Atividades Treinamento 01 para Adrian

Professor: Luís Cláudio Oliveira-Lopes

1. Matrizes e Vetores

Estudos Recomendados

- VanderPlas, J. Python for Data Science Handbook, Capítulos 1 a 3.
- NumPy Documentation: https://numpy.org/doc/

Exercícios Propostos

- 1. Crie uma matriz 3×3 com valores aleatórios e calcule sua transposta, determinante e inversa.
- 2. Multiplique duas matrizes compatíveis e verifique a comutatividade.
- 3. Represente vetores de entrada e saída em um sistema de mistura com duas correntes de entrada.

Exemplo de Código Python

import numpy as np

```
A = np.array([[1, 2], [3, 4]])
```

B = np.array([[5, 6], [7, 8]])

C = np.dot(A, B)

print("Produto Matricial:\n", C)

print("Determinante de A:", np.linalg.det(A))

print("Transposta de A:\n", A.T)

2. Sistemas Algébricos Lineares

Estudos Recomendados

• Kiusalaas, J. Numerical Methods in Engineering with Python 3, Capítulo 6.

• Strang, G. Introduction to Linear Algebra (MIT OpenCourseWare).

Exercícios Propostos

1. Resolva o sistema:

$$\begin{cases} 3x + y - z = 1 \ 2x - 2y + 4z = -2 \ -x + 0.5y - z = 0 \end{cases}$$

com np.linalg.solve().

- 2. Calcule o número de condição da matriz dos coeficientes.
- 3. Implemente o método de Gauss-Seidel para o mesmo sistema.
- 4. Estudo de caso: balanço de massa em uma malha de tanques conectados.

Exemplo de Código Python

import numpy as np

A = np.array([[3, 1, -1], [2, -2, 4], [-1, 0.5, -1]])
b = np.array([1, -2, 0])
$$x = np.linalg.solve(A, b)$$

print("Solução do sistema:", x)
print("Número de condição:", np.linalg.cond(A))

3. Sistemas Algébricos Não Lineares

Estudos Recomendados

- Chapra, S. Applied Numerical Methods with MATLAB, Capítulo 5.
- SciPy Documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.ht ml

Exercícios Propostos

1. Resolva a equação

$$f(x) = x^3 - x - 2 = 0$$

usando o método de Newton.

2. Resolva o sistema não linear:

$$\begin{cases} x^2 + y^2 = 4 \\ x - y = 1 \end{cases}$$

com scipy.optimize.fsolve.

3. Estudo de caso: equilíbrio líquido-vapor isobárico com dois componentes e KiK_iKi constante.

Exemplo de Código Python

from scipy.optimize import fsolve

def equations(vars):

sol = fsolve(equations, [1, 1]) print("Solução:", sol)

4. Sistema de Equações Diferenciais Ordinárias (EDOs)

Estudos Recomendados

- Kiusalaas, J. *Numerical Methods in Engineering with Python 3*, Capítulo 8.
- SciPy Documentation: https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint .html e https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.solve_ivp.html

Exercícios Propostos

1. Resolva:

$$\frac{dC}{dt} = -kC, \quad C(0) = 1, \quad k = 0.1$$

com solve_ivp.

def cstr(t, y):

- 2. Modele um reator CSTR com reação exotérmica e calcule a evolução de Ca(t) e T(t).
- 3. Compare as soluções com odeint e solve_ivp.
- 4. Realize análise de sensibilidade para perturbações de vazão ou temperatura.

Exemplo de Código Python

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt
```

```
Ca, T = y

k = 10**(10 - 5000 / T)

dCadt = 1.0 - k * Ca

dTdt = 0.5 - 1500 * k * Ca

return [dCadt, dTdt]

sol = solve_ivp(cstr, [0, 10], [1.0, 350.0], t_eval=np.linspace(0, 10, 100))
```

```
plt.plot(sol.t, sol.y[0], label='Ca')
plt.plot(sol.t, sol.y[1], label='T')
plt.xlabel('Tempo')
plt.ylabel('Variáveis de Estado')
plt.legend()
```

```
plt.title('Resposta Dinâmica do CSTR')
plt.grid(True)
plt.show()
```

Plano Semanal: Atividades de Treinamento 01 para Adrian

Duração: 4 semanas

Objetivo Geral: Desenvolver domínio em modelagem e resolução computacional de problemas envolvendo álgebra linear, sistemas não lineares e equações diferenciais, com aplicações reais da engenharia química.

Semana 1: Matrizes e Vetores com Aplicações

Objetivo da semana: Compreender e aplicar operações matriciais com NumPy e analisar situações reais simples.

Metas:

- Estudar capítulos 1 a 3 do Python Data Science Handbook (VanderPlas).
- Executar exemplos de criação, transposição, inversão e determinante de matrizes.
- Resolver o problema de mistura de duas correntes com balanço de massa.
- Reproduzir e modificar o código de multiplicação matricial com np.dot.

Entregável da semana:

- Notebook Python com 3 exemplos resolvidos envolvendo vetores, produtos e operações com matrizes.
- Relatório de 1 página explicando a aplicação em uma planta de mistura.

✓ Semana 2: Sistemas Lineares e Estudo de Casos

Objetivo da semana: Resolver sistemas lineares por métodos diretos e iterativos, com interpretação em contextos industriais.

Metas:

- Estudar Cap. 6 do livro de Kiusalaas.
- Resolver 3 sistemas com np.linalg.solve() e verificar condicionamento.
- Estudar e implementar o método de Gauss-Seidel.
- Aplicar em uma malha de 3 tanques interligados com entrada e saída conhecidas.

Entregável da semana:

- Scripts Python com soluções de sistemas.
- Análise numérica e interpretação física dos resultados.

Semana 3: Sistemas Não Lineares e Métodos Numéricos

Objetivo da semana: Resolver equações não lineares e sistemas com fsolve e Newton-Raphson, com aplicações em equilíbrio de fases.

Metas:

- Estudar Cap. 5 do livro de Chapra.
- Resolver equação cúbica por métodos numéricos.
- Resolver sistema n\u00e3o linear com fsolve.
- Estudar o equilíbrio líquido-vapor com aproximação de K-values constantes.

Entregável da semana:

- Código com pelo menos 2 sistemas não lineares resolvidos.
- Modelo simplificado de sistema flash isobárico implementado em Python.

Semana 4: EDOs Aplicadas ao Controle de Reatores

Objetivo da semana: Implementar simulação dinâmica de reatores com equações diferenciais ordinárias.

Metas:

- Estudar Cap. 8 do livro de Kiusalaas.
- Simular decaimento de reagente com solve_ivp.
- Implementar simulação de CSTR com balanço de massa e energia.
- Analisar resposta do sistema a perturbações.

Entregável da semana:

- Código com simulação de CSTR.
- Gráficos de Ca(t) e T(t) com e sem perturbação.
- Análise descritiva da estabilidade e desempenho.

Avaliação Final (Opcional):

- Redação de um relatório técnico com os resultados obtidos.
- Discussão comparativa dos métodos utilizados.
- Sugestões de melhoria ou extensão dos modelos (ex.: inclusão de controle).