Física Computacional

ano letivo: 2021/2022

docente: Nuno Castro [nuno.castro@fisica.uminho.pt]

Licenciatura em Física e Licenciatura / Mestrado Integrado em Engenharia Física

5ª aula (04/11/2021)

Universidade do Minho
Escola de Ciências

Equações não lineares: o método da relaxação

$$x = 2 - e^{-x}$$

$$x = 1$$

$$x' = 2 - e^{-1} \sim 1.632$$

$$x'' = 2 - e^{-1.632} \sim 1.804$$

$$\ldots$$

```python
from math import exp
x = 1.0
for k in range(10):
    x = 2 - exp(-x)
    print(x)
```

```
1.6321205588285577
1.8044854658474119
1.8354408939220457
1.8404568553435368
1.841255113911434
1.8413817828128696
1.8414018735357267
1.8414050598547234
1.8414055651879888
1.8414056453310121
```

Sistemas de equações não lineares

$$
\begin{aligned}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0
\end{aligned}
$$

Sistemas de equações não lineares

Método de Newton–Raphson

$$f_i(\mathbf{x} + \mathbf{\Delta x}) = f_i(\mathbf{x}) + \sum_{j=1}^{n} \frac{\partial f_i}{\partial x_j} \mathbf{\Delta} x_j + \cancel{O(\Delta x^2)}$$

aproximação linear

$$\mathbf{f}(\mathbf{x} + \mathbf{\Delta x}) = \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\, \mathbf{\Delta x}$$

sendo **J**(**x**) a *matriz Jacobiana* (de dimensão *n* × *n*)

$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

Método de Newton–Raphson

- assumindo $\mathbf{x}$ como uma aproximação de $\mathbf{f}(\mathbf{x}) = \mathbf{0}$, e sendo $\mathbf{x} + \Delta\mathbf{x}$ uma aproximação melhor:
  - Queremos agora encontrar $\Delta\mathbf{x}$, pelo que inserimos $\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{0}$ em

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\,\Delta\mathbf{x}$$

  - Obtemos assim um conjunto de equações lineares para $\Delta\mathbf{x}$:

$$\mathbf{J}(\mathbf{x})\,\Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})$$

Sistemas de equações não lineares

Método de Newton–Raphson

1. estimar o vetor de soluções **x**

2. avaliar **f(x)**

3. Calcular a matrix Jacobiana J(x)

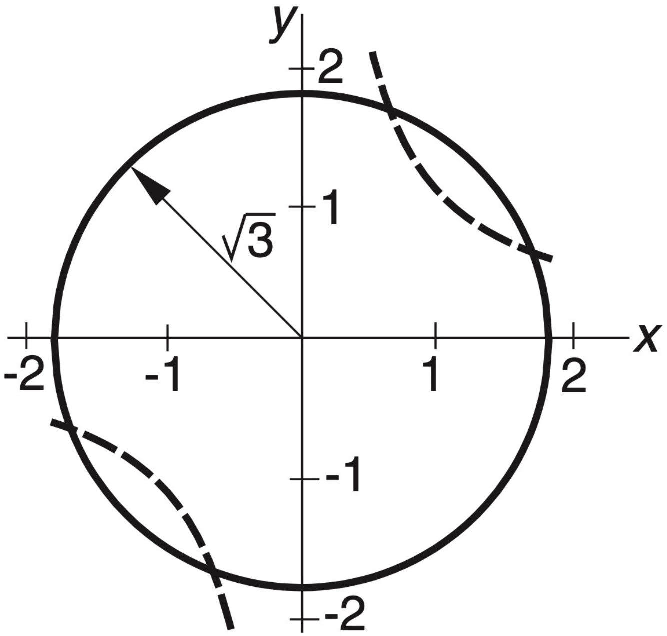
$$J_{ij} = \frac{\partial f_i}{\partial x_j}$$

4. Definir o conjunto de equações simultâneas **J(x) Δx = –f(x)** e resolver em ordem a **Δx**

5. Fazer **x ← x + Δx** e repetir os passos 2 a 5 até que **|Δx| < ε**

Método de Newton–Raphson: exemplo

Determinar os pontos de interseção do círculo $x^2 + y^2 = 3$ e da hipérbola $xy = 1$



$$f_1(x, y) = x^2 + y^2 - 3 = 0$$

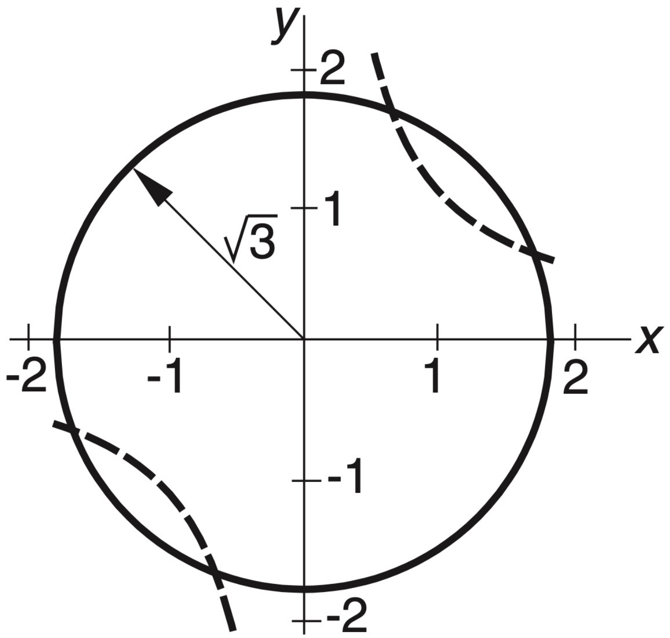
$$f_2(x, y) = xy - 1 = 0$$

Sistemas de equações não lineares

$$\mathbf{J}(x, y) = \begin{bmatrix} \partial f_1/\partial x & \partial f_1/\partial y \\ \partial f_2/\partial x & \partial f_2/\partial y \end{bmatrix} = \begin{bmatrix} 2x & 2y \\ y & x \end{bmatrix}$$

$$\boxed{\mathbf{J}(\mathbf{x})\ \Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})}$$

$$\begin{bmatrix} 2x & 2y \\ y & x \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -x^2 - y^2 + 3 \\ -xy + 1 \end{bmatrix}$$



$x = 0.5$, $y = 1.5$

$$\begin{bmatrix} 1.0 & 3.0 \\ 1.5 & 0.5 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} 0.50 \\ 0.25 \end{bmatrix}$$

cuja solução é $\Delta x = \Delta y = 0.125$

assim, podemos corrigir a nossa estimative inicial:

$x = 0.5 + 0.125 = 0.625$
$y = 1.5 + 0.125 = 1.625$

Sistemas de equações não lineares

2

$$\begin{bmatrix} 1.250 & 3.250 \\ 1.625 & 0.625 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -0.031250 \\ -0.015625 \end{bmatrix}$$

$$\Delta x = \Delta y = -0.00694$$

$$x = 0.625 - 0.006\,94 = 0.618\,06 \qquad y = 1.625 - 0.006\,94 = 1.618\,06$$

Sistemas de equações não lineares

3

$$\begin{bmatrix} 1.236\,12 & 3.23612 \\ 1.618\,06 & 0.61806 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -0.000\,116 \\ -0.000\,058 \end{bmatrix}$$

$$\Delta x = \Delta y = -0.00003$$

$$x = 0.618\,06 - 0.000\,03 = 0.618\,03$$

$$y = 1.618\,06 - 0.000\,03 = 1.618\,03$$

Sistemas de equações não lineares

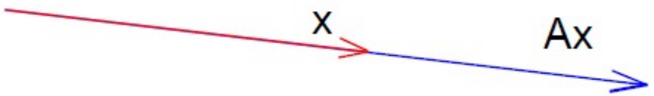
```
from scipy.optimize import fsolve
import math

def equations(p):
    x, y = p
    return (x**2+y**2-4, x*y - 1)

x, y =  fsolve(equations, (1, 1))

print(x,y)
print(equations((x, y)))
```

```
1.9318516525779594 0.5176380902053392
(-3.7614356074300304e-13, 4.833911049217932e-13)
```

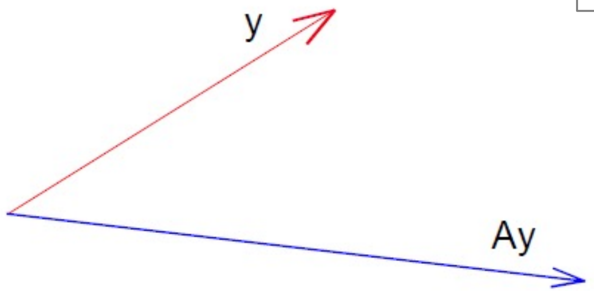
Valores e vectores próprios

Uma matriz quadrada A, quando aplicada sobre certos vetores (vetores próprios), obtém-se o próprio vetor multiplicado por uma constante λ (valor próprio):

$$\boxed{Ax = \lambda x}$$



- $Ax = \lambda x$
- Ax tem a mesma direcção de x
- comprimento e sentido de Ax depende de λ



- y não é valor próprio de A
- $Ay \neq \lambda y$
- Ay não tem a mesma direcção de y

Seja A uma matriz de ordem $n \times n$, queremos determinar um vector x, de ordem $n \times 1$ tal que

$$
\begin{aligned}
& Ax = \lambda x \\
\Leftrightarrow & Ax - \lambda x = 0 \\
\Leftrightarrow & Ax - \lambda I x = 0 \\
\Leftrightarrow & (A - \lambda I)x = 0
\end{aligned}
$$

Temos de resolver o sistema homogéneo $(A - \lambda I)x = 0$ de maneira a encontrar a solução não trivial, pois $x \neq 0$, para tal o determinante da matriz dos coeficientes tem de ser nulo:

$$|A - \lambda I| = 0$$

Valores e vectores próprios

Desenvolvimento do determinante característico $|A - \lambda I| = 0$ origina o polinómio característico

$$c_0\lambda^n + c_1\lambda^{n-1} + \ldots + c_{n-1}\lambda + c_n = 0$$
$$(\lambda - \lambda_1)(\lambda - \lambda_2)\cdots(\lambda - \lambda_n) = 0$$

cujas n raízes $\lambda_1, \lambda_2, \ldots \lambda_n$ são os valores próprios de A
Resolução dos n sistemas

$$(A - \lambda_i I)x_i = 0 \quad \text{para} \quad i = 1, 2, \ldots, n$$

permite obter os n vectores próprios correspondentes $x_1, x_2, \ldots, x_n$
Como $|A - \lambda I| = 0$ cada sistema $(A - \lambda_i I)x_i = 0$ tem solução múltipla, pelo que cada x_i representa uma solução particular extraída arbitrariamente da solução geral

Valores e vectores próprios

Determinar os valores próprios de $\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$

$$|A - \lambda I| = 0 \Leftrightarrow \left| \begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right| = 0$$

$$\Leftrightarrow \begin{vmatrix} 3-\lambda & -1 \\ -1 & 3-\lambda \end{vmatrix} = 0$$

$$\Leftrightarrow (3-\lambda)(3-\lambda) - (-1)(-1) = 0$$

$$\Leftrightarrow \lambda^2 - 6\lambda + 8 = 0$$

$$\Leftrightarrow \lambda = \frac{6 \pm \sqrt{36-32}}{2}$$

$$\Leftrightarrow \lambda_1 = 2, \quad \lambda_2 = 4$$

Valores e vectores próprios

Determinar os vectores próprios de $\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix}$

$$(A - \lambda_1 I)\, x_1 = 0 \Leftrightarrow \left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - 2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 3-2 & -1 \\ -1 & 3-2 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} x_{11} - x_{12} = 0 \\ 0 = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{11} = x_{12} \\ x_{12} \in \mathbb{R} \end{cases} \Leftrightarrow x_1 = \begin{bmatrix} \alpha \\ \alpha \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \text{com} \quad \alpha \in \mathbb{R}$$

$$\text{se} \quad \alpha = 1 \Rightarrow x_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Valores e vectores próprios

$$(A-\lambda_2 I)\,x_2 = 0 \Leftrightarrow \left(\begin{bmatrix} 3 & -1 \\ -1 & 3 \end{bmatrix} - 4\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} 3-4 & -1 \\ -1 & 3-4 \end{bmatrix}\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} -1 & -1 \\ -1 & -1 \end{bmatrix}\begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Leftrightarrow \begin{cases} -x_{11} - x_{12} = 0 \\ 0 = 0 \end{cases}$$

$$\Leftrightarrow \begin{cases} x_{11} = -x_{12} \\ x_{12} \in \mathbb{R} \end{cases} \Leftrightarrow x_2 = \begin{bmatrix} -\alpha \\ \alpha \end{bmatrix} = \alpha\begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \text{com} \quad \alpha \in \mathbb{R}$$

$$\text{se} \quad \alpha = 1 \Rightarrow x_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$$

Valores e vectores próprios

- Valores próprios de matriz diagonal são os próprios elementos da diagonal
- Valores próprios de matriz triangular são os elementos da diagonal
- Matriz A de ordem n possui exactamente n valores próprios
- Se A for simétrica ($A = A^T$) os seus valores próprios são todos reais e os respectivos vectores próprios são ortogonais

Valores e vectores próprios

```python
import numpy as np
from numpy.linalg import eig

a=np.array([[1,2,3],[4,5,6],[7,8,9]])

values , vectors = eig(a)
print(values)
print()
print(vectors)
```

```
[ 1.61168440e+01 -1.11684397e+00 -1.30367773e-15]

[[-0.23197069 -0.78583024  0.40824829]
 [-0.52532209 -0.08675134 -0.81649658]
 [-0.8186735   0.61232756  0.40824829]]
```

Derivada de 1ª ordem

As várias fórmulas de derivação numérica obtêm-se fazendo combinações simples da expansão em série de Taylor da função nos pontos próximos do ponto onde se pretende calcular a derivada...

$$f_{k+1} = f_k + f_k'(x_{k+1} - x_k) + \frac{1}{2} f_k''(x_{k+1} - x_k)^2 + \cdots$$

Derivada de 1ª ordem
(2 pontos)

$$f_{k+1} = f_k + f'_k(x_{k+1} - x_k) + \frac{1}{2} f''_k(x_{k+1} - x_k)^2 + \cdots$$

$$\Downarrow \quad h = x_{k+1} - x_k$$

$$f'_k = \frac{f_{k+1} - f_k}{h} + \mathcal{O}(h)$$