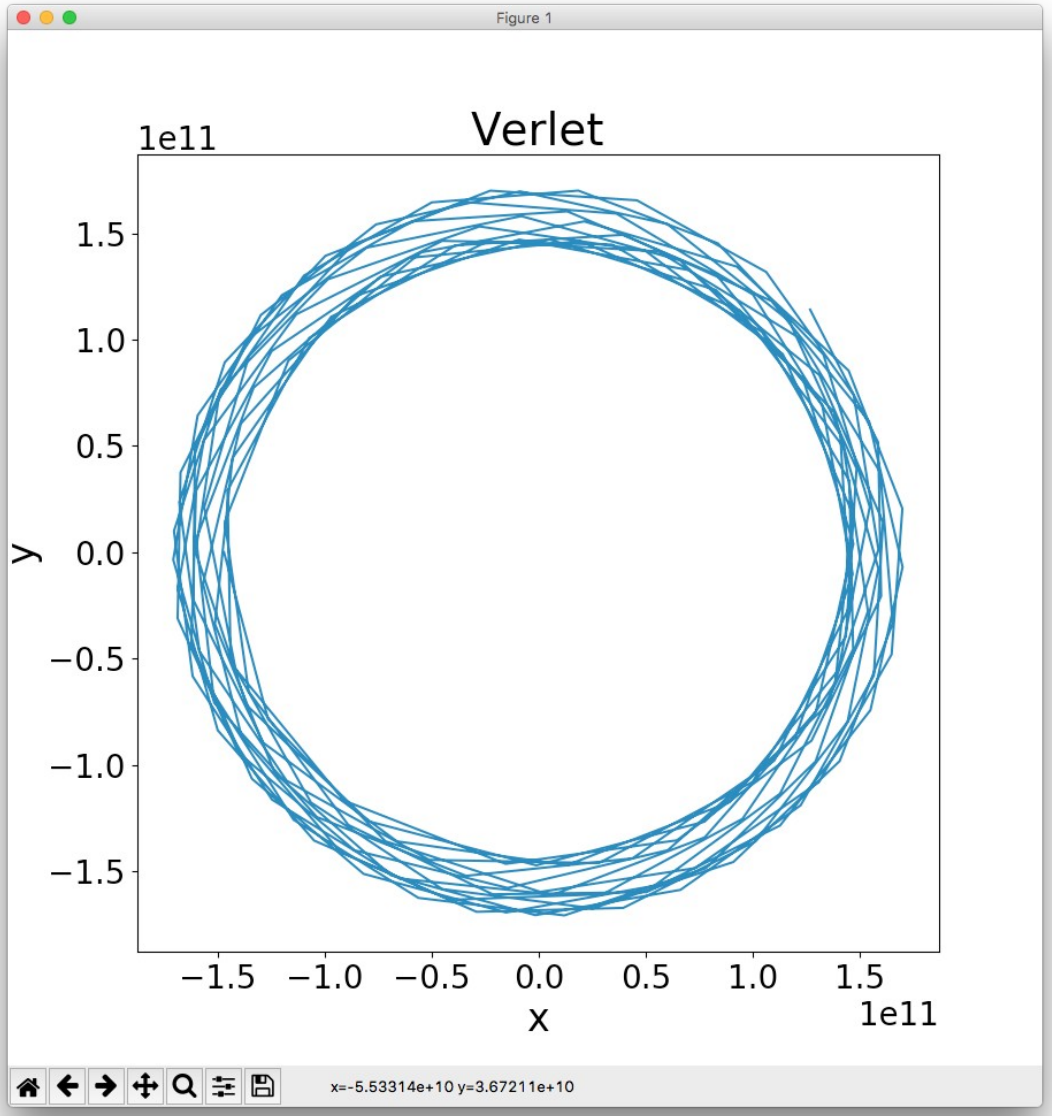
$$\frac{dv_x}{dt} = -GM\frac{x}{r^3}$$

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

$$\frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

$$h = 30\,\text{dias}$$



Note que utilizamos um passo de tempo muito longo, mas estamos executando o programa pelo equivalente a 16 anos. Embora a órbita não seja fechada como deveria, ela permanece em um intervalo estreito de distância ao Sol.

Exemplo 4: órbita da Terra

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

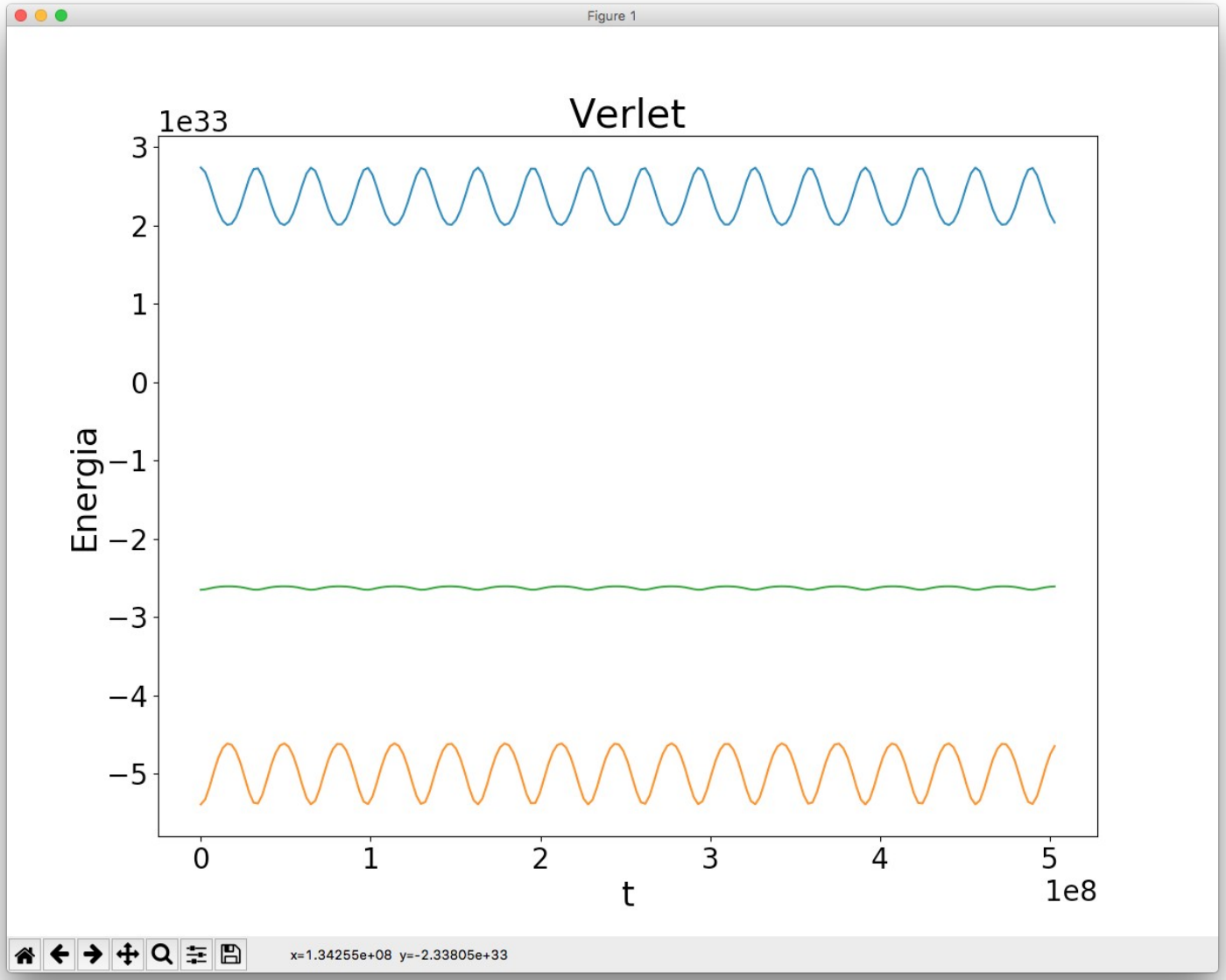
$$\frac{dv_x}{dt} = -GM\frac{x}{r^3}$$

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

$$\frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

$$h = 30\,\text{dias}$$



Como consequência, mesmo o algoritmo de Verlet tendo um erro acumulado proporcional ao quadrado do passo de tempo, a energia mecânica oscila em torno de uma reta horizontal. O gráfico mostra também a energia cinética e a energia potencial.

Exemplo 4: órbita da Terra

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

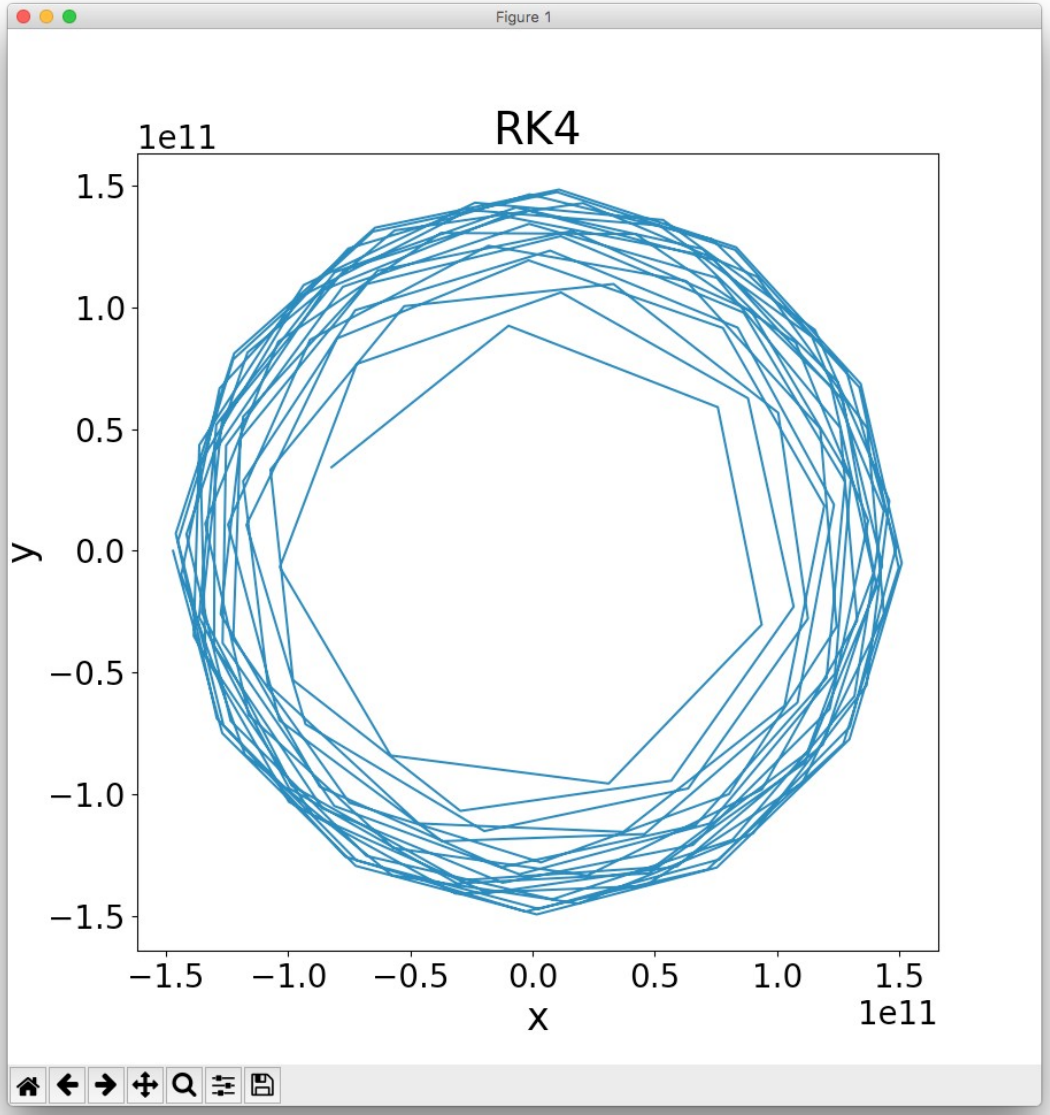
$$\frac{dv_x}{dt} = -GM\frac{x}{r^3}$$

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

$$\frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

$$h = 30\,\text{dias}$$



Com exatamente os mesmos parâmetros e condições iniciais, mas utilizando para a integração numérica o método de Runge-Kutta de quarta ordem, a trajetória espirala em direção ao Sol.

Exemplo 4: órbita da Terra

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

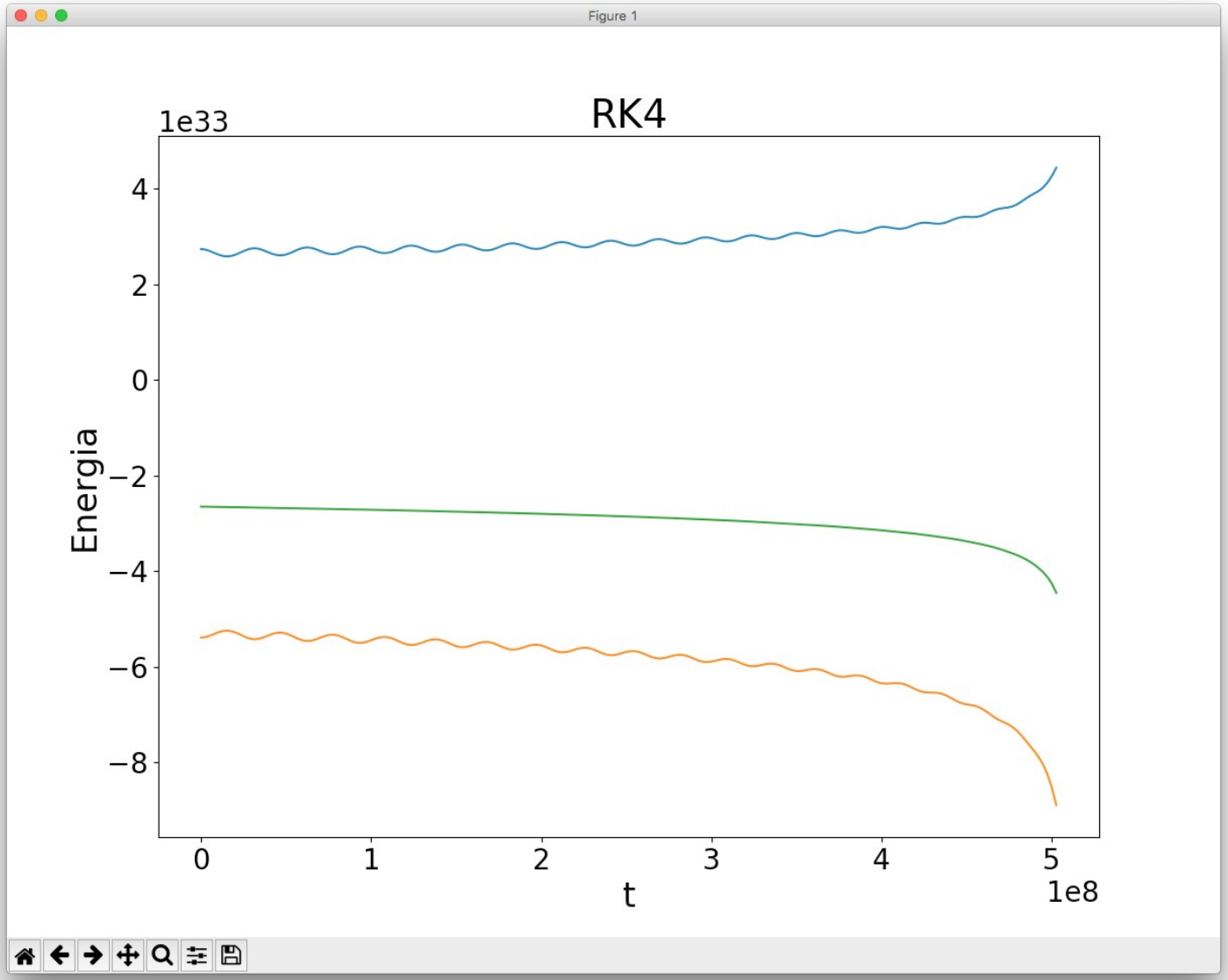
$$\frac{dv_x}{dt} = -GM\frac{x}{r^3}$$

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

$$\frac{dv_y}{dt} = -GM\frac{y}{r^3}$$

$$r = \sqrt{x^2 + y^2}$$

$$h = 30\,\text{dias}$$



A energia mecânica decresce ao longo do tempo. O gráfico mostra também a energia cinética e a energia potencial.

Exercícios no moodle

- São 2 exercícios, explorando o conteúdo da aula de hoje, que podem ser feitos com base nos programas dos exemplos, que estão disponíveis no moodle. Os exercícios incluem figuras que mostram os resultados esperados, para que você possa verificar se seus códigos estão corretos. O segundo exercício serve de bônus para o EP3.

A data para entrega é o dia **22 de abril**.