

# Ciência Computacional: Modelagem e Simulação

Roberto M. Cesar Jr.  
rmcesar@usp.br

# **SISTEMAS DINÂMICOS**

# Sistemas Dinâmicos

- Processos que evoluem em função de alguma dimensão (ex. Tempo)
- A lei de evolução é definida por um modelo matemático como uma função ou equação diferencial
- Contínuos ou discretos
- Determinísticos ou estocásticos

# Sistemas dinâmicos

- $x(n+1) = f(x(n))$
- $x$  pode ser vetorial  
 $\mathbf{x} = (x_1, x_2, \dots, x_n)$
- $\mathbf{x}$  é denominado **vetor de estados**
- Espaço de estados
- Trajetórias

# Sistemas dinâmicos, implementação com vetor de estados e erro quadrático definido pela norma

ODE do movimento:

$$\begin{aligned}\frac{dy}{dt} &= v(t) \\ \frac{dv}{dt} &= a(t)\end{aligned}$$

Euler:

$$\begin{aligned}y(t + \Delta t) &= y(t) + \Delta t v(t) \\ v(t + \Delta t) &= v(t) + \Delta t a(t)\end{aligned}$$

Implementação com vetor de estados em uma modelagem por sistemas dinâmicos:

$$\vec{S}(t + \Delta t) = \vec{S}(t) + \Delta t \vec{R}(t)$$

em que  $\vec{S}$  é o vetor de estados e  $\vec{R}$  é o vetor de taxas de variação.

## Exemplo

Abaixo está a solução para  $\frac{d^2x}{dt^2} = 6t$  usando vetor de estados. Temos:

$$\frac{d^2x}{dt^2} = 6t$$

# **Exercício Jupyter – 2**

# Simple Harmonic Oscillator - SHO

---

$$\frac{d^2x}{dt^2} = -\omega^2 x$$

$$\omega = \sqrt{\frac{k}{m}}$$

ou

$$\frac{dv}{dt} = a(t) = -\omega^2 x(t)$$

$$\frac{dx}{dt} = v(t)$$

e a formulação de Euler:

$$x(t + \Delta t) = x(t) + \Delta t v(t)$$

$$v(t + \Delta t) = v(t) - \Delta t \omega^2 x$$

Podemos definir o vetor de estados e o vetor de taxas de variação:

$$\vec{S} = [\vec{S}[0], \vec{S}[1], \vec{S}[2]] = [x, v, t]$$

$$\vec{R} = [\vec{S}[1], -\omega^2 \vec{S}[0], 1]$$

# **Movimentos e forças multidimensionais**

# Vetores

# Capítulo 3

---

## MOVIMENTO BIDIMENSIONAL

### 3.1 – Descrição em termos de coordenadas

Neste capítulo, vamos passar do movimento retilíneo à descrição do movimento num plano, que inclui muitos casos importantes, como o movimento dos projéteis e o movimento da Terra em torno do Sol.

Conforme já foi mencionado na Seção 1.6, podemos especificar a posição de um ponto num plano através de 2 parâmetros, que são suas *coordenadas* em relação a um dado referencial. Se adotarmos coordenadas cartesianas, por exemplo, a posição de uma partícula em movimento no plano será descrita pelo par de funções

$$(x(t), y(t)) \tag{3.1.1}$$

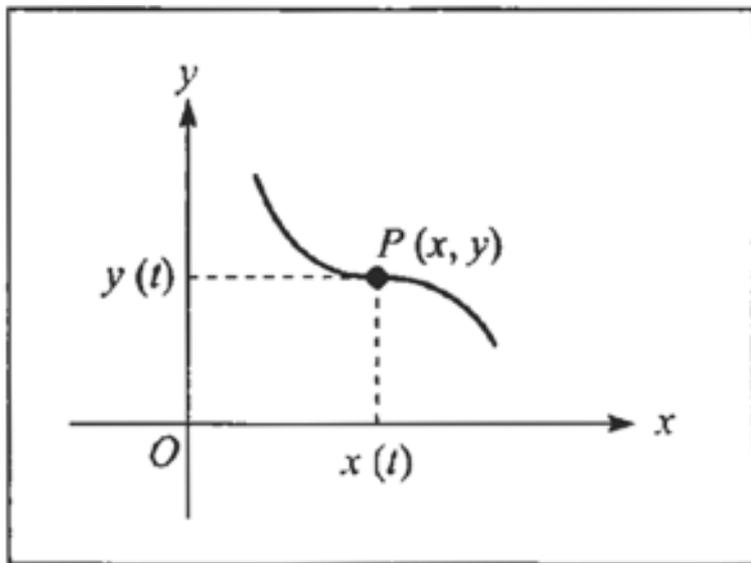


Figura 3.1 Movimento num plano.

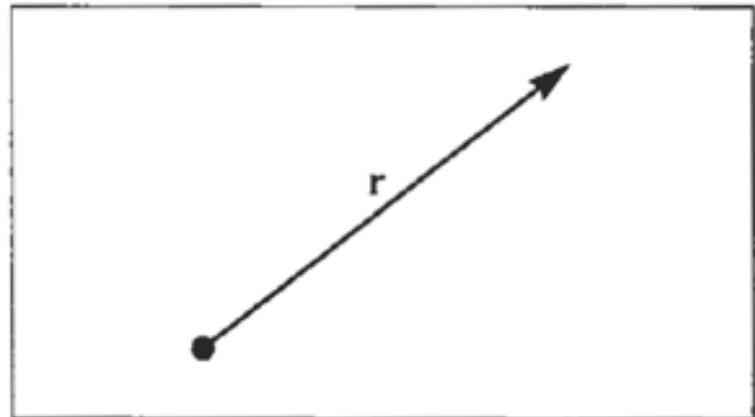


Figura 3.3 Deslocamento como vetor.

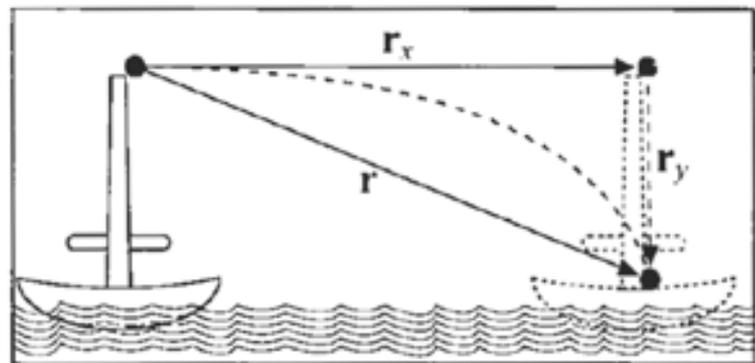
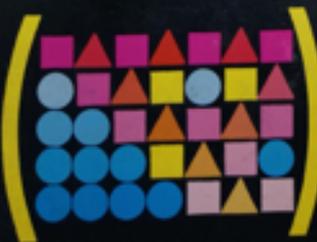
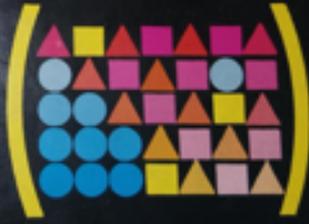


Figura 3.4 Composição de deslocamentos.



# ÁLGEBRA LINEAR E APLICAÇÕES

CARLOS A. CALLIOLI  
HYGINO H. DOMINGUES  
ROBERTO C. F. COSTA



## 2. ESPAÇOS VETORIAIS

Vamos introduzir agora o conceito de *espaço vetorial*. Os espaços vetoriais constituem os objetos de estudo da Álgebra Linear.

**Definição 1** – Dizemos que um conjunto  $V \neq \emptyset$  é um *espaço vetorial sobre*  $\mathbb{R}$  quando, e somente quando:

I – Existe uma adição  $(u, v) \mapsto u + v$  em  $V$ , com as seguintes propriedades:

a)  $u + v = v + u, \forall u, v \in V$  (*comutativa*);

b)  $u + (v + w) = (u + v) + w, \forall u, v, w \in V$  (*associativa*);

c) Existe em  $V$  um *elemento neutro* para essa adição o qual será simbolizado genericamente por  $o$ . Ou seja:

$$\exists o \in V \mid u + o = u, \forall u \in V; (*)$$

d) Para todo elemento  $u$  de  $V$  existe o *oposto*; indicaremos por  $(-u)$  esse oposto. Assim:

$$\forall u \in V, \exists (-u) \in V \mid u + (-u) = o. (**)$$

II – Está definida uma multiplicação de  $\mathbb{R} \times V$  em  $V$ , o que significa que a cada par  $(\alpha, u)$  de  $\mathbb{R} \times V$  está associado um único elemento de  $V$  que se indica por  $\alpha u$ , e para essa multiplicação tem-se o seguinte:

a)  $\alpha(\beta u) = (\alpha\beta)u$

b)  $(\alpha + \beta)u = \alpha u + \beta u$

c)  $\alpha(u + v) = \alpha u + \alpha v$

d)  $1u = u$

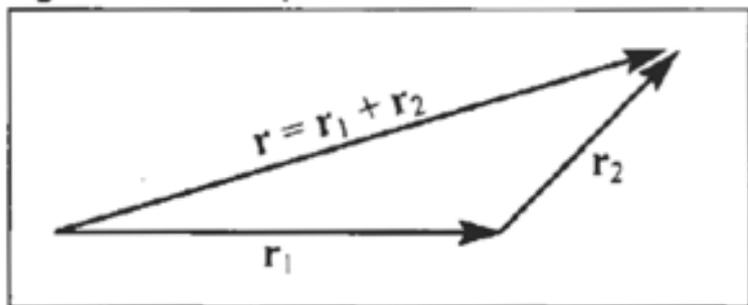


Figura 3.5 Deslocamento resultante.

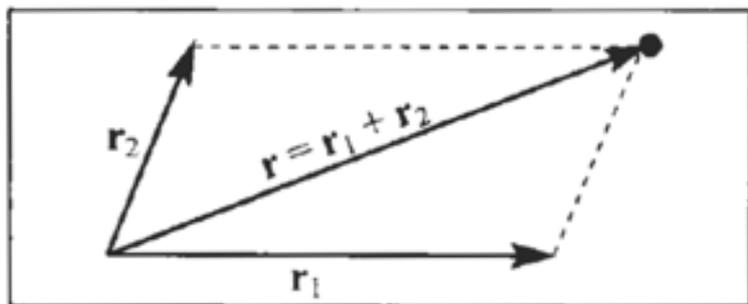


Figura 3.6 Regra do paralelogramo.

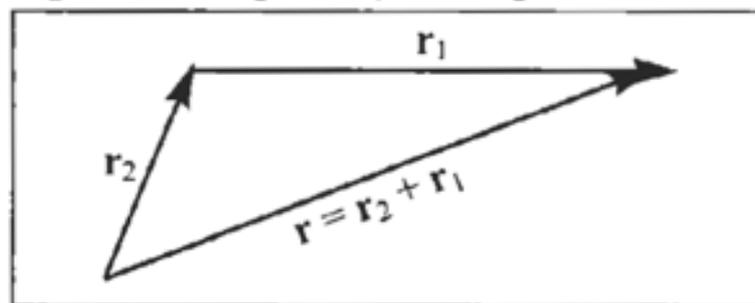


Figura 3.7 Comutatividade da soma.

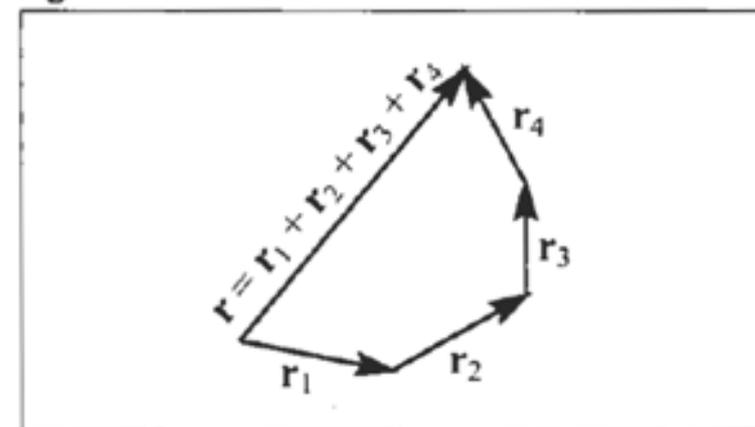


Figura 3.8 Soma de vários deslocamentos.

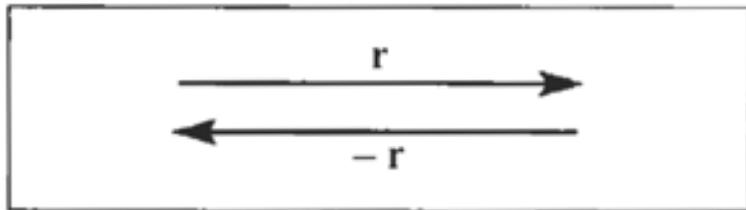


Figura 3.9 Deslocamento oposto.

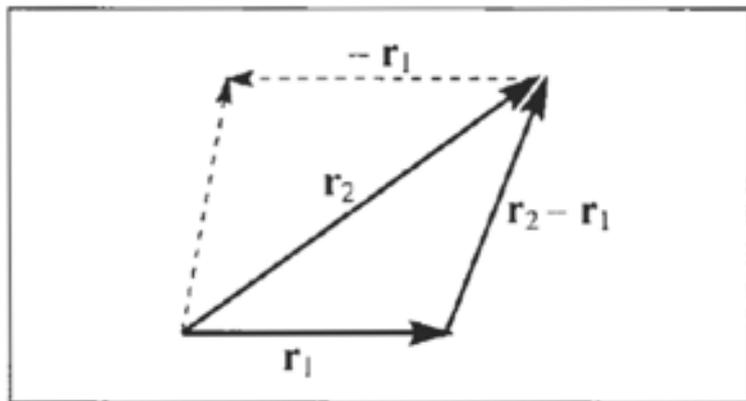


Figura 3.10 Diferença de deslocamentos.

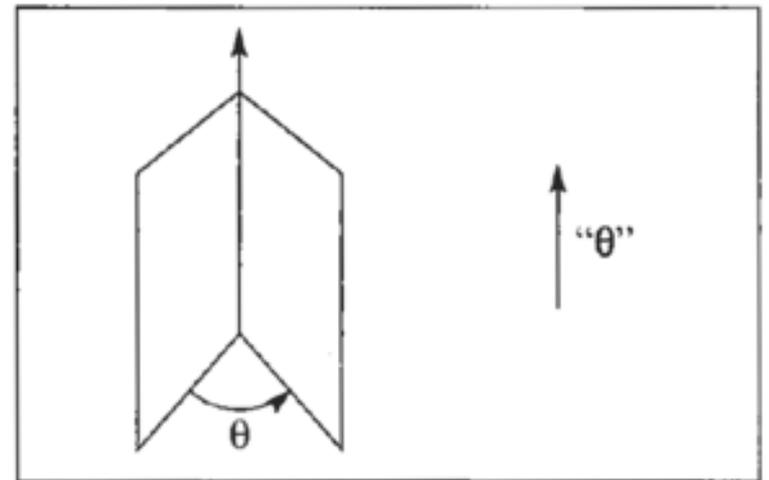


Figura 3.11 Representação de rotação finita.

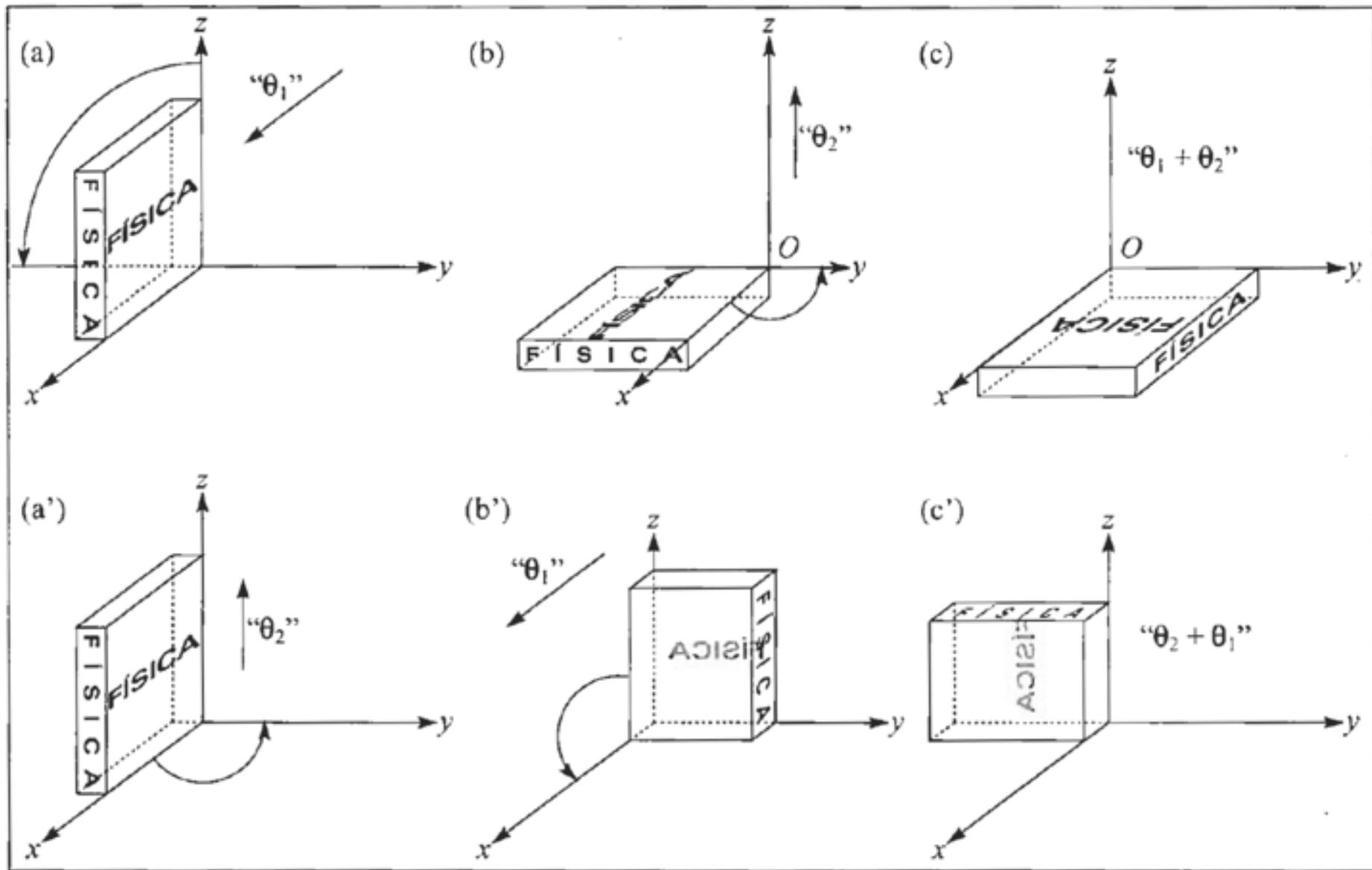


Figura 3.12 Não comutatividade da resultante de rotações finitas.

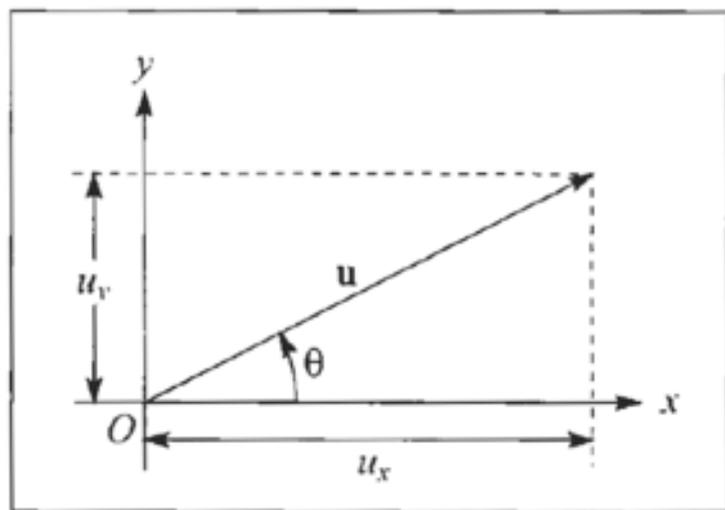


Figura 3.13 Componentes de um vetor.

Seja  $\mathbf{u}$  um vetor qualquer (na Fig. 3.13, tomamos a origem de  $\mathbf{u}$  no ponto  $O$ , origem das coordenadas, o que não tem nada de restritivo, porque um vetor não está associado a uma origem determinada: um vetor obtido de  $\mathbf{u}$  por uma translação é igual a  $\mathbf{u}$ ).

Chamam-se *componentes de  $\mathbf{u}$*  segundo os eixos  $Ox$  e  $Oy$  as projeções  $u_x$  e  $u_y$  de  $\mathbf{u}$  sobre esses eixos (Fig.). A magnitude de  $\mathbf{u}$  (ou *módulo de  $\mathbf{u}$* ) é dada por

$$|\mathbf{u}| = \sqrt{u_x^2 + u_y^2} \quad (3.3.1)$$

Chama-se *vetor unitário* um vetor de módulo = 1. Costuma-se designar um vetor unitário na direção de  $\mathbf{u}$  por  $\hat{\mathbf{u}}$ , de forma que

$$\hat{\mathbf{u}} = \mathbf{u} / |\mathbf{u}| \quad (3.3.2)$$

# **Forças bidimensionais**

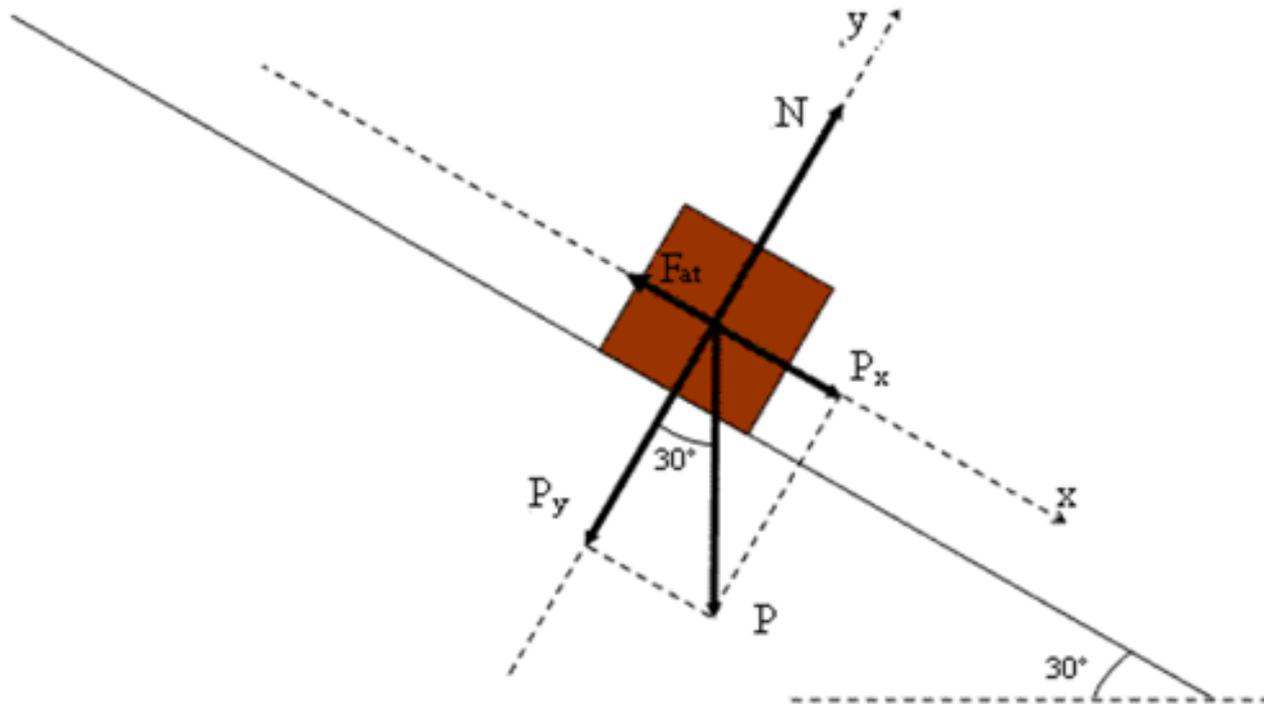
# CAPÍTULO 3



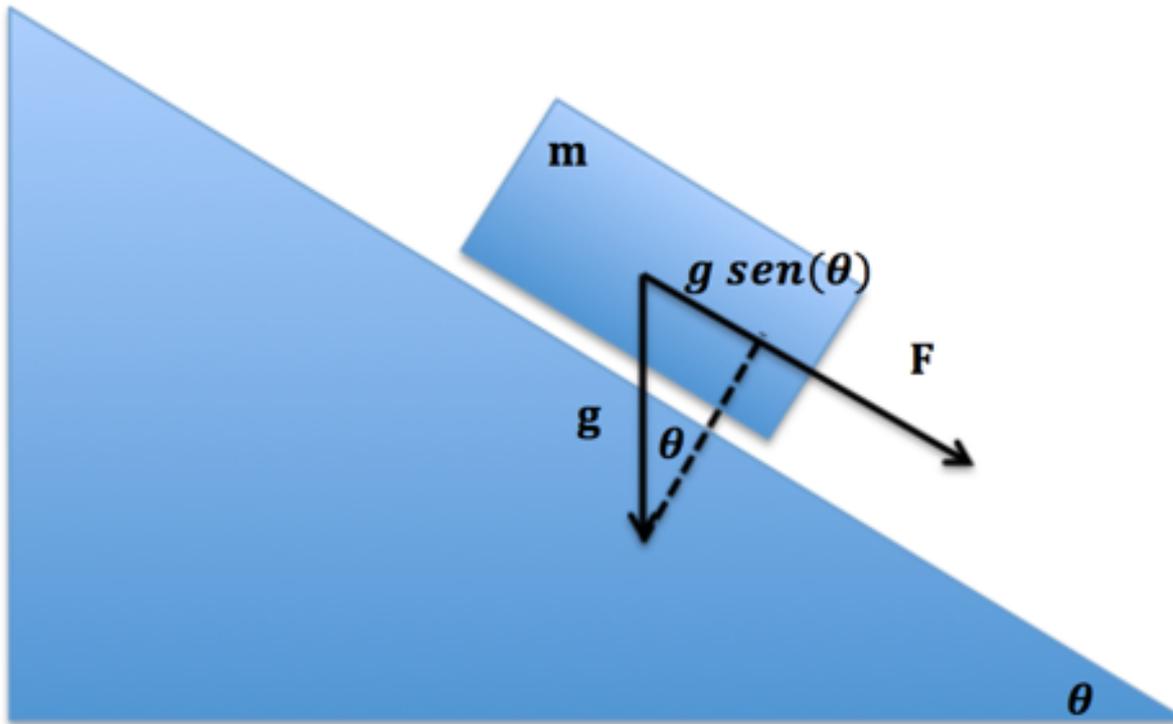
É HORA DE ...

**...EXPERIMENTO!**

# Rampa inclinada



Descida na rampa: Escreva o algoritmo de Euler



$$F = m g \text{ sen } \theta = m \frac{d^2 y}{dt^2}$$

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

$$\frac{dv}{dt} = g \text{ sen } \theta$$

# Exercício

- Rampa inclinada com celular no carrinho
- Sinal de luz para calcular a posição
- Derivar as equações

# Exercício

- Analise os dados do acelerômetro:
- Use o algoritmo de Euler para calcular o intervalo de movimento na rampa.
- Calcule e plote os gráficos de velocidade e posição ao longo do tempo.
- Calcule e compare a velocidade média experimental com a aproximada numericamente.
- Use modelagem por sistemas dinâmicos

# **Movimentos bidimensionais**

### 3.4 — Velocidade e aceleração vetoriais

Consideremos uma partícula, em movimento num plano, que descreve uma trajetória APB, em relação a um sistema de referência Oxy. Seja  $\mathbf{r}(t) = \mathbf{OP}$  o deslocamento da partícula em relação à origem O no instante  $t$ , onde P é a posição ocupada pela partícula no instante  $t$ ; seja  $\mathbf{r}(t + \Delta t) = \mathbf{OP}'$  o deslocamento no instante  $t + \Delta t$ . Pela (3.3.10), o deslocamento relativo da partícula entre os instantes  $t$  e  $t + \Delta t$  é o vetor (Fig. 3.18).

$$\mathbf{PP}' = \Delta\mathbf{r} = \mathbf{r}(t + \Delta t) - \mathbf{r}(t) \quad (3.4.1)$$

Por analogia com a (2.1.5), é natural definirmos a *velocidade média* entre os instantes  $t$  e  $t + \Delta t$  por

$$\mathbf{v}_{t \rightarrow t + \Delta t} = \frac{\mathbf{r}(t + \Delta t) - \mathbf{r}(t)}{\Delta t} = \frac{\Delta\mathbf{r}}{\Delta t} \quad (3.4.2)$$

Como a diferença entre dois vetores e o produto de um vetor por um escalar são vetores, a (3.4.2) mostra que a velocidade média é um vetor, cuja direção e sentido são os da corda PP' que liga as posições nos instantes  $t$  e  $t + \Delta t$  sobre a trajetória.

As componentes da velocidade média são

$$\left. \begin{aligned} v_{x(t \rightarrow t + \Delta t)} &= \frac{\Delta x}{\Delta t} \\ v_{y(t \rightarrow t + \Delta t)} &= \frac{\Delta y}{\Delta t} \end{aligned} \right\} \quad (3.4.3)$$

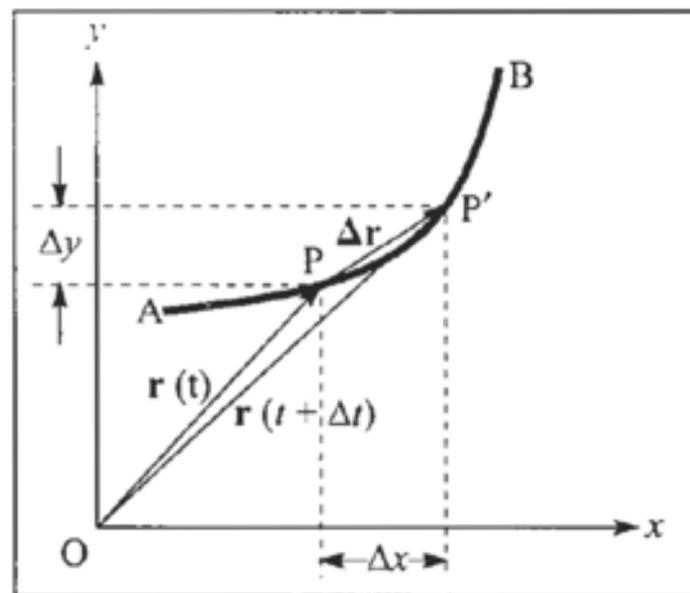


Figura 3.18 Trajetória plana.

## Velocidade Instantânea

$$\mathbf{v}(t) = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta \mathbf{r}}{\Delta t} \right) = \frac{d\mathbf{r}}{dt} = \frac{dx}{dt} \mathbf{i} + \frac{dy}{dt} \mathbf{j} = v_x(t) \mathbf{i} + v_y(t) \mathbf{j} \quad (3.4.5)$$

## Aceleração média

$$\bar{\mathbf{a}}_{t \rightarrow t+\Delta t} = \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t} = \frac{\Delta \mathbf{v}}{\Delta t} \quad (3.4.6)$$

A aceleração *instantânea* no instante  $t$  é o vetor

$$\mathbf{a}(t) = \lim_{\Delta t \rightarrow 0} \left[ \frac{\mathbf{v}(t+\Delta t) - \mathbf{v}(t)}{\Delta t} \right] = \lim_{\Delta t \rightarrow 0} \left( \frac{\Delta \mathbf{v}}{\Delta t} \right) = \frac{d\mathbf{v}}{dt} \quad (3.4.7)$$

ou seja, é a derivada do vetor velocidade instantânea em relação ao tempo. Pela (3.4.5), também podemos escrever

$$\mathbf{a}(t) = \frac{d^2 \mathbf{r}}{dt^2} = \frac{d^2 x}{dt^2} \mathbf{i} + \frac{d^2 y}{dt^2} \mathbf{j} \quad (3.4.8)$$

### 3.5 – Movimento uniformemente acelerado

Um movimento qualquer chama-se *uniformemente acelerado* quando a aceleração é constante (independente do tempo):

$$\boxed{a(t) = a = \text{constante}} \quad (3.5.1)$$

onde "constante", para um vetor, significa *constante em módulo, direção e sentido*.

Analogamente à discussão da Seç. 2.5, para determinar o movimento é preciso ainda dar as *condições iniciais*:

$$\boxed{\begin{array}{l} v(t_0) = v_0 \\ r(t_0) = r_0 \end{array}} \quad (3.5.2)$$

No instante  $t_0 + \Delta t$ ,  $v$  e  $r$  terão variado respectivamente de  $\Delta v$  e  $\Delta r$ , onde, para  $\Delta t$  suficientemente pequeno (de modo que possamos confundir aceleração e velocidade médias e instantâneas), teremos

$$\left. \begin{array}{l} \Delta v = a\Delta t \\ \Delta r = v_0\Delta t \end{array} \right\} \quad (3.5.3)$$

$$\left. \begin{aligned} v_y(t) &= v_{0y} + a(t - t_0) \\ v_x(t) &= v_{0x} \end{aligned} \right\} \quad (3.5.6)$$

$$\left. \begin{aligned} y(t) &= y_0 + v_{0y}(t - t_0) + \frac{1}{2}a(t - t_0)^2 \\ x(t) &= x_0 + v_{0x}(t - t_0) \end{aligned} \right\} \quad (3.5.7)$$

$$\mathbf{v}(t) = \mathbf{v}_0 + \mathbf{a}(t - t_0) \quad (3.5.8)$$

$$\mathbf{r}(t) = \mathbf{r}_0 + \mathbf{v}_0(t - t_0) + \frac{1}{2}\mathbf{a}(t - t_0)^2 \quad (3.5.9)$$

# Equação da trajetória

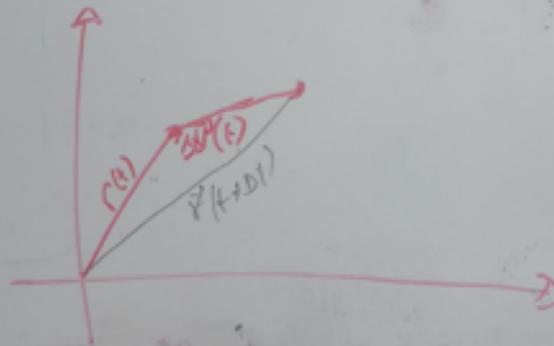
$$y - y_0 = \left( \frac{v_{0y}}{v_{0x}} \right) (x - x_0) + \frac{1}{2} \frac{a}{v_{0x}^2} (x - x_0)^2 \quad (3.5.13)$$

# Euler vetorial

$$\frac{d\vec{r}}{dt} = \vec{v} = \frac{\vec{r}(t+\Delta t) - \vec{r}(t)}{\Delta t}$$

$$\frac{d\vec{v}}{dt} = \vec{a}$$

$$\vec{r}(t+\Delta t) = \vec{v} \Delta t + \vec{r}(t)$$



# Simulação usando Jupyter

- Proponha uma representação do movimento uniformemente acelerado usando sistemas dinâmicos e vetor de estados.
- Crie um documento jupyter implementando o movimento estudado.

# Movimento de projéteis

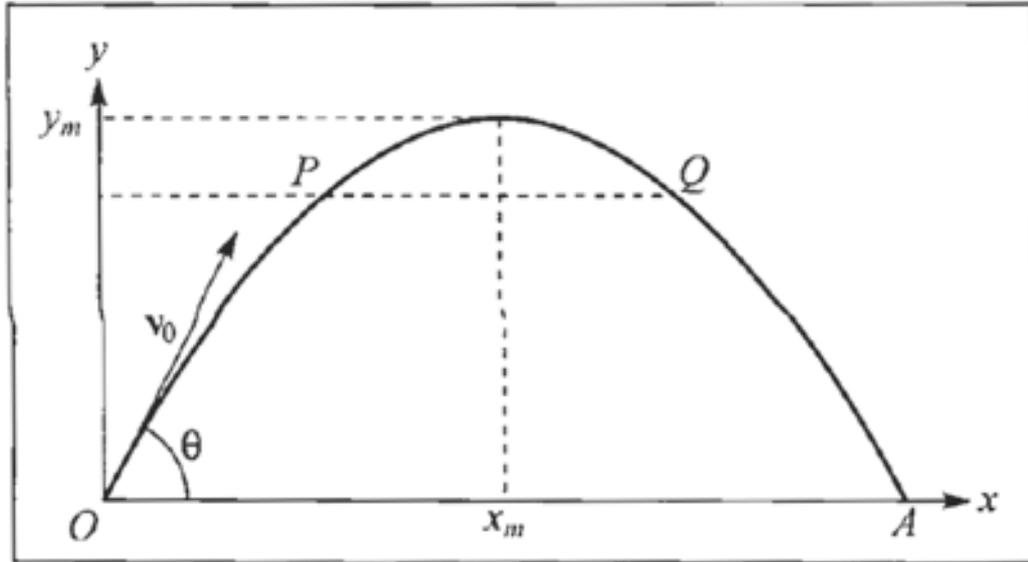


Figura 3.25 Trajetória parabólica.

$$a = -g j \quad (3.6.1)$$

$$v_{0x} = v_0 \cos \theta \quad v_{0y} = v_0 \sin \theta \quad (3.6.2)$$

As (3.5.6) e (3.5.7) ficam

$$v_y = v_0 \sin \theta - gt \quad v_x = v_0 \cos \theta \quad (3.6.3)$$

$$\left. \begin{aligned} y &= v_0 \sin \theta t - \frac{1}{2} g t^2 \\ x &= v_0 \cos \theta t \end{aligned} \right\} \quad (3.6.4)$$

e a equação da trajetória (3.5.13) fica

$$\boxed{y = \operatorname{tg} \theta \cdot x - \frac{gx^2}{2v_0^2 \cos^2 \theta}} \quad (3.6.5)$$

# Simulação usando Jupyter

- Crie um documento jupyter implementando o movimento estudado.

### 3.7 – Movimento circular uniforme

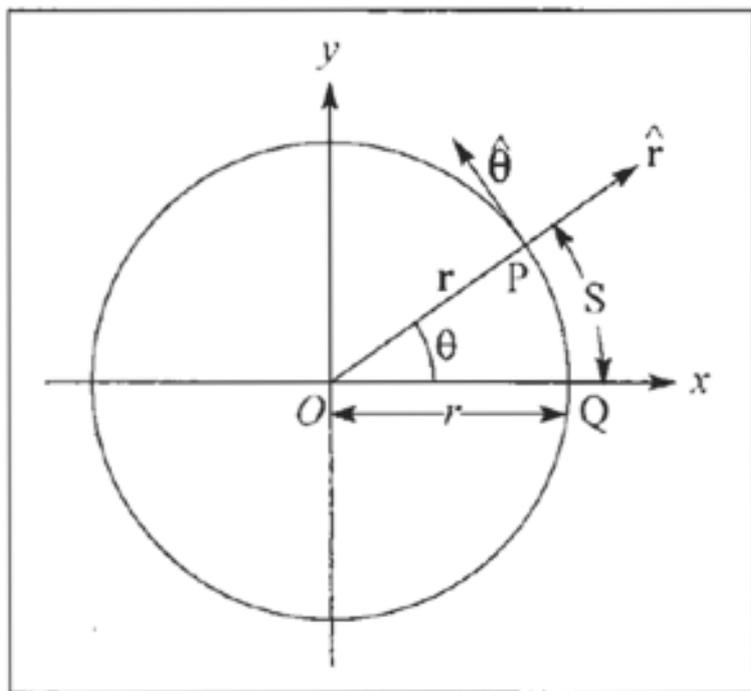


Figura 3.27 Movimento circular.

$$s = r\theta$$

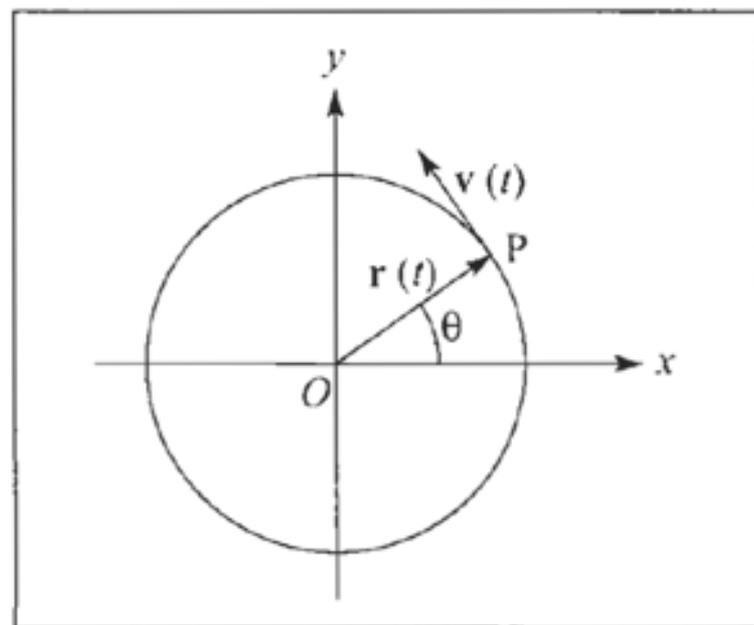


Figura 3.28 Velocidade instantânea.

$$\mathbf{v} = v\hat{\theta}$$

$$v = ds / dt$$

O período  $T$  do movimento é o tempo para dar uma volta completa, ou seja,

$$T = 2\pi r / |v| \quad (3.7.5)$$

Chama-se *freqüência*  $\nu$  o inverso do período:

$$\nu = 1/T \quad (3.7.6)$$

Podemos empregar a (3.7. 1) para exprimir a lei horária (3.7.2) em termos do ângulo  $\theta$  descrito em função do tempo:

$$\theta = \theta_0 + \omega(t - t_0) \quad (3.7.7)$$

onde

$$\omega = v / r \quad (3.7.8)$$

chama-se *velocidade angular*. Temos, analogamente à (3.7.4),

$$\omega = \frac{d\theta}{dt} \quad (3.7.9)$$

e as (3.7.5) e (3.7.8) mostram que

$$|\omega| = \frac{2\pi}{T} = 2\pi\nu \quad (3.7.10)$$

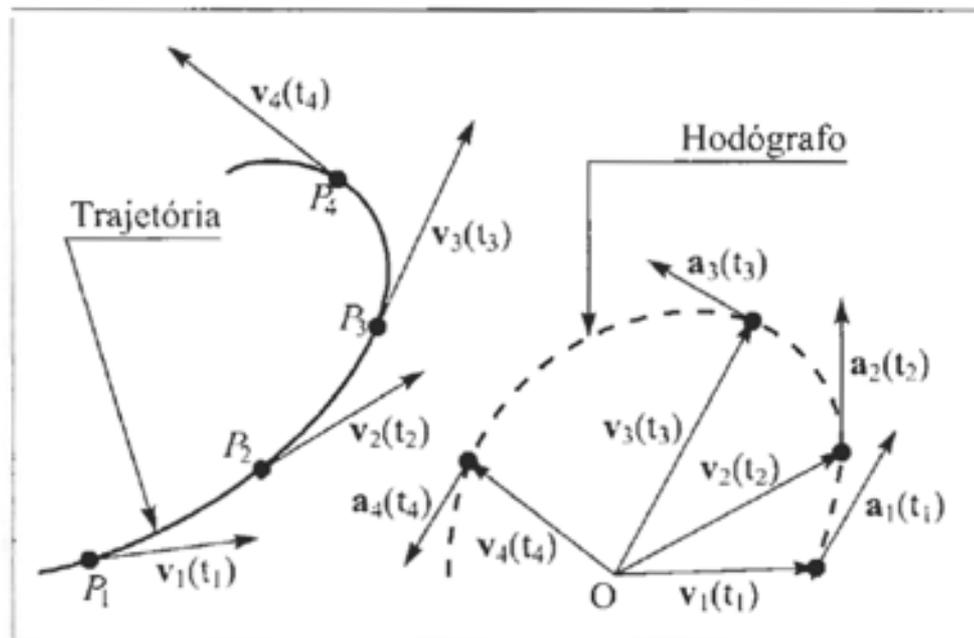


Figura 3.21 Hodógrafo.

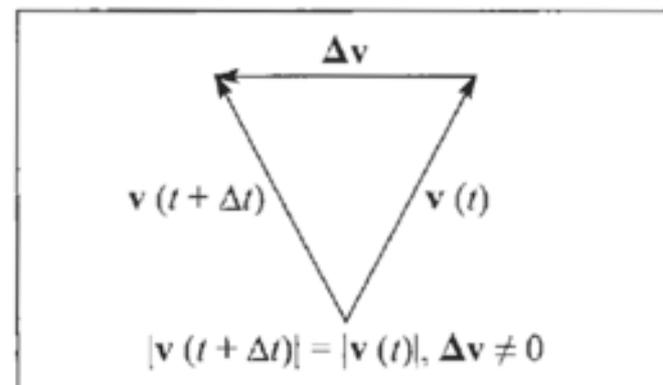


Figura 3.22 Variação da direção da velocidade.

$$\mathbf{v} = \omega r \hat{\theta}$$

(3.7.11)

Embora o movimento circular uniforme tenha uma velocidade de *módulo* constante, a *direção* da velocidade  $v$  varia de ponto a ponto da trajetória. Logo, conforme foi mencionado no fim da Seç. 3.4, ele é um movimento *acelerado*, ou seja, a aceleração é  $\neq 0$ . Vamos agora ver como se obtém a aceleração  $a$ .

Uma forma possível de determinar  $a$  é pelo processo geométrico do hodógrafo, descrito na Seç. 3.4. O hodógrafo de um movimento circu-

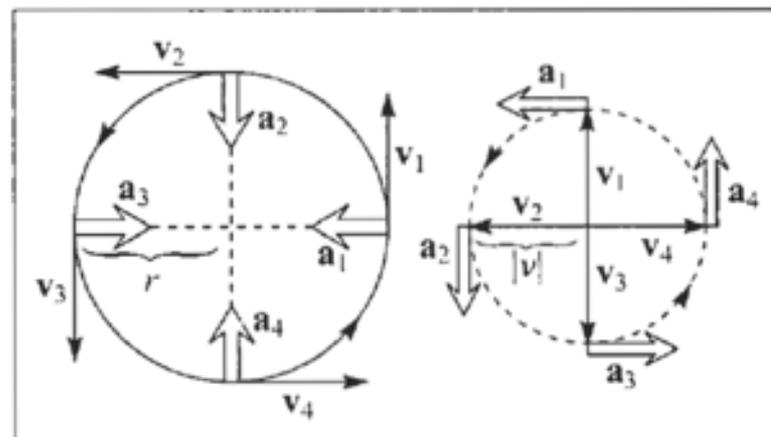


Figura 3.29 Hodógrafo do movimento circular uniforme.

ângulo  $\Delta\theta$  radianamente para dentro do círculo original (trajetória). Logo

$$\mathbf{a} = -|\mathbf{a}|\hat{\mathbf{r}} = -\omega^2 r\hat{\mathbf{r}} = -\frac{v^2}{r}\hat{\mathbf{r}} \quad (3.7.13)$$

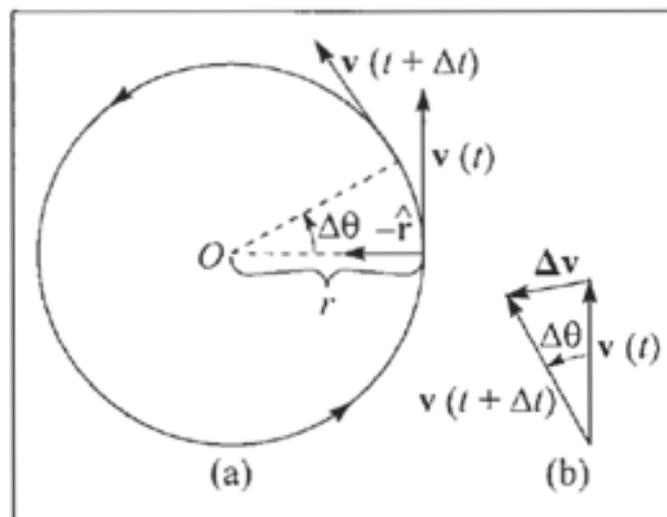


Figura 3.30 Incremento de velocidade.  
ângulo  $|\Delta\theta|$ :

Esta é a chamada *aceleração centrípeta* (porque aponta para o centro do círculo).

Podemos também obter o mesmo resultado de outra forma, empregando diretamente a definição (3.4.7) do vetor  $\mathbf{a}$ . A Fig. 3.30 (a) ao lado mostra os vetores  $\mathbf{v}(t)$  e  $\mathbf{v}(t + \Delta t)$ , onde  $\Delta t$  corresponde a um incremento  $\Delta\theta$ . A (b) ilustra a construção de  $\Delta\mathbf{v}$ , mostrando que, no limite em que  $\Delta t \rightarrow 0$ ,  $\Delta\mathbf{v}$  tende a apontar na direção de  $-\hat{\mathbf{r}}$ . Além disto, o ângulo entre  $\mathbf{v}(t)$  e  $\mathbf{v}(t + \Delta t)$  é também  $\Delta\theta$ , e, no limite em que  $\Delta t \rightarrow 0$ , podemos confundir o comprimento de  $\Delta\mathbf{v}$  (corda) com o do arco de círculo de raio  $|\mathbf{v}|$  que subentende o

$$|\Delta\mathbf{v}| \approx |\mathbf{v}| |\Delta\theta| \left\{ \begin{array}{l} \frac{|\Delta\mathbf{v}|}{\Delta t} \approx |\mathbf{v}| \frac{|\Delta\theta|}{\Delta t} \end{array} \right. \quad (3.7.14)$$

o que se torna exato no limite em que  $\Delta t \rightarrow 0$ , levando novamente à (3.7.12) (cf. (3.7.9)).

# Simulação usando Jupyter

- Crie um documento jupyter implementando o movimento estudado.

## 3.10 \*Visualizing Three-Dimensional Motion

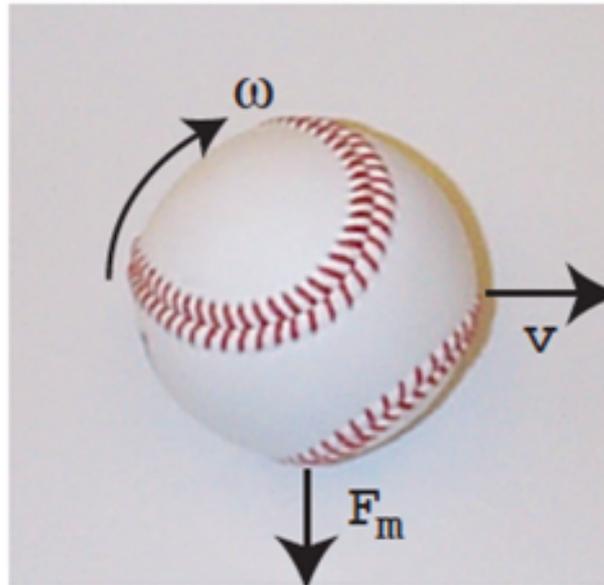


Figure 3.5: The Magnus force on a spinning ball pushes a ball with topspin down.

Of particular interest to baseball fans is the curve of balls in flight due to their rotation. This force was first investigated in 1850 by G. Magnus and the curvature of the trajectories of spinning objects is now known as the *Magnus effect*. It can be explained qualitatively by observing that the speed of the ball's surface relative to the air is different on opposite edges of the ball. If the drag force has the form  $F_{\text{drag}} \sim v^2$ , then the unbalanced force due to the difference in the velocity on opposite sides of the ball due to its rotation is given by

$$F_{\text{magnus}} \sim v\Delta v. \quad (3.22)$$

We can express the velocity difference in terms of the ball's angular velocity and radius and write

$$F_{\text{magnus}} \sim vr\omega. \quad (3.23)$$

The direction of the Magnus force is perpendicular to both the velocity and the rotation axis. For example, if we observe a ball moving to the right and rotating clockwise (that is, with topspin), then the velocity of the ball's surface relative to the air at the top,  $v + \omega r$ , is higher than the velocity at the bottom,  $v - \omega r$ . Because the larger velocity will produce a larger force, the Magnus effect will contribute a force in the downward direction. These considerations suggest that the Magnus force can be expressed as a vector product:

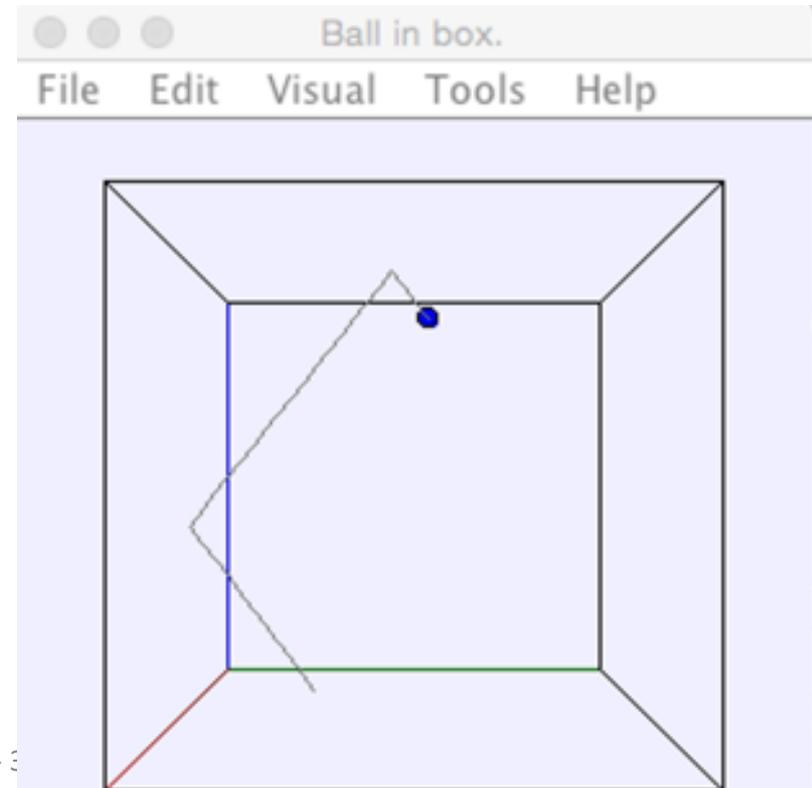
$$F_{\text{magnus}}/m = C_M(\boldsymbol{\omega} \times \mathbf{v}), \quad (3.24)$$

$$\mathbf{F}/m = \mathbf{g} - C_D|\mathbf{v}|\mathbf{v} + C_M(\boldsymbol{\omega} \times \mathbf{v}).$$

$$\frac{dv_x}{dt} = -C_Dvv_x + C_M(\omega_yv_z - \omega_zv_y)$$

$$\frac{dv_y}{dt} = -C_Dvv_y + C_M(\omega_zv_x - \omega_xv_z)$$

$$\frac{dv_z}{dt} = -C_Dvv_z + C_M(\omega_xv_y - \omega_yv_x) - g,$$



# EP2

- Realizar experimentos de:
  - Queda livre
  - Descida na rampa
  - Pêndulo
  - Lançamento de projétil
  - Movimento circular

# EP2

- Em todos:
  - Obter dados com celular
  - Simular e comparar com os resultados experimentais
  - Comparar as variações dos algoritmos de Euler

# EP2

- Entrega - pacote zip no paca contendo:
  - **1- Relatório** no formato definido no Apendix 1A, capítulo 1, pg 9 do livro. O Relatório deverá ter as seções indicadas nesse apêndice.
    - Uma última seção deve ser anexada: Contribuições dos Autores, em que deve constar as responsabilidades de cada membro da equipe. Como exemplo, veja a seção Author's Contributions em:  
<http://www.biomedcentral.com/1471-2105/16/35>

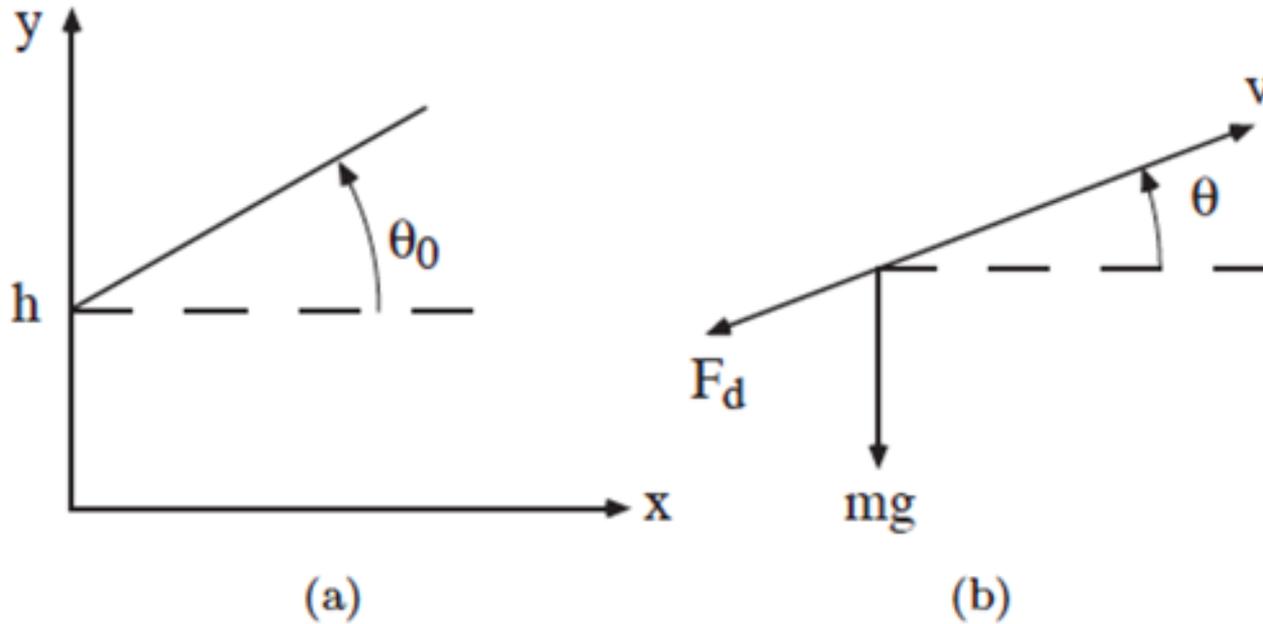
# EP2

- Entrega: pacote zip no paca contendo:
  - **2- Códigos fonte**
  - **3- Planilha csv dos dados**
  - **4- Vídeo do experimento:** Faça um vídeo da realização do experimento, coloque online e inclua a URL no relatório.

# Material de trabalho

- Leia o Capítulo 3 do livro texto do Gould.
- Leia o Capítulo 3 do livro texto do Moysés.
- Resolva os exercícios desses capítulos.
- Procure o monitor ou o professor para suas dúvidas.

# Two-Dimensional Trajectories



$$m \frac{dv_x}{dt} = -F_d \cos \theta$$

$$m \frac{dv_y}{dt} = -mg - F_d \sin \theta.$$

For example, let us maximize the range of a round steel ball of radius 4 cm. A reasonable assumption for a steel ball of this size and typical speed is that  $F_d = C_2 v^2$ . Because  $v_x = v \cos \theta$  and  $v_y = v \sin \theta$ , we can rewrite (3.17) as

$$m \frac{dv_x}{dt} = -C_2 v v_x \quad (3.18a)$$

$$m \frac{dv_y}{dt} = -mg - C - 2v v_y. \quad (3.18b)$$

Note that  $-C_2 v v_x$  and  $-C_2 v v_y$  are the  $x$  and  $y$  components of the drag force  $-C_2 v^2$ . Because (3.18a) and (3.18b) for the change in  $v_x$  and  $v_y$  involve the square of the velocity,  $v^2 = v_x^2 + v_y^2$ , we cannot calculate the vertical motion of a falling body without reference to the horizontal component, that is, the motion in the  $x$  and  $y$  direction is *coupled*.

## Problem 3.10. Trajectory of a steel ball

- a. Use `Projectile` and `ProjectileApp` to compute the two-dimensional trajectory of a ball moving in air without air friction, and plot  $y$  as a function of  $x$ . Compare your computed results with the exact results. For example, assume that a ball is thrown from ground level at an angle  $\theta_0$  above the horizontal with an initial velocity of  $v_0 = 15 \text{ m/s}$ . Vary  $\theta_0$  and show that the maximum range occurs at  $\theta_0 = \theta_{\max} = 45^\circ$ . What is  $R_{\max}$ , the maximum range, at this angle? Compare your numerical result to the analytical result  $R_{\max} = v_0^2/g$ .
- b. Suppose that a steel ball is thrown from a height  $h$  at an angle  $\theta_0$  above the horizontal with the same initial speed as in part (a). If you neglect air resistance, do you expect  $\theta_{\max}$  to be larger or smaller than  $45^\circ$ ? What is  $\theta_{\max}$  for  $h = 2 \text{ m}$ ? By what percent is the range  $R$  changed if  $\theta$  is varied by 2% from  $\theta_{\max}$ ?
- c. Consider the effects of air resistance on the range and optimum angle of a steel ball. For a ball of mass  $7 \text{ kg}$  and cross-sectional area  $0.01 \text{ m}^2$ , the parameter  $C_2 \approx 0.1$ . What are the units of  $C_2$ ? It is convenient to exaggerate the effects of air resistance so that you can more easily determine the qualitative nature of the effects. Hence, compute the optimum angle for  $h = 2 \text{ m}$ ,  $v_0 = 30 \text{ m/s}$ , and  $C_2/m = 0.1$ , and compare your answer to the value found in part (b). Is  $R$  more or less sensitive to changes in  $\theta_0$  from  $\theta_{\max}$  than in part (b)? Determine the optimum launch angle and the corresponding range for the more realistic value of  $C_2 = 0.1$ . A detailed discussion of the maximum range of the ball has been given by Lichtenberg and Wills.

# Variações do algoritmo de Euler

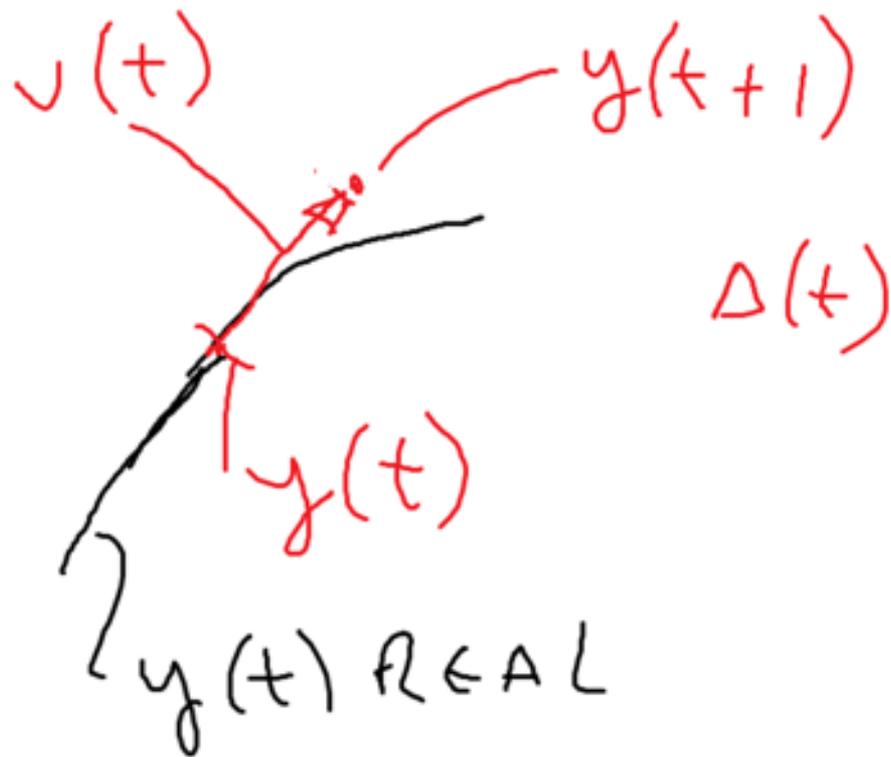
$$\begin{aligned}v(t + \Delta t) &= v(t) + a(t)\Delta t \\y(t + \Delta t) &= y(t) + v(t)\Delta t,\end{aligned}$$

Algoritmo de Euler modificado: Euler-Cromer

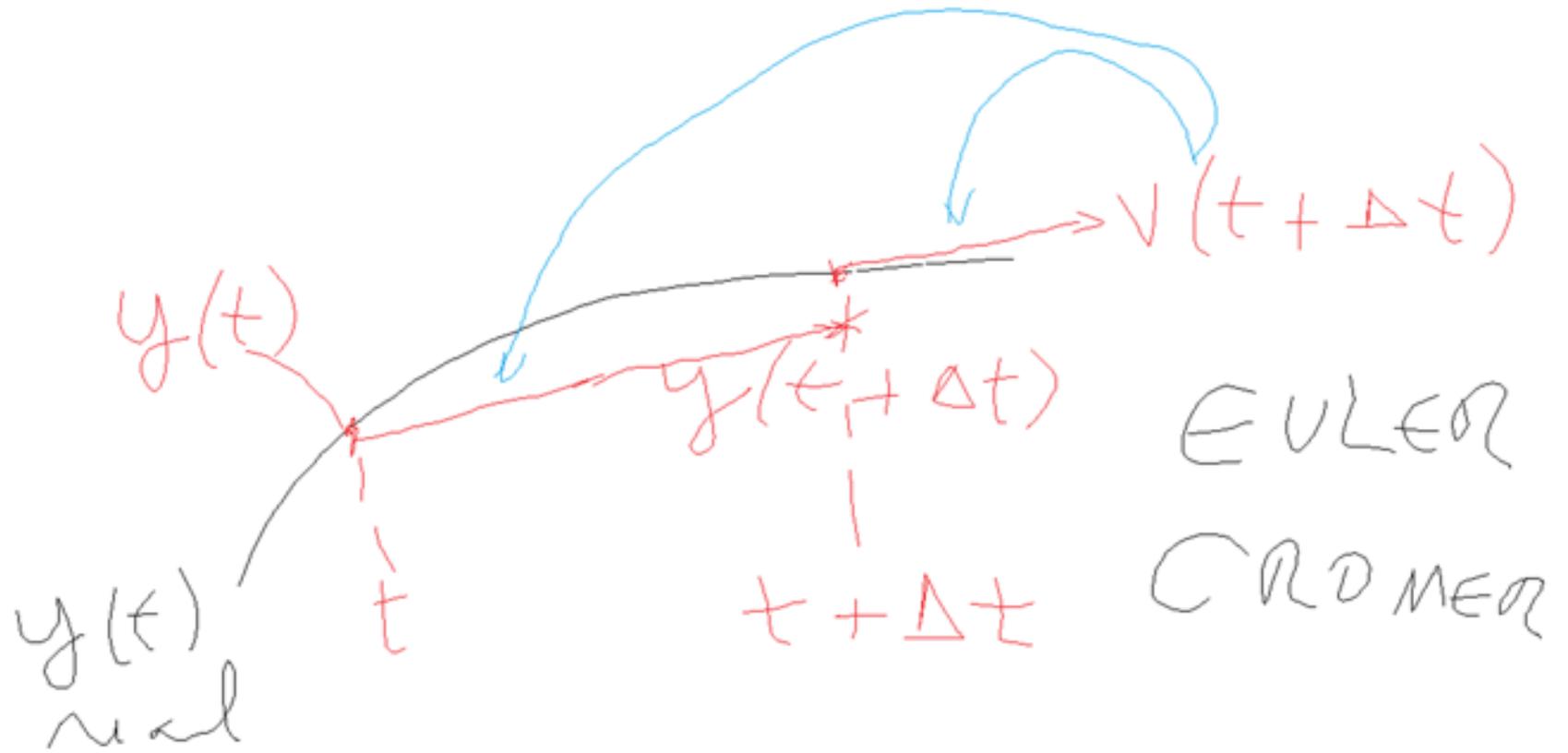
To illustrate why we need algorithms other than the simple Euler algorithm, we make a very simple change in the Euler algorithm and write

$$v(t + \Delta t) = v(t) + a(t)\Delta t \tag{3.1a}$$

$$y(t + \Delta t) = y(t) + v(t + \Delta t)\Delta t, \tag{3.1b}$$



ALGORITMO  
DE  
EVLER



### Problem 3.1. Comparing Euler algorithms

- a. Write a class that extends `Particle` and models a simple harmonic oscillator for which  $F = -kx$ . For simplicity, choose units such that  $k = 1$  and  $m = 1$ . Determine the numerical error in the position of the simple harmonic oscillator after the particle has evolved for several cycles. Is the original Euler algorithm stable for this system? What happens if you run for longer times?
- b. Repeat part (a) using the Euler-Cromer algorithm. Does this algorithm work better? If so, in what way?
- c. Modify your program so that it computes the total energy,  $E_{\text{sho}} = v^2/2 + x^2/2$ . How well is the total energy conserved for the two algorithms? Also consider the quantity  $\tilde{E} = E_{\text{sho}} + (\Delta t/2)xp$ . What is the behavior of this quantity for the Euler-Cromer algorithm?

# Analizando Algoritmos



- Existe um algoritmo que resolve meu problema?
- Exemplo: o algoritmo vai parar algum dia?

Algoritmo simples

1- enquanto x for diferente de 1 faça

$$x = x - 2$$

2- Imprima "Ola, cheguei!"

Fim

- Se x for inicialmente um número ímpar e positivo, o algoritmo irá imprimir a mensagem. Senão for, não irá imprimir nunca.
- Nós sabemos exatamente o que vai acontecer!

# Analizando Algoritmos



- Exemplo 2: o algoritmo vai parar algum dia?

Algoritmo nao tao simples

1- enquanto x for diferente de 1 faca

    Se x for par,

~~$x = x - 2$~~   $x = x / 2$

    senao

$x = 3x + 1$

2- Imprima "Ola, cheguei!"

Fim

# O que é fácil, o que é difícil...

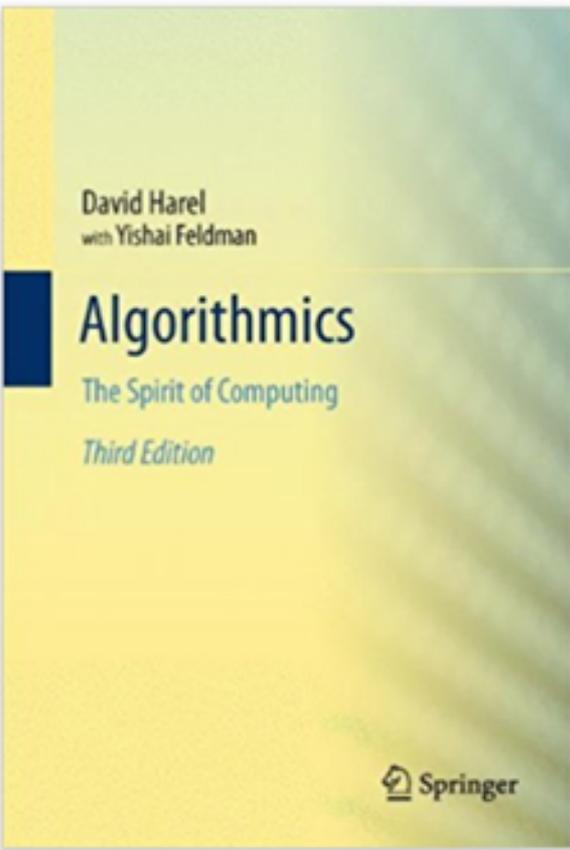


- Está aqui um exemplo de seqüência produzida pelo algoritmo: 7, 22, 11, 34, 17, 52, 26, 13, 40, 20, 10, 5, 16, 8, 4, 2, 1
- Esse algoritmo já foi testado inúmeras vezes com diversos números de entrada, e sempre parou, mas...
- até hoje, ninguém conseguiu provar que ele sempre parará para qualquer número de entrada!

# Algorithmics: The Spirit of Computing 3rd ed. 1987 Edition

by [David Harel](#) (Author), [Yishai Feldman](#) (Author)

★★★★☆ 9 customer reviews



ISBN-13: 978-3642272653

ISBN-10: 3642272657

[Why is ISBN important?](#)

**Hardcover**  
\$66.49

**Paperback**  
\$65.70 - \$79.95

**Other Sellers**  
See all 3 versions

Buy new

**Only 3 left in stock (more on the way).**

Ships from and sold by Amazon.com. Gift-wrap available.

This item ships to **São Paulo, Brazil**. Want it **Thursday, April 19**? Order within **23 hrs 42 mins** and choose **AmazonGlobal Priority Shipping** at checkout. [Learn more](#)

## More Buying Choices

22 New from \$62.49 | 20 Used from \$60.01

## 3.4 Specifying the state of a system using arrays

The numerical solution of an ordinary differential equation (frequently called an ODE) begins by expressing the equation as several first-order differential equations. If the highest derivative in the ODE is order  $n$  (for example,  $d^n x/dt^n$ ), then it can be shown that the ODE can be written equivalently as  $n$  first-order differential equations. For example, Newton's equation of motion is a second-order differential equation and can be written as two first-order differential equations for the position and velocity in each spatial dimension. For example, in one dimension we can write

$$\frac{dy}{dt} = v(t) \tag{3.5a}$$

$$\frac{dv}{dt} = a(t) = F(t)/m. \tag{3.5b}$$

If we have more than one particle, there are additional first-order differential equations for each particle. It is convenient to have a standard way of handling all these cases.

Let us assume that each differential equation is of the form:

$$\frac{dx_i}{dt} = r_i(x_0, x_1, x_2, \dots, x_{n-1}, t), \tag{3.6}$$

## 3.4 Specifying the state of a system using arrays

Now we show some examples:

```
// one particle in one dimension:  
state[0] // stores x  
state[1] // stores v  
state[2] // stores t (time)  
// one particle in two dimensions:  
state[0] // stores x  
state[1] // stores vx  
state[2] // stores y  
state[3] // stores vy  
state[4] // stores t  
// two particles in one dimension:  
state[0] // stores x1  
state[1] // stores v1  
state[2] // stores x2  
state[3] // stores v2  
state[4] // stores t
```

## 3.5 The ODE interface

To introduce the ODE interface, we again consider the equations of motion for a falling particle. We use a state array ordered as  $s = (y, v, t)$ , so that the dynamical equations can be written as:

$$\dot{s}_0 = s_1 \tag{3.7a}$$

$$\dot{s}_1 = -g \tag{3.7b}$$

$$\dot{s}_2 = 1. \tag{3.7c}$$

```
package org.opensourcephysics.sip.ch03;
import org.opensourcephysics.numerics.*;

public class FallingParticleODE implements ODE {
    final static double g = 9.8;
    double[] state = new double[3];

    public FallingParticleODE(double y, double v) {
        state[0] = y;
        state[1] = v;
        state[2] = 0;           // initial time
    }

    public double[] getState() { // required to implement ODE interface
        return state;
    }

    public void getRate(double[] state, double[] rate) {
        rate[0] = state[1]; // rate of change of y is v
        rate[1] = -g;
        rate[2] = 1;       // rate of change of time is 1
    }
}
```

Listing 3.6: The ODE solver interface. Note the four methods that must be defined.

```
package org.opensourcephysics.numerics;  
public interface ODESolver {  
    public void initialize(double stepSize);  
  
    public double step();  
  
    public void setStepSize(double stepSize);  
  
    public double getStepSize();  
}
```

```

public class FallingParticleODEApp extends AbstractCalculation {
    public void calculate() {
        // gets initial conditions
        double y0 = control.getDouble("Initial y");
        double v0 = control.getDouble("Initial v");
        // creates ball with initial conditions
        FallingParticleODE ball = new FallingParticleODE(y0, v0);
        // creates ODE solver
        ODESolver solver = new Euler(ball); // note how particular algorithm is chosen
        // sets time step dt in the solver
        solver.setStepSize(control.getDouble("dt"));
        while(ball.state[0]>0) {
            solver.step();
        }
        control.println("final time = "+ball.state[2]);
        control.println("y = "+ball.state[0]+" v = "+ball.state[1]);
    }

    public void reset() {
        control.setValue("Initial y", 10); // sets default input values
        control.setValue("Initial v", 0);
        control.setValue("dt", 0.01);
    }

    public static void main(String[] args) { // creates a calculation control
        CalculationControl.createApp(new FallingParticleODEApp());
    }
}

```

```

public class Projectile implements Drawable, ODE {
    static final double g = 9.8;
    double[] state = new double[5]; // {x, vx, y, vy, t}
    int pixRadius = 6; // pixel radius for drawing of projectile
    EulerRichardson odeSolver = new EulerRichardson(this);

    public void setStepSize(double dt) {
        odeSolver.setStepSize(dt);
    }

    public void step() {
        odeSolver.step(); // do one time step using selected algorithm
    }

    public void setState(double x, double vx, double y, double vy) {
        state[0] = x;
        state[1] = vx;
        state[2] = y;
        state[3] = vy;
        state[4] = 0;
    }

    public double[] getState() {
        return state;
    }
}

```

```

public double[] getState() {
    return state;
}

public void getRate(double[] state, double[] rate) {
    rate[0] = state[1]; // rate of change of x
    rate[1] = 0;       // rate of change of vx
    rate[2] = state[3]; // rate of change of y
    rate[3] = -g;      // rate of change of vy
    rate[4] = 1;       // dt/dt = 1
}

public void draw(DrawingPanel drawingPanel, Graphics g) {
    int xpix = drawingPanel.xToPix(state[0]);
    int ypix = drawingPanel.yToPix(state[2]);
    g.setColor(Color.red);
    g.fillOval(xpix-pixRadius, ypix-pixRadius, 2*pixRadius, 2*pixRadius);
    g.setColor(Color.green);
    int xmin = drawingPanel.xToPix(-100);
    int xmax = drawingPanel.xToPix(100);
    int y0 = drawingPanel.yToPix(0);
    g.drawLine(xmin, y0, xmax, y0); // draw a line to represent the ground
}
}

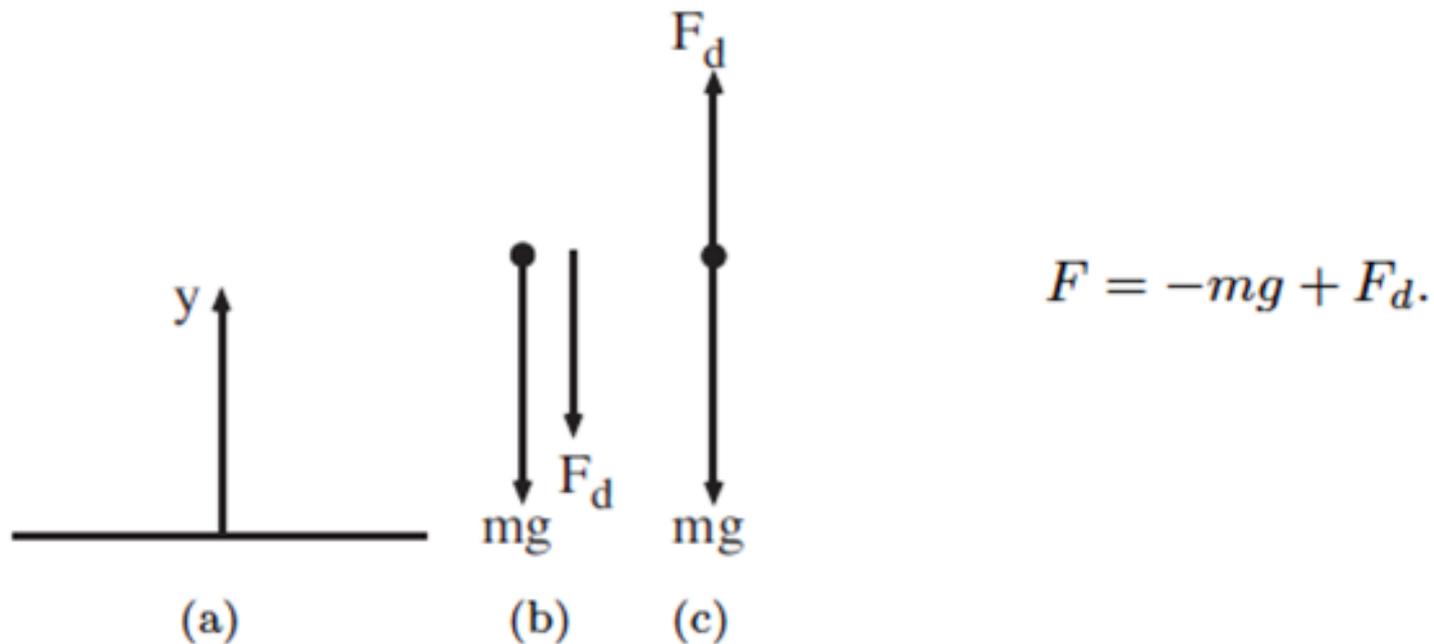
```

**Modelos mais realistas**

# Effects of Drag Resistance

$$F = \frac{GMm}{(R+y)^2} = \frac{GMm}{R^2(1+y/R)^2} = mg\left(1 - 2\frac{y}{R} + \dots\right), \quad (3.8)$$

where  $y$  is measured from the Earth's surface,  $R$  is the radius of the Earth,  $M$  is the mass of the Earth,  $G$  is the gravitational constant, and  $g = GM/R^2$ .



# Modelo com resistência do ar

- É necessário determinar a  $F_d$  como função de  $v$ , i.e.  $F_d(v)$
- Duas abordagens:
  - Empiricamente: medindo  $y(t)$  e calculando numericamente  $v(t)$ ,  $a(t)$ .
  - Assumindo um modelo *a priori* para  $v(t)$

$$F_{1,d}(v) = C_1 v,$$

$$F_{2,d}(v) = C_2 v^2,$$

Because  $F_d(v)$  increases as  $v$  increases, there is a limiting or *terminal velocity* (speed) at which the net force on a falling object is zero. This terminal speed can be found from (3.9) and (3.10) by setting  $F_d = mg$  and is given by

$$v_{1,t} = \frac{mg}{C_1}, \quad (\text{linear drag}) \quad (3.11a)$$

$$v_{2,t} = \left(\frac{mg}{C_2}\right)^{1/2}, \quad (\text{quadratic drag}) \quad (3.11b)$$

$$F_{1,d} = C_1 v_{1,t} \left(\frac{v}{v_{1,t}}\right) = mg \frac{v}{v_{1,t}}, \quad F_1(v)/m = -g \left(1 - \frac{v}{v_{1,t}}\right),$$
$$F_{2,d} = C_2 v_{2,t}^2 \left(\frac{v}{v_{2,t}}\right)^2 = mg \left(\frac{v}{v_{2,t}}\right)^2. \quad F_2(v)/m = -g \left(1 - \frac{v^2}{v_{2,t}^2}\right).$$

# Simulação usando Jupyter

- Crie um documento jupyter implementando o movimento estudado.