



UNIDADE CURRICULAR: Computação Numérica

CÓDIGO: 21180

DOCENTE: Yves Robert e Filipe Pais

A preencher pelo estudante

NOME: Luís Carlos Crispim Pereira

N.º DE ESTUDANTE: 2300163

CURSO: LEI – Licenciatura em Engenharia Informática

DATA DE ENTREGA: 03/01/2026

TRABALHO / RESOLUÇÃO:

Introdução:

O presente trabalho insere-se no âmbito da unidade curricular de Computação Numérica e tem como objetivo o estudo e aplicação de métodos iterativos para a resolução de sistemas lineares. Em particular, é analisado o método SOR (Successive Over-Relaxation), um método amplamente utilizado para a resolução eficiente de sistemas do tipo $Ax = b$, especialmente quando a dimensão do sistema é elevada.

O trabalho procura não só implementar o método de forma correta, respeitando os critérios de convergência associados, mas também analisar o impacto dos parâmetros envolvidos no seu desempenho. Um aspeto central desta análise é a influência do parâmetro de relaxamento ω na velocidade de convergência do método, avaliada através do número de iterações necessárias e do tempo de execução.

Para esse efeito, considera-se um sistema linear de dimensão fixa, construído de forma a garantir condições favoráveis à convergência dos métodos iterativos. Os resultados obtidos permitem comparar o comportamento do método para diferentes valores de ω , evidenciando a importância de uma escolha adequada deste parâmetro para a eficiência do processo iterativo.

1.1: Enunciado - Verificação da dominância diagonal: Pretende-se desenvolver uma função que permita verificar se uma matriz quadrada é estritamente dominante diagonalmente, condição que garante a convergência de vários métodos iterativos clássicos para a resolução de sistemas lineares. A função deve ainda fornecer, para cada linha da matriz, uma medida quantitativa do grau de dominância diagonal, permitindo uma análise mais detalhada da matriz em estudo.

Descrição e implementação: Para dar resposta ao enunciado, foi implementada a função *verif_dominancia(A)*, cuja finalidade é analisar a matriz de coeficientes do sistema linear e verificar se satisfaz a condição de dominância diagonal estrita.

Numa primeira fase, a função verifica se a matriz fornecida é quadrada. Esta verificação é essencial como salvaguarda, uma vez que a definição de dominância diagonal e a própria formulação dos sistemas lineares do tipo $Ax = b$ pressupõem matrizes quadradas. Caso esta condição não se verifique, a execução da função é interrompida.

De seguida, são calculados os valores absolutos dos elementos da diagonal principal da matriz, representados pelo vetor $d = |a_{ii}|$, bem como a soma dos valores absolutos dos restantes elementos de cada linha, representados pelo vetor $s = \sum_{j \neq i} |a_{ij}|$. Estes cálculos são efetuados recorrendo a operações vetoriais e matriciais, evitando ciclos explícitos, de modo a tornar a implementação mais clara e eficiente. Em particular, o vetor d é obtido a partir da diagonal principal da matriz, enquanto o vetor s resulta da soma dos valores absolutos de cada linha, subtraindo a contribuição da diagonal.

A condição de dominância diagonal estrita é então avaliada comparando, para cada linha, o valor absoluto do elemento diagonal com a soma dos valores absolutos dos elementos extra-diagonais, isto é, verificando se $|a_{ii}| > \sum_{j \neq i} |a_{ij}|$. Se esta condição se verificar para todas as linhas da matriz, a função devolve o valor lógico *true*, caso contrário, devolve *false*.

Adicionalmente, a função calcula um vetor r , cujos elementos correspondem à razão entre a soma dos elementos extra-diagonais e o valor absoluto do elemento diagonal em cada linha, $r_i = \frac{s_i}{d_i}$. Este vetor permite avaliar o grau de dominância diagonal da matriz, sendo valores de $r_i < 1$ indicativos de dominância diagonal nessa linha e valores mais próximos de 1 associados a situações potencialmente mais problemáticas para a convergência dos métodos iterativos.

Desta forma, a função fornece não apenas uma verificação lógica da condição de dominância diagonal, mas também uma informação quantitativa complementar que pode ser útil na análise do comportamento dos métodos iterativos.

1.2 - Enunciado - Método SOR: Pretende-se desenvolver uma função que permita resolver um sistema linear do tipo $Ax = b$ recorrendo ao método iterativo SOR (Successive Over-Relaxation). A função deve receber como argumentos a matriz dos coeficientes, o vetor do termo independente, o parâmetro de relaxamento ω , uma estimativa inicial da solução, uma tolerância de paragem e um número máximo de iterações, devolvendo a aproximação da solução e o número de iterações realizadas. O critério de paragem deve basear-se na norma infinito da diferença entre duas iterações consecutivas.

Descrição e implementação: Para dar resposta ao enunciado, foi implementada a função `solve_sor(A, b, omega, x0, tol, max_iter)`, que aplica o método SOR de forma iterativa à resolução do sistema linear $Ax = b$.

A função começa por definir a dimensão do sistema a partir do vetor b e por garantir que a estimativa inicial é tratada como um vetor coluna. De seguida, inicia-se um ciclo iterativo, limitado superiormente pelo número máximo de iterações especificado.

Em cada iteração, é guardada a aproximação da solução da iteração anterior, de modo a permitir a avaliação do critério de paragem. O método SOR procede então à atualização sequencial de cada componente do vetor solução. Para cada índice i , a soma associada ao produto matricial é dividida em duas partes, a contribuição dos elementos já atualizados na iteração corrente e a contribuição dos elementos ainda não atualizados, provenientes da iteração anterior. Esta separação permite implementar corretamente a fórmula do método SOR.

A atualização de cada componente da solução segue a expressão:

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

onde ω representa o parâmetro de relaxamento.

Como medida de salvaguarda, é verificado se o elemento diagonal a_{ii} é nulo antes de efetuar a atualização, uma vez que tal situação tornaria o método inaplicável devido à divisão por zero. Caso essa situação ocorra, a execução da função é interrompida.

Após a atualização completa do vetor solução numa dada iteração, é avaliado o critério de paragem, baseado na norma infinito da diferença entre a solução atual e a solução da iteração anterior, $\|x^{(k+1)} - x^{(k)}\|_{\infty} < tol$.

Se esta condição for satisfeita, a função termina, devolvendo a aproximação da solução e o número de iterações efetuadas. Caso contrário, o processo iterativo prossegue até ser atingido o número máximo de iterações definido.

A tolerância de paragem é fornecida como argumento da função, permitindo controlar o compromisso entre precisão da solução e custo computacional sem necessidade de alterar a implementação do método. Caso o número máximo de iterações seja atingido sem que o critério de paragem seja satisfeito, é emitido um aviso informativo.

Desta forma, a função implementa o método SOR de forma genérica e flexível, permitindo a sua aplicação a diferentes sistemas lineares e diferentes escolhas do parâmetro de relaxamento ω .

1.3 – Enunciado - Aplicação do método SOR e análise da influência do parâmetro ω : Pretende-se desenvolver um script que utilize as funções desenvolvidas nas Questões 1.1 e 1.2 para construir e resolver um sistema linear do tipo $Ax = b$ de dimensão $n = 100$ recorrendo ao método SOR. O script deve aplicar o método para diferentes valores do parâmetro de relaxamento ω , comparar o número de iterações e o tempo de execução obtidos em cada caso e permitir analisar a influência de ω na velocidade de convergência do método.

Descrição e implementação: Para dar resposta ao enunciado, foi desenvolvido o script *efb_sor.m*, que integra as funções previamente implementadas nas Questões 1.1 e 1.2, assegurando uma abordagem modular e estruturada à resolução do problema.

Numa primeira fase, o script procede à limpeza do ambiente de trabalho, garantindo que a execução decorre sem interferência de variáveis anteriormente definidas. Em seguida, é definida a dimensão do sistema, $n = 100$, e é construída a matriz dos coeficientes A , com estrutura tridiagonal, composta por uma diagonal principal de valor 4 e duas diagonais adjacentes de valor -1 . Esta matriz é estritamente dominante diagonalmente, condição favorável à convergência dos métodos iterativos considerados.

Com o objetivo de permitir a validação dos resultados obtidos, é considerada uma solução exata conhecida, definida por um vetor cujas componentes são todas unitárias. O vetor do termo independente é então calculado como $b = Ax_{exato}$, assegurando que o sistema linear possui uma solução conhecida.

Antes da aplicação do método SOR, o script recorre à função desenvolvida na Questão 1.1 (*verif_dominancia*) para confirmar a dominância diagonal da matriz A , funcionando esta verificação como uma validação das condições teóricas de convergência do método.

De seguida, são definidos os parâmetros necessários à aplicação do método SOR. A estimativa inicial da solução é escolhida como o vetor nulo, opção comum em métodos iterativos quando não existe informação prévia sobre a solução, permitindo analisar o comportamento do método a partir de uma aproximação neutra.

A tolerância de paragem é definida como $tol = 10^{-6}$, valor suficientemente pequeno para garantir uma boa aproximação da solução exata, sem comprometer de forma significativa o custo computacional, tendo em conta a dimensão do sistema considerado.

O número máximo de iterações é fixado como um valor suficientemente elevado, $max_iter = 10000$, para garantir que o método tenha oportunidade de convergir sempre que tal seja possível, funcionando simultaneamente como uma salvaguarda que impede a execução indefinida do algoritmo caso o critério de paragem não seja satisfeito.

O script aplica então a função desenvolvida na Questão 1.2 (*solve_sor*) para diferentes valores do parâmetro de relaxamento ω . Em particular, considera-se $\omega = 1$ como caso base (equivalente a Gauss–Seidel) e valores $\omega > 1$ para avaliar o efeito da sobre-relaxação na velocidade de convergência. Em cada caso, são registados o número de iterações efetuadas e o tempo de execução, permitindo comparar o impacto deste parâmetro no desempenho do método.

Os resultados são apresentados sob a forma de tabela, por permitir uma comparação direta entre valores de ω , número de iterações e tempo de execução, conforme solicitado no enunciado. Adicionalmente, é calculado o erro final em norma infinito relativamente à solução exata apenas como verificação da eficácia do critério de paragem adotado na Questão 1.2, não sendo este valor utilizado como métrica principal de comparação.

Resultados e Conclusões:

A execução do script desenvolvido na Questão 1.3 foi realizada no ambiente Octave, tendo sido obtida a seguinte tabela de resultados, onde se apresentam o número de iterações, o tempo de execução e o erro final para diferentes valores do parâmetro de relaxamento ω .

```
A matriz é estritamente dominante diagonalmente.
```

omega	iteracoes	tempo (s)	erro
1.00	14	0.0149	2.09e-07
1.25	16	0.0164	2.78e-07
1.80	93	0.0953	4.57e-07

```
>> |
```

Os resultados mostram que, para $\omega = 1.00$, correspondente ao método de Gauss–Seidel, o método converge rapidamente, necessitando de um número reduzido de iterações e apresentando um tempo de execução baixo. Para $\omega = 1.25$, o comportamento é semelhante, verificando-se uma convergência eficiente com um custo computacional comparável.

Por outro lado, para $\omega = 1.80$, observa-se um aumento significativo do número de iterações e do tempo de execução, indicando que valores excessivamente elevados do parâmetro de relaxamento podem prejudicar a velocidade de convergência do método SOR. Estes resultados confirmam que a escolha de ω influencia de forma determinante o desempenho do método.

O erro final, calculado em norma infinito relativamente à solução exata, é da ordem de 10^{-7} em todos os casos, confirmando que o critério de paragem baseado na diferença entre iterações consecutivas foi corretamente satisfeito. Este erro serve apenas como validação adicional da convergência do método, não sendo utilizado como métrica principal de comparação.

Em síntese, os resultados obtidos evidenciam de forma clara a importância da escolha adequada do parâmetro de relaxamento ω , sendo que valores moderados conduzem a uma convergência eficiente, enquanto valores excessivamente elevados podem resultar num aumento significativo do número de iterações e do tempo de execução. O trabalho permitiu, assim, consolidar a compreensão do método SOR e a sua aplicação prática à resolução de sistemas lineares, evidenciando a relação entre a escolha dos parâmetros do método e o seu desempenho computacional.

De um modo geral, a metodologia adotada revelou-se adequada aos objetivos propostos, permitindo analisar de forma sistemática o comportamento do método SOR e reforçando a importância de uma escolha criteriosa dos seus parâmetros na resolução numérica de sistemas lineares.