



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA



SILVANA RODRIGUES ALVES

Métodos para resolução de sistemas lineares de grande porte

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. Dr. Paulo José da Silva e Silva .

Campinas
Novembro de 2022

1 Espaços de Krylov

Os espaços de Krylov se destacam no desenvolvimento de alguns métodos importantes para a resolução de sistemas de grande porte. Definindo o espaço:

Seja $A \approx I, b \neq 0$:

$$K(A, b, k) = \{b, Ab, A^2b, \dots, A^{k-1}b\},$$

Um método baseado em espaços de Krylov para aproximar um vetor x tem sua k -ésima interação de x no k -ésimo subespaço de Krylov $K(A, b, k)$.

Ao estudar a solução de sistemas lineares é possível encontrar diferentes resoluções. As técnicas possuem como objetivo resolver $Ax = b$, tendo A como uma matriz quadrada b um vetor chamado do lado direito. Alguns métodos diretos são baseado em fatoração LU ou QR que possuem um custo computacional $O(n^3)$, tornando inviável o uso para problemas de grande porte.

Para melhor visualização das ideias dos métodos que utilizam de espaços de Krylov, tome a princípio métodos iterativos clássicos, tais como o método de Jacobi e Gauss-Seide, que possuem como ideia base a "quebra" da matriz A como forma de encontrar a solução de um sistema $Ax = b$. A técnica principal é escrever $A = M - N$ onde M é fácil de inverter. Assim a solução x , para $Ax = b$ satisfaz $Mx = b + Nx$. Portanto, para encontrar o ponto fixo x , é usada a regra de atualização:

$$Mx^{(k)} = b + Nx^{(k-1)}.$$

O vetor inicial x^0 normalmente é zero.

Essa regra de atualização funciona bem se M for aproximadamente A . Para isso a condição de convergência é tomada como $\rho(M^{-1}N) < 1$. Ou seja, o maior autovalor de $M^{-1}N$ deve ser menor que 1. No caso de A ser aproximadamente a matriz identidade, é possível escolher $M = I$. Transformando assim a interação:

$$x^{(k)} = b + (I - A)x^{(k-1)} = x^{(k-1)} + b - Ax^{(k-1)}.$$

Dessa equação podemos ver que o próximo iterado, x^k , está no próximo espaço de Krylov com relação a x^{k-1} .

Os métodos de Krylov basicamente se baseiam na busca de uma aproximação $x^{(k)}$ nesses subespaços. Esta aproximação acaba tornando os métodos que utilizam esse espaço mais eficientes para a resolução de sistemas com matrizes esparsas.

Como a proposta é encontrar uma estimativa $x^{(k)} \in K(A, b, k)$ para a solução de $Ax = b$, isso naturalmente leva à questão de como é possível encontrar este vetor solução dentro do espaço que seja o mais próximo possível de x . Infelizmente, encontrar $x^{(k)} \in K(A, b, k)$ que possa minimizar $\|x^{(k)} - x\|_2$ é inviável, pois requer conhecimento da solução exata x . Em vez disso, é possível recorrer à minimização de diferentes medidas de erro, o que levará a uma ampla variedade de métodos.

Existem duas estratégias principais para o desenvolvimento dos métodos. Primeiro tentar minimizar o resíduo $r^{(k)} = b - Ax^{(k)}$ em alguma norma. Segundo busca-se garantir que o resíduo $r^{(k)}$ seja ortogonal ao último subespaço de Krylov. Ao decorrer do texto iremos apresentar diferentes métodos que utilizam dos dois pontos de vista e as consequências de se optar por um dos caminhos na resolução de problemas.

Como o estudo gira em torno dos métodos iterativos, observa-se que a cada iteração tem-se como resultado uma aproximação nova. É esperado que a cada iteração este resultado esteja “melhorando” até que o ponto seja uma aproximação aceitável.

2 CG - Gradientes Conjugados

Inicialmente, para uma melhor compreensão desse método é preciso estabelecer algumas ideias. Iremos tomar $x^{(k)} \in K(A, b, k)$ como sendo a k-esima iteração, em seguida determinar Q_k como sendo uma base ortogonal para o espaço $K(A, b, k)$, conseqüentemente é possível escrever $x^{(k)} = Q_k y$. Dessa forma foi definido o resíduo como :

$$x^{(k)} = b - Ax^{(k)} = b - AQ_k y.$$

Esse método pode ser compreendido de duas maneiras: tanto através da minimização de $r^{(k)}$ em uma norma apropriada, quanto forçando a situação de uma ortogonalidade entre o resíduo e o último subespaço de Krylov. Assumindo que a matriz A é simétrica positiva definida (SPD) é possível definir uma norma na qual conseguimos minimizar o resíduo na escolha de $x^{(k)} \in K(A, b, k)$:

$$\|r^{(k)}\|_{A^{-1}}^2 = (b - AQ_k y)^T A^{-1} (b - AQ_k y) = b^T x - 2y^T Q_k^T b + y^T Q_k^T A Q_k y,$$

$$\|r^2\|_{A^{-1}}^2 = (b - AQ_k y)^T A^{-1} (b - AQ_k y),$$

$$= bA^{-1}b - 2b^T A^{-1}AQ_k + y^T Q_k^T A^T A^{-1}AQ_k y,$$

$$= b^T x - 2(b^T Q_k)y + y^T Q_k^T A Q_k y.$$

Lembrando que queremos encontrar o melhor $x^{(k)}$ que aqui está representado pelo vetor y, queremos minimizar:

$$\|r^2\|_{A^{-1}}^2 = cte - 2v^T y + y^T M y = f(y).$$

Para isso, precisamos calcular:

$$\nabla f(x) = 2My - 2v = 0 \implies My = v,$$

Em que,

$$M = Q_k^T A Q_k \text{ e } v = (Q_k^T b).$$

Sendo assim,

$$Q_k^T A Q_k y = Q_k^T b \implies Q_k^T (b - A Q_k y) = 0.$$

Um ponto chave dessa equação é que a operação ocorre somente com informações do vetor b e da matriz A. Além disso, lembrando que a expressão dentro do parêntesis é o resíduo da iteração k, essa expressão equivale:

$$Q_k^T r^{(k)} = 0 \text{ ou } r^{(k)} \perp K(A, b, k).$$

Isso prova que a condição de minimizar o resíduo na norma A^{-1} é equivalente a uma condição de ortogonalidade. Isso ocorre de duas maneiras:

1. O método encontra $x^{(k)} \in K(A, b, k)$ tal que $r^{(k)} \perp K(A, b, k)$.
2. O método encontra $x^{(k)} \in K(A, b, k)$ tal que $\|r^{(k)}\|_{A^{-1}}$ é minimizado.

Qualquer uma delas define completamente as iterações.

2.1 Formulação de Hestenes-Stiefel

Uma importante derivação do algoritmo CG, é chamada de derivação de Hestenes-Stiefel, possuindo como base a condição de ortogonalidade proposta $r^{(k)} \perp K(A, b, k)$. Agora que entendemos que o objetivo do CG é encontrar uma iteração $x^{(k)} \in K(A, b, k)$ que satisfaça essa condição de ortogonalidade, o próximo passo é encontrar essa iteração considerando:

$$\Delta x^{(k)} = x^{(k+1)} - x^{(k)}.$$

Os vetores de atualização $\Delta x^{(k)}$ satisfazem uma propriedade especial: eles são A-conjugados. Os vetores w e z são A-conjugados, se satisfazem a condição de que $w^T A z = 0$. A ideia geométrica chave do CG é que as direções de busca são A-conjugadas, logo os vetores de atualização $\Delta x^{(k)}$ e $\Delta x^{(l)}$ devem obedecer a uma condição do tipo:

$$(\Delta x^{(k)})^T A x^{(l)} = 0, \text{ para } k \neq l.$$

A prova se dá porque $r^{(k)} - r^{(k+1)} = A \Delta x^{(k)}$ ambos perpendiculares a Q_k . Provando que $A \Delta x^{(k)} \perp Q_k$, então $A \Delta x^{(k)} \perp \Delta x^{(l)}$, $l \leq k - 1$.

Como A é definida positiva é possível escrever:

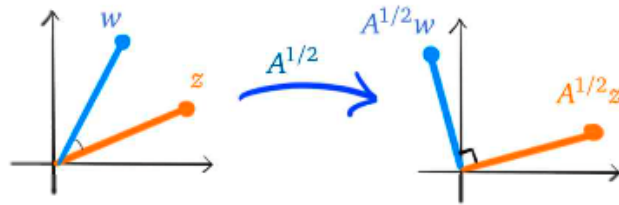
$$A = A^{(1/2)} A^{(1/2)},$$

Logo,

$$w^T A z = (A^{(1/2)} w)^T (A^{(1/2)} z).$$

A multiplicação feita é uma transformação linear. Assim, w e z são A-conjugadas se suas imagens por A se tornarem ortogonais após a transformação linear.

Figura 1: Transformação linear



Fonte: DARVE, Eric; WOOTTERS, Mary. Numerical Linear Algebra with Julia. SIAM, 2021, p270.

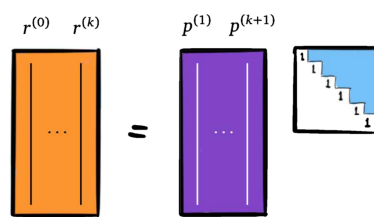
Após a transformação o algoritimo funciona como uma serie de passos de ortogonalização $\Delta x^{(1)}, \Delta x^{(2)}, \dots$ que a cada rodada aproxima-se mais da solução, vemos que não é possível realizar mais do que n passos, quando a solução exata deve ser encontrada.

Queremos encontrar uma expressão para o passo $A\Delta x^{(k)}$, e também para os vetores $p^{(1)}, \dots, p^{(k+1)}$ e em escalares μ_1, \dots, μ_{k+1} de modo que $\Delta x^{(l)} = \mu_{l+1}p^{(l+1)}$ para $l = 1, \dots, k$ Ou seja, $p^{(l+1)}$ corresponde a uma "direção de busca" e μ_{l+1} corresponde ao tamanho do passo que CG dá nessa direção. Observe que existe um caso degenerado quando $\Delta x^{(l)} = 0$. Nesse caso, a escolha de $p^{(l+1)} = 0$ funciona.

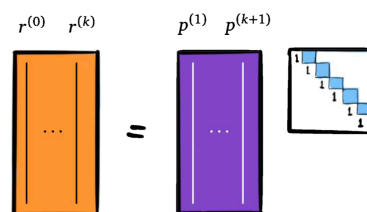
A razão pela qual é possível escolher 1 na diagonal se deve a não especificação da norma de $p^{(k)}$, apenas a direção. Logo, a norma é determinada de modo que 1 esteja na diagonal.

Recordando algumas regras ligadas a ortogonalidade de $p^{(k)}$:

Figura 2: Transformação linear



O resultado torna-se:



Fonte: DARVE, Eric; WOOTTERS, Mary. Numerical Linear Algebra with Julia. SIAM, 2021, p270.

1 - Vimos antes que as atualizações Δx_k e portanto as direções de busca são:

$$(p^{(l)})^T A p^{(k)} = 0 \text{ tendo } l \neq k.$$

2 - Uma outra condição é:

$$(p^{(l)})^T Ar^{(k)} = 0, l \leq k + 1.$$

Para poder observar esse caso, tem-se que o vetor p é paralelo a $\Delta x^{(l-1)} = x^{(l)} - x^{(l-1)} \in K(A, b, l)$ então $Ap^{(l)} \in K(A, b, l)$ consequentemente $Ap^{(l)} \in K(A, b, l + 1)$ então $Ap^{(l)} \in K(A, b, k)$ sempre que $l \leq k - 1$. A ortogonalidade pontua que o resíduo é ortogonal ao espaço $K(A, b, k)$ de modo que $Ap^{(l)}$ é ortogonal à $r^{(k)}$.

Da segunda condição é possível aferir que para qualquer $l \leq k - 1$:

Figura 3: Operação do vetor

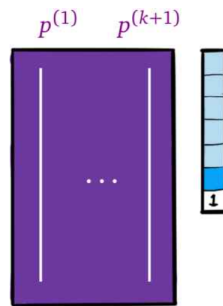


Figura 4: Fonte: DARVE, Eric; WOOTTERS, Mary. Numerical Linear Algebra with Julia. SIAM, 2021, p270.

$$0 = (p^{(l)})^T Ar^{(k)}.$$

Quando $r^{(k)}$ é substituído por algo em termos de p , é legítimo usar da condição de ortogonalidade 1 para observar que a maioria dos termos $(p^{(l)})^T Ap^{(j)}$ são zerados, levando ao seguinte resultado:

$$0 = a_l (p^{(l)})^T Ap^{(l)}.$$

Em que a_l é a l -ésima entrada, desde que $(p^{(l)})^T Ap^{(j)}$ seja diferente de zero, isso implica em $a_l = 0$ para qualquer $l \leq k - 1$ trazendo:

$$r^{(k)} = p^{(k+1)} - \tau_k p^{(k)}.$$

para todo k , e para algum coeficiente $\tau^{(k)}$.

Como o objetivo é calcular a direção de busca e o tamanho dos passos para que μ_k seja válido, é permitido supor que já foram completados k etapas do algoritmo, então haverá iterações $x^{(0)}, \dots, x^{(k-1)}$ e resíduos $r^{(0)}, \dots, r^{(k-1)}$, juntamente com a direção de busca $p^{(1)}, \dots, p^{(k)}$ e o tamanho dos passos μ_1, \dots, μ_k .

Sendo assim o objetivo é calcular $p^{(k+1)}$ e μ_{k+1} usando essa informação, o que dirá como obter a próxima iteração e o próximo resíduo. Começando com τ_k :

$$(p^{(k)})^T Ar^{(k)} = (p^{(k)})^T A(p^{(k+1)} - \tau_k (p^{(k)})^T) = -\tau_k (p^{(k)})^T Ap^{(k)}.$$

usando a condição de ortogonalidade 1 acima novamente. Isso significa que:

$$\tau_k = -\frac{(p^{(k)})^T A r^{(k)}}{(p^{(k)})^T A p^{(k)}}.$$

O que leva a possibilidade de começar a computar a direção de busca $p^{(k+1)}$ dada por $p^{(k)}$ e $r^{(k)}$ tendo como resultado $p^{(k+1)} = r^{(k)} + \tau_k p^{(k)}$. Agora, observando os passos na direção $x^{(k+1)} - x^{(k)} = \mu_{k+1} p^{(k+1)}$, ao multiplicar ambos os lados por uma matriz A:

$$r^{(k)} - r^{(k+1)} = \mu_{k+1} A p^{(k+1)}.$$

Usando a ortogonalidade novamente e multiplicando o lado esquerdo por $(p^{(k+1)})^T$ o resultado passa a ser:

$$(p^{(k+1)})^T r^{(k)} = \mu_{k+1} (p^{(k+1)})^T A p^{(k+1)}.$$

O lado esquerdo pode ser simplificado como:

$$(p^{(k+1)})^T r^{(k)} = (r^{(k)} + \tau_k p^{(k)})^T r^{(k)} = (r^{(k)})^T r^{(k)}.$$

Em que é possível usar um argumento semelhante ao usado acima para estabelecer a condição de ortogonalidade 2 para dizer que $(p^{(k)})^T r^{(k)} = 0$. Logo, existe uma forma de obter μ_{k+1} como:

$$\mu_{k+1} = \frac{(r^{(k)})^T r^{(k)}}{(p^{(k+1)})^T A p^{(k+1)}}.$$

Agora, considerando a direção de busca e o tamanho dos passos, é possível calcular a nova direção de busca e o novo tamanho dos passos. Usando esses resultados, a interação e o resíduo podemos encontrar.

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \mu_{k+1} p^{(k+1)}, \\ r^{(k+1)} &= r^{(k)} + \mu_{k+1} p^{(k+1)}. \end{aligned}$$

Neste ponto, já existe a possibilidade de escrever um algoritmo iterativo para calcular as estimativas x^k , no entanto, primeiro é preciso simplificar a equação τ_k para evitar a multiplicação por A. Isso permitirá realizar atualizações mais rapidamente.

$$A p^{(k)} = -\frac{1}{\mu_k} (r^{(k)} - r^{(k+1)}).$$

Substituindo na equação de τ_k onde é usado que $r^{(k+1)} \perp r^{(k)}$.

$$\tau_k = -\frac{(p^{(k)})^T A r^{(k)}}{(p^{(k)})^T A p^{(k)}} = -\frac{1}{\mu_k} \frac{(r^{(k)} - r^{(k+1)})^T r^{(k)}}{(p^{(k)})^T A p^{(k)}} = \frac{(r^{(k)})^T r^{(k)}}{((p^{(k)})^T A p^{(k)}) \mu_k} = \frac{(r^{(k)})^T r^{(k)}}{(r^{(k-1)})^T r^{(k-1)}}$$

Levando ao no algoritmo iterativo final.

Usando uma matriz simétrica positiva definida A e um vetor b:

1. Tome um $x^{(0)}$ por exemplo $x^{(0)} = 0$.

2. Tomando $r^{(0)} = b - Ax^{(0)}, p^{(0)} = 0, k = 1$.

3. Enquanto $r^{(k+1)} \neq 0$,

$$\tau_{k-1} = -\frac{(r^{(k-1)})^T r^{(k-1)}}{(r^{(k-2)})^T r^{(k-2)}},$$

$$p^{(k)} = r^{(k-1)} + \tau_{k-1} p^{(k-1)},$$

$$\mu_k = \frac{(r^{(k-1)})^T r^{(k-1)}}{(p^{(k)})^T A p^{(k)}},$$

$$x^{(k)} = x^{(k-1)} + \mu_k p^{(k)},$$

$$r^{(k)} = r^{(k-1)} - \mu_k A p^{(k)},$$

$$k \leftarrow k + 1.$$

4. Retorna $x^{(k+1)}$.

Esse código é feito para que ao final como resultado seja possível visualizar $x^{(k)}$.

2.2 Convergência do método Gradiente Conjugado

Como os subespaços de Krylov são definidos por potências da matriz A, estudar polinômios de A é uma das formas de estudar a convergência de métodos iterativos baseados em Krylov. Considere o método CG, que encontra $x^{(k)} \in K(A, b, k)$ minimizando:

$$\|r^{(k)}\|_{A^{-1}} = \|x - x^{(k)}\|_A,$$

O vetor $x^{(k)}$ está contido em $K(A, b, k)$, resultando em uma combinação linear de $b, Ab, \dots, A^{k-1}b$. Assim, é possível escrever $x^{(k)}$ como $Aq(A)x$, onde q é um polinômio de grau k-1:

$$\|r^{(k)}\|_{A^{-1}} = \|(I - Aq(A))x\|_A.$$

Seja $p(z) = 1 - zq(z)$, podemos reescrever o problema de minimizar $\|r^{(k)}\|_{A^{-1}}$ como um problema que busca encontrar p de grau k, tal que $p(0) = 1$ em que $p(A)x$ será minimizado. Ou seja:

$$\|x - x^{(k)}\|_A = \|r^{(k)}\|_{A^{-1}} = \min_{\substack{p(0)=1 \\ p \text{ de grau } k}} \|p(A)x\|_A.$$

Com as restrições $p(0) = 1$ e p de grau k.

A expressão acima não é tão útil porque a solução x não é conhecida. Porém é possível se obter polinômios para os quais limitantes úteis podem ser obtidos. De fato isso pode ser feito com polinômios de Chebyshev como feito em [92], chegando ao teorema de convergência:

Seja $x^{(k)}$ a k -ésima iteração de CG, e seja $r^{(k)} = b - Ax^{(k)}$ o resíduo. Então:

$$\|x - x^{(k)}\|_A = \|r^{(k)}\|_{A^{-1}} \leq \frac{\|r^{(0)}\|_{A^{-1}}}{T_k^{cb}(1 + 2/(k-1))} \leq 2\|r^{(0)}\|_{A^{-1}} \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k.$$

Onde T_k^{cb} é o polinômio de Chebyshev de ordem k , e $k = \lambda_{\min}/\lambda_{\max}$ um número de condição de A .

Quando os autovalores extremos de A estão bem separados dos demais, ocorre algo chamado super-convergência. A super-convergência significa que a taxa de convergência aumenta à medida que o número de iterações aumenta. Para ver por que isso pode acontecer suponha que alguns autovalores de A estejam bem separados dos outros. Os primeiros passos de CG criam um subespaço de Krylov que aproxima bem os autovetores correspondentes. Depois disso, no que diz respeito ao algoritmo, parecerá que esses autovalores foram "removidos" de A e como resultado o número de condição do sistema linear é reduzido, acarretando em uma convergência mais rápida[1,2].

3 MINRES e SYMMLQ

Ao estudar os métodos MINRES E SYMMLQ é possível diminuir as hipóteses sobre a matriz A em relação ao CG, nesse basta que A seja uma matriz simétrica não singular. Existem dois critérios equivalentes em que ambos os métodos utilizam para resolver $Ax = b$.

Objetivos do método MINRES:

1 - Encontrar $x^{(k)} \in K(A, b, k)$ tal que $\|r^{(k)}\|_2$ é minimizado.

2 - Encontrar $x^{(k)} \in K(A, b, k)$ tal que $A^T r^{(k)} \perp K(A, b, k)$.

O ponto principal em torno do MINRES é minimizar o resíduo, o que foca no primeiro objetivo. Existe um algoritmo diferente que trabalha o segundo objetivo que é conhecido como Conjugado resíduo (CR). Apesar de ambos terem como resultado as mesmas interações, alguns detalhes no CR faz com que o a interação pare quando A é indefinida. Já o método MINRES é sempre capaz de convergir para a solução exata em um número finito de interações.

Método SYMMLQ:

1 - Ao resolver $Ax = b$ para uma matriz A simétrica não singular SYMMLQ encontra $x^{(k)} \in K(A, b, k)$ tal que $r^{(k)} \perp K(A, b, k-1)$ e $\|r^{(k)}\|_2$ é minimizado. Essas propriedades definem unicamente $x^{(k)}$.

Estabelecendo algumas notações para melhor compreensão, seja Q_k a matriz $n \times k$ descoberta por Lanczos, cujo colunas formam uma base $K(A, b, k)$. Considere $(k+1) \times k$ a matriz:

$$\underline{T}_k := Q_{(k+1)}^T A Q_k.$$

Essa matriz é similar a matriz $T_k = Q_k^T A Q_k$ entretanto não é quadrada pois não possui a última coluna de T_{k+1} .

3.1 Método Método resíduo Mínimo (MINRES)

O objetivo principal é encontrar $x^{(k)} \in K(A, b, k)$ que minimize $\|Ax^{(k)} - b\|_2$. Ou, de maneira equivalente, encontrar $x^{(k)} \in K(A, b, k)$ tal que $Ar^{(k)} \perp K(A, b, k)$.

Para visualizar que essas condições são equivalentes, seja o vetor $y^{(k)}$ que minimiza:

$$\|AQ_k y^{(k)} - b\|_2^2.$$

Usando argumento de quadrados mínimos $y^{(k)}$ satisfaz a equação normal:

$$Q_k^T A^T (AQ_k y^{(k)} - b) = 0.$$

Como A simétrica, isso é $Q_k^T Ar^{(k)} = 0$, é equivalente a dizer que $Ar^{(k)} \perp K(A, b, k)$.

Diferente do método CG que possuía um único algoritmo nesse método existem dois algoritmos relativamente parecidos com o mesmo objetivo. Ambos produzem as mesmas interações, porém com diferentes suposições sobre A . MINRES, é válido para uma matriz simétrica A e foca na minimização do resíduo, enquanto CR requer que A seja simétrica positiva definida e tem como foco a relação de conjugação.

- *MINRES* : Minimizando $\|r^{(k)}\|_2$

Ao escrever $x^{(k)} \in K(A, b, k)$ como $Q_k y^{(k)}$ o objetivo se torna a minimização de $\|AQ_k y^{(k)} - b\|_2^2$, gerando um problema de quadrados mínimos.

Seja $\beta_0 = \|b\|_2$ e $AQ = QT$ e como T é tri-diagonal, então:

$$AQ_k = Q_{k+1} Q_{k+1}^T AQ_k = Q_{k+1} \underline{T}_k.$$

Assim, a expressão que deve ser minimizada fica:

$$\|b - AQ_k y^{(k)}\|_2 = \|\beta_0 Q_{k+1} e_1 - Q_{k+1} \underline{T}_k y^{(k)}\|_2 = \|\beta_0 e_1 - \underline{T}_k y^{(k)}\|_2.$$

Devido a estrutura de \underline{T}_k , é possível obter uma decomposição usando rotação de Givens:

$$G_k^T \dots G_1^T \underline{T}_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}.$$

Em que G_1, \dots, G_k são rotação de Givens G_1, \dots, G_k e R_k é uma matriz triangular superior, reduzindo o problema de quadrados mínimos à encontrar $y^{(k)}$ que minimize:

$$\|\beta_0 G_k^T \dots G_1^T e_1 - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y^{(k)}\|_2.$$

Figura 5: Subtração dos termos



Resolvendo:



Fonte: DARVE, Eric; WOOTTERS, Mary. Numerical Linear Algebra with Julia. SIAM, 2021, p273

Isso é, se,

$$G_k^T \dots G_1^T \beta_0 e_1 = (v_k),$$

Então o minimizador é o $y^{(k)}$ que satisfaz,

$$R_k y^{(k)} = v_k.$$

Assim, basta encontrar $y^{(k)}$, e a partir disso obter $x^{(k)} = Q_k y^{(k)}$.

Neste ponto, é plausível, que a ideia acima possa ser transformada em um algoritmo eficiente usando a iteração Lanczos para construir Q_k e T_k . Ao longo do caminho são realizadas as etapas acima afim de encontrar a próxima iteração $x^{(k)}$. Tal como no método CG, é possível implementar essas etapas com eficiência. O espaço extra necessário é $O(n)$ e o número de flops é igual ao custo de calcular um produto matriz-vetor (Ax) e realizar algumas operações lineares de vetores. Há uma descrição mais detalhada sobre o algoritmo de MINRES nos textos de Paige e Saunders.^[3, 4]

A taxa de convergência para MINRES tem uma forma semelhante a encontrada em CG, mas considerando a norma do resíduo em vez da A-norma do erro $\|x - x^{(k)}\|_A$. Assumindo $x^{(0)} = 0$ para MINRES:

$$\|r^{(k)}\|_2 = \min_{\substack{p(0)=1 \\ p \text{ de grau } k}} \|p(A)b\|_2,$$

Se A for definida positiva simétrica, obtém-se um limite para MINRES que se assemelha ao de CG (que pode ser derivado usando polinômios de Chebyshev escalados e deslocados):

$$\|r^{(k)}\|_2 \leq 2\|r^{(0)}\|_2 \left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k.$$

Outros limites também podem ser derivados quando A é simétrica indefinida.[5]

• *Resíduo conjugado (CR)* : aplicando $Ar^{(k)} \perp K(A, b, k)$

CR produz as mesmas interações $x^{(k)}$, mas a derivação é diferente de MINRES. Ao contrário do MINRES, CR pode falhar quando A é indefinida (ou seja, não singular, mas nem positiva definida nem negativa definida) que funcione, mas às vezes pode acabar dividindo por zero e ter que parar. CR foi proposto por Saa[6] e busca encontrar $x^{(k)} \in K(A, b, k)$ de modo que:

$$Ar^{(k)} \perp K(A, b, k).$$

Assim como é feito para CG, é possível derivar uma relação de conjugação para as direções de busca $\Delta x^{(k)} = x^{(k+1)} - x^{(k)}$ De modo que $\Delta x^{(k)} = r^{(k+1)} - r^{(k)}$. Assim:

$$A\Delta x^{(k)} = -\Delta x^{(k)},$$

Logo, para qualquer $l < k$,

$$(\Delta x^{(l)})^T A^T A \Delta x^{(k)} = -(\Delta x^{(l)})^T A \Delta r^{(k)} = 0.$$

Porque $\Delta x^{(l)} \in K(A, b, l+1) \subseteq K(A, b, k)$, tendo que $\Delta r^{(k)} \perp K(A, b, k)$. Isso implica que as direções de busca $\Delta x^{(k)}$ são $A^T A$ -conjugado:

Conjugação das direções de busca no CR/MINRES:

Seja $x^{(k)}$ a k -ésima iteração de CR/MINRES tentando resolver $Ax = b$. Então as direções de busca $\Delta x^{(k)}$ são $A^T A$ -conjugado. Ou seja, ambas as direções satisfazem:

$$(\Delta x^{(k)})^T A^T A \Delta x^{(l)} = 0.$$

para qualquer $k \neq l$.

Agora reproduzindo a abordagem para CG:

$$\Delta x^{(k)} = \mu_{k+1} p^{(k+1)},$$

em que é assumido que $\Delta x^{(k)} \neq 0$ e portanto $\mu_{k+1} \neq 0$. Tendo A como uma matriz simétrica e não singular ainda é possível reconhecer a matriz triangular superior:

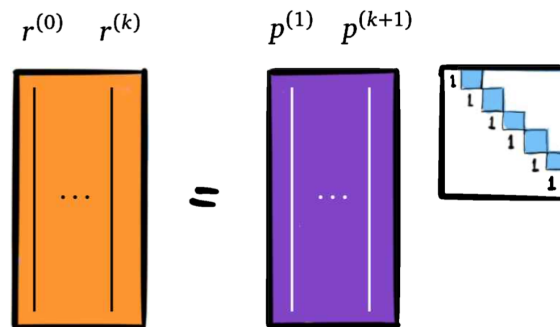


Figura 6: fonte: DARVE, Eric; WOOTTERS, Mary. Numerical Linear Algebra with Julia. SIAM, 2021, p.308

Para obter isso, podemos usar condições de ortogonalidade:

1 - As direções de busca são $A^T A$ -conjugado o que significa que:

$$(p^{(l)})^T A^T A p^{(k)} = 0.$$

para qualquer

$$l \neq k.$$

2 - Tem-se ainda, $(p^{(l)})^T A^T A p^{(k)} = 0$ para qualquer $l \leq k-1$. Para visualizar isso, usando a simetria de A :

$$(p^{(l)})^T A^T A r^{(k)} = (p^{(l)})^T A^T A^T r^{(k)}.$$

Observe que $A p^{(l)} \in K(A, b, l+1)$. Pela condição de ortogonalidade $A r^{(k)} \perp K(A, b, k)$ para $l \leq k-1$ e, portanto, a expressão acima é nula.

Foi provando a seguinte fórmula de recorrência para $p^{(k+1)}$ (Considerando $\mu \neq 0$):

$$p^{(k+1)} = r^{(k)} + \tau_k p^{(k)}.$$

Para que o restante do algoritmo funcione, é preciso que $p^{(k+1)}$ seja diferente de zero. Caso contrário, as divisões por zero ocorrerão no algoritmo posteriormente. Isso é garantido se:

$$r^{(k)} = b - A x^{(k)}.$$

pertencer a $K(A, b, k+1)$ mas não $K(A, b, k)$. Nesse caso, como $p^{(k)} \in K(A, b, k)$ tem-se que $p^{(k+1)} \neq 0$. No entanto, essa hipótese sobre $r^{(k)}$ falha sempre que $x^{(k)} = x^{(k-1)}$ (o que implica que $\mu_k = 0$). Isso não pode acontecer se A for positiva (ou negativa) definida, mas pode acontecer ocasionalmente quando A for indefinida. Quando isso acontece, o método CR não tem como encontrar uma nova direção de busca

$p^{(k+1)}$ e o algoritmo para. Em contraste, como o MINRES que é baseado no cálculo de uma base ortogonal para o subespaço Krylov, ele pode continuar as iterações até que o erro seja pequeno o suficiente (ou encontre uma solução exata), mesmo que $x^{(k)} = x^{(k-1)}$ ocorra em algum ponto durante as iterações.

Agora consideramos que A é positiva definida ou que $\Delta x^{(k)} \neq 0$. Em ambos os casos, é possível percorrer uma derivação muito semelhante à que foi feita para CG, usando $\Delta A^T A$ em vez de A para resolver o sistema linear. Chegando ao seguinte algoritmo:

Usando uma matriz simétrica positiva definida $n \times n$ e um vetor b :

1. Tome um $x^{(0)}$ por exemplo $x^{(0)} = 0$.
2. Tomando $r^{(0)} = b - Ax^{(0)}$, $p^{(1)} = r^{(0)}$ e $k = 1$.
3. Enquanto $r^{(k+1)} \neq 0$,

$$\mu_k = -\frac{(r^{(k-1)})^T A r^{(k-1)}}{(p^{(k)})^T A^T A p^{(k)}},$$

$$x^{(k)} = x^{(k-1)} + \mu_k p^{(k)},$$

$$r^{(k)} = r^{(k-1)} + \mu_k A p^{(k)},$$

$$\tau_k = \frac{(r^{(k)})^T A r^{(k)}}{(r^{(k-1)})^T A r^{(k-1)}},$$

$$p^{(k+1)} = r^{(k)} + \tau_k p^{(k)},$$

$$k \leftarrow k + 1.$$

4. Retorna $x^{(k-1)}$.

3.2 Método SYMMLQ

Symmlq é um algoritmo diferente que funciona para matrizes simétricas gerais (não necessariamente definidas positivas). A ideia com SYMMLQ é impor a condição de ortogonalidade:

$$r^{(k)} \perp K(A, b, k - 1).$$

Observe que isso é mais flexível do que a exigência de CG de que $r^{(k)} \perp K(A, b, k)$. Na verdade, é um pouco flexível demais: não existe um único $x^{(k)} \in K(A, b, k)$ que satisfaça a condição de ortogonalidade. Para ver isso, é preciso contar as dimensões: $K(A, b, k)$ possui dimensão k , enquanto o requisito de que $r^{(k)} \perp K(A, b, k - 1)$ leva apenas a $k-1$ restrições. Então resta um grau de liberdade. Para decidir qual $x^{(k)} \in K(A, b, k)$ escolher, a melhor opção é aquela com a menor norma 2. Essa escolha pode ser interpretada como uma espécie de regularização para tornar o problema matematicamente bem posto.

Como foi feito com o MINRES, em que foi usado a estrutura de \underline{T}_{k-1} para ajudar a encontrar o $x^{(k)}$. A condição de ortogonalidade pode ser reescrita como:

$$Q_{k-1}^T r^{(k)} = 0,$$

$$Q_{k-1}^T Q T Q^T x^{(k)} = Q_{k-1}^T b,$$

$$\underline{T}_{k-1}^T y^{(k)} = \beta_0 e_1,$$

Onde $x^{(k)} = Q_k y^{(k)}$. Minimizar $\|x^{(k)}\|_2$ é o mesmo que minimizar $\|y^{(k)}\|_2$ então o objetivo passa a ser encontrar $y^{(k)} \in R(\text{reais})$ de modo que $T_{k-1}^T y^{(k)} = \beta_0 e_1$, e $\|y^{(k)}\|_2$ a menor possível. Para resolver o problema é possível usar a decomposição QR $T_{k-1} = UR$ e considerar:

$$R^T U^T y^{(k)} = \beta_0 e_1,$$

Denote:

$$z^{(k)} = U^T y^{(k)}.$$

Como $\|z^{(k)}\|_2 = \|y^{(k)}\|_2$ é possível minimizar $\|z^{(k)}\|_2$ com a seguinte restrição:

$$R^T z^{(k)} = \beta_0 e_1.$$

Entretando, R^T tem uma forma simples,

$$R^T = (L \ 0),$$

em que L é não singular. Assim as primeiras k -componentes de $z^{(k)}$ são unicamente definidas por essa equação. Como o objetivo é minimizar $\|z^{(k)}\|_2$ a única opção para a última componente é $[z^{(k)}]_k = 0$. Em termos de $y^{(k)}$:

$$U^{(T)} y^{(k)} = \begin{pmatrix} \beta_0 L^{-1} e_1 \\ 0 \end{pmatrix} \implies y^{(k)} = U \begin{pmatrix} \beta_0 L^{-1} e_1 \\ 0 \end{pmatrix}.$$

Isso fornece uma fórmula para calcular $y^{(k)}$ em que a cada etapa $x^{(k)} = Q_k y^{(k)}$ pode ser obtido. A última parte do desenvolvimento de SYMMLQ é encontrar recorrências para que não seja necessário calcular explicitamente Q_k e $y^{(k)}$ a cada passo. O algoritmo final é muito próximo do MINRES, mas usa diferentes direções de busca e tamanhos dos passos. O custo computacional em cada etapa é a quantidade de tempo necessária para multiplicar A por um vetor, mais um custo $O(n)$. O armazenamento extra também é $O(n)$. [7]

4 Método Resíduos Mínimos Gernalizado (GMRES)

O método resíduo mínimo generalizado, ou GMRES se assemelha ao MINRES. Ele procura $x^{(k)} \in K(A, b, k)$ que minimiza $\|Ax^{(k)} - b\|_2$ isso é equivalente à condição $A^T r^{(k)} \perp K(A, b, k)$. Essa condição pode ser escrita em termos da matriz $Q_{k+1}^T A Q_k$. Anteriormente, essa matriz era denotada por T_k porque era tri-diagonal. No entanto, esta matriz é agora Hessenberg superior porque A não é mais considerada simétrica. Assim, será usada a notação:

$$H_k = Q_{k+1}^T A Q_k.$$

Critério de otimização GMRES:

Para matriz quadrada não singular A , GMRES encontra $x^{(k)} = Q_K y^{(K)}$ de modo que a norma-2 do resíduo :

$$\|r^{(k)}\|_2 = \|\beta_0 e_1 - H_k y^{(k)}\|_2.$$

é minimizado.

Como mencionado, MINRES trabalha com uma matriz tri-diagonal, mas, sem simetria, H_k é agora Hessenberg superior. Assim, no GMRES a iteração usada é a de Arnoldi em vez da iteração de Lanczos. Por causa disso, o GMRES é geralmente mais lento que o MINRES.

O objetivo é encontrar um y para minimizar $\|\beta_0 e_1 - H_k y^{(k)}\|_2$ onde H_k é Hessenberg superior. Portanto, é usado uma sequência de rotações de Givens G_k para tornar H_k triangular superior. Obtendo:

$$G_k^T \dots G_1^T H_k = \begin{pmatrix} R_k \\ 0 \end{pmatrix}.$$

O novo problema de minimização torna-se:

$$\|G_k^T \dots G_1^T (\beta_0 e_1) - \begin{pmatrix} R_k \\ 0 \end{pmatrix} y^{(k)}\|_2.$$

Que pode ser resolvido escrevendo:

$$\beta_0 G_k^T \dots G_1^T e_1 = \begin{pmatrix} p_k \\ \rho_k \end{pmatrix}.$$

Em que p_k é um vetor com as primeiras k -entradas e ρ_k a última entrada. Então a solução $y^{(k)}$ satisfaz:

$$R_k y^{(k)} = p_k,$$

E a norma do resíduo é:

$$\|\beta_0 e_1 - H_k y^{(k)}\|_2 = \rho_k.$$

O algoritmo segue essas ideias. Ele monta a matriz H_k usando a iteração de Arnoldi, calcula uma sequência de rotações de Givens e calcula a norma do resíduo usando ρ_k . A solução $x^{(k)}$ é obtida no final, uma vez que o critério de convergência é satisfeito.

GMRES funciona para matrizes A não singulares arbitrárias, A cada iteração, é preciso atualizar a fatoração QR de uma matriz de Hessenberg superior, que leva $O(kn)$ flops. Também há uma multiplicação de um vetor por A , mas a esperança para esses métodos é que, se A for esparsa ou estruturada de outra forma, isso pode ser feito rapidamente, no tempo $O(n)$.

Entretanto conforme k cresce o custo por iteração também cresce, por essa razão o processo GMRES é frequentemente reiniciado. Portanto, é preciso redefinir o subespaço de Krylov para o resíduo na iteração atual e redefinir $k = 1$, se o número de iterações se tornar muito grande.[8,9]

4.1 Convergência do método MINRES

A análise da convergência para GMRES é um pouco mais complicada do que a análise para CG. Como a matriz é assimétrica, é de se esperar autovalores reais e complexos. Como resultado, algumas das análises realizadas para CG não se aplicam mais. Ainda é preciso considerar o argumento baseado em polinômios como foi feito anteriormente.

Desta vez, GMRES minimiza :

$$\|r^{(k)}\|_2 = \|b - AQ_k y\|_2 = \|(I - Aq(A))b\|_2.$$

Em que q é um polinômio de grau k-1. Como CG, podemos definir um polinômio p(x) de grau k tal que p(0)=1 e que minimize $\|p(A)b\|_2$. O resultado final que pode ser obtido é[10]:

Suponha que A tenha uma decomposição de autovalor $A = X\Lambda X^{-1}$. Então o erro GMRES pode ser limitado por:

$$\|r^{(k)}\|_2 \leq \|b\|_2 \kappa(X).$$

Com $\min (p \text{ de grau } k \text{ } p(0) = 1)$ e $\max (\lambda \text{ autovalor de } A) |p(\lambda)|$.

Em que $\kappa(X) = \|X\|_2 \|X^{-1}\|_2$ é o número de condição de X.

Para comparação, pode ocorrer a derivação dos limites semelhantes para CG e MINRES. Se A é SPD, então para CG:

$$\|x - x^{(k)}\|_A \leq \|x\|_A \xrightarrow[\substack{\text{p de grau } k \\ p(0)=1}]{\min} \xrightarrow[\substack{\lambda \text{ autovalor de } A}]{\max} |p(\lambda)|.$$

Se A é simétrico, então para MINRES:

$$\|r^{(k)}\|_2 \leq \|b\|_A \xrightarrow[\substack{\text{p de grau } k \\ p(0)=1}]{\min} \xrightarrow[\substack{\lambda \text{ autovalor de } A}]{\max} |p(\lambda)|.$$

Em muitos casos, espera-se uma convergência rápida se os autovalores de A estiverem agrupados dentro de alguns discos bem separados da origem. Intuitivamente, isso ocorre porque o polinômio p que minimiza o lado direito da equação $\|x - x^{(k)}\|_A \leq \|x\|_A \min \max |p(\lambda)|$ tem raízes próximas aos autovalores de A. Portanto, se os autovalores de A estão agrupados em discos, então uma única raiz de p pode "pegar" vários autovalores de A, tornando o lado direito da equação:

$$\|r^{(k)}\|_2 \leq \|b\|_2 \kappa(X).$$

menor.

Quando A é normal, $\kappa(X) = 1$, conforme A se desvia da normalidade, $\kappa(X)$ pode crescer. A Equação da convergência fornece apenas um limite superior para a norma do resíduo. Se $\kappa(X)$ permanecer próximo de 1 ou pequeno, espera-se um pequeno resíduo (desde que o termo com min e max decaia rapidamente à medida que k aumenta). Quando $\kappa(X)$ é grande, o limite é menos útil. A convergência pode permanecer rápida [13].

Um exemplo de um caso complexo é quando todos os autovalores de A estão no disco unitário (o disco de raio 1 centrado na origem) de maneira uniforme [11,12]. Por exemplo, suponha que A seja uma matriz de permutação, representando uma permutação $\pi : (1, \dots, n) \rightarrow (1, \dots, n)$, e tentando resolver $Ax = e_1$. A solução é o vetor de base padrão e_r , de modo que $\pi(r) = 1$. Enquanto isso, $K(A, e_1, k)$ é um subespaço de coordenadas gerado por: $(e_1, e_{\pi(1)}, e_{\pi(2)} \dots e_{\pi^{k-1}(1)})$. Portanto, a convergência ocorre apenas para k de

modo que $e_r \in K(A, e, k)$, o que só acontece quando $\pi^{(k)}(1) = 1$. Isso não pode ocorrer até que $k = n$ se, por exemplo, π é a permutação $\pi(i) = i + 1 \pmod n$.

e54

5 Resultados e conclusão

Para testar os métodos usamos 3 problemas diferentes que são matrizes simétricas positivas definidas com diferentes dimensões, para que fosse possível observar uma na casa das centenas, outra com dimensão mil por mil e uma última na casa dos milhares. Essa escolha tem como princípio observar como os métodos se comportam quando expostos a um padrão de matriz para uma observação sobre qual melhor método nas condições impostas.

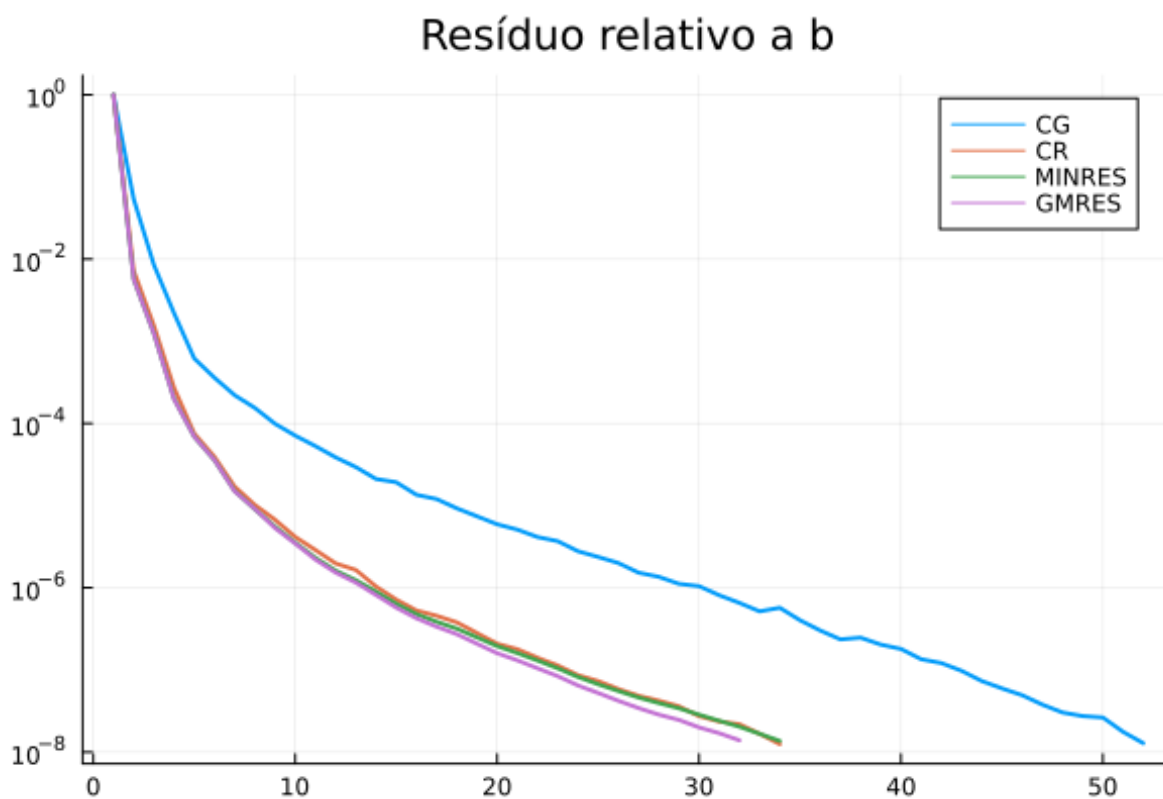


Figura 7: Problema 1: matriz simétrica positiva definida na ordem de 100
fonte: Suite Sparse <https://sparse.tamu.edu/>, nome:Pothen/mesh2em5

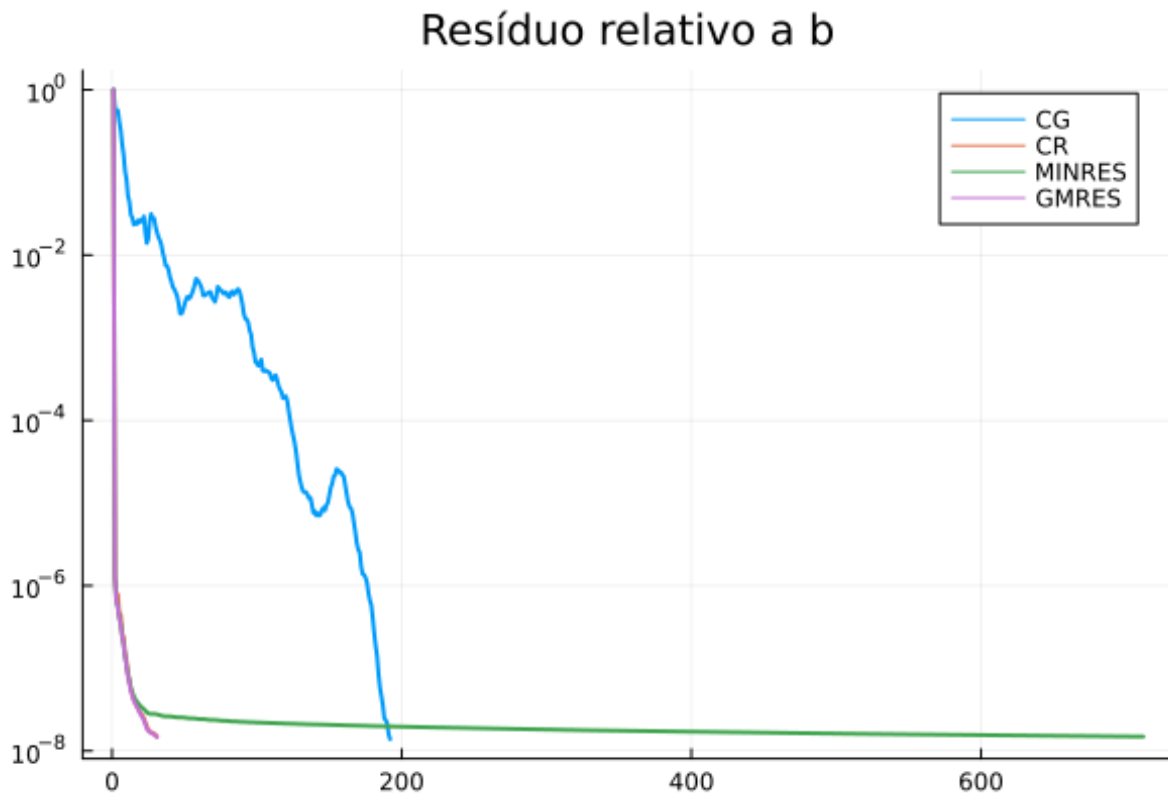


Figura 8: Problema 2: matriz simétrica positiva definida na ordem de 1000
 fonte: Suite Sparce <https://sparse.tamu.edu/>, nome: Oberwolfach/LFAT5000

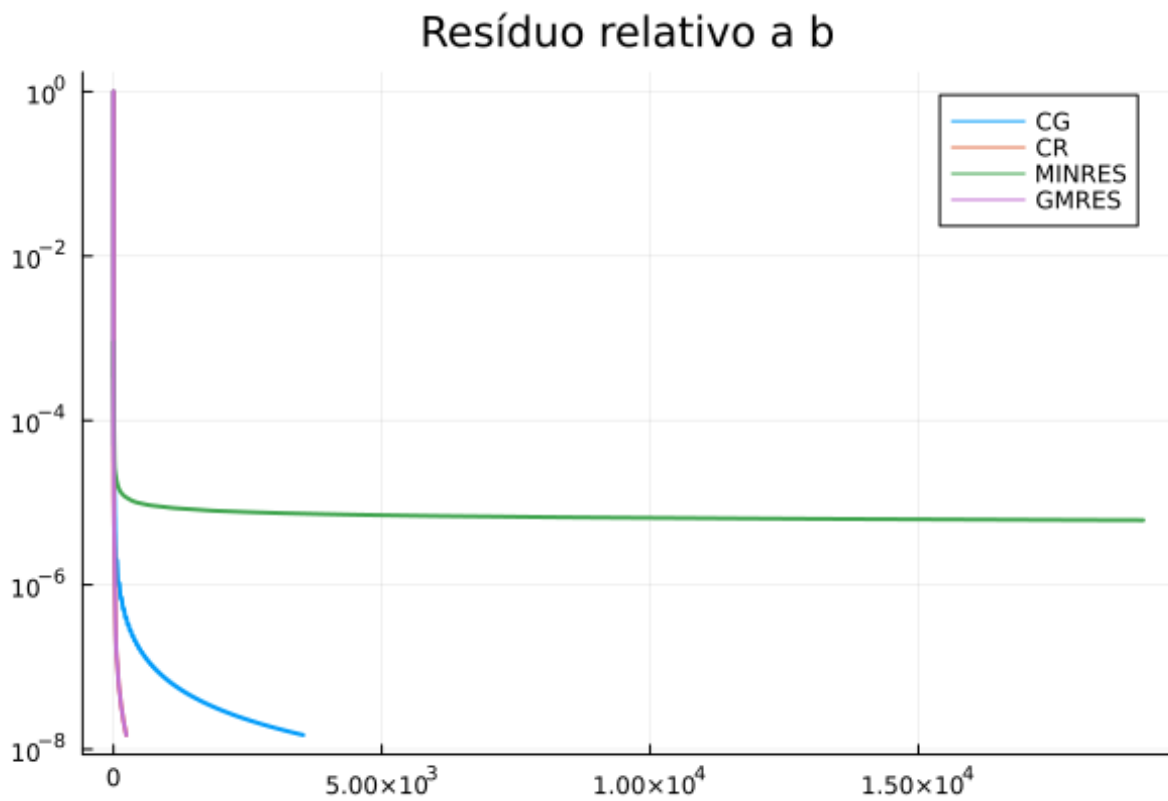


Figura 9: Problema 1: matriz simétrica positiva definida na ordem de 10000
 fonte: Suite Sparce <https://sparse.tamu.edu/>, nome: TKK/plbuckle.

Esperava-se que o método CG em relação aos outros métodos, tivesse um melhor desempenho pois as matrizes possuem características que favorecem o método. Entretanto o resíduo do método é calculado na norma de A, que vem da matriz, já os outros métodos buscam na norma 2.[14]

Ao observarmos o comportamento dos métodos nos diferentes problemas é possível observar alguns pontos. Nota-se que apesar do método GMRES fazer menos operações nesses testes é preciso lembrar que suas operações possuem um custo maior a medida que ocorre o aumento da complexidade das matrizes. Assim, apenas o número de operações pode ser enganoso.

Sobre o método MINRES depois de um número determinada de iterações, os dados acumulados são eliminados e os resultados intermediários são usados como os dados iniciais para as próximas iterações.[16] Este procedimento é repetido até que a convergência seja alcançada, entretanto na matriz de ordem 10000 o método quebrou e não conseguiu atingir a convergência esperada.

Conclui-se que os métodos que se utilizam dos subespaços de Krylov possuem diferentes abordagens com o mesmo propósito, convergir para uma solução mais próxima possível, seja através de abordagem que buscam minimizar o resíduo ou através da construção de uma base ortogonal para o subespaço. Apesar do estudo dos métodos trazer uma boa base teórica sobre a convergência de cada abordagem e os critérios usados, ao desenvolver o algoritmo necessário e utilizar para a resolução de problemas enfrentamos alguns desafios relacionados a funcionalidade do algoritmo perante matrizes mais complexas.

6 Bibliografia

- [1] - Concus, P., Golub, G. H. O'Leary, D. P. *A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations. In Sparse Matrix Computations*, 309-332 (Academic Press, 1976) (cit. on p. 288).
- [2] - Van der Sluis, A. van der Vorst, H. A. *The rate of convergence of conjugate gradients. Numerische Mathematik* 48, 543-560 (1986).
- [3] - Paige, C. Saunders, M. *Solution of Sparse Indefinite Systems of Equations and Least Squares Problems*. Tech. rep. (Department of Computer Science, Stanford University, 1973).
- [4] - Paige, C. C. Saunders, M. A. *Solution of sparse indefinite systems of linear equations*. SIAM Journal on Numerical Analysis 12, 617-629 (1975).
- [5] - Greenbaum, A. *Iterative Methods for Solving Linear Systems* (SIAM, 1997).
- [6] - Embree, M. *How Descriptive Are GMRES Convergence Bounds?* Tech. rep. (Numerical Analysis Group, The Mathematical Institute, University of Oxford, 1999).
- [7] - Choi, S.-C. *Iterative Methods for Singular Linear Equations and Least-Squares Problems*, PhD thesis (ICME, Stanford University, 2006).
- [8] - Saad, Y. Schultz, M. H. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. SIAM Journal on Scientific and Statistical Computing 7, 856-869 (1986)

- [9] - Axelsson, O. Barker, V. A. *Finite Element Solution of Boundary Value Problems: Theory and Computation*.
- [10] - Briggs, W. L., Henson, V. E. McCormick, S. F. *A Multigrid Tutorial*, 2nd ed. (SIAM, 2000).
- [11] - Liesen, J. Strakoš, Z. *Krylov Subspace Methods: Principles and Analysis* (Oxford University Press, 2013).
- [12] - Tebbens, J. D. Meurant, G. Any Ritz value behavior is possible for Arnoldi and for GMRES. *SIAM Journal on Matrix Analysis and Applications* 33, 958-978 (2012).
- [13] - Xia, J., Chandrasekaran, S., Gu, M. Li, X. S. *Fast algorithms for hierarchically semiseparable matrices. Numerical Linear Algebra with Applications* 17, 953-976 (2010).
- [14] - DARVE, Eric; WOOTTERS, Mary. *Numerical Linear Algebra with Julia*. SIAM, 2021.
- [15] - GHIDETTI, Kamila; BOERES, Maria Claudia; CATABRIGA, Lucia. UM ESTUDO COMPARATIVO DE MÉTODOS HEURÍSTICOS PARA REORDENAMENTO DE MATRIZES ESPARSAS.