Introdução à Física Computacional II (4300318)

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

Aula 7

Problemas de valores de contorno

Problemas de valores de contorno

- Até aqui, investigamos técnicas para integrar equações diferenciais ordinárias a partir das condições iniciais.
- Outra classe de problemas envolve fixar não as condições iniciais, mas o que se costuma chamar de **valores de contorno**. Por exemplo, podemos fixar o valor da variável dependente no início e no final de um intervalo da variável independente.

Problemas de valores de contorno

- Como exemplo concreto, imagine uma bola lançada verticalmente para cima a partir do solo. Qual deve ser a velocidade de lançamento para que a bola alcance uma certa altura h exatamente t_1 segundos depois?
- Em termos matemáticos, queremos integrar a equação diferencial

$$\frac{d^2 y}{d t^2} = -g$$

entre $t = 0$ e $t = t_1$, com $y(0) = 0$ e $y(t_1) = h$.

Problemas de valores de contorno

- As técnicas que estudamos até aqui trabalham integrando a equação diferencial a partir das condições iniciais. No contexto do exemplo, essas condições correspondem à altura e à velocidade da bola no mesmo instante $t = 0$. Como utilizá-las quando o que se quer justamente é descobrir a velocidade inicial?

O método do chute

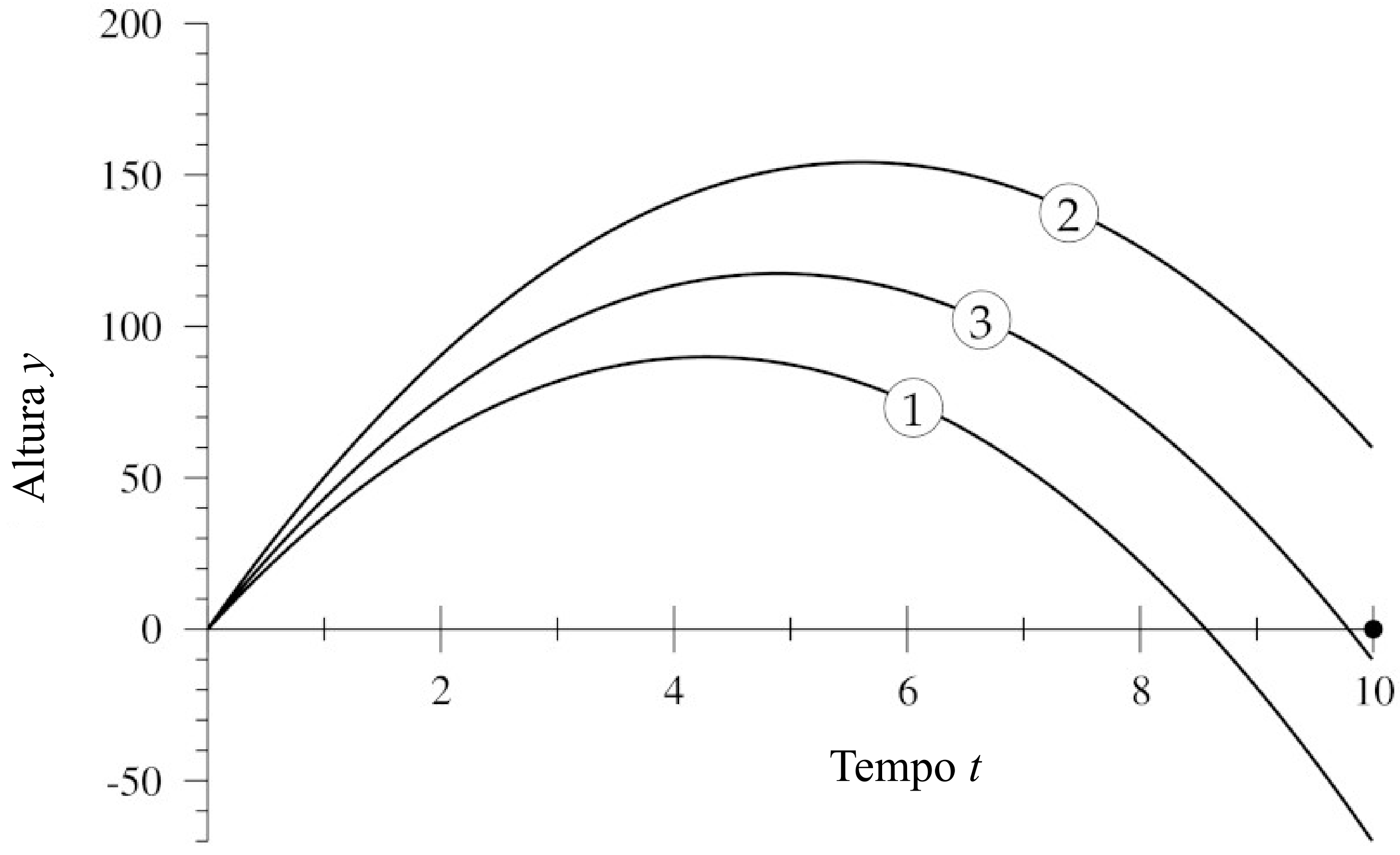
- Uma possível solução é simplesmente supor um valor v para a velocidade inicial, verificar qual é a altura final $y(t_1;v)$ obtida e usar o resultado para reajustar o valor da velocidade inicial até obter a altura final correta, dentro de uma certa precisão. Isso corresponde a resolver para v a equação

$$y(t_1;v)-h=0.$$

- Exceto pelo fato de que podemos não ter uma expressão fechada para a função $y(t_1;v)$, a equação acima pode ser resolvida pelas técnicas aprendidas em Introdução à Física Computacional I.

O método do chute

- Uma representação gráfica da abordagem, no caso específico em que $h = 0$, é oferecida pela figura abaixo.



Exemplo 1

$$\frac{d^2 y}{d t^2} = -g$$

$$y(0) = 0, \quad y(10) = 0$$

$$g = 9.81$$

$$\delta = 10^{-10}$$

```
# Constantes
g = 9.81        # Aceleração da gravidade

ta = 0.0          # Início do intervalo da variável independente
tb = 10.0         # Final do intervalo da variável independente
ya = 0.0          # Valor da variável independente no início do intervalo
yb = 0.0          # Valor da variável independente no final do intervalo
dt = 1e-2         # Tamanho do passo de integração
prec = 1e-10      # Precisão do resultado para a velocidade inicial

def f(r,t):
    y, v = r[0], r[1]
    f0, f1 = v, -g
    return array([f0,f1],float)

def passo_rk4(f,r,t,dt):             # Calcula um passo no método de RK4
    k1 = dt*f(r,t)
    k2 = dt*f(r+0.5*k1,t+0.5*dt)
    k3 = dt*f(r+0.5*k2,t+0.5*dt)
    k4 = dt*f(r+k3,t+dt)
    return (k1+2.0*(k2+k3)+k4)/6.0

# Função que integra a equação diferencial definida por f entre 'ta' e 'tb'
# com condição inicial 'ra' e retorna o valor final do vetor 'r'
def r_final(f,ta,tb,ra,dt):
    r = ra
    for t in arange(ta,tb,dt):
        r += passo_rk4(f,r,t,dt)
    return r
```

Exemplo 1a

$$\frac{d^2 y}{d t^2} = -g$$

$$y(0) = 0, \quad y(10) = 0$$

```
# Solução via método da bissecção (busca binária)
v1 = 1e-3   # Palpite de uma velocidade com altura final abaixo da desejada
v2 = 1e3    # Palpite de uma velocidade com altura final acima da inicial
r1 = array([ya,v1],float)   # Vetor 'r' inicial correspondente a 'v1'
r2 = array([ya,v2],float)   # Vetor 'r' inicial correspondente a 'v2'
altura1 = r_final(f,ta,tb,r1,dt)[0] - yb # Alt. relativa final partindo com 'v1'
altura2 = r_final(f,ta,tb,r2,dt)[0] - yb # Alt. relativa final partindo com 'v2'
if (altura1*altura2 > 0):
    sys.exit("Alturas relativas com mesmo sinal. Modifique velocidades.")
while abs(v2-v1) > prec:
    vp = (v1+v2)/2                              # Média entre 'v1' e 'v2'
    rp = array([ya,vp],float)                   # Vetor 'r' inicial correspondente
    alturap = r_final(f,ta,tb,rp,dt)[0] - yb # Altura relativa correspondente
    if (altura1 * alturap) > 0:                 # Altura final menor que desejada?
        v1, altura1 = vp, alturap               # Sim; aumentamos o palpite 'v1'
    else:
        v2, altura2 = vp, alturap               # Não; diminuímos o palpite 'v2'
v = (v1+v2)/2       # Resultado final do cálculo
```

$$g = 9.81$$

$$\delta = 10^{-10}$$

```
Tempo de execução (em segundos): 2.0661580562591553
A velocidade inicial necessária é 49.049999999996786 m/s
>>>
```

Digressão: o método de Newton

- Se quisermos encontrar a raiz x_0 de uma equação (linear ou não linear)

$$f(x_0)=0,$$

um método em geral mais eficiente do que o da busca binária é o método de Newton, cuja base é a expansão em série de Taylor da função $f(x)$. Partindo de um palpite x_1 para a raiz, temos

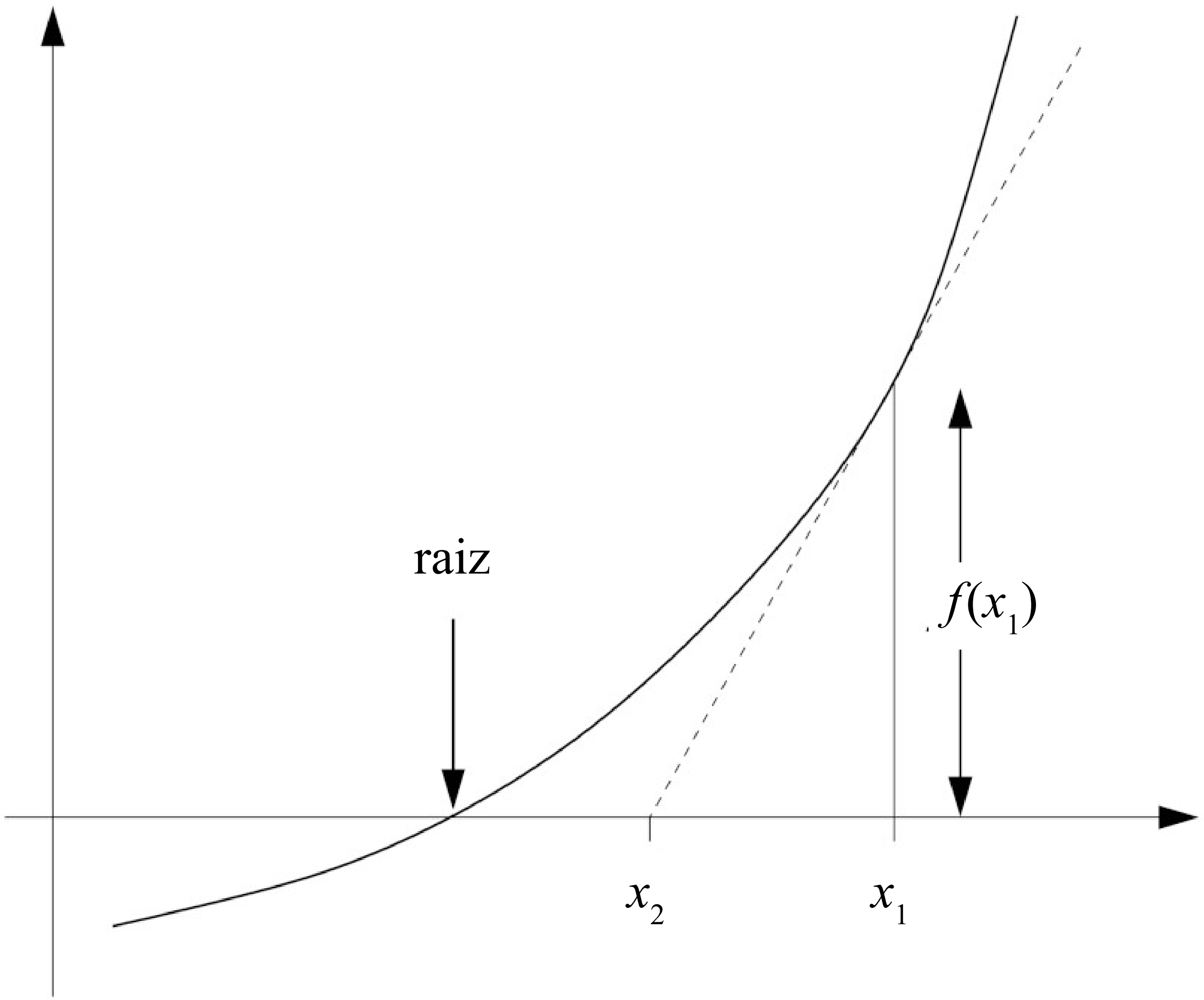
$$0=f(x_0)\simeq f(x_1)+(x_0-x_1)f'(x_1)$$

$$\Rightarrow x_0\simeq x_1-\frac{f(x_1)}{f'(x_1)}=x_2$$

Digressão: o método de Newton

$$x_0 \simeq x_1 - \frac{f(x_1)}{f'(x_1)} = x_2$$

Iterar a equação acima (substituindo x_1 por x_2 para obter um x_3, e assim por diante) leva rapidamente ao valor correto de x_0 (se o palpite inicial for suficientemente próximo de x_0).



Digressão: o método da secante

- O método de Newton requer o conhecimento da derivada da função $f(x)$:

$$0 = f(x_0) \simeq f(x_1) + (x_0 - x_1) f'(x_1).$$

- Se não dispomos da derivada analiticamente, podemos estimá-la numericamente:

$$f'(x_1) \simeq \frac{f(x_2) - f(x_1)}{x_2 - x_1}.$$

- Essa formulação dá origem ao método da secante, baseado na iteração da equação

$$x_0 \simeq x_1 - f(x_1) \frac{x_2 - x_1}{f(x_2) - f(x_1)} = x_3.$$

Exemplo 1b

$$\frac{d^2 y}{dt^2} = -g$$

$$y(0) = 0,$$
$$y(10) = 0$$

$$g = 9.81$$

$$\delta = 10^{-10}$$

```
# Solução via método da secante
v1 = 1   # Primeiro palpite de uma velocidade
v2 = 2   # Segundo palpite de uma velocidade
r1 = array([ya,v1],float)   # Vetor 'r' inicial correspondente a 'v1'
altura2 = r_final(f,ta,tb,r1,dt)[0] - yb # Altura final relativa com 'v1'
while abs(v2-v1) > prec:
    altura1 = altura2
    r2 = array([ya,v2],float)
    altura2 = r_final(f,ta,tb,r2,dt)[0] - yb # Altura final relativa com 'v2'
    v1, v2 = v2, v2 - altura2*(v2-v1)/(altura2-altura1)
v = v2      # Resultado final do cálculo
```

```
Tempo de execução (em segundos): 0.14392518997192383
A velocidade inicial necessária é 49.050000000000225 m/s
>>>
```

Problemas de autovalores

- **ATENÇÃO: este tópico é apresentado apenas como uma ilustração do que é possível fazer com as técnicas que estudamos. O conteúdo não será cobrado.**
- No curso de Física 2 oferecido em 2019-2, fizemos uso do fato de que os valores permitidos de energia para um oscilador harmônico quântico são igualmente espaçados. Vamos utilizar as técnicas de integração que aprendemos para “demonstrar” esse fato.

A equação de Schrödinger independente do tempo

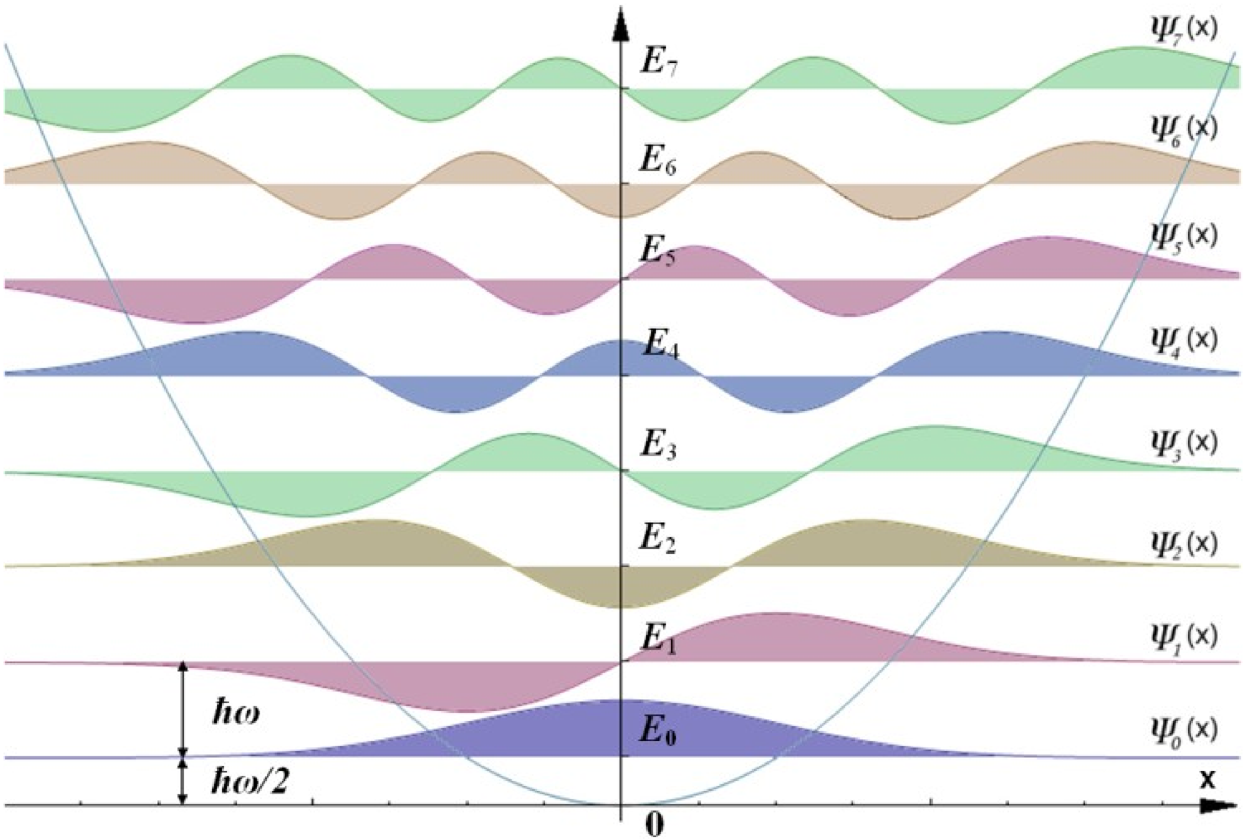
- Em uma dimensão, a equação de Schrödinger independente do tempo é

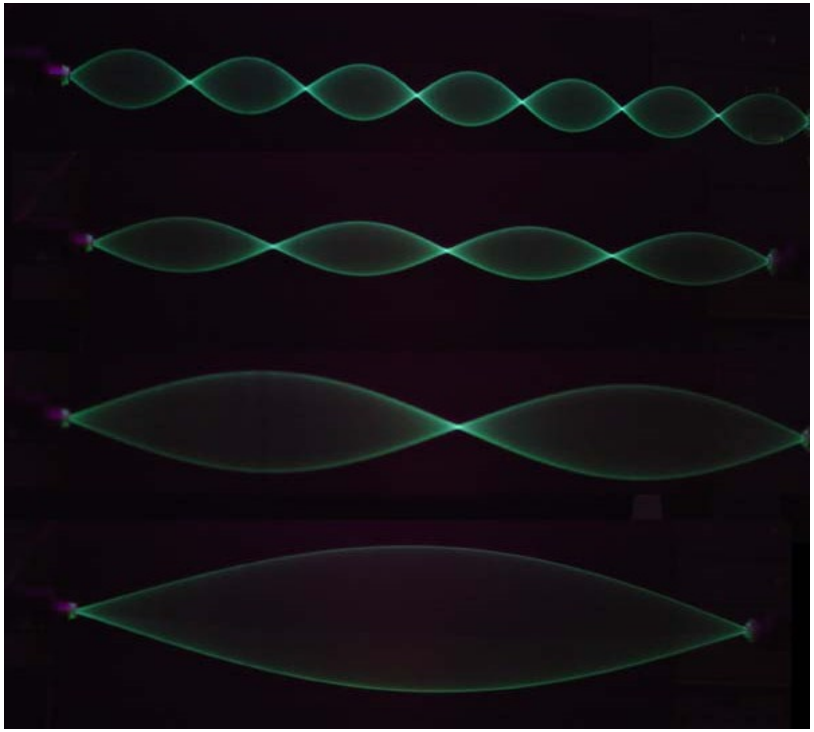
$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}+V(x)\psi=E\psi,$$

sendo $V(x)$ a energia potencial e E a energia total do sistema. Assim como em uma corda presa nas extremidades, nem todos os valores da energia são permitidos, mas apenas aqueles que satisfazem as **condições de contorno** de que a função de onda se anula nos limites do intervalo de x (que agora é a variável independente).

O oscilador harmônico quântico

$$V(x)=\frac{1}{2}kx^2, \quad \omega=\sqrt{\frac{k}{m}}, \quad E_n=\left(n+\frac{1}{2}\right)\hbar\omega$$





Níveis de energia igualmente espaçados são característicos do oscilador harmônico.

O oscilador harmônico quântico

- Nesse caso, a equação de Schrödinger independente do tempo pode ser escrita como

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}+\frac{1}{2}kx^2\psi=E\psi \;\Rightarrow\; \frac{d^2\psi}{dx^2}=\frac{2m}{\hbar^2}\left(\frac{m\omega^2}{2}x^2-E\right)\psi.$$

- Com as grandezas $\hbar$, m e ω, podemos construir combinações l e ε com dimensões de comprimento e de energia:

$$l=\sqrt{\frac{\hbar}{m\omega}} \quad , \quad \varepsilon=\hbar\omega.$$

O oscilador harmônico quântico

- Nesse caso, a equação de Schrödinger independente do tempo pode ser escrita como

$$-\frac{\hbar^2}{2m}\frac{d^2\psi}{dx^2}+\frac{1}{2}kx^2\psi=E\psi \;\Rightarrow\; \frac{d^2\psi}{dx^2}=\frac{2m}{\hbar^2}\left(\frac{m\omega^2}{2}x^2-E\right)\psi.$$

- Se medirmos os comprimentos em unidades de l e as energias em unidades de ε, a equação de Schrödinger torna-se

$$\frac{d^2\psi}{d(x/l)^2}=\left[\left(\frac{x}{l}\right)^2-2\frac{E}{\varepsilon}\right]\psi.$$

O oscilador harmônico quântico

- Nosso problema numérico agora é descobrir os valores de E e as funções ψ correspondentes que satisfazem a equação

$$\frac{d^2\psi}{d(x/l)^2}=\left[\left(\frac{x}{l}\right)^2-2\frac{E}{\varepsilon}\right]\psi$$

com as condições de contorno de que ψ vai a zero quando x tende a $\pm\infty$. Sendo impossível atingir numericamente esses limites, utilizamos em vez disso o intervalo entre $x_a=-6l$ e $x_b=+6l$, supondo $\psi_a\equiv\psi(x_a)=0$ e $\psi_b\equiv\psi(x_b)=0$.

O oscilador harmônico quântico

- Nosso problema numérico agora é descobrir os valores de E e as funções ψ correspondentes que satisfazem a equação

$$\frac{d^2\psi}{d\left(x/l\right)^2}=\left[\left(\frac{x}{l}\right)^2-2\frac{E}{\varepsilon}\right]\psi.$$

- Portanto, queremos resolver para E a equação

$$\psi\left(x_b;E\right)=0,$$

com $\psi\left(x_b;E\right)$ obtida, dada E, pela integração da equação de Schrödinger entre x_a e x_b.

O oscilador harmônico quântico

- Nosso problema numérico agora é descobrir os valores de E e as funções ψ correspondentes que satisfazem a equação

$$\frac{d^2\psi}{d(x/l)^2}=\left[\left(\frac{x}{l}\right)^2-2\frac{E}{\varepsilon}\right]\psi .$$

- E quanto às condições iniciais? Além de $\psi_a \equiv \psi(x_a)=0$, precisamos da derivada de ψ em relação a x no ponto x_a. Mas a equação de Schrödinger é linear em ψ, logo um múltiplo da solução também é solução. Assim, qualquer derivada não nula em x_a pode ser utilizada.

O oscilador harmônico quântico

- Nosso problema numérico agora é descobrir os valores de E e as funções ψ correspondentes que satisfazem a equação

$$\frac{d^2\psi}{d(x/l)^2}=\left[\left(\frac{x}{l}\right)^2-2\frac{E}{\varepsilon}\right]\psi.$$

- Ao final, podemos “normalizar” a função de onda, uma vez que seu módulo quadrado associa-se à probabilidade de encontrar a partícula em torno de um certo ponto.

$$\int_{x_a}^{x_b}|\psi(x)|^2dx=1$$

Exemplo 2

```python
# Supomos aqui que os comprimentos são medidos em unidades de
# l = sqrt(hbar/(m*omega)) e as energias são medidas em unidades
# de hbar*omega, sendo hbar a constante de Planck dividida por
# 2*pi e omega = sqrt(k/m).

xa = -6.             # Início do intervalo da variável independente
xb = 6.              # Final do intervalo da variável independente
psia = 0.0           # Valor da variável independente no início do intervalo
psib = 0.0           # Valor alvo da variável independente no final do intervalo
h = 0.01             # Tamanho do passo de integração
prec = 1e-10         # Precisão do resultado para a energia

def V(x):
    return x**2/2

def f(r,x,E):
    psi, dpsi = r[0], r[1]
    f0, f1 = dpsi, 2*(V(x)-E)*psi
    return array([f0,f1],float)

def passo_rk4(f,r,x,h,E):                  # Calcula um passo no método de RK4
    k1 = h*f(r,x,E)
    k2 = h*f(r+0.5*k1,x+0.5*h,E)
    k3 = h*f(r+0.5*k2,x+0.5*h,E)
    k4 = h*f(r+k3,x+h,E)
    return (k1+2.0*(k2+k3)+k4)/6.0

# Função que integra a equação diferencial definida por f entre 'xa' e 'xb'
# com condição inicial 'ra' e retorna o valor final do vetor 'r'
def r_final(f,xa,xb,ra,h,E):
    r = ra.copy() # Não utilizar 'copy' faz com que mudar 'r' afete 'ra'
    for x in arange(xa,xb,h):
        r += passo_rk4(f,r,x,h,E)
    return r
```

Exemplo 2

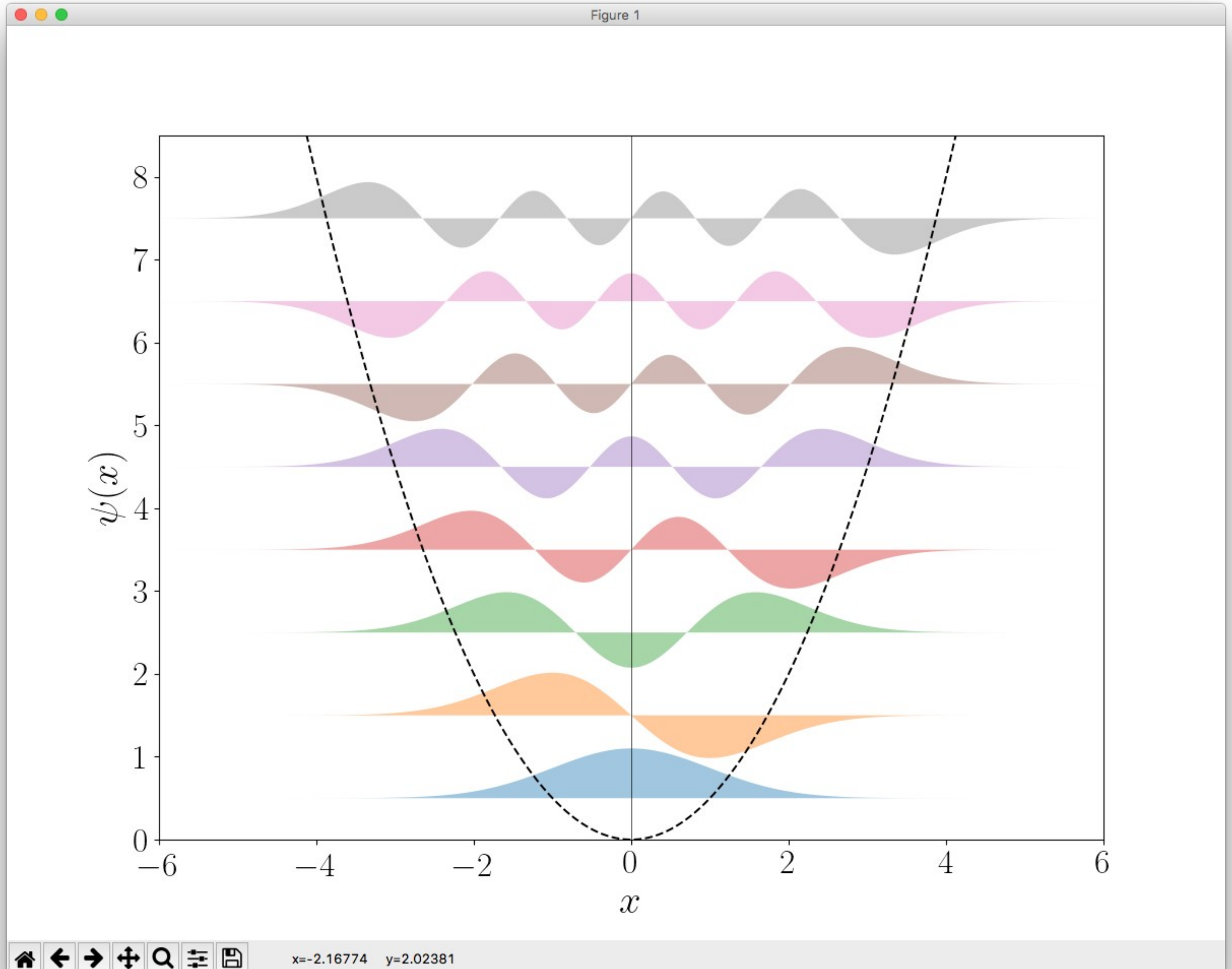
```python
# Função que calcula a função de onda associada a uma energia E.
# O resultado é normalizado para que a integral do quadrado da
# função de onda seja igual a 1. Há também uma variável de
# reescala ('esc') que permite acomodar várias curvas em um só gráfico.
def produz_lista_psi(x_lista,ra,E,esc):
    r = ra
    psi_lista = []
    norm = 0.0
    for x in x_lista:
        psi_lista.append(r[0])
        norm += h*r[0]**2 # Calculando a integral do quadrado da função de onda
        r += passo_rk4(f,r,x,h,E)
    norm = sqrt(norm)
    psi_lista[:] = [esc*psi/norm + E for psi in psi_lista]
    return psi_lista

# Função que determina uma energia permitida, impondo que a função de onda
# seja nula nos extremos do intervalo de x
def determina_E(E1,E2,ra,prec): # Solução via método da secante
    psi2 = r_final(f,xa,xb,ra,h,E1)[0] # Valor de psib com energia E1
    while abs((E2-E1)/((E1+E2)/2)) > prec:
        psi1 = psi2
        psi2 = r_final(f,xa,xb,ra,h,E2)[0] # Valor de psib com energia 'E2'
        E1, E2 = E2, E2 - psi2*(E2-E1)/(psi2-psi1)
    return (E1+E2)/2       # Resultado final do cálculo
```

Exemplo 2

```python
dpsi = 1e-6
ra = array([psia,dpsi],float)
E_lista = []

#
# Determinando as energias
#
nmax = 8 # Número de energias a determinar
for n in range(nmax):
    E1 = n+0.01       # Dois palpites para iniciar a solução
    E2 = E1 + 0.1
    E_lista.append(determina_E(E1,E2,ra,prec))
    print("A energia do nível",n,"é",E_lista[n])

# Produzindo gráficos das funções de onda e do potencial
esc=0.8
plt.figure(figsize=(12,9))
x_lista = arange(xa,xb,h)
V_lista = []
for x in x_lista:
    V_lista.append(V(x))
plt.plot(x_lista,V_lista,'k--')
plt.xlabel("$x$")
plt.ylabel("$\psi(x)$")
plt.axvline(0,0,E_lista[-1]+1,color='k',linewidth=0.5)
plt.xlim(xa,xb)
plt.ylim(0.0,1+E_lista[-1])
for E in E_lista:
    plt.fill_between(x_lista,E,produz_lista_psi(x_lista,ra,E,esc),alpha=0.5)
plt.show()
```

```
A energia do nível 0 é 0.5000000000504714
A energia do nível 1 é 1.5000000007811645
A energia do nível 2 é 2.50000003380626
A energia do nível 3 é 3.500000009192828
A energia do nível 4 é 4.500000020434955
A energia do nível 5 é 5.500000506696125
A energia do nível 6 é 6.500000212865547
A energia do nível 7 é 7.5000013518996616
```

Exemplo 2



Compare com a figura no slide 12

Exemplo 3

- Um oscilador quártico pode ser descrito pela "equação de Schrödinger"

$$\frac{d^2\psi}{d(x/l)^2} = \left[\beta\left(\frac{x}{l}\right)^4 - 2\frac{E}{\varepsilon}\right]\psi,$$

em que agora l e ε não mais correspondem aos valores associados à constante de mola. Vamos supor que definem unidades arbitrárias.

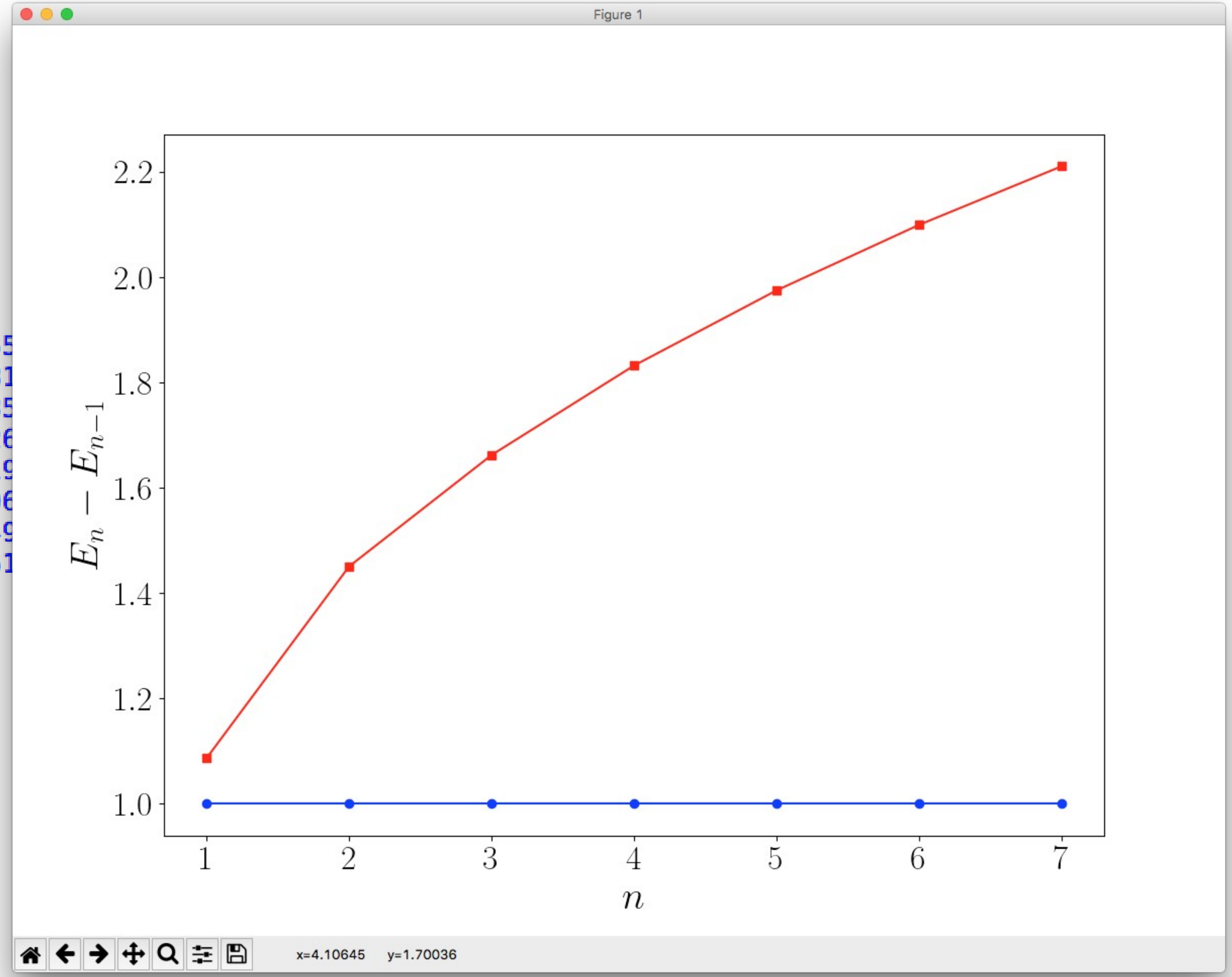
Exemplo 3

$$\frac{d^2\psi}{d(x/l)^2} = \left[\beta\left(\frac{x}{l}\right)^4 - 2\frac{E}{\epsilon}\right]\psi,$$

```
# Construímos uma equação semelhante à de Schrödinger, mas com unidades
# arbitrárias

xa = -4.          # Início do intervalo da variável independente
xb = 4.           # Final do intervalo da variável independente
psia = 0.0        # Valor da variável independente no início do intervalo
psib = 0.0        # Valor alvo da variável independente no final do intervalo
h = 0.01          # Tamanho do passo de integração
prec = 1e-14      # Precisão do resultado para a energia

beta=0.25
def V(x):
    return beta*x**4

def f(r,x,E):
    psi, dpsi = r[0], r[1]
    f0, f1 = dpsi, 2*(V(x)-E)*psi
    return array([f0,f1],float)

# Produzindo gráficos da energia e comparando com a do oscilador harmônico
plt.figure(figsize=(12,9))
n_lista = array(range(1,nmax))
DeltaEohs, DeltaE = [], []
DeltaEohs[:] = [1 for i in range(1,nmax)]
DeltaE[:] = [E_lista[i]-E_lista[i-1] for i in range(1,nmax)]
plt.xlabel("$n$")
plt.ylabel("$E_n - E_{n-1}$")
plt.plot(n_lista,DeltaEohs,'b-',n_lista,DeltaE,'r-')
plt.plot(n_lista,DeltaEohs,'bo',n_lista,DeltaE,'rs')
plt.show()
```

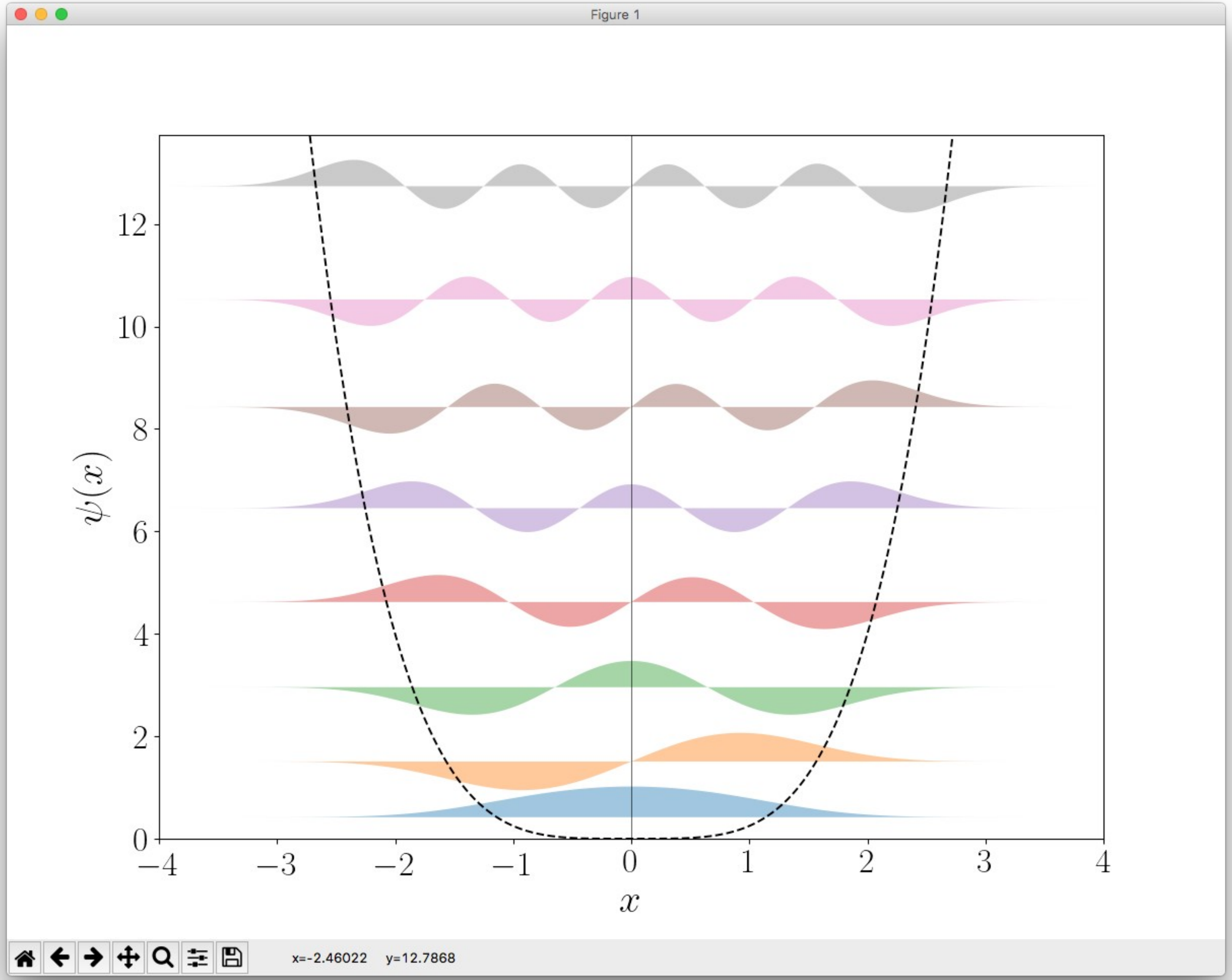
Exemplo 3

```
A energia do nível 0 é 0.420804974455
A energia do nível 1 é 1.507901242481
A energia do nível 2 é 2.958795696935
A energia do nível 3 é 4.621220353926
A energia do nível 4 é 6.453510027719
A energia do nível 5 é 8.428454092006
A energia do nível 6 é 10.52783119549
A energia do nível 7 é 12.73833777161
```



Diferença de energia entre níveis consecutivos, tanto para o oscilador harmônico quanto para o oscilador quártico.

Exemplo 2



Compare com a figura no slide 22

Exercícios no moodle

- Há um exercício, explorando o conteúdo da aula de hoje, que pode ser feito com base nos programas do exemplo 1, que estão disponíveis no moodle. A data para entrega é o dia **6 de maio**.
- O EP 3 está liberado. São dois problemas, o primeiro valendo 70% e o segundo, 30% da nota. A data para entrega é o dia **13 de maio**.