

”

E-fólio B | Folha de resolução para E-fólio



UNIDADE CURRICULAR: Computação Numérica

CÓDIGO: 21180

DOCENTE: Ivo Robert

A preencher pelo estudante

NOME: Ruben Brites

N.º DE ESTUDANTE: 2402116

CURSO: Engenharia Informática

DATA DE ENTREGA: 05/01/25

TRABALHO / RESOLUÇÃO:

Introdução

Os sistemas lineares surgem frequentemente em problemas de engenharia e ciência, sobretudo quando se tratam de modelos matemáticos discretizados. Quando a dimensão do sistema é elevada e a matriz associada é esparsa, os métodos diretos tornam-se pouco eficientes. Nestes casos, os métodos iterativos assumem um papel fundamental.

Neste e-fólio é estudado o método SOR (Successive Over-Relaxation), um método iterativo que generaliza o método de Gauss-Seidel através da introdução de um fator de relaxamento, permitindo acelerar a convergência em determinadas situações.

Exercício 1 — Método SOR

1.1 Verificação da dominância diagonal

A convergência dos métodos iterativos clássicos é garantida quando a matriz de coeficientes é estritamente dominante diagonalmente por linhas. Uma matriz $A = [a_{ij}]$ verifica esta condição se, para cada linha i , se cumprir:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$$

Com o objetivo de verificar esta condição, foi implementada a função `verif_dominancia(A)`, que devolve um valor lógico `is_dom`, indicando se a matriz é dominante, e um vetor `r` com os rácios entre a soma dos valores extra-diagonais e o valor absoluto da diagonal em cada linha.

O rácio r_i é definido por:

$$r_i = (\sum_{j \neq i} |a_{ij}|) / |a_{ii}|.$$

Valores inferiores a 1 indicam que a linha respeita a condição de dominância.

A implementação recorreu a operações vetoriais, nomeadamente à extração da diagonal com `diag(A)` e à soma por linhas com `sum(abs(A),2)`, evitando ciclos desnecessários e tornando o código mais eficiente.

1.2 Implementação do método SOR

O método SOR é uma extensão do método de Gauss-Seidel, introduzindo um fator de relaxamento ω . A atualização de cada componente da solução é feita de forma sequencial, utilizando valores já atualizados.

A fórmula de atualização utilizada foi:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + (\omega/a_{ii})(b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)})$$

Quando $\omega = 1$, o método coincide com o método de Gauss-Seidel. Para valores de ω superiores a 1, pode verificar-se uma aceleração da convergência, embora valores demasiado elevados possam prejudicar o desempenho.

O critério de paragem adotado foi baseado na norma infinito da diferença entre duas iterações consecutivas, ou seja:

$$\|x^{(k+1)} - x^{(k)}\|_{\infty} < \text{tol.}$$

Foi ainda imposto um número máximo de iterações, sendo emitido um aviso ao utilizador caso este limite seja excedido. Não foi utilizada a inversão de matrizes, cumprindo integralmente as restrições do enunciado.

1.3 Script de teste e análise dos resultados

Para testar o método implementado, foi criado o script `efb_sor.m`. Neste script construiu-se uma matriz tridiagonal de dimensão 100×100 , com diagonal principal igual a 4 e diagonais adjacentes iguais a -1 .

O vetor b foi definido de modo a que a solução exata do sistema fosse um vetor de uns, o que permitiu avaliar facilmente o erro da solução aproximada.

Foram testados três valores do fator de relaxamento: $\omega = 1.0$, $\omega = 1.25$ e $\omega = 1.8$. Para cada caso foram medidos o número de iterações necessárias para convergência e o tempo de execução.

Os resultados mostraram que, para este problema, o método de Gauss-Seidel ($\omega = 1.0$) apresentou a convergência mais eficiente. O valor $\omega = 1.25$ não trouxe melhorias significativas, enquanto $\omega = 1.8$ resultou num aumento considerável do número de iterações e do tempo de execução.

Conclusão

Neste e-fólio foram implementadas e testadas com sucesso as funções pedidas para o método SOR. Os resultados obtidos confirmam a influência determinante do fator de relaxamento ω na velocidade de convergência e evidenciam a importância de uma escolha adequada deste parâmetro.