

---

# Folhas de problemas de Física computacional

Problemas seleccionados e traduzidos do livro

Computational Physics

Mark Newman

Ano lectivo 2018/2019

Docentes:

João Manuel Viana Parente Lopes

José Miguel Nunes da Silva

João Manuel Borregana Lopes dos Santos

---

# Física Computacional

---

Ano lectivo 2017/2018

---

## Folha 1 - Programação em Python para Físicos

### 1. Exercício

A órbita espacial de um corpo em torno de outro, como um planeta em torno do Sol, não é necessariamente circular. Em geral assume a forma de uma elipse, com o corpo por vezes mais próximo e por vezes mais afastado. Se na máxima aproximação de um planeta ao Sol, também chamado de *periélio*, a distância é  $\ell_1$  e sua velocidade linear  $v_1$ , então qualquer outra propriedade da órbita pode ser calculado a partir destes dois valores como se segue.

1. A segunda lei de Kepler diz-nos que a distância  $\ell_2$  e a velocidade  $v_2$  do planeta no ponto de máximo afastamento, o *afélio*, satisfaz  $\ell_2 v_2 = \ell_1 v_1$ . Também é sabido que a energia mecânica total, cinética mais potencial gravítica, do planeta com velocidade  $v$  a uma distância  $r$  do Sol é dada por

$$E = \frac{1}{2}mv^2 - G\frac{mM}{r},$$

onde  $m$  é a massa do planeta,  $M = 1.9891 \times 10^{30}$  kg a massa do Sol, e  $G = 6.6738 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2}$  a constante de gravitação universal. Dado que a energia mecânica deve ser conservada, mostre que  $v_2$  é a menor das raízes da equação quadrática

$$v_2^2 - \frac{2GM}{v_1 \ell_1} v_2 - \left[ v_1^2 - \frac{2GM}{\ell_1} \right] = 0.$$

Uma vez obtido o valor de  $v_2$  podemos calcular  $\ell_2$  usando a relação  $\ell_2 = \ell_1 v_1 / v_2$ .

2. Conhecidos os valores de  $v_1$ ,  $\ell_1$ , e  $\ell_2$ , outros parâmetros da órbita são dados por expressões simples obtíveis da geometria elíptica da órbita e das leis de Kepler:

$$\text{Semi-eixo maior: } a = \frac{1}{2}(\ell_1 + \ell_2),$$

$$\text{Semi-eixo menor: } b = \sqrt{\ell_1 \ell_2},$$

$$\text{Período orbital: } T = \frac{2\pi ab}{\ell_1 v_1},$$

$$\text{Excentricidade orbital: } e = \frac{\ell_2 - \ell_1}{\ell_2 + \ell_1}.$$

Desenvolva um programa que peça ao utilizador a distância do planeta ao Sol e a sua velocidade no periélio e, depois, calcule e mostre os valores das quantidades  $\ell_2$ ,  $v_2$ ,  $T$ , e  $e$ .

3. Teste o programa fazendo o cálculo das propriedades orbitais da Terra (para a qual  $\ell_1 = 1.4710 \times 10^{11}$  m e  $v_1 = 3.0287 \times 10^4 \text{ m s}^{-1}$ ) e do cometa Halley ( $\ell_1 = 8.7830 \times 10^{10}$  m e  $v_1 = 5.4529 \times 10^4 \text{ m s}^{-1}$ ). Entre outras coisas, deve obter para a Terra um período orbital de um ano, e de 76 anos para o cometa Halley.

## 2. Exercício

Os números de Catalan  $C_n$  são uma sequência de inteiros 1, 1, 2, 5, 14, 42, 132... com um papel importante em Mecânica Quântica e na Teoria dos Sistemas Desordenados. (Estão na base da demonstração por Eugene Wigner da famosa 'Lei do Semicírculo'.) Estes números podem ser definidos por recorrência:

$$C_0 = 1, \quad C_{n+1} = \frac{4n+2}{n+2} C_n.$$

Desenvolva um programa que mostre, por ordem crescente, todos os números de Catalan até um milhar de milhão.

## 3. Exercício

Em Física da Matéria Condensada a constante de Madelung representa a energia potencial elétrica total de um ião na rede cristalina. Ela depende das cargas dos outros iões vizinhos e das suas localizações. Considere, por exemplo, o caso do Cloreto de Sódio (sal das cozinhas) onde os iões estão dispostos alternadamente numa rede cúbica simples: o de sódio com uma carga positiva  $+e$  e o de cloro com uma carga negativa  $-e$ , onde  $e$  é a carga do elétron. Se etiquetarmos cada nodo da rede por três números inteiros  $(i, j, k)$ , então os iões sódio ficam nas posições onde  $i + j + k$  é par, e os iões cloro nas posições onde  $i + j + k$  é ímpar.

Consideremos o ião sódio situado na origem,  $i = j = k = 0$ , e calculemos a constante de Madelung. Se o espaçamento entre os iões for  $a$  (constante da rede), então a distância da origem ao ião na posição  $(i, j, k)$  é

$$\sqrt{(ia)^2 + (ja)^2 + (ka)^2} = a\sqrt{i^2 + j^2 + k^2},$$

e o potencial na origem criado por esse ião é

$$V(i, j, k) = \pm \frac{e}{4\pi\epsilon_0 a \sqrt{i^2 + j^2 + k^2}},$$

( $\epsilon_0$  é a permissividade do vácuo e o sinal da expressão depende de  $i + j + k$  ser par ou ímpar). O potencial total na origem  $V_{\text{total}}$  é então a soma sobre todas as restantes posições. Supondo um cristal de formato cúbico, e centrado na origem, com  $L$  iões em todas as direções, vem

$$V_{\text{total}} = \sum_{\substack{i, j, k = -L \\ \text{not } i=j=k=0}}^L V(i, j, k) = \frac{e}{4\pi\epsilon_0 a} M,$$

onde  $M$  é (no limite  $L \rightarrow \infty$ ) a constante de Madelung.

Desenvolva um programa que calcule a constante de Madelung do Cloreto de Sódio. Use um valor suficientemente grande para  $L$  mas que lhe permita correr o programa num tempo não superior a um minuto. Compare com o valor exato para o NaCl:  $-1,748$ .

## 4. Exercício

O coeficiente binomial  $\binom{n}{k}$  é um número inteiro igual a

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} = \frac{n \times (n-1) \times (n-2) \times \dots \times (n-k+1)}{1 \times 2 \times \dots \times k}$$

com  $k \geq 1$ , ou  $\binom{n}{0} = 1$  quando  $k = 0$ .

1. Usando este formulário para o coeficiente binomial, escreva uma função *binomial*( $n, k$ ) que calcula o coeficiente binomial para  $n$  e  $k$ . Certifique-se de que sua função retorna a resposta na forma de

---

um inteiro (não um float) e forneça o valor correto de 1 para o caso em que  $k = 0$ .

2. Usando sua função, escreva um programa para imprimir as primeiras 20 linhas do “triângulo de Pascal”. A linha  $n$  do triângulo de Pascal contém  $n + 1$  números, que correspondem aos coeficientes  $\binom{n}{0}$ ,  $\binom{n}{1}$ , até  $\binom{n}{n}$ . As primeiras linhas são,

1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1

3. A probabilidade de uma moeda justa, lançada  $n$  vezes, apresenta  $k$  resultados de caras é  $\binom{n}{k}/2^n$ . Escreva um programa que calcula (a) A probabilidade de uma moeda ser lançada 100 vezes e ter como resultado 60 caras, e (b) a probabilidade de ter como resultado pelo menos 60 caras.

## 5. Exercício

O programa no Exemplo 2.8 do livro adotado não corresponde ao modo mais eficiente de calcular números primos: verifica cada número para ver se é divisível por qualquer número menor que ele. Podemos desenvolver um programa muito mais rápido para números primos usando as seguintes observações:

1. Um número  $n$  é primo se não tiver nenhum fator primo menor que  $n$ . Por isso, só precisamos verificar se é divisível por outros primos.
2. Se um número  $n$  é não primo, tendo um fator  $r$ , então  $n = rs$ , onde  $s$  também é um fator. Se  $r \geq \sqrt{n}$  então  $n = rs \geq s\sqrt{n}$ , o que implica que  $s \leq \sqrt{n}$ . Em outras palavras, qualquer não-primo deve ter fatores e, portanto, também fatores primos, menores ou iguais a  $\sqrt{n}$ . Assim, para determinar se um número é primo, temos que verificar seus fatores primos apenas até e incluindo  $\sqrt{n}$  - se não houver nenhum, o número é primo.
3. Se encontrarmos um único fator primo menor que  $\sqrt{n}$ , então sabemos que o número não é primo e, portanto, não há necessidade de verificar mais nada - podemos abandonar esse número e passar para outra coisa..

Escreva um programa Python que encontre todos os primos até dez mil. Crie uma lista para armazenar os primos, contém inicialmente apenas um elemento, o número primo 2. Então, para cada número  $n$  de 3 a 10 000, verifique se o número é divisível por qualquer um dos primos da lista, até e incluindo  $\sqrt{n}$ . Assim que encontrar um único fator primo, poderá parar de verificar os restantes - sabe que  $n$  não é um primo. Se não encontrar fatores primos  $\sqrt{n}$  ou menos, então  $n$  é primo e você deve adicioná-lo à lista. Pode imprimir a lista de uma só vez no final do programa, ou pode imprimir os números individuais à medida que os encontrar.

## 6. Exercício

Um recurso útil de funções definidas pelo utilizador é a capacidade de uma função se chamar recursivamente. Por exemplo, considere a seguinte definição do fatorial  $n!$  de um inteiro positivo  $n$ :

$$n! = \begin{cases} 1 & \text{se } n = 1, \\ n \times (n-1)! & \text{se } n > 1. \end{cases}$$

Isto constitui uma definição completa do fatorial que nos permite calcular o valor de  $n!$  para qualquer inteiro positivo. Podemos empregar essa definição diretamente para criar uma função Python para fatoriais, como este:

```
def factorial(n):
    if n==1:
        return 1
    else:
        return n*factorial(n-1)
```

Note como, se  $n$  não for igual a 1, a função chama a si própria para calcular o fatorial de  $n - 1$ . A este procedimento chamamos recursão. Se executarmos `factorial(5)` o computador irá imprimir corretamente a resposta 120.

1. Encontramos os números de Catalan  $C_n$  no Exercício 2. Com apenas um pequeno rearranjo, a definição pode ser reescrita na forma

$$C_n = \begin{cases} 1 & \text{se } n = 0, \\ \frac{4n-2}{n+1} C_{n-1} & \text{se } n > 0. \end{cases}$$

Escreva uma função Python, usando recursão, que calcule  $C_n$ . Use essa função para calcular e imprimir  $C_{100}$ .

2. Euclides mostrou que o maior divisor comum  $g(m, n)$  de dois inteiros não negativos  $m$  e  $n$  satisfaz

$$g(m, n) = \begin{cases} m & \text{if } n = 0, \\ g(n, m \bmod n) & \text{if } n > 0. \end{cases}$$

Escreva uma função Python que empregue recursão para calcular o maior divisor comum de  $m$  e  $n$  usando essa fórmula. Use sua função para calcular e imprimir o maior divisor comum de 108 e 192.

Comparando o cálculo dos números de Catalan na parte (a) acima com o do Exercício 2, vemos que é possível fazer o cálculo de duas maneiras, diretamente ou usando recursão. Na maioria dos casos, se uma quantidade puder ser calculada sem recursão, será mais rápido fazer isso, e normalmente recomendamos que siga essa rota, se possível. Existem alguns cálculos, no entanto, que são essencialmente impossíveis (ou pelo menos muito mais difíceis) sem recursividade. Veremos alguns exemplos mais adiante no livro.

---

# Física Computacional

Ano lectivo 2017/2018

---

## Folha 2 - Representação gráfica de dados experimentais

### 1. Exercício

Nos recursos on-line do livro adotado existe um ficheiro de nome `sunspots.txt` que contém o número de manchas solares observadas em cada mês desde Janeiro de 1749. Este ficheiro tem duas colunas de números em que a primeira refere o mês e a segunda refere o número de manchas solares.

1. Desenvolva um programa que leia esses dados e trace um gráfico do número de manchas solares em função do tempo.
2. Modifique o programa para que mostre apenas os primeiros 1000 pontos do gráfico.
3. Adapte esse programa de forma a que também calcule e represente num gráfico as médias móveis definidas por

$$Y_k = \frac{1}{2r+1} \sum_{m=-r}^r y_{k+m},$$

onde  $r = 5$  (e os  $y_k$  são os números de manchas solares). Trace no mesmo gráfico os dados originais e as médias móveis, limitando a representação aos primeiros 1000 pontos.

### 2. Exercício

Embora a função `plot` seja primeiramente entendida para a construção de gráficos  $y$  em função de  $x$ , também pode ser adaptada a outros tipos de gráficos.

1. Obtenha um gráfico da curva conhecida por *deltoide* e que é definida parametricamente pelas equações

$$x = 2 \cos \theta + \cos 2\theta, \quad y = 2 \sin \theta - \sin 2\theta,$$

onde  $0 \leq \theta < 2\pi$ . Considere um conjunto de valores para  $\theta$  entre zero e  $2\pi$ , calcule os correspondentes  $x$  e  $y$  para cada das equações anteriores e, então, represente  $y$  como função de  $x$ .

2. Indo mais além, faça uma representação polar  $r = f(\theta)$ , com uma função  $f$  adequada, de forma a obter a conhecida lei de transformação de coordenadas cartesianas para coordenadas polares:  $x = r \cos \theta$ ,  $y = r \sin \theta$ . Use este método para obter o gráfico da Espiral Galileana  $r = \theta^2$  for  $0 \leq \theta \leq 10\pi$ .

3. Usando esse mesmo método, obtenha o gráfico da 'função de Fey'

$$r = e^{\cos \theta} - 2 \cos 4\theta + \sin^5 \frac{\theta}{12}$$

no intervalo  $0 \leq \theta \leq 24\pi$ .

### 3. Exercício

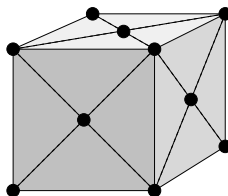
Nos recursos on-line do livro adotado existe um ficheiro de nome `stm.txt`, que contém uma grelha de valores medidos num microscópio de efeito-túnel da superfície (111) do silício. Estes valores representam as alturas na referida superfície. Desenvolva um programa que leia esses dados e depois trace um gráfico de densidades. Estude as melhores opções e variantes de forma a melhor representar a estrutura dessa superfície do silício.

### 4. Exercício

Usando como ponto de partida o programa do Exemplo 3.2 do livro adoptado, ou se preferir começando do início, faça o seguinte:

1. Um cristal de cloreto de sódio tem átomos de sódio e cloro dispostos em uma rede cúbica, mas os átomos alternam entre o sódio e o cloro, de modo que cada átomo sódio é cercado por seis átomos de cloro e cada átomo cloro é cercado por seis átomos de sódio. Crie uma visualização da estrutura de cloreto de sódio usando duas cores diferentes para representar os dois tipos de átomos.

2. A rede cúbica de face centrada (FCC), que é a rede cristalina mais comum na natureza, consiste de uma estrutura cúbica com átomos posicionados não apenas nos cantos de cada cubo, mas também no centro de cada face:



Faça uma visualização de uma rede do FCC com uma única espécie de átomo (como ocorre no ferro metálico, por exemplo).

### 5. Exercício

Os seis planetas mais profundos do nosso sistema solar giram em torno do Sol em órbitas aproximadamente circulares, todas localizadas aproximadamente no mesmo plano (elíptico). No seguinte quadro podemos observar os seus parâmetros:

Objecto	Raio (km)	Raio da órbita (milhões de km)	Período da órbita (dias)
Mercúrio	2440	57.9	88.0
Vénus	6052	108.2	224.7
Terra	6371	149.6	365.3
Marte	3386	227.9	687.0
Júpiter	69173	778.5	4331.6
Saturno	57316	1433.4	10759.2
Sol	695500	–	–

Usando as facilidades fornecidas pelo módulo *visual*, crie uma animação do sistema solar que mostre o seguinte:

1. O Sol e planetas como esferas nas suas posições apropriadas e com tamanhos proporcionais aos seus tamanhos reais. Como os raios dos planetas são minúsculos em comparação com as distâncias entre eles, represente os planetas por esferas com raios  $c_1$  vezes maiores que seus valores originais, para que possam ser visualmente distinguida a diferença de raios. Encontre um bom valor para  $c_1$  que torne os planetas visíveis. Também é necessário encontrar um bom raio para o sol. Escolha

qualquer valor que forneça uma visualização clara. Não funciona para reescalar o raio do Sol pelo mesmo fator dos planetas porque ficará muito grande. Para um realismo adicional, atribua cores diferentes às esferas. Por exemplo, a Terra pode ser azul e o Sol pode ser amarelo.

2. O movimento dos planetas em torno do Sol (fazendo as esferas dos planetas moverem-se). No interesse de aliviar o tédio, construa seu programa para que o tempo de animação seja um fator de  $c_2$  mais rápido que o tempo real. Encontre um bom valor de  $c_2$  que torne o movimento das órbitas facilmente visível, mas não excessivamente rápido. Faça uso da função *rate* para a animação funcionar sem problemas.

**Sugestão:** Pode ser útil armazenar as coordenadas das esferas numa matriz do tipo descrito na página 115 do livro adotado.

## 6. Exercício

Um dos exemplos mais famosos de caos é o *mapa logístico*, definido pela equação

$$x' = rx(1 - x)$$

Para um dado valor da constante  $r$ , escolhemos um valor de  $x$  - digamos  $x = 1/2$  - e avaliamos o lado direito equação para calcular  $x'$ . Fazendo  $x = x'$  recalculamos o lado direito, e obtemos um novo valor  $x'$  e assim por diante. Este é um *mapa iterativo*. Fazendo a mesma operação repetidamente, acontece uma de três coisas a  $x$ :

1. O valor de  $x$  deixa de variar (ponto fixo). Por exemplo,  $x = 0$  é sempre um ponto fixo do mapa logístico (colocando  $x = 0$  no lado direito e obtemos  $x' = 0$  à esquerda).
2. O valor de  $x$  não converge para um valor único, mas estabelece um padrão periódico, percorrendo periodicamente um conjunto de valores (ciclo limite).
3. O mapa gera uma sequência aparentemente aleatória de números que parecem não ter nenhum padrão. Este é o caos determinista. *Caos* porque parece *caótico* e *determinista* porque, embora os valores pareçam aleatórios, não o são. São totalmente previsíveis, porque são gerados por uma equação a partir de uma condição inicial. O comportamento é determinado, embora possa não o parecer.

Escreva um programa que calcule e exiba o comportamento do mapa logístico. Para um determinado valor de  $r$ , comece com  $x = 1/2$  e itere a equação do mapa logístico mil vezes. Esta iteração inicial permite chegar a um ponto fixo ou ciclo limite. De seguida, execute por mais de 1.000 iterações e represente os pontos  $(r, x)$  num gráfico com  $r$  no eixo horizontal e  $y$  no eixo vertical (pode usar a função *plot* com as opções “*ko*” ou “*k*.” para representar pontos ou a função *scatter*). Repita o cálculo para diferentes valores de  $r$  de 1 a 4 em passos de 0,01, representando os pontos para todos os valores de  $r$  na mesma figura e, finalmente, usando a função *show* uma vez para exibir a figura completa.

O seu programa deve gerar um gráfico muito típico que se parece com uma árvore inclinada para o lado. Esta famosa figura é chamada de *Feigenbaum plot*, depois da sua descoberta por Mitchell Feigenbaum, ou às vezes por *figtree plot*, uma alusão ao fato às semelhanças com uma árvore e ao facto que Feigenbaum significar *figueira* em alemão.

Responda às seguintes perguntas:

1. Para um dado valor de  $r$ , como seria um ponto fixo no gráfico de Feigenbaum? E um ciclo limite? E como seria o caos?
2. Com base no gráfico que obteve, em que valor de  $r$  o sistema move-se de um comportamento ordenado (pontos fixos ou ciclos de limite) para um comportamento caótico? Este ponto é às vezes chamado de *fronteira do caos*.



O mapa logístico é um sistema matemático muito simples, mas o caos determinístico é observado em muitos sistemas físicos mais complexos, incluindo a dinâmica de fluidos e o clima. Por causa de sua natureza aparentemente aleatória, o comportamento de sistemas caóticos é difícil de prever e é fortemente afetado por pequenas perturbações nas condições externas. Provavelmente já ouviu falar do exemplo clássico do caos na meteorologia, o *efeito borboleta*, que foi popularizado pelo físico Edward Lorenz em 1972, numa palestra dada na *Associação Americana para o Avanço da Ciência*, intitulada “*O bater de asas de uma borboleta no Brasil desencadeou um tornado no Texas?*” (embora a primeira pessoa a sugerir o efeito borboleta não fosse um físico, mas o escritor de ficção científica Ray Bradbury no seu famoso conto de 1952, *A Sound of Thunder*, em que a destruição descuidada de uma borboleta pelo viajante do tempo durante uma viagem turística à era jurássica muda o curso da história).

**Comentário:** Existe outra abordagem para calcular o gráfico Feigenbaum, que é mais puro e rápido, fazendo uso da capacidade do Python de realizar aritmética com arrays. Crie um array  $r$  com um elemento contendo cada valor distinto de  $r$  que quer investigar:  $[1.0, 1.01, 1.02, \dots]$ . Em seguida, crie outra matriz  $x$  do mesmo tamanho para manter os valores correspondentes de  $x$ , que devem ser inicialmente definidos como 0, 5. Numa iteração do mapa logístico podemos executar para todos os valores de  $r$  de uma só vez com uma declaração da forma  $x = r * x * (1 - x)$ . Devido à velocidade com a qual o Python pode executar cálculos em matrizes, esse método deve ser significativamente mais rápido do que o método mais básico acima.

## 7. Exercício

O conjunto de Mandelbrot, batizado com o nome do matemático francês Benoît Mandelbrot, é um fractal, um objeto matemático infinitamente ramificado que contém estrutura dentro da estrutura e dentro da estrutura, tão fundo quanto observemos. A definição do conjunto Mandelbrot pode ser feita através de números complexos.

Considere a equação

$$z' = z^2 + c,$$

onde  $z$  é um número complexo e  $c$  é uma constante complexa. Para qualquer valor dado de  $c$ , esta equação transforma um número de entrada  $z$  num número de saída  $z'$ . A definição do conjunto Mandelbrot envolve a iteração repetida dessa equação: a partir de um valor inicial inicial de  $z$ , calculamos a equação para obter um novo valor  $z'$ . Com esse novo valor o fornecemos novamente para obter outro valor, e assim por diante. O conjunto Mandelbrot é o conjunto de pontos no plano complexo que satisfaz a seguinte definição:

Para um dado valor complexo de  $c$ , comece com  $z = 0$  e repita o procedimento iterativo.

Se a magnitude  $|z|$  do valor resultante alguma vez for maior que 2, então o ponto no plano complexo na posição  $c$  não está no conjunto Mandelbrot, caso contrário, ele está no conjunto.

Para usar essa definição, seria necessário, em princípio, iterar infinitas vezes para provar que um ponto está no conjunto Mandelbrot, já que um ponto está no conjunto apenas se a iteração *nunca* passar  $|z| = 2$ . Na prática, no entanto, apenas executamos um grande número de iterações, digamos 100, e se  $|z|$  não excedeu 2, assumimos que não o fará. Escreva um programa para fazer uma imagem do conjunto Mandelbrot executando a iteração para todos os valores de  $c = x + iy$  numa grelha  $N \times N$  abrangendo a região onde  $-2 \leq x \leq 2$  e  $-2 \leq y \leq 2$ . Faça um gráfico de densidade em que os pontos da grelha dentro do conjunto de Mandelbrot são coloridos em preto e os de fora são coloridos em branco. O conjunto Mandelbrot tem uma forma muito distinta que se parece com um escaravelho com um focinho comprido - saberá quando o vir.

**Sugestão:** Provavelmente achará útil começar com uma grelha bastante grossa, ou seja, com um pequeno valor de  $N$  - talvez  $N = 100$  - para que seu programa seja executado rapidamente enquanto testa. Quando tiver certeza de que está funcionando corretamente, aumente o valor de  $N$  para produzir uma imagem final de alta qualidade da forma do conjunto.

---

Se estiver entusiasmado, pode fazer outra variante do mesmo exercício que produz imagens incríveis. Em vez de pontos de cor apenas preto ou branco, atribua a cor de acordo com o número de iterações da equação antes de  $|z|$  se tornar maior que 2 (ou o número máximo de iterações se  $|z|$  nunca se tornar maior que 2). Se usar um dos esquemas de cores que o Python fornece para gráficos de densidade, como os esquemas *“hot”* ou *“jet”*, pode fazer algumas imagens espetaculares. Outra variante interessante é colorir de acordo com o logaritmo do número de iterações, o que ajuda a revelar algumas das estruturas mais finas fora do conjunto.

---

# Física Computacional

Ano lectivo 2017/2018

---

## Folha 3 - Exactidão e velocidade

### 1. Exercício

Escreva um programa para calcular e imprimir o fatorial de um número digitado pelo utilizador. Se quiser, pode basear seu programa na função para fatorial na Seção 2.6, mas escreva seu programa para que calcule o fatorial usando variáveis inteiras, não variáveis de ponto flutuante. Use seu programa para calcular o fatorial de 200.

Modifique seu programa para usar variáveis de ponto flutuante e, novamente, calcule o fatorial de 200? Discuta o resultado.

### 2. Exercício

1. Escreva um programa que tome como dados três números,  $a$ ,  $b$  e  $c$ , e imprima as duas soluções na equação quadrática  $ax^2 + bx + c = 0$  usando a fórmula resolvente,

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Use seu programa para calcular as soluções de  $0.001x^2 + 1000x + 0.001 = 0$ .

2. Existe outra maneira de escrever as soluções para uma equação quadrática. Multiplicando a parte superior e inferior da solução acima por  $-b \mp \sqrt{b^2 - 4ac}$ , mostre que as soluções também podem ser escritas como

$$x = \frac{2c}{-b \mp \sqrt{b^2 - 4ac}}.$$

Adicione código ao seu programa para imprimir esses valores além dos anteriores e use novamente o programa para resolver  $0.001x^2 + 1000x + 0.001 = 0$ . O que vê? Como o explica ?

3. Usando o que aprendeu, escreva um novo programa que calcule ambas as raízes de uma equação quadrática com precisão em todos os casos.

Este é um bom exemplo de como os computadores nem sempre funcionam como se espera. Se aplicar simplesmente a fórmula padrão para a equação quadrática, pode receber uma resposta errada. Na prática, o método que elaborou aqui é a maneira correta de resolver uma equação quadrática num computador, mesmo que seja mais complicado do que a fórmula padrão. Se estivesse em escrever um programa que envolvesse a solução de muitas equações quadráticas, este método poderia ser um bom candidato para definir função: poderia colocar os detalhes do método dentro de uma função para evitar o problema de passar passo a passo todas as vezes que tiver uma nova equação para resolver.

### 3. Exercício

Suponha que temos função  $f(x)$  e queremos calcular sua derivada num ponto  $x$ . Podemos fazer isso com lápis e papel, se soubermos a forma matemática da função, ou podemos fazê-lo no computador, fazendo uso da definição da derivada:

$$\frac{df}{dx} = \lim_{\delta \rightarrow 0} \frac{f(x + \delta) - f(x)}{\delta}.$$

No computador, não podemos realmente tomar o limite, pois  $\delta$  vai para zero, mas podemos obter uma aproximação razoável apenas fazendo  $\delta$  pequeno.

1. Escreva um programa que defina uma função  $f(x)$  que devolve o valor  $x(x - 1)$ , calcula a derivada da função no ponto  $x = 1$  usando a fórmula acima com  $\delta = 10^{-2}$ . Calcule o valor real da mesma derivada analiticamente e compare com a resposta do seu programa. Os dois não concordarão perfeitamente. Por que não?
2. Repita o cálculo para  $\delta = 10^{-4}, 10^{-6}, 10^{-8}, 10^{-10}, 10^{-12}$ , e  $10^{-14}$ . Verá que a precisão do cálculo inicialmente melhora à medida que o delta diminui, mas piora novamente. Porquê?

Vamos examinar derivadas numéricas em mais detalhe na Seção 5.10, onde estudaremos técnicas para lidar com esses problemas e maximizar a precisão de nossos cálculos.

### 4. Exercício

#### Cálculo de integrais

Suponha que queremos calcular o valor do integral

$$I = \int_{-1}^1 \sqrt{1 - x^2} dx.$$

o gráfico da função integranda é um semicírculo de raio um: e, portanto, o valor do integral—a área sob a curva—deve ser  $\pi/2 = 1.57079632679 \dots$

Alternativamente, podemos avaliar o integral no computador dividindo o domínio de integração em um grande número  $N$  de fatias de largura  $h = 2/N$  cada, e usando a definição de Riemann da integral:

$$I = \lim_{N \rightarrow \infty} \sum_{k=1}^N h y_k,$$

em que

$$y_k = \sqrt{1 - x_k^2} \quad \text{and} \quad x_k = -1 + hk.$$

Na prática, não podemos tomar o limite  $N \rightarrow \infty$ , mas podemos fazer uma aproximação razoável tornando  $N$  grande.

1. Escreva um programa para avaliar a integral acima com  $N = 100$  e compare o resultado com o valor exato. Os dois não concordarão muito bem, porque  $N = 100$  não é um número suficientemente grande de fatias.
2. Aumente o valor de  $N$  para obter um valor mais preciso para o integral. Se precisar que o programa seja executado em cerca de um segundo ou menos, qual a precisão que consegue?

O cálculo de integrais é uma tarefa comum em física computacional. Estudaremos técnicas para fazer integrais em detalhe no próximo capítulo. Como veremos, existem métodos substancialmente mais rápidos e precisos do que o que usamos aqui.

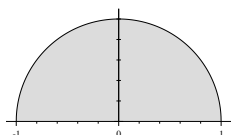


Figura 3.1:

---

# Física Computacional

Ano lectivo 2017/2018

---

## Folha 4 - Integrais e derivadas

### 1. Exercício

Nos recursos on-line encontra um ficheiro com o nome `velocities.txt` que contém duas colunas de números, a primeira representando o tempo  $t$  em segundos e a segunda a velocidade  $x$  em metros por segundo de uma partícula, medida uma vez por segundo desde o tempo  $t = 0$  até  $t = 100$ . As primeiras linhas têm o aspecto:

```
0 0
1 0.069478
2 0.137694
3 0.204332
4 0.269083
5 0.331656
```

Escreva um programa para fazer o seguinte:

1. Ler os dados e, usando a regra trapezoidal, calcular a distância aproximada percorrida pela partícula na direção  $x$  em função do tempo. Veja a Seção 2.4.3 na página 57 se quiser recordar como ler dados de um arquivo.
2. Fazer um gráfico que mostre, na mesma janela, a curva de velocidade original e a distância percorrida em função do tempo.

### 2. Exercício

1. Escreva um programa para calcular um valor aproximado para o integral  $\int_0^2 (x^4 - 2x + 1)dx$  do Exemplo 5.1, mas usando a regra de Simpson com 10 intervalos, em vez da regra trapezoidal. Pode querer basear seu programa no programa de regras trapezoidais na página 142.
2. Execute o programa e compare seu resultado com o valor correto conhecido de 4.4. Qual é o erro fraccionário no seu cálculo?
3. Modifique o programa para usar uma centena de intervalos, e depois mil. Observe a melhoria no resultado. Como se comparam os resultados aos do Exemplo 5.1 para a regra trapezoidal com o mesmo número de intervalos?

---

### 3. Exercício

---

Considere o integral

$$E(x) = \int_0^x e^{-t^2} dt.$$

1. Escreva um programa para calcular  $E(x)$  para valores de  $x$  de 0 a 3 em passos de 0.1. Escolha o método que você usará para executar a integral e um número adequado de intervalos.
2. Quando estiver convencido de que seu programa está a funcionar, complete-o para fazer um gráfico de  $E(x)$  como uma função de  $x$ . Se quiser lembrar-se de como fazer um gráfico, consulte a Seção 3.1, na página 88.

### 4. Exercício

---

#### Limite de difração de um telescópio

Nossa capacidade para resolver detalhes em observações astronômicas é limitada pela difração de luz nos telescópios. A luz das estrelas pode ser tratada efetivamente como proveniente de uma fonte pontual no infinito. Quando essa luz, com comprimento de onda  $\lambda$ , passa pela abertura circular de um telescópio (que assumimos ter raio unitário) e é focada pelo telescópio no plano focal, não produz um único ponto, mas um padrão circular de difração, consistindo num ponto central rodeado por uma série de anéis concêntricos. A intensidade da luz neste padrão de difração é dada por

$$I(r) = \left( \frac{J_1(kr)}{kr} \right)^2,$$

onde  $r$  é a distância no plano focal do centro do padrão de difração,  $k = 2\pi/\lambda$ , e  $J_1(x)$  é uma função de Bessel. As funções de Bessel,  $J_m(x)$ , são dadas por

$$J_m(x) = \frac{1}{\pi} \int_0^\pi \cos(m\theta - x \sin \theta) d\theta,$$

onde  $m$  é um inteiro não negativo e  $x \geq 0$ .

1. Escreva uma função Python  $J(m, x)$  que calcule o valor de  $J_m(x)$  usando a regra de Simpson com  $N = 1000$  pontos. Use a sua função num programa para fazer um gráfico, numa única janela, das funções de Bessel  $J_0, J_1$  e  $J_2$  como um funções de  $x$ , de  $x = 0$  a  $x = 20$ .
2. Escreva um segundo programa que faça um gráfico de densidade da intensidade do padrão de difração circular de uma fonte de luz pontual com  $\lambda = 500$  nm, num quadrado do plano focal, usando a fórmula dada acima. A sua imagem deve cobrir valores de  $r$  de zero a cerca de  $1 \mu m$ .

Sugestão 1: Você pode achar útil saber que  $\lim_{x \rightarrow 0} J_1(x)/x = 1/2$ . Sugestão 2: O ponto central no padrão de difração é tão brilhante que pode ser difícil ver os anéis ao seu redor no ecrã do computador. Se se deparar com este problema, uma maneira simples de lidar com ele é usar um dos outros esquemas de cores para gráficos de densidade descritos na Seção 3.3. O esquema `hot` funciona bem. Para uma solução mais sofisticada do problema, o `imshow` tem um argumento adicional `vmax` que permite definir o valor que corresponde ao ponto mais claro da gráfico. Por exemplo, se disser `imshow(x, vmax = 0.1)`, os elementos com valor 0,1, ou qualquer valor maior, produzirão a cor mais brilhante (mais positiva) na tela. Baixando `vmax`, pode reduzir o intervalo total de valores entre o brilho mínimo e máximo e, portanto, aumentar a sensibilidade do gráfico, tornando detalhes subtis visíveis. (Há também um argumento `vmin` que pode ser usado para definir o valor que corresponde à cor mais escura (mais negativa).) Para este exercício, um valor `vmax = 0.01` parece funcionar bem.

## 5. Exercício

### Erro na regra de Simpson

Seguindo o mesmo argumento que levou à Eq. (5.28), mostre que o erro num integral usando a regra de Simpson é dado, na ordem dominante em  $h$ , por Eq. (5.29).

## 6. Exercício

Escreva um programa, ou modifique um programa já feito, para mais uma vez calcular o valor do integral  $\int_0^2 (x^4 - 2x + 1) dx$  do Exemplo (5.28), usando a regra do trapézio com 20 intervalos, mas desta vez o programa deve escrever a estimativa do erro do resultado, calculado através do método da Eq. (5.28). Para isto precisa de efectuar o integral duas vezes, uma com  $N_1 = 10$  intervalos e novamente com  $N_2 = 20$ . Através da Eq. (5.28) obtemos o erro. Como se compara o erro calculado deste modo com o cálculo directo do erro como a diferença entre o valor do integral e o valor exacto de 4.4? Porque é que não têm um acordo perfeito?

## 7. Exercício

Considere o integral

$$I = \int_0^1 \sin^2 \sqrt{100x} dx$$

1. Escreva um programa que use o método de regra trapezoidal adaptativa da Seção 5.3 e Eq. (5.34) para calcular o valor deste integral com uma precisão aproximada de  $\epsilon = 10^{-6}$  (isto é, correcto até seis dígitos após o ponto decimal). Comece com um único intervalo de integração e aumente a partir daí para dois, quatro, oito e assim por diante. Peça ao seu programa para imprimir o número de intervalos, a sua estimativa do integral e a sua estimativa do erro no integral, para cada valor do número de intervalos  $N$ , até que a precisão desejada seja atingida. (Dica: deverá encontrar o resultado em torno de  $I = 0,45$ )
2. Agora modifique seu programa para avaliar o mesma integral usando a técnica de integração Romberg descrita nesta seção. Peça ao seu programa para imprimir uma tabela triangular de valores, como na página 161, de todas as estimativas de Romberg do integral. Calcule o erro em suas estimativas usando Eq. (5.49) e continue novamente o cálculo até chegar a uma precisão de  $\epsilon = 10^{-6}$ . Deverá descobrir que o método Romberg atinge a precisão requerida consideravelmente mais rápido que a regra trapezoidal por si só.

## 8. Exercício

### Aplicação do método adaptativo de Romberg

Escreva um programa que use o método adaptativo da regra de Simpson apresentados na secção 5.3 e Eqs. (5.35) a (5.39) para calcular o mesmo integral do último problema da folha anterior, novamente para uma precisão aproximada de  $\epsilon = 10^{-6}$ . Desta vez, comece com dois intervalos de integração, e em seguida com quatro, oito e assim por diante. Mostre os resultados em cada etapa até que a precisão necessária seja atingida. Deve obter que alcança essa precisão para um número significativamente menor de intervalos do que com o a regra trapezoidal, mas com um número um pouco maior do que com a integração de Romberg.

## 9. Exercício

### Capacidade calorífica de um sólido

A teoria de sólidos de Debye dá a capacidade calorífica de um sólido à temperatura  $T$  como um integral

$$C_V = 9V\rho k_B \left(\frac{T}{\theta_D}\right)^3 \int_0^{\theta_D/T} \frac{x^4 e^x}{(e^x - 1)^2} dx,$$

onde  $V$  é o volume do sólido,  $\rho$  é a densidade de átomos,  $k_B$  é a constante de Boltzmann, e  $\theta_D$  é a chamada *temperatura de Debye*, uma propriedade dos sólidos que depende da sua densidade e da velocidade de propagação do som nesse sólido.

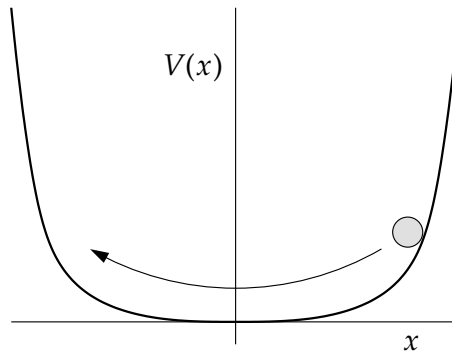
1. Escreva uma função Python,  $cv(T)$ , que calcule  $C_V$  para um dado valor da temperatura, para uma amostra de 1000 centímetros cúbicos de alumínio sólido, que tem uma densidade atômica de  $\rho = 6.022 \times 10^{28} \text{ m}^{-3}$  e uma temperatura de Debye de  $\theta_D = 428 \text{ K}$ . Use a quadratura de Gauss para avaliar o integral, com  $N = 50$  pontos de amostra.
2. Use essa função para fazer um gráfico da capacidade calorífica em função da temperatura de  $T = 5 \text{ K}$  até  $T = 500 \text{ K}$ .

## 10. Exercício

### Período de oscilador anarmônico

O oscilador harmônico simples pode ser estudado facilmente usando métodos analíticos e tem a importante propriedade de que seu período de oscilação é uma constante, independente de sua amplitude, tornando-o útil, por exemplo, para manter o tempo dos relógios. Frequentemente, em Física, no entanto, também nos deparamos com osciladores anarmônicos, cujo período varia com a amplitude e cujo comportamento geralmente não pode ser calculado analiticamente.

Um oscilador clássico geral pode ser considerado uma partícula num potencial côncavo. Quando perturbada, a partícula vai oscilar para frente e para trás nesse poço de potencial:



O oscilador harmônico corresponde a um potencial quadrático  $V(x) \propto x^2$ . Qualquer outra forma dá um oscilador anarmônico. (Assim, existem muitos tipos diferentes de osciladores anarmônicos, dependendo da forma exata do potencial.)

Uma maneira de calcular o movimento de um oscilador é escrever a equação para a conservação de energia no sistema. Se a partícula tiver massa  $m$  e posição  $x$ , então a energia total é igual à soma das energias cinética e potencial, assim:

$$E = \sqrt{m \left(\frac{dx}{dt}\right)^2 + V(x)}.$$

Como a energia deve ser constante ao longo do tempo, esta equação é efetivamente uma equação diferencial (não-linear) de evolução de  $x$  em  $t$ .



Vamos supor que o potencial  $V(x)$  é simétrica relativamente a  $x = 0$  e vamos impor uma amplitude  $a$ . Isto é, em  $t = 0$  libertamos a massa em repouso na posição  $x = a$  e seguimos a sua evolução até que volta à origem. Então a  $t = 0$  temos  $dx/dt = 0$  e a equação acima lê-se  $E = V(a)$ , o que nos dá a energia total da partícula em termos de amplitude.

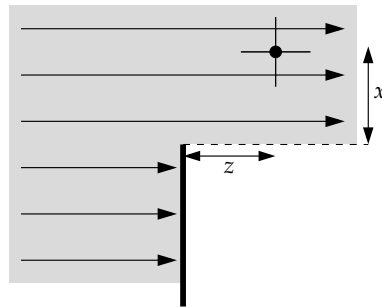
1. Quando a partícula atinge a origem pela primeira vez, passou por um quarto de um período do oscilador. Reorganizando a equação acima para  $dx/dt$  e, em seguida, integrando em relação a  $t$  de 0 a  $\frac{1}{4}T$ , mostre que o período  $T$  é dado por

$$T = \sqrt{8m} \int_0^a \frac{dx}{\sqrt{V(a) - V(x)}}.$$

2. Suponha que o potencial é  $V(x) = x^4$  e a massa da partícula é  $m = 1$ . Escreva uma função Python que calcula o período do oscilador para uma dada amplitude usando a quadratura de Gauss com  $N = 20$  pontos, em seguida, use sua função para fazer um gráfico do período para amplitudes que vão desde  $a = 0$  a  $a = 2$ .
3. Deve obter que o oscilador fica mais rápido à medida que a amplitude aumenta, mesmo que a partícula tenha que viajar ainda mais para uma amplitude maior. Deve obter que o período diverge à medida que a amplitude vai para zero. Como explica esses resultados?

## 11. Exercício

Suponha que uma onda plana, como luz ou onda sonora, é bloqueada por um objeto com uma fronteira rectilínea, representada pela linha sólida na parte inferior desta figura:



A onda será difractada na fronteira e a intensidade resultante na posição  $(x, z)$  marcada pelo ponto, é dada pela teoria de difracção de campo próximo,

$$I = \frac{I_0}{8} \left( [2C(u) + 1]^2 + [2S(u) + 1]^2 \right),$$

onde  $I_0$  é a intensidade da onda antes da difracção e

$$u = x\sqrt{\frac{2}{\lambda z}}, \quad C(u) = \int_0^u \cos \frac{1}{2}\pi t^2 dt, \quad S(u) = \int_0^u \sin \frac{1}{2}\pi t^2 dt.$$

Escreva um programa que calcule  $I/I_0$  e faça o gráfico como função de  $x$  na gama  $-5$  m a  $5$  m no caso da velocidade do som com comprimento de onda  $\lambda = 1$  m, medido  $z = 3$  m após a fronteira rectilínea. Calcule os integrais usando a quadratura gaussiana com  $N = 50$  pontos. Deve encontrar uma variação significativa na intensidade de som difractada – suficiente para que se possa facilmente ouvir o efeito se o som for difractado na fronteira de um edifício alto.

---

## 12. Exercício

### Derivada de função

Crie uma função  $f(x)$  em Python que devolva o valor de  $1 + \frac{1}{2} \tanh 2x$ . Usando o esquema da diferença central calcule a derivada dessa função no intervalo  $-2 \leq x \leq 2$ . Calcule analiticamente a derivada dessa função e trace num mesmo gráfico esta derivada (linha contínua) e a obtida anteriormente por via numérica (pontos a cheio).

## 13. Exercício

Mesmo quando podemos encontrar o valor de  $f(x)$  para qualquer valor de  $x$  a diferença progressiva pode ser mais precisa do que a diferença centrada para  $h$  suficientemente grande. Para que valores de  $h$  o erro de aproximação na diferença progressiva da Eq. (5.87) é menor que na diferença centrada da Eq. (5.95)?

---

# Física Computacional

Ano lectivo 2017/2018

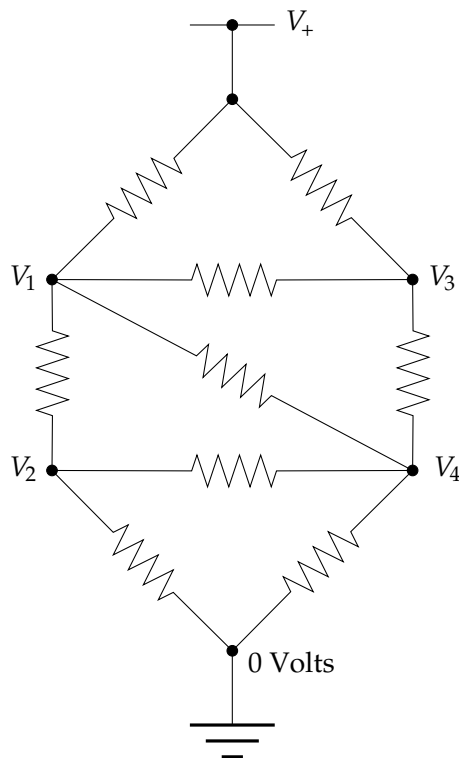
---

## Folha 5 - Equações Lineares e Não Lineares

### 1. Exercício

#### Circuito de resistências

Considere o seguinte circuito de resistências:



Todas as resistências têm o mesmo valor  $R$ . O barramento de energia no topo está na voltagem  $V_+ = 5\text{ V}$ . Quais são as outras quatro voltagens,  $V_1$  a  $V_4$ ?

Para responder a esta pergunta, usamos a lei de Ohm e a lei de nós de Kirchhoff, que diz que o fluxo de corrente líquido total de (ou em) qualquer junção em um circuito deve ser zero. Assim, para o nó do potencial  $V_1$ , por exemplo, temos

$$\frac{V_1 - V_2}{R} + \frac{V_1 - V_3}{R} + \frac{V_1 - V_4}{R} + \frac{V_1 - V_+}{R} = 0,$$

ou, noutra forma,

$$4V_1 - V_2 - V_3 - V_4 = V_+.$$

1. Escreva equações semelhantes para os outros três nós com os potenciais desconhecidos.
2. Escreva um programa para resolver as quatro equações resultantes usando a eliminação de Gauss determinando os quatro potenciais (pode modificar um programa pré-existente `gausselim.py` no Exemplo 6.1).

---

## 2. Exercício

1. Modifique o programa `gausselim.py` do Exemplo 6.1 para incorporar o pivotamento parcial. Execute esse programa e mostre que fornece as mesmas respostas que o programa original quando aplicado à Eq. (6.1).
2. Modifique o programa para resolver a equação (6.17) e mostre que ele também pode encontrar a solução para essa equação vetorial, mesmo que a eliminação de Gauss sem pivotar falhe.

---

## 3. Exercício

### Decomposição LU

Este exercício convida-o a desenvolver o seu próprio programa para resolver equações simultâneas usando o método de decomposição LU.

1. Se desejar, partindo do programa para eliminação de Gauss do Exemplo 6.1, escreva uma função Python que calcule a decomposição da LU de uma matriz. O cálculo é o mesmo que para a eliminação de Gauss, exceto que em cada etapa do cálculo precisa extrair os elementos apropriados da matriz e juntá-los para formar a matriz diagonal inferior  $L$  da Eq. (6.32). Teste essa função calculando a decomposição LU da matriz da Eq. (6.2), e depois multiplique  $L$  por  $U$  para verificar que obtém a matriz original.
2. Use a sua função de decomposição de LU para criar um programa completo para resolver Eq. (6.2) através de uma dupla substituição conforme é descrito no texto. Resolva as mesmas equações usando a função `solve` do pacote `numpy` e verifique se obtém a mesma resposta.
3. Tente a sua incursão na decomposição da LU com pivotamento parcial. O pivotamento parcial funciona da mesma forma para a decomposição da LU, tal como ocorre na eliminação de Gauss, trocando linhas para obter o maior elemento diagonal, conforme explicado na Seção 6.1.3, mas a extensão para a decomposição da LU requer duas etapas adicionais. Primeiro, sempre que trocar duas linhas também terá que trocar as mesmas linhas na matriz  $L$ . Segundo, quando usar a decomposição da LU para resolver um conjunto de equações  $A\bar{x} = \bar{v}$ , também precisa executar a mesma sequência de trocas no vetor  $\bar{v}$  no lado direito. Isso significa que precisa registrar as trocas enquanto está fazendo a decomposição para que possa recriá-los mais tarde. A maneira mais simples de fazer isso é configurar uma lista ou matriz na qual o valor do  $i$ -ésimo elemento registra a linha com a qual trocou na  $i$ -ésima etapa do processo. Por exemplo, se trocou a primeira linha com a segunda e a segunda com a quarta, os dois primeiros elementos da lista seriam 2 e 4. Resolver um conjunto de equações para determinado  $\bar{v}$  envolve primeiro executar a sequência necessária de trocas nos elementos de  $\bar{v}$ , em seguida, executando uma substituição dupla, como de costume. (Na eliminação gaussiana ordinária com pivotamento, troca-se os elementos de  $\bar{v}$  à medida que o algoritmo avança, ao invés de todos de uma só vez, mas a diferença não tem efeito sobre os resultados, então é bom realizar todas as trocas de uma só vez se assim o desejarmos.)

Modifique a função que escreveu para a parte (a) para executar a decomposição da LU com pivotamento parcial. A função deve retornar as matrizes  $L$  e  $U$  para a decomposição da LU da matriz trocada, além de uma lista das trocas realizadas. Em seguida, modifique o resto do programa para resolver equações da forma  $A\bar{x} = \bar{v}$  usando a decomposição da LU com pivotamento. Teste o seu programa no exemplo da Eq. (6.17), que não pode ser resolvido sem pivotamento devido ao zero no primeiro elemento da matriz. Verifique os seus resultados com uma solução das mesmas equações usando a função `solve` de `numpy`.

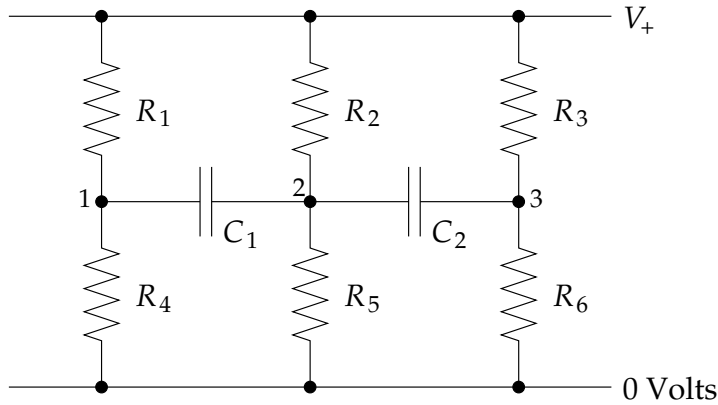
#### 4. Exercício

Escreva um programa que resolva a rede de resistências do problema do Exercício 6.1 na página 220 usando a função `solve` do `numpy.linalg`. Se também fez o Exercício 6.1, pode verificar que obtém a mesma resposta em ambos os casos.

#### 5. Exercício

##### Um circuito resistivo-capacitivo

Aqui está um problema de circuito mais complicado:



$$\begin{aligned} \left( \frac{1}{R_1} + \frac{1}{R_4} + i\omega C_1 \right) x_1 - i\omega C_1 x_2 &= \frac{x_+}{R_1}, \\ -i\omega C_1 x_1 + \left( \frac{1}{R_2} + \frac{1}{R_5} + i\omega C_1 + i\omega C_2 \right) x_2 - i\omega C_2 x_3 &= \frac{x_+}{R_2}, \\ -i\omega C_2 x_2 + \left( \frac{1}{R_3} + \frac{1}{R_6} + i\omega C_2 \right) x_3 &= \frac{x_+}{R_3}. \end{aligned}$$

A tensão  $V_+$  é variável no tempo e sinusoidal na forma  $V_+ = x_+ e^{i\omega t}$ , com  $x_+$  constante. As resistências no circuito podem ser tratados usando a lei de Ohm como de costume. Para os condensadores, a carga  $Q$  e a tensão  $V$  através deles são relacionadas pela lei dos condensadores  $Q = CV$ , onde  $C$  é a capacidade. Diferenciando ambos os lados dessa expressão, vem a corrente  $I$  que passa pelo condensador é:

$$I = \frac{dQ}{dt} = C \frac{dV}{dt}.$$

1. Assumindo que as tensões nos pontos marcados 1, 2 e 3 são da forma  $V_1 = x_1 e^{i\omega t}$ ,  $V_2 = x_2 e^{i\omega t}$ , e  $V_3 = x_3 e^{i\omega t}$ , aplique a lei de Kirchhoff em cada um dos três nodos junto com a lei de Ohm e a lei dos condensadores, para mostrar que as constantes  $x_1$ ,  $x_2$ , e  $x_3$  satisfazem as equações

$$\begin{aligned} \left( \frac{1}{R_1} + \frac{1}{R_4} + i\omega C_1 \right) x_1 - i\omega C_1 x_2 &= \frac{x_+}{R_1}, \\ -i\omega C_1 x_1 + \left( \frac{1}{R_2} + \frac{1}{R_5} + i\omega C_1 + i\omega C_2 \right) x_2 - i\omega C_2 x_3 &= \frac{x_+}{R_2}, \\ -i\omega C_2 x_2 + \left( \frac{1}{R_3} + \frac{1}{R_6} + i\omega C_2 \right) x_3 &= \frac{x_+}{R_3}. \end{aligned}$$

$$\begin{aligned}\left(\frac{1}{R_1} + \frac{1}{R_4} + i\omega C_1\right)x_1 - i\omega C_1 x_2 &= \frac{x_+}{R_1}, \\ -i\omega C_1 x_1 + \left(\frac{1}{R_2} + \frac{1}{R_5} + i\omega C_1 + i\omega C_2\right)x_2 - i\omega C_2 x_3 &= \frac{x_+}{R_2}, \\ -i\omega C_2 x_2 + \left(\frac{1}{R_3} + \frac{1}{R_6} + i\omega C_2\right)x_3 &= \frac{x_+}{R_3}.\end{aligned}$$

2. Escreva um programa para resolver  $x_1$ ,  $x_2$ , e  $x_3$  quando

$$\begin{aligned}R_1 = R_3 = R_5 &= 1 \text{ k}\Omega, \\ R_2 = R_4 = R_6 &= 2 \text{ k}\Omega, \\ C_1 &= 1 \mu\text{F}, \quad C_2 = 0.5 \mu\text{F}, \\ x_+ &= 3 \text{ V}, \quad \omega = 1000 \text{ s}^{-1}.\end{aligned}$$

Observe que a matriz para esse problema possui elementos complexos. Precisar  definir uma matriz complexa para a acomodar, mas ainda pode usar a fun  o `solve`, como anteriormente, para resolver as equa  es - ela funciona com argumentos reais ou complexos. Usando essa solu   o, calcule e imprima as amplitudes das tr s tens  es  $V_1$ ,  $V_2$ , e  $V_3$  e as suas fases em graus. (Nota: Pode achar  teis as fun   es `polar` ou `phase` no pacote `cmath`. Se  $z$    um n mero complexo ent  o "`r, theta = polar(z)`" retornar  o m dulo e a fase (em radianos) de  $z$  e "`r, theta = polar(z)`" retornar  somente a fase.)

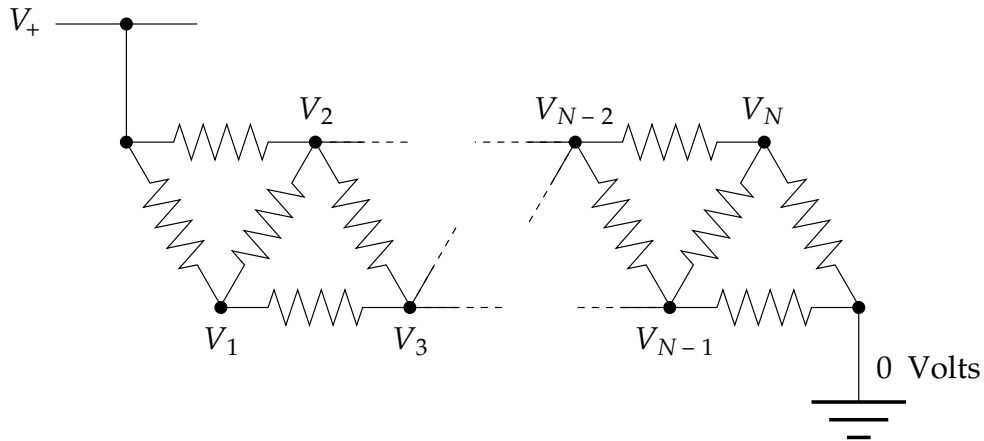
## 6. Exerc cio

Partindo ou do programa `springs.py` na p gina 237 ou `springsb.py` na p gina 238, remova o c digo que produz o gr fico dos resultados e substitua-o pelo c digo que gera uma anima   o das massas na sua vibra   o, os seus deslocamentos relativamente  s suas posi   es de equil brio s o dados pela parte real da Eq. (6.53). Para maior clareza, assuma que as suas posi   es de equil brio est  o separadas por duas unidades na linha horizontal. Pelo menos a anima   o sever  mostrar cada uma das massas individuais, por exemplo como pequenas esferas (de raio 0.2 ou 0.3).

## 7. Exerc cio

### Cadeia de resist ncias

Considere a cadeia de resist ncias da figura



Todas as resist ncias t m o mesmo valor  $R$ . O barramento do topo est  ao potencial  $V_+ = 5 \text{ V}$ . O problema   encontrar os potenciais,  $V_1 \dots V_N$ , nos n dos internos do circuito.

1. Usando a lei de Ohm e a lei de nodos de Kirchhoff, que diz que o fluxo de corrente líquido total de (ou em) qualquer nodo num circuito deve ser zero, mostre que os potenciais  $V_1, \dots, V_N$  satisfazem as equações

$$\begin{aligned} 3V_1 - V_2 - V_3 &= V_+, \\ -V_1 + 4V_2 - V_3 - V_4 &= V_+, \\ &\vdots \\ -V_{i-2} - V_{i-1} + 4V_i - V_{i+1} - V_{i+2} &= 0, \\ &\vdots \\ -V_{N-3} - V_{N-2} + 4V_{N-1} - V_N &= 0, \\ -V_{N-2} - V_{N-1} + 3V_N &= 0. \end{aligned}$$

2. Exprima essas equações em forma vetorial  $A\vec{v} = \vec{w}$  e encontre os valores da matriz  $A$  e do vetor  $\vec{w}$ .
3. Escreva um programa para resolver os valores do  $V_i$  quando houver  $N = 6$  junções internas com potenciais desconhecidas. (Dica: Todos os valores de  $V_i$  devem estar entre zero e 5 V. Se não, algo está errado.)
4. Agora repita o seu cálculo para o caso em que existem  $N = 10\,000$  junções internas. Esta parte não é possível usando ferramentas padrão como a função `linalg.solve`. Tem de usar o facto de a matrix  $\vec{A}$  ser em banda, e usar função `banded` do ficheiro `banded.py`, discutido Appendix E, ou a função `linalg.solve_banded`.

## 8. Exercício

### O algoritmo QR

Neste exercício escreverá um programa para calcular os valores próprios e vetores próprios de uma matriz simétrica real usando o algoritmo QR. O primeiro desafio é escrever um programa que encontre a decomposição QR de uma matriz. Então iremos usar essa decomposição para encontrar os valores próprios.

Como descrito acima, a decomposição QR expressa uma matriz quadrada real  $A$  na forma  $A = QR$ , onde  $Q$  é uma matriz ortogonal e  $R$  é uma matriz triangular superior. Dado uma matriz  $A$ ,  $N \times N$ , podemos calcular a decomposição QR da seguinte forma.

Vamos pensar na matriz como um conjunto de  $N$  vetores de coluna  $\vec{a}_0 \dots \vec{a}_{N-1}$  então:

$$A = \begin{pmatrix} | & | & | & \dots \\ \vec{a}_0 & \vec{a}_1 & \vec{a}_2 & \dots \\ | & | & | & \dots \end{pmatrix},$$

Onde numeramos os vetores em Python, começando do zero, o que será conveniente ao escrever o programa. Agora definimos dois novos conjuntos de vetores  $\vec{u}_0 \dots \vec{u}_{N-1}$  e  $\vec{q}_0 \dots \vec{q}_{N-1}$  do seguinte modo:

$$\begin{aligned} \vec{u}_0 &= \vec{a}_0, & \vec{q}_0 &= \frac{\vec{u}_0}{|\vec{u}_0|}, \\ \vec{u}_1 &= \vec{a}_1 - (\vec{q}_0 \cdot \vec{a}_1)\vec{q}_0, & \vec{q}_1 &= \frac{\vec{u}_1}{|\vec{u}_1|}, \\ \vec{u}_2 &= \vec{a}_2 - (\vec{q}_0 \cdot \vec{a}_2)\vec{q}_0 - (\vec{q}_1 \cdot \vec{a}_2)\vec{q}_1, & \vec{q}_2 &= \frac{\vec{u}_2}{|\vec{u}_2|}, \end{aligned}$$

e assim por diante. As fórmulas gerais para calcular  $\vec{u}_i$  e  $\vec{q}_i$  são

$$\vec{u}_i = \vec{a}_i - \sum_{j=0}^{i-1} (\vec{q}_j \cdot \text{vetores próprios } \vec{a}_i) \vec{q}_j, \quad \vec{q}_i = \frac{\vec{u}_i}{|\vec{u}_i|}.$$

1. Mostre, por indução ou de outra forma, que os vetores  $\vec{q}_i$  são ortonormais, isto é, que satisfazem

$$\vec{q}_i \cdot \vec{q}_j = \begin{cases} 1 & \text{se } i = j, \\ 0 & \text{se } i \neq j. \end{cases}$$

Agora, reorganizando as definições dos vetores, temos

$$\begin{aligned} \vec{a}_0 &= |\vec{u}_0| \vec{q}_0, \\ \vec{a}_1 &= |\vec{u}_1| \vec{q}_1 + (\vec{q}_0 \cdot \vec{a}_1) \vec{q}_0, \\ \vec{a}_2 &= |\vec{u}_2| \vec{q}_2 + (\vec{q}_0 \cdot \vec{a}_2) \vec{q}_0 + (\vec{q}_1 \cdot \vec{a}_2) \vec{q}_1, \end{aligned}$$

e assim por diante. Ou podemos agrupar os vetores  $\vec{q}_i$  juntos como as colunas de uma matriz e escrever todas essas equações como uma equação matricial única

$$A = \begin{pmatrix} | & | & | & \cdots \\ \vec{a}_0 & \vec{a}_1 & \vec{a}_2 & \cdots \\ | & | & | & \cdots \end{pmatrix} = \begin{pmatrix} | & | & | & \cdots \\ \vec{q}_0 & \vec{q}_1 & \vec{q}_2 & \cdots \\ | & | & | & \cdots \end{pmatrix} \begin{pmatrix} |\vec{u}_0| & \vec{q}_0 \cdot \vec{a}_1 & \vec{q}_0 \cdot \vec{a}_2 & \cdots \\ 0 & |\vec{u}_1| & \vec{q}_1 \cdot \vec{a}_2 & \cdots \\ 0 & 0 & |\vec{u}_2| & \cdots \end{pmatrix}.$$

(Se isso parece complicado, vale a pena multiplicar as matrizes à direita para verificar por si mesmo que obtém as expressões corretas para o  $\vec{a}_i$ .)

Observe agora que a primeira matriz no lado direito desta equação, a matriz com colunas  $\vec{q}_i$ , é ortogonal, porque os vetores  $\vec{q}_i$  são ortonormais e a segunda matriz é triangular superior. Por outras palavras, encontramos a decomposição QR  $A = QR$ . As matrizes  $Q$  e  $R$  são

$$Q = \begin{pmatrix} | & | & | & \cdots \\ \vec{q}_0 & \vec{q}_1 & \vec{q}_2 & \cdots \\ | & | & | & \cdots \end{pmatrix}, \quad R = \begin{pmatrix} |\vec{u}_0| & \vec{q}_0 \cdot \vec{a}_1 & \vec{q}_0 \cdot \vec{a}_2 & \cdots \\ 0 & |\vec{u}_1| & \vec{q}_1 \cdot \vec{a}_2 & \cdots \\ 0 & 0 & |\vec{u}_2| & \cdots \end{pmatrix}.$$

1. Escreva uma função Python que tome como argumento uma matriz quadrada real  $A$  e retorne as duas matrizes  $Q$  e  $R$  que formam a decomposição QR. Como teste, experimente essa função na matriz

$$A = \begin{pmatrix} 1 & 4 & 8 & 4 \\ 4 & 2 & 3 & 7 \\ 8 & 3 & 6 & 9 \\ 4 & 7 & 9 & 2 \end{pmatrix}.$$

Verifique os resultados multiplicando  $Q$  e  $R$  juntos para recuperar a matriz original  $A$ .

2. Usando essa função, escreva um programa completo para calcular os valores próprios e vetores próprios de uma matriz simétrica real usando o algoritmo QR. Continue o cálculo até que a magnitude de todos os elementos fora da diagonal da matriz seja menor que  $10^{-6}$ . Teste o programa na matriz do exemplo acima. Deve obter que os valores próprios são 1, 21,  $-3$ , e  $-8$ .

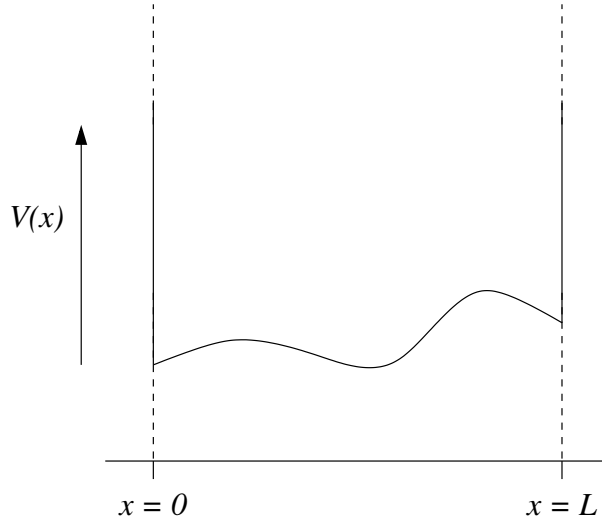
## 9. Exercício

### Poço de potencial quântico assimétrico

A mecânica quântica pode ser formulada como um problema de matriz e resolvida num computador usando métodos da álgebra linear. Suponha, por exemplo, que temos uma partícula de massa  $M$  num



poço de potencial uni-dimensional de largura  $L$ . Suponha que o potencial  $V(x)$  varia de alguma forma dentro do poço:



Não podemos resolver tais problemas analiticamente em geral, mas podemos resolvê-los no computador.

Num estado puro de energia  $E$ , a parte espacial da função de onda obedece a equação de Schrödinger independente do tempo  $\hat{H}\psi(x) = E\psi(x)$ , onde o operador hamiltoniano  $\hat{H}$  é dado por

$$\hat{H} = -\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x).$$

Por simplicidade, vamos supor que as paredes do poço são infinitamente altas, de modo que a função de onda seja zero fora do poço, o que significa que ele deve ir a zero  $x = 0$  e  $x = L$ . Nesse caso, a função de onda pode ser expressa como uma série de senos de Fourier:

$$\psi(x) = \sum_{n=1}^{\infty} \psi_n \sin \frac{\pi n x}{L},$$

onde  $\psi_1, \psi_2, \dots$  são os coeficientes de Fourier.

1. Observando que, para  $m, n$  inteiros positivos

$$\int_0^L \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx = \begin{cases} L/2 & \text{se } m = n, \\ 0 & \text{se } m \neq n, \end{cases}$$

mostre que a equação de Schrödinger  $\hat{H}\psi = E\psi$  implica que

$$\sum_{n=1}^{\infty} \psi_n \int_0^L \sin \frac{\pi m x}{L} \hat{H} \sin \frac{\pi n x}{L} dx = \frac{1}{2} L E \psi_m.$$

Assim, definindo uma matriz  $H$  com elementos

$$\begin{aligned} H_{mn} &= \frac{2}{L} \int_0^L \sin \frac{\pi m x}{L} \hat{H} \sin \frac{\pi n x}{L} dx \\ &= \frac{2}{L} \int_0^L \sin \frac{\pi m x}{L} \left[ -\frac{\hbar^2}{2M} \frac{d^2}{dx^2} + V(x) \right] \sin \frac{\pi n x}{L} dx, \end{aligned}$$

mostre que a equação de Schrödinger pode ser escrita em forma de  $H\psi = E\psi$ , onde  $\psi$  é o vetor  $(\psi_1, \psi_2, \dots)$ . Então  $\psi$  é um vetor próprio da *matriz hamiltoniana*  $H$  com valor próprio  $E$ . Se pudermos calcular os valores próprios desta matriz, então sabemos as energias permitidas da partícula no poço.

2. Para o caso  $V(x) = ax/L$ , avalie analiticamente o integral em  $H_{mn}$  e assim encontre uma expressão geral para o elemento da matriz  $H_{mn}$ . Mostre que a matriz é real e simétrica. Provavelmente achará útil saber que

$$\int_0^L x \sin \frac{\pi m x}{L} \sin \frac{\pi n x}{L} dx = \begin{cases} 0 & \text{se } m \neq n \text{ e ambos pares ou ímpares,} \\ -\left(\frac{2L}{\pi}\right)^2 \frac{mn}{(m^2 - n^2)^2} & \text{se } m \neq n \text{ e um é par, e outro é ímpar,} \\ L^2/4 & \text{se } m = n. \end{cases}$$

Escreva um programa em Python para avaliar a expressão para  $H_{mn}$  para arbitrários  $m$  e  $n$  quando a partícula no poço é um elétron, o poço tem largura 5 Å, e  $a = 10$  eV. (A massa e carga de um elétron são  $9.1094 \times 10^{-31}$  kg e  $1.6022 \times 10^{-19}$  C respectivamente.)

3. A matriz  $H$  é teoricamente infinitamente grande, então não podemos calcular todos os seus valores próprios. Mas podemos obter uma solução bastante precisa para os primeiros, cortando a matriz após os primeiros elementos. Modifique o programa que escreveu para a parte (2) acima para criar uma matriz dos  $10 \times 10$  elementos de  $H$  até  $m, n = 10$ . Calcule os valores próprios dessa matriz usando a função apropriada de `numpy.linalg` e, portanto, imprima, em unidades de elétron-volt, os dez primeiros níveis de energia do poço quântico, dentro dessa aproximação. Deve descobrir, por exemplo, que a energia do estado fundamental do sistema está próxima 5.84 eV. (Nota: lembre-se de que os índices matriciais em Python começam em zero, enquanto os índices em expressões algébricas padrão, como as acima, começam em um. Precisarás fazer adaptações para isso no programa.)
4. Modifique o programa para usar em alternativa um array  $100 \times 100$  e que calcule novamente os dez primeiros valores próprios de energia. Comparando com os valores calculados na parte (3), o que conclui sobre a precisão do cálculo?
5. Agora modifique o programa mais uma vez para calcular a função de onda  $\psi(x)$  para o estado fundamental e os dois primeiros estados excitados no poço. Use esses resultados para criar um gráfico com três curvas mostrando a densidade de probabilidade  $|\psi(x)|^2$  como função de  $x$  em cada um desses três estados. Preste especial atenção à normalização da função de onda - ela deve satisfazer a condição  $\int_0^L |\psi(x)|^2 dx = 1$ . Isso é verdade da sua função de onda?

## 10. Exercício

Considere a equação  $x = 1 - e^{-cx}$ , em que  $c$  é um parâmetro conhecido and  $x$  uma incógnita. Esta equação surge numa variedade de situações, incluindo a física dos processos de contacto, os modelos matemáticos de epidemias e a teoria de gráficos aleatórios.

- Escreva um programa para resolver esta equação para  $x$  usando o método de relaxação para o caso  $c = 2$ . Calcule a sua solução com uma exactidão de pelo menos  $10^{-6}$ .
- Modifique seu programa para calcular a solução para valores de  $c$  de 0 a 3 em passos de 0.01 e faça um gráfico de  $x$  em função de  $c$ . Deve ver uma transição clara de um regime em que  $x = 0$  para um regime de  $x \neq 0$ . Este é outro exemplo de transição de fase. Na física, essa transição é conhecida como *transição de percolação*; em epidemiologia é o *limiar epidémico*.

## II. Exercício

### Sobre-relaxação sucessiva

Se não fez o Exercício 6.10, deve fazê-lo antes de iniciar este.

O método de sobre-relaxação envolve iterar a equação  $x' = f(x)$ , começando de um palpite inicial, até convergir. Como vimos, essa é uma maneira rápida e fácil de encontrar soluções para equações não-lineares. No entanto, é possível, em alguns casos, tornar o método ainda mais rápido usando a técnica de *sobre-relaxação*. Suponha que nosso palpite inicial na solução de uma equação seja, digamos,  $x = 1$ , e que a verdadeira solução é  $x = 5$ . Após o primeiro passo do processo iterativo, podemos ver um valor de, digamos,  $x = 3$ . No método de sobre-relaxação, observamos esse valor e notamos que  $x$  está a aumentar, então deliberadamente ultrapassamos o valor calculado, na esperança de que isso nos aproxime da solução final — neste caso podemos passar  $x = 3$  e atingir por exemplo o valor de  $x = 4$ , que está mais próximo da verdadeira solução  $x = 5$  e, portanto, deve nos levar à solução mais rapidamente. O método de sobre-relaxação fornece uma fórmula para executar esse tipo de aceleração de modo controlado e, muitas vezes, embora nem sempre, leva-nos à solução mais rapidamente. Em detalhe, funciona da seguinte forma.

Podemos reescrever a equação  $x' = f(x)$  na forma  $x' = x + \Delta x$ , onde

$$\Delta x = x' - x = f(x) - x.$$

O método de sobre-relaxação envolve a iteração da equação modificada

$$x' = x + (1 + \omega) \Delta x,$$

(mantendo a mesma definição de  $\Delta x$ ). Se o parâmetro  $\omega$  é nulo, então este método é igual à relaxação habitual, contudo para  $\omega > 0$  o método pega na quantidade  $\Delta x$  da qual o valor  $x$  iria ser alterado e incrementa-a um pouco. Usando  $\Delta x = f(x) - x$ , nós também podemos escrever  $x'$  como

$$x' = x + (1 + \omega)[f(x) - x] = (1 + \omega)f(x) - \omega x,$$

que é a sua forma habitual.

Para que o método funcione, o valor de  $\omega$  deve ser escolhido corretamente, embora haja algum espaço de manobra - existe um valor ótimo, mas valores na sua vizinhança também fornecem bons resultados. Infelizmente, não há uma teoria geral que nos diga qual é o valor ideal. Geralmente é encontrado por tentativa e erro.

1. Derive um equivalente de Eq. (6.81) para o método de sobre-relaxação e, portanto, mostrem que o erro em  $x'$ , o equivalente da Eq. (6.83), é dado por

1.

$$\epsilon' \simeq \frac{x - x'}{1 - 1/[(1 + \omega)f'(x) - \omega]}.$$

2. Considere novamente a equação  $x = 1 - e^{-cx}$  que resolvemos no Exercício 6.10. Usando o programa que fez para a alínea (a) do exercício, que resolveu a equação para o caso  $c = 2$ , e modificando-o para que imprima o número de iterações necessárias para que a solução convirja com uma precisão de  $10^{-6}$ .
3. Agora escreva um novo programa (o modifique o já existente) para resolver a mesma equação  $x = 1 - e^{-cx}$  para  $c = 2$ , novamente com um precisão de  $10^{-6}$ , mas usando a sobre-relaxação. Programe o seu programa para imprimir as respostas que encontrar junto com o número de iterações necessárias para encontrá-las. Experimente valores diferentes de  $\omega$  para ver o quão rápido você pode fazer com que o método convirja. Um valor de  $\omega = 0.5$  é um ponto de partida razoável. Com alguma tentativa e erro, você deve conseguir que o cálculo convirja duas vezes mais rápido que o método de relaxação simples, ou seja, em cerca de metade das iterações.

4. Há alguma circunstância em que usar um valor  $\omega < 0$  nos ajudaria a encontrar uma solução mais rápido do que podemos com o método de relaxação comum? (Dica: a resposta é sim, mas por quê?)

## 12. Exercício

O processo bioquímico de glicólise, a quebra da glicose no corpo para liberar energia, pode ser modelado pelas equações

$$\frac{dx}{dt} = -x + ay + x^2y, \quad \frac{dy}{dt} = b - ay - x^2y.$$

As variáveis  $x$  e  $y$  representam concentrações de dois químicos, ADP e F6P, e  $a$  e  $b$  são constantes positivas. Uma das características importantes de equações lineares não-lineares como estas são os seus pontos fixos, significando valores de  $x$  e  $y$  nos quais as derivadas de ambas as variáveis se tornam zero simultaneamente, de modo que os valores das variáveis deixam de mudar e se tornam constantes no tempo. Pondo as derivadas a zero acima, os pontos fixos das equações de glicólise são soluções de

$$-x + ay + x^2y = 0, \quad b - ay - x^2y = 0.$$

1. Demonstre, analiticamente, que a solução dessas equações é

$$x = b, \quad y = \frac{b}{a + b^2}.$$

2. Mostre que as equações podem ser reorganizadas na forma

$$x = y(a + x^2), \quad y = \frac{b}{a + x^2}$$

e escreva um programa para determinar o ponto estacionário usando o método de relaxação com  $a = 1$  e  $b = 2$ . Deve encontrar que o método não converge para uma solução neste caso.

3. Encontre uma maneira diferente de organizar as equações de forma que, quando aplicar novamente o método de relaxação, convirja agora para um ponto fixo. Verifique se a solução obtida concorda com a solução analítica.

## 13. Exercício

### Constante de Wien

A lei de radiação de Planck diz-nos que a intensidade da radiação por unidade de área e por unidade de comprimento de onda  $\lambda$  de um corpo negro à temperatura  $T$  é

$$I(\lambda) = \frac{2\pi hc^2 \lambda^{-5}}{e^{hc/\lambda k_B T} - 1},$$

em que  $h$  é a constante de Planck,  $c$  a velocidade da luz, e  $k_B$  a constante de Boltzmann.

1. Mostre que o máximo de intensidade radiada é obtida para um comprimento de onda  $\lambda$  determinado pela equação

$$5e^{-hc/\lambda k_B T} + \frac{hc}{\lambda k_B T} - 5 = 0.$$

Com a substituição  $x = hc/\lambda k_B T$  prove que o comprimento de onda do máximo de intensidade satisfaz a *lei de deslocamento de Wien*:

$$\lambda = \frac{b}{T},$$

em que a chamada *constante de Wien* é  $b = hc/k_B x$ , sendo  $x$  a solução da equação não linear

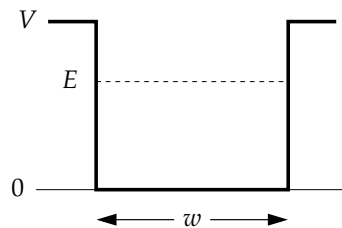
$$5e^{-x} + x - 5 = 0.$$

Escreva um programa para resolver essa equação com uma exactidão de  $\epsilon = 10^{-6}$ , usando o método de bisecção e determine o valor da constante de Wien.

2. A lei do deslocamento de Wien é a base para o método de *pirometria óptica*, um método para medir as temperaturas dos objetos, observando a cor da radiação térmica que emitem. O método é comumente usado para estimar as temperaturas da superfície de corpos astronómicos, como o Sol. O pico de comprimento de onda na radiação emitida pelo Sol cai em  $\lambda = 502 \text{ nm}$ . Das equações acima e do seu valor da constante de deslocamento, estime a temperatura da superfície do Sol.

## 14. Exercício

Considere um potencial quadrado de largura  $w$ , com paredes de altura  $V$ :



Usando a equação de Schrödinger, é possível mostrar-se que as energias  $E$  permitidas de uma partícula única quântica de massa  $m$  presa no poço são soluções de,

$$\tan\left(\sqrt{w^2 m E / 2\hbar^2}\right) = \begin{cases} \sqrt{(V - E)/E} & \text{índice par} \\ -\sqrt{E/(V - E)} & \text{índice ímpar} \end{cases}$$

onde os estados são numerados a partir de zero, com o estado fundamental correspondendo ao estado zero, o primeiro estado excitado numerado por um, etc.

1. Para um electrão (massa  $9.1094 \times 10^{-31} \text{ kg}$ ) num poço com  $V = 20 \text{ eV}$  e  $w = 1 \text{ nm}$ , escreva um programa de Python que represente as três quantidades

$$y_1 = \tan\left(w^2 m E / 2\hbar^2\right) \quad y_2 = \sqrt{\frac{V - E}{E}} \quad y_3 = -\sqrt{\frac{E}{V - E}}$$

no mesmo gráfico, como função de  $E$  desde  $E = 0$  até  $E = 20 \text{ eV}$ . A partir do gráfico faça uma estimativa aproximada das energias dos primeiros seis níveis de energia da partícula.

2. Escreva um segundo programa que calcule os valores dos primeiros seis níveis de energia em electrão-volt com uma precisão de  $0.001 \text{ eV}$  usando o método de bisecção.

## 15. Exercício

### Raízes de um polinómio

Considere o polynómio de sexta ordem

$$P(x) = 924x^6 - 2772x^5 + 3150x^4 - 1680x^3 + 420x^2 - 42x + 1.$$

Não existe uma fórmula geral para as raízes de um polinómio de sexta ordem, mas é possível encontrá-las facilmente usando um computador.

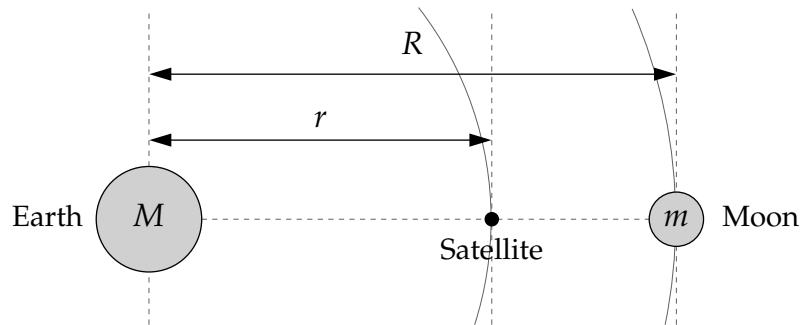
1. Faça um gráfico de  $P(x)$  de  $x = 0$  a  $x = 1$  e, por inspeção, encontre valores aproximados para as seis raízes do polinômio—os pontos nos quais a função é zero.
2. Escreva um programa em Python para determinar as posições de todas as seis raízes com pelo menos dez casas decimais, usando o método de Newton.

Note que o polinômio deste exemplo é o sexto polinômio de Legendre (mapeado no intervalo de zero a um), pelo que o cálculo executado aqui é o mesmo que encontrar os pontos de integração para a quadratura gaussiana de 6 pontos (ver Seção 5.6.2). De facto, o método de Newton é o método de eleição para calcular os pontos de quadratura de Gauss.

## 16. Exercício

### Pontos de Lagrange

Existe um ponto mágico entre a Terra e a Lua, chamado de  $L_1$  Ponto de Lagrange, no qual um satélite orbitará a Terra em perfeita sincronia com a Lua, permanecendo sempre entre os dois. Isso funciona porque a atração interior da Terra e a atração externa da Lua se combinam para criar exatamente a força centrípeta necessária que mantém o satélite em sua órbita. Aqui está a configuração:



1. Assumindo órbitas circulares, e assumindo que a Terra é muito mais massiva que a Lua ou o satélite, mostre que a distância  $r$  do centro da Terra para o  $L_1$  ponto satisfaz

$$\frac{GM}{r^2} - \frac{Gm}{(R-r)^2} = \omega^2 r,$$

onde  $M$  e  $m$  são as massas da Terra e da Lua,  $G$  é a constante gravitacional de Newton, e  $\omega$  é a velocidade angular da Lua e do satélite.

2. A equação acima é uma equação polinomial de quinta ordem em  $r$  (também chamado de equação quártica). Tais equações não podem ser resolvidas exatamente de forma fechada, mas é simples resolvê-las numericamente. Escreva um programa que use o método de Newton ou o método da secante para obter a distância  $r$  da Terra para o  $L_1$  ponto. Calcule uma solução com precisão de pelo menos quatro números significativos.

Os valores dos vários parâmetros são:

$$G = 6.674 \times 10^{-11} \text{ m}^3 \text{ kg}^{-1} \text{ s}^{-2},$$

$$M = 5.974 \times 10^{24} \text{ kg},$$

$$m = 7.348 \times 10^{22} \text{ kg},$$

$$R = 3.844 \times 10^8 \text{ m},$$

$$\omega = 2.662 \times 10^{-6} \text{ s}^{-1}.$$

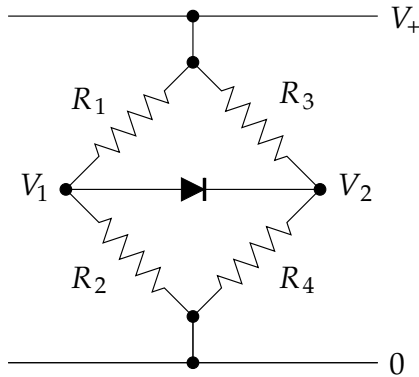
Também precisará escolher um valor inicial adequado para  $r$ , ou dois valores iniciais se usar o método da secante.

## 17. Exercício

### Circuitos não lineares

O Exercício 6.1 usou equações lineares simultâneas para resolver o comportamento dos circuitos de resistências. As resistências são componentes lineares - a corrente é proporcional à tensão - e as equações resultantes que precisamos resolver são, portanto, também lineares e podem ser resolvidas por métodos matriciais padrão. Circuitos reais, no entanto, geralmente incluem componentes não-lineares. Para resolver o comportamento desses circuitos, precisamos resolver equações não-lineares.

Considere o seguinte circuito simples, uma variação da clássica ponte de Wheatstone:



As resistências obedecem à lei de Ohm normal, mas o diodo obedece à equação do diodo:

$$I = I_0(e^{V/V_T} - 1),$$

onde  $V$  é a tensão através do diodo e  $I_0$  e  $V_T$  são constantes.

1. A lei dos nodos de Kirchhoff diz que o saldo das correntes fluem para dentro ou para fora de cada nodo de um circuito deve ser zero. Aplicando a lei ao nodo com tensão  $V_1$  temos

$$\frac{V_1 - V_+}{R_1} + \frac{V_1}{R_2} + I_0[e^{(V_1 - V_2)/V_T} - 1] = 0.$$

Derive a equação correspondente para tensão  $V_2$ .

2. Resolva as duas equações não-lineares para as tensões  $V_1$  e  $V_2$  com as condições

$$\begin{aligned} V_+ &= 5 \text{ V}, \\ R_1 &= 1 \text{ k}\Omega, \quad R_2 = 4 \text{ k}\Omega, \quad R_3 = 3 \text{ k}\Omega, \quad R_4 = 2 \text{ k}\Omega, \\ I_0 &= 3 \text{ nA}, \quad V_T = 0.05 \text{ V}. \end{aligned}$$

Pode usar o método de relaxamento ou o método de Newton para resolver as equações. Se usar o método de Newton, então pode resolver a Eq. (6.108) para  $\Delta \vec{x}$  usando a função `solve()` de `numpy.linalg` se assim quiser, mas neste caso a matriz é apenas uma matriz  $2 \times 2$ , por isso é fácil calcular o inverso diretamente também.

3. A regra prática do engenheiro eletrônico para os diodos é que a tensão num diodo (polarizado diretamente) é sempre de aproximadamente 0,6 volts. Confirme se os resultados estão de acordo com essa regra.

---

## 18. Exercício

---

### Temperatura de uma lâmpada

Uma lâmpada incandescente é um dispositivo simples - ela contém um filamento, geralmente feito de tungstênio, aquecido pelo fluxo de eletricidade até ficar suficientemente quente para irradiar luz visível. Essencialmente toda a energia consumida por tal lâmpada é irradiada como energia eletromagnética, mas parte da radiação não está nos comprimentos de onda visíveis, o que significa que é inútil para fins de iluminação.

Vamos definir a eficiência de uma lâmpada para ser uma fração da energia irradiada que cai na faixa visível. É uma boa notícia a radiação emitida por um filamento à temperatura  $T$  obedecer à lei de radiação de Planck, o que significa que a potência irradiada por unidade de comprimento de onda  $\lambda$  obedece

$$I(\lambda) = 2\pi Ahc^2 \frac{\lambda^{-5}}{e^{hc/\lambda k_B T} - 1},$$

onde  $A$  é a área da superfície do filamento,  $h$  é a constante de Planck,  $c$  é a velocidade da luz e  $k_B$  é a constante de Boltzmann. Os comprimentos de onda visíveis são de  $\lambda_1 = 390$  nm a  $\lambda_2 = 750$  nm, pelo que a energia total irradiada na janela visível é  $\int_{\lambda_1}^{\lambda_2} I(\lambda) d\lambda$  e a energia total em todos os comprimentos de onda é  $\int_0^\infty I(\lambda) d\lambda$ . Dividindo uma expressão pela outra e substituindo  $I(\lambda)$ , temos uma expressão para a eficiência  $\eta$  da lâmpada:

$$\eta = \frac{\int_{\lambda_1}^{\lambda_2} \lambda^{-5} / (e^{hc/\lambda k_B T} - 1) d\lambda}{\int_0^\infty \lambda^{-5} / (e^{hc/\lambda k_B T} - 1) d\lambda}.$$

Fazendo a substituição  $x = hc/\lambda k_B T$ , também pode ser escrita como

$$\eta = \frac{\int_{hc/\lambda_2 k_B T}^{hc/\lambda_1 k_B T} x^3 / (e^x - 1) dx}{\int_0^\infty x^3 / (e^x - 1) dx} = \frac{15}{\pi^4} \int_{hc/\lambda_2 k_B T}^{hc/\lambda_1 k_B T} \frac{x^3}{e^x - 1} dx,$$

onde fizemos uso do valor exato conhecido do integral no denominador.

1. Escreva uma função Python que tem a temperatura  $T$  como argumento e que calcule o valor de  $\eta$  para essa temperatura a partir da fórmula acima. O integral na fórmula não pode ser feito analiticamente, mas também pode fazê-lo numericamente usando qualquer método de sua escolha. (Por exemplo, a quadratura gaussiana com 100 pontos de amostra funciona bem.) Use essa função para fazer um gráfico de  $\eta$  em função da temperatura entre 300 K e 10 000 K. Deve verificar que há uma temperatura intermediária em que a eficiência é máxima.
2. Calcule a temperatura de eficiência máxima da lâmpada dentro de 1 K usando a pesquisa da proporção áurea. (Nota: uma precisão de 1 K é o equivalente a algumas partes em dez mil, neste caso. Para obter esse tipo de precisão, neste cálculo, precisará usar valores para as constantes fundamentais que sejam adequadamente precisas, ou seja, precisará de valores precisos com vários algarismos significativos.)
3. É prático usar uma lâmpada de filamento de tungstênio na temperatura que encontrou? Se não, por que não?