

U.C. 21180

Computação Numérica

19 de dezembro de 2025 a 5 de janeiro de 2026

”

E-fólio B | Instruções para a realização do E-fólio



- Leia estas instruções na totalidade antes de iniciar a resolução da prova.
- Este enunciado constitui o elemento de avaliação designado por "e-fólio B" no âmbito da avaliação contínua e tem a cotação total de 4 valores.
- A resolução deve ser entregue através de um único ficheiro compactado .zip contendo os ficheiros .m e o ficheiro relatorio.pdf.
- Nota ética: O trabalho deve ser original. Não é permitida a consulta a agentes de IA para geração de código.

Grupo I [4 valores]

1. Os métodos iterativos são essenciais para a resolução de grandes sistemas lineares esparsos, onde os métodos diretos (como a Eliminação de Gauss) são computacionalmente custosos. O objetivo deste trabalho é implementar e analisar o Método SOR (*Successive Over-Relaxation*).

1.1. [1.0] A convergência dos métodos iterativos clássicos é garantida se a matriz de coeficientes A for estritamente dominante diagonalmente por linhas. Escreva a função `verif_dominancia(A)`.

```
function [is_dom, r] = verific_dominancia(A)
%
% Verifica se a matriz A é estritamente dominante diagonalmente
% A: Matriz quadrada de coeficientes
% is_dom: booleano (true se for dominante, false caso contrário)
% r: vetor com a razão (soma extra-diagonal / diagonal) por linha
```

A função deve retornar **true** apenas se $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ para todas as linhas i . O vetor de saída r deve conter, para cada linha, o rácio de ocupação, permitindo identificar as linhas mais problemáticas para a convergência. Utilize operações vetoriais sempre que possível.

1.2. [1.5] Escreva a função `solve_sor(A, b, omega, x0, tol, max_iter)` que implemente o método SOR para resolver $Ax = b$. A atualização de cada componente x_i na iteração $k + 1$ segue a expressão:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)} \right)$$

```
function [x, iter] = solve_sor(A, b, omega, x0, tol, max_iter)
%
% Resolve Ax=b pelo método SOR
% omega: fator de relaxamento
% x0: estimativa inicial
% tol: tolerância de paragem (norma infinito do erro entre iterações)
% x: vetor solução; iter: número de iterações realizadas
```

Requisitos:

- Implemente o critério de paragem $\|x^{(k+1)} - x^{(k)}\|_\infty < tol$.
- Se exceder `max_iter`, a função deve avisar o utilizador e devolver a última aproximação.
- Não utilize a inversão de matrizes (`inv`).

1.3. [1.5] Elabore um script de teste `efb_sor.m` que:

- Crie uma matriz A de dimensão 100×100 tridiagonal e dominante (ex: diagonal principal = 4, diagonais superior/inferior = -1). Defina b tal que a solução exata seja um vetor de uns.
- Utilize a função da alínea 1.1 para validar a matriz.
- Resolva o sistema usando a função da alínea 1.2 para três valores de ω :
 - $\omega = 1.0$ (Gauss-Seidel)
 - $\omega = 1.25$ (Sobrerrelaxamento otimizado)
 - $\omega = 1.8$ (Alto relaxamento)
- Compare o número de iterações assim como o tempo de execução obtido em cada caso e comente os resultados no relatório, discutindo a influência do parâmetro ω na velocidade de convergência.

Todos os ficheiros devem ter o cabeçalho de identificação padrão.

Cr terios de corre  o

- N o   permitida a utiliza  o de vari veis globais (uso da palavra chave "global") nem da fun  o `fplot()`.
- N o   permitida a utiliza  o de fun  es j  existentes no Octave an logas  s que se pretendem desenvolver nem a utiliza  o de fun  es de pacotes de software (packages) adicionais   instala  o base do Octave.
- As fun  es pedidas devem implementar rigorosamente a interface de argumentos de entrada e de sa da indicados. N o devem pedir dados ao utilizador nem imprimir dados ou gr ficos n o solicitados no enunciado.
- Os programas devem empregar sempre que poss vel opera  es vetoriais e/ou matriciais (ex. produto interno de vetores, etc) em detrimento de ciclos que manipulam simples escalares.
- O c digo dos programas deve estar correta e uniformemente indentado de modo a permitir a sua leitura f cil.
- Os programas devem estar estruturados, comentados e em conjunto com o relat rio explicados de modo   f cil compreens o da sua estrutura e funcionamento.
- Os programas que n o funcionem corretamente ou n o cumpram todas as especifica  es ou sejam demasiado complexos \Rightarrow de 0 a 100% valores, sendo cada programa avaliado como um todo e tendo em conta a implementa  o das caracter sticas pedidas.

FIM