

ANA PAULA PIANTONI GONÇALVES

PROCESSO DE BIORTOGONALIZAÇÃO DE LANCZOS
COM ESTRATÉGIA DE LOOK-AHEAD

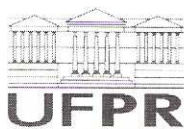
ANA PAULA PIANTONI GONÇALVES

PROCESSO DE BIORTOGONALIZAÇÃO DE LANCZOS
COM ESTRATÉGIA DE LOOK-AHEAD

Dissertação apresentada ao Programa de Pós-Graduação em Matemática Aplicada da Universidade Federal do Paraná, como requisito parcial à obtenção do grau de Mestre em Matemática Aplicada.

Orientador: Prof. Dr. Saulo Pomponet Oliveira.

Curitiba
Setembro de 2013



Ministério da Educação
Universidade Federal do Paraná
Setor de Ciências Exatas/Departamento de Matemática
Programa de Pós-Graduação em Matemática Aplicada - PPGMA

ATA DA 53ª DEFESA DE DISSERTAÇÃO DE MESTRADO

Aos treze dias do mês de setembro de 2013, no Anfiteatro A, Prédio das PCs, da Universidade Federal do Paraná, foi instalada pelo Professor Saulo Pomponet Oliveira, a Banca Examinadora para a quinquagésima terceira Defesa de Dissertação de Mestrado em Matemática Aplicada. Estiveram presentes ao Ato, professores, alunos e visitantes.

A banca examinadora, homologada pelo Colegiado do Programa de Pós-Graduação em Matemática Aplicada, ficou constituída pelos professores: Prof. Dr. Fermín Sinfioriano Viloche Bazán, da Universidade Federal de Santa Catarina, Prof. Dr. Yuan Jin Yun, da Universidade Federal do Paraná, e o Prof. Dr. Saulo Pomponet Oliveira, orientador da dissertação, a quem coube a presidência dos trabalhos.

Às dez horas, a banca iniciou seus trabalhos, convidando a candidata **ANA PAULA PIANTONI GONÇALVES** a fazer a apresentação do tema da dissertação intitulada "PROCESSO DE BIORTOGONALIZAÇÃO DE LANCZOS COM ESTRATÉGIA DE LOOK-AHEAD". Encerrada a apresentação, iniciou-se a fase de argüição pelos membros participantes. Após a argüição, a banca com pelo menos 03 (três) membros, reuniu-se para apreciação do desempenho do pós-graduando.

A banca considerou que o pós-graduando fez uma apresentação com a necessária concisão. A Dissertação apresenta contribuição à área de estudos e não foram registrados problemas fundamentais de estrutura e redação, resultando em plena e satisfatória compreensão dos objetivos pretendidos.

Tendo em vista a dissertação e a argüição, os membros presentes da banca decidiram pela sua aprovação.

Curitiba, 13 de setembro de 2013.

Prof. Dr. Saulo Pomponet Oliveira
Presidente

Prof. Dr. Fermín Sinfioriano Viloche Bazán
Titular

Prof. Dr. Yuan Jin Yun
Titular



Ministério da Educação
Universidade Federal do Paraná
Setor de Ciências Exatas/Departamento de Matemática
Programa de Pós-Graduação em Matemática Aplicada - PPGMA

PARECER DA BANCA EXAMINADORA

Após a apresentação, a banca deliberou pela aprovação da dissertação da candidata **ANA PAULA PIANTONI GONÇALVES** devendo, para tanto, incorporar as sugestões feitas pelos membros da banca, no prazo estabelecido pelo regimento correspondente.

Curitiba, 13 de setembro de 2013.

Prof. Dr. Saulo Pomponet Oliveira
Presidente

Prof. Dr. Fermin Sinforiano Vileche Bazán
Titular

Prof. Dr. Yuan Jin Yun
Titular

Agradecimentos

Meu especial agradecimento ao meu orientador Prof. Saulo Pomponet Oliveira por todos os ensinamentos ao longo desta trajetória. Obrigada pelo apoio, dedicação, discussões e trocas de ideias que culminaram neste trabalho.

Aos professores Yuan Jin Yuan e Fermín Sinforiano Viloche, membros da banca examinadora, por aceitarem ao convite e pelas sugestões enriquecedoras apontadas.

Aos professores e alunos do Programa de Pós Graduação em Matemática Aplicada pelos conhecimentos compartilhados e pelas trocas de experiências, sejam no âmbito profissional ou pessoal.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior pelo apoio financeiro.

Ao Moisés dos Santos pelo incentivo, compreensão e companheirismo durante toda esta etapa.

Por último, agradeço àqueles que, dia a dia, enchem a minha vida de alegria: meus pais e irmãos. Sem vocês, a vida não teria graça e as vitórias conquistadas não teriam o mesmo sabor. Muito obrigada pelo imenso companheirismo, carinho e apoio incondicional. Amo vocês!

Resumo

Esta dissertação trata de métodos de projeção baseados em subespaços de Krylov, tendo como principal enfoque o método de Biortogonalização de Lanczos para matrizes não simétricas. Este método gera duas bases biortogonais para subespaços de Krylov associados a uma matriz e sua transposta. Por ser um método suscetível a possíveis quebras e instabilidades numéricas, apresenta-se uma versão do algoritmo com a estratégia *look ahead*, que supera esses problemas pulando os passos em que ocorrem quebras ou que estão na iminência de quebras. Para tanto, esta estratégia constrói dois tipos de vetores (regulares e internos), que são agrupados de forma a serem biortogonais por blocos. Um estudo da relação entre tais vetores com polinômios ortogonais formais (FOPs) também é apresentado, e exemplos numéricos comparam o algoritmos de Biortogonalização de Lanczos com e sem a estratégia *look ahead*.

Palavras chave: *Álgebra linear numérica, Biortogonalização de Lanczos, Look ahead, Polinômios ortogonais formais, Métodos de projeção.*

Abstract

This work concerns projection methods based on Krylov subspaces, with emphasis on the Lanczos biorthogonalization method for unsymmetric matrices. This method generates two biorthogonal bases for Krylov subspaces associated with a matrix and its transpose. However, this method is susceptible to possible breakdowns and numerical instabilities. We present a version of the algorithm with the *look ahead* strategy, which overcomes these problems skipping the steps where breakdown or near-breakdown occurs. This strategy constructs two types of vectors (regular and inner vectors) which are grouped in blocks that are set to be biorthogonal. A study of the relationship between such vectors and formal orthogonal polynomials (FOPs) is also presented, and numerical examples compare the Lanczos Biorthogonalization algorithm with and without *look ahead*.

Keywords: *Numerical linear algebra, Lanczos Biorthogonalization, Look ahead, Formal orthogonal polynomials, Projection Methods*

Lista de Algoritmos

1	Método de Máxima Descida	13
2	Iteração MR	13
3	Método de Arnoldi	16
4	Método de Ortogonalização Completa (FOM)	20
5	Método GMRES	21
6	Algoritmo de Lanczos Simétrico	23
7	Algoritmo de Biortogonalização de Lanczos	26
8	Algoritmo de Lanczos Normalizado	55
9	Algoritmo de Lanczos <i>Look Ahead</i>	62

Sumário

Introdução	1
1 Métodos de Projeção	4
1.1 Projetores	4
1.2 Projetores Oblíquos e Ortogonais	7
1.2.1 Propriedades de Projetores Ortogonais	8
1.3 Métodos de Projeção	9
1.3.1 Representação Matricial dos Métodos de Projeção	10
1.4 Métodos de Projeção Unidimensionais	12
2 Método de Arnoldi	14
2.1 Subespaços de Krylov	14
2.2 Método de Arnoldi	15
2.2.1 Métodos de Projeção Baseados no Método de Arnoldi	19
2.3 Algoritmo de Lanczos Simétrico	22
3 Processo de Biortogonalização de Lanczos	24
3.1 Descrição do Algoritmo	24
3.1.1 Métodos de Projeção Baseados na Biortogonalização de Lanczos	32
3.2 Quebras no Algoritmo	33
4 Algoritmo de Lanczos com Estratégia <i>Look Ahead</i>	38
4.1 FOPs Regulares e Singulares	43
4.2 Lanczos <i>Look Ahead</i>	49
4.2.1 Algoritmo de Lanczos Normalizado	53
4.2.2 Algoritmo Sequencial	57
Conclusão	70
Referências Bibliográficas	70

Introdução

A solução de problemas em ciências e engenharias por meio de métodos computacionais de otimização e métodos numéricos para equações diferenciais parciais geralmente requer a resolução de sistemas lineares de grande porte. Por exemplo, a simulação numérica de modelos eletromagnéticos tridimensionais em geofísica pode envolver centenas de milhões de variáveis [25]. Nestes casos, torna-se inviável o uso de métodos diretos em sua resolução, e precisamos recorrer a métodos iterativos.

Para a solução iterativa de sistemas lineares simétricos em que a matriz é positiva definida, tornou-se consensual [29] a utilização do método de gradientes conjugados, propostos por Hestenes and Stiefel [23]. Embora a sequência gerada pelo método de gradientes conjugados convirja para a solução exata em n passos, sendo n a ordem da matriz [27], desejamos que, em problemas de grande porte, a convergência ocorra muito antes do n -ésimo passo. Por este motivo, usualmente o método de gradientes conjugados é utilizado em conjunto com um condicionador, por exemplo o condicionador gerado pela decomposição incompleta de Choleski [19, 29].

Para sistemas lineares não simétricos, não é clara a existência de um método consensual [15, Sec. 5.7], embora o método GMRES, proposto por Saad e Schultz [28], seja um dos métodos mais populares. Este método tem a desvantagem do seu custo computacional crescer (sobretudo em termos de memória RAM) com o número de iterações [29].

Além do GMRES, foram propostos diversos métodos baseados no processo de biortogonalização de Lanczos [21], tais como o método de gradientes biconjugados (BGG, [9, 22]), o método de resíduo quase-minimal (QMR, [13]), e o método BGC estabilizado (Bi-CGSTAB, [31]). Dada uma matriz A e dois vetores não-ortogonais v_1 e w_1 , o processo de biortogonalização de Lanczos gera duas sequências biortogonais que formam bases para subespaços de Krylov $K_m(A, v_1)$ e $K_m(A^T, w_1)$. Entretanto, o cálculo destas sequências pode ser abortado devido a divisões por zero, causadas por um problema conhecido como *quebra* no algoritmo.

Há muitos estudos que seguem na linha de evitar quebras no algoritmo de biortogonalização de Lanczos. Em [5] Brezinski, Sadok e Zaglia propõem duas variantes do método de zoom recursivo (MRZ, [6]), que pulam quebras que ocorrem nos algoritmos tipo Lanczos. Em seguida, eles estendem tais métodos para os casos de “quase-quebras” (near-breakdowns). Ye e Tong [32, 30] propuseram substituir os vetores que geram as quebras

por vetores que a evitam (*new-start vectors*).

Vários estudos envolvendo métodos para resolver sistemas lineares estão ligados a teoria de polinômios ortogonais formais (FOPs) [8]. Brezinski e Sadok [7] sintetizam os algoritmos *tipo Lanczos* para resolver sistemas lineares, mostrando que esses algoritmos consistem em utilizar diferentes relações de recorrência para calcular os FOPs envolvidos no método de Lanczos. Baheux [2] apresenta todas as combinações possíveis de obter algoritmos de Lanczos que são caracterizados pela escolha de uma ou duas relações de recorrência. Brezinski e Zaglia [3] também estudam casos de quebras no caso de cálculos recursivos de membros de uma família de FOPs, analisando e classificando tais quebras, além de mostrarem como evitar algumas quebras particulares. Já em [4], eles mostram que a teoria de FOPs pode ser usada para curar quebras e quase-quebras na implementação da transformação-G.

Parlett, Taylor e Liu [24] propuseram uma estratégia de biortogonalização em blocos que denominaram *look ahead*. Em [16], Gutknecht relaciona a teoria de FOPs com aproximação de Padé e aplica tais resultados para generalizar a estratégia *look ahead*. Gutknecht, Freund e Nachtigal [17] desenvolveram a implementação da estratégia apresentada em [16].

Freund [11] apresenta uma versão do algoritmo de Lanczos com estratégia *look ahead* baseada na conexão com FOPs associados a formas bilineares Hankel, vindo ao encontro da abordagem que trazemos neste trabalho. Ele também descreve algoritmos relacionados a construção de FOPs. Baseado no algoritmo de Lanczos não simétrico com estratégia *look ahead*, Freund, Roland e Nachtigal [13] propõem uma implementação do método QMR que supera os problemas de quebras e instabilidades numéricas apresentadas pelo método BCG.

Em alguns casos, utiliza-se estratégias *look ahead* para calcular elementos de uma família de polinômios ortogonais. Ayachour [1] mostra como evitar utilizar estratégias *look ahead* usando um novo método chamado ALA (*Avoiding Look Ahead*) e discute a aplicação do ALA ao método de Lanczos para resolver sistemas lineares.

Neste trabalho, veremos a dedução do método de Biortogonalização de Lanczos do ponto de vista dos métodos de projeção [27]. Para superar os problemas de quebras e instabilidades numéricas, consideramos a versão do algoritmo de Lanczos com estratégia *look ahead* apresentada em [17]. A dissertação está organizada da seguinte maneira:

No Capítulo 1, estudamos métodos de projeções gerais e algumas de suas propriedades. Vimos que tais métodos extraem soluções aproximadas para o sistema linear $Ax = b$ em um determinado subespaço do \mathbb{R}^n e expressamos a solução aproximada do sistema linear na forma matricial.

No Capítulo 2, apresentamos o Método de Arnoldi, que corresponde ao processo de ortogonalização de Gram-Schmidt para os vetores que geram um subespaço de Krylov. Ilustramos dois métodos de projeção baseados neste processo, o Método de Ortogonalização Completa (FOM) e o Método de Resíduo Mínimo Generalizado (GMRES). Em

seguida, particularizamos esse método para o caso da matriz A ser simétrica, obtendo assim o Método de Lanczos Simétrico.

Iniciamos o Capítulo 3 estendendo para matrizes não simétricas o Método de Lanczos Simétrico, resultando no método de Biortogonalização de Lanczos. Assim como no Método de Arnoldi, podemos obter métodos de projeção a partir da Biortogonalização de Lanczos, em particular os métodos BCG e QMR. Em seguida, discutimos dois tipos de quebra no algoritmo de biortogonalização, a quebra favorável e a quebra séria. No caso de quebra favorável, podemos garantir que a solução aproximada pelos métodos de projeção associados ao processo de Biortogonalização de Lanczos coincidem com a solução exata do sistema linear $Ax = b$. Isso já não ocorre no caso de quebra séria. Neste caso, introduzimos a estratégia *look ahead*, cuja idéia é “pular” a quebra e continuar a calcular vetores Lanczos no algoritmo.

Para entendermos a idéia do algoritmo com a estratégia *look ahead*, veremos no Capítulo 4 um estudo sobre a relação entre o método de Lanczos e a teoria de polinômios ortogonais formais (FOPs). Vimos que, em termos de FOPs, a estratégia *look ahead* consiste em pular polinômios ortogonais regulares não existentes, associando vetores internos ao algoritmo nesses casos e vetores regulares nos casos da existência. Vimos ainda que para verificação da existência de um FOP regular de grau n , basta analisarmos a singularidade de uma determinada matriz de Hankel de dimensão n . Em seguida, apresentamos o algoritmo de Lanczos com *look ahead*, construindo vetores internos e regulares de forma a serem agrupados em blocos biortogonais. Por fim, apresentamos exemplos numéricos que nos permitiram comparar o algoritmo de Lanczos não simétrico com o algoritmo Lanczos *look ahead*, bem como validar os resultados obtidos pela teoria de polinômios ortogonais. Tais exemplos foram implementados no pacote computacional *Octave for Windows*.

Capítulo 1

Métodos de Projeção

Neste capítulo, abordaremos inicialmente o conceito de um projetor, bem como algumas de suas propriedades. Em seguida, veremos algumas técnicas de projeção, que são utilizadas para extrairmos soluções aproximadas de um sistema linear em um dado subespaço.

1.1 Projetores

Definição 1.1 *Um projetor $P : \mathbb{R}^n \rightarrow \mathbb{R}^n$ é uma aplicação linear que é idempotente, ou seja, $P^2 = P$.*

Proposição 1.2 *Se P é um projetor, então:*

- (i) $(I - P)$ é um projetor;
- (ii) $\text{Ker}(P) = \text{Im}(I - P)$;
- (iii) $\text{Ker}(P) \cap \text{Im}(P) = \{0\}$.

Demonstração.

(i) A linearidade de $I - P$ segue da linearidade de I e P . Além disso, de $P^2 = P$, temos

$$(I - P)^2 = (I - P)(I - P) = I - P - P + P^2 = I - P - P + P = I - P.$$

(ii) (\subseteq) se $x \in \text{Ker}(P)$, então $Px = 0$, de modo que

$$x = x - Px = (I - P)x \Rightarrow x \in \text{Im}(I - P).$$

(\supseteq) se $y \in \text{Im}(I - P)$, então existe $x \in \mathbb{R}^n$ tal que $(I - P)x = y$. Aplicando P a ambos os lados desta igualdade, obtemos

$$\begin{aligned} P(x - Px) &= Py \\ Px - P^2x &= Py, \end{aligned}$$

o que resulta em $0 = Py$, pois $P^2x = Px$. Portanto, $y \in Ker(P)$.

(iii) Por hipótese, P é um projetor. Então, da linearidade de P , segue que $P(0) = 0$, donde $0 \in Ker(P) \cap Im(P)$. Agora, seja $x \in Ker(P) \cap Im(P)$. Então, $Px = 0$ e existe $y \in \mathbb{R}^n$ tal que $Py = x$. Assim, $P(Py) = Px = 0 \Rightarrow x = Py = P^2y = 0$. \square

Note que, se P é um projetor, todo elemento $x \in \mathbb{R}^n$ pode ser escrito como

$$x = \underbrace{Px}_{\in Im(P)} + \underbrace{x - Px}_{\in Ker(P)}.$$

Como $Ker(P) \cap Im(P) = \{0\}$, temos que $\mathbb{R}^n = Im(P) \oplus Ker(P)$.

Assim, se P é um projetor, temos que $\mathbb{R}^n = Im(P) \oplus Ker(P)$. Veremos agora que, se M e S são subespaços quaisquer tais que $\mathbb{R}^n = M \oplus S$, então existe um único projetor P tal que $Im(P) = M$ e $Ker(P) = S$. Dizemos que a aplicação linear P projeta x em M e paralelamente ao subespaço S .

Proposição 1.3 *Sejam M e S subespaços quaisquer tais que $\mathbb{R}^n = M \oplus S$. Então, existe um único projetor P tal que $Im(P) = M$ e $Ker(P) = S$.*

Demonstração. Dado $x \in \mathbb{R}^n$, temos que x é escrito de forma única como $x = x_m + x_s$, com $x_m \in M$ e $x_s \in S$. Considere $P : \mathbb{R}^n \rightarrow \mathbb{R}^n$ dado por $Px = x_m$, ou seja, P associa cada elemento $x \in \mathbb{R}^n$ ao componente pertencente a M da decomposição associada à soma direta. Mostremos que P assim definido é projetor. De fato, dados $x, y \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}$, temos que $x = x_m + x_s$ e $y = y_m + y_s$, com $x_m, y_m \in M$ e $x_s, y_s \in S$. Assim,

$$\begin{aligned} P(\alpha x + y) &= P[\alpha(x_m + x_s) + (y_m + y_s)] \\ &= P[(\alpha x_m + y_m) + (\alpha x_s + y_s)] \\ &= (\alpha x_m + y_m) = \alpha Px + Py. \end{aligned}$$

Logo, P é linear. Além disso, como $x_m \in M$, temos $x_m = x_m + 0$, com $x_m \in M$ e $0 \in S$ e então, $Px_m = x_m$. Assim,

$$P^2x = P(Px) = P(x_m) = x_m = Px \quad \forall x \in \mathbb{R}^n \Rightarrow P^2 = P.$$

Agora, mostremos que $Im(P) = M$ e $Ker(P) = S$. Para tanto, note que qualquer $x \in \mathbb{R}^n$ satisfaz as seguintes condições:

$$\begin{cases} Px \in M \\ x - Px \in S. \end{cases} \quad (1.1)$$

De fato, dado $x \in \mathbb{R}^n$, temos que $x = x_m + x_s$, com $x_m \in M$ e $x_s \in S$. Assim, temos que $Px = x_m \in M$ e $x - Px = (x_m + x_s) - x_m = x_s \in S$.

Suponha $x \in \text{Ker}(P)$, ou seja, $Px = 0$. Então, $x - Px = x \in S$. Logo, $\text{Ker}(P) \subseteq S$. Por outro lado, se $x \in S$, então $Px = x - (x - Px) \in S$, já que S é subespaço. Assim, $Px \in M \cap S$ e então $Px = 0$, já que $M \oplus S \Rightarrow M \cap S = \{0\}$. Daí, $x \in \text{Ker}(P)$ e temos $S \subseteq \text{Ker}(P)$. Portanto, $S = \text{Ker}(P)$.

Como $M \oplus S = \mathbb{R}^n$, temos que $\dim(M) = n - \dim(S)$. Por outro lado, $\dim \text{Ker}(P) + \dim \text{Im}(P) = n$ pelo Teorema do Núcleo e da Imagem, e portanto, $\dim \text{Im}(P) = n - \dim(S) = \dim(M)$. Por outro lado, de (1.1) temos que $Px \in M \forall x \in \mathbb{C}$, donde $\text{Im}(P) \subset M$. Como esses conjuntos têm mesma dimensão, segue que $M = \text{Im}(P)$.

Resta mostrarmos a unicidade de P .

Suponha que Q é um projetor que satisfaz $\text{Im}(Q) = M$ e $\text{Ker}(Q) = S$ e seja $x \in \mathbb{R}^n$ dado por $x = x_m + x_s$, com $x_m \in M$ e $x_s \in S$. Então, de $x_s \in S$ e $\text{Ker}(Q) = S$, segue que $Q(x_s) = 0$. De $x_m \in M$ e $\text{Im}(Q) = M$, segue que existe $y \in \mathbb{R}^n$ tal que $Qy = x_m$. Assim, $Q(Qy) = Qx_m \Rightarrow Q^2y = Qx_m \Rightarrow Qy = Qx_m$. Logo, $Qx_m = x_m$.

Da linearidade de P e Q , temos:

$$\begin{cases} Qx = Q(x_m + x_s) = Q(x_m) + Q(x_s) = x_m + 0 = x_m \\ Px = P(x_m + x_s) = x_m. \end{cases}$$

Assim, $Qx = Px \forall x \in \mathbb{R}^n$. Logo, $P = Q$. □

Seja $\text{posto}(P) = m$ e $\{p_1, p_2, \dots, p_m\}$ base de $\text{Im}(P) = M$. Então, dado $x \in \mathbb{R}^n$, podemos escrever $Px = \alpha_1 p_1 + \alpha_2 p_2 + \dots + \alpha_m p_m$, com $\alpha_i \in \mathbb{R} \forall i = 1, \dots, m$. Assim, note que em (1.1), na primeira expressão temos m coeficientes a determinar para obtermos Px . Para determiná-los, precisamos impor m condições. Do Teorema do Núcleo e Imagem, temos que

$$\begin{aligned} \dim \text{Ker}(P) + \text{posto}(P) &= \dim \mathbb{R}^n \\ \dim S + m &= n \\ \dim S &= n - m. \end{aligned}$$

Como $\dim S^\perp = m$, é interessante reescrevermos a segunda expressão de (1.1) em função de $R = S^\perp$, pois assim teremos as m condições de que precisamos. Podemos fazer isso da seguinte maneira:

$$x - Px \in S \Leftrightarrow x - Px \in S^{\perp\perp} \Leftrightarrow x - Px \perp S^\perp = R.$$

Assim, (1.1) pode ser reescrita como

$$\begin{cases} Px \in M \\ x - Px \perp R. \end{cases} \quad (1.2)$$

As condições (1.2) definem um projetor P em M ortogonal ao subespaço R .

Agora, o que podemos nos perguntar é se, dados dois subespaços arbitrários M e R com $\dim M = \dim R = m$, será sempre possível definir um projetor em M e ortogonal a R através das condições (1.2)? Veremos na proposição a seguir que isto é possível desde que $M \cap R^\perp = \{0\}$.

Proposição 1.4 *Dados dois subespaços M e R do \mathbb{R}^n com $\dim M = \dim R = m$, temos que $M \cap R^\perp = \{0\}$ se, e somente se, para qualquer $x \in \mathbb{R}^n$, existe um único vetor $u = Px$ que satisfaz as condições (1.2).*

Demonstração. Como R é de dimensão m , temos que R^\perp é de dimensão $n - m$ e então, a condição $M \cap R^\perp = \{0\}$ é equivalente a condição $\mathbb{R}^n = M \oplus R^\perp$, que é equivalente a afirmação que qualquer $x \in \mathbb{R}^n$ pode ser escrito de forma única como $x = u + w$, onde $u \in M$ e $w = x - u \in R^\perp$. \square

Assim, se M e R são subespaços de mesma dimensão que satisfazem $M \cap R^\perp = \{0\}$, então existe um projetor em M e ortogonal a R definido através das condições (1.2). Além disso, vale a seguinte propriedade:

$$Px = 0 \Leftrightarrow x \perp R. \quad (1.3)$$

De fato, a segunda condição de (1.2) nos diz que $\langle x - Px, y \rangle = 0 \forall y \in R$. Assim, se $Px = 0$, temos que $\langle x, y \rangle = 0 \forall y \in R$, ou seja, $x \perp R$. Por outro lado, se $x \perp R$ então

$$\begin{aligned} \langle x - Px, y \rangle &= 0 \quad \forall y \in R \\ \langle Px, y \rangle &= 0 \quad \forall y \in R \Rightarrow Px \in R^\perp. \end{aligned}$$

Assim, $Px \in M \cap R^\perp = \{0\}$, logo $Px = 0$.

Observe que se $M \cap R^\perp \neq \{0\}$, então não temos a unicidade do vetor $u = Px$ que satisfaz (1.2). De fato, se $z \in M \cap R^\perp$, então $z \in M$ e $\langle z, y \rangle = 0 \forall y \in R$. Agora, dado $x \in \mathbb{R}^n$, seja u tal que $u \in M$ e $x - u \in R^\perp$ e considere $\tilde{u} = u + \alpha z$, com $\alpha \in \mathbb{C}$. Então,

$$z \in M \text{ e } u \in M \Rightarrow \tilde{u} = u + \alpha z \in M \text{ e } x - \tilde{u} = (x - u) - \alpha z \in R^\perp,$$

ou seja, \tilde{u} também satisfaz (1.2).

1.2 Projetores Oblíquos e Ortogonais

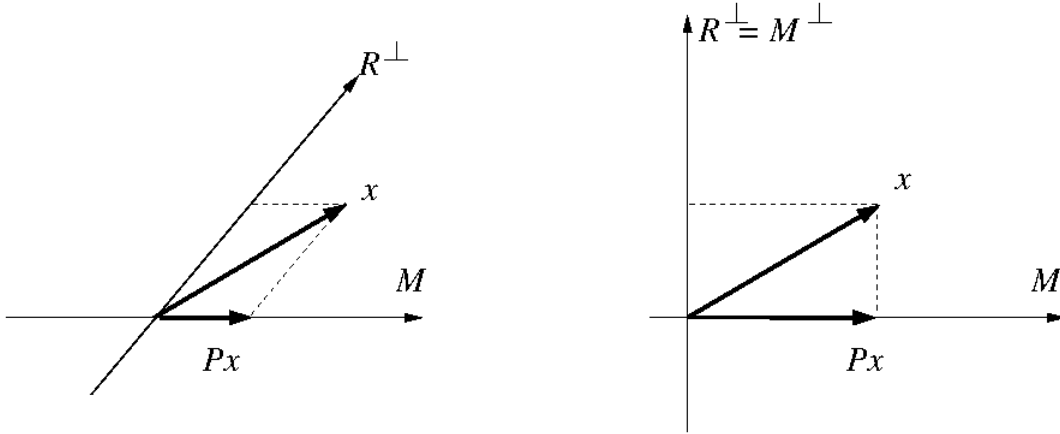
Definição 1.5 *Um projetor P é dito **ortogonal** se os subespaços $M = \text{Im}(P)$ e $R = (\text{Ker}P)^\perp$ são iguais. Caso contrário ($R \neq M$), o projetor é dito **oblíquo**.*

Assim, um projetor ortogonal é definido pelas seguintes condições:

$$\begin{cases} Px \in M \\ x - Px \perp M, \end{cases} \quad (1.4)$$

ou, equivalentemente,

$$\begin{cases} Px \in M \\ \langle (I - P)x, y \rangle = 0 \quad \forall y \in M. \end{cases}$$



1.2.1 Propriedades de Projetores Ortogonais

Se P é um projetor ortogonal, então, por (1.4), temos que Px e $(I - P)x$ são ortogonais. Assim, aplicando a norma-2 em ambos os lados da decomposição $x = Px + (I - P)x$, temos

$$\begin{aligned} \|x\|_2^2 &= \|Px + (I - P)x\|_2^2 \\ &= \langle Px + (I - P)x, Px + (I - P)x \rangle \\ &= \langle Px, Px \rangle + 2\langle Px, (I - P)x \rangle + \langle (I - P)x, (I - P)x \rangle \\ &= \|Px\|_2^2 + \|(I - P)x\|_2^2 \geq \|Px\|_2^2. \end{aligned} \quad (1.5)$$

Assim,

$$\frac{\|Px\|_2^2}{\|x\|_2^2} \leq 1 \quad \forall x \in \mathbb{R}^n - \{0\} \implies \|P\|_2^2 = \sup_{x \neq 0} \frac{\|Px\|_2^2}{\|x\|_2^2} \leq 1.$$

Além disso, o valor 1 é atingido para qualquer $x \in \text{Im}(P) = M$, pois neste caso teremos $Px = x$. Logo, se P é um projetor ortogonal, então $\|P\|_2^2 = 1$.

Teorema 1.6 *Seja P um projetor ortogonal em um subespaço M . Então, para qualquer $x \in \mathbb{R}^n$, vale a seguinte relação:*

$$\min_{y \in M} \|x - y\|_2 = \|x - Px\|_2.$$

Demonstração. Seja $y \in M$ arbitrário. Como $Px - y \in M$ e, por (1.4), $x - Px \perp M$, temos que $x - Px$ é ortogonal a $Px - y$. Analogamente a (1.5), encontramos

$$\|x - y\|_2^2 = \|(x - Px) + (Px - y)\|_2^2 = \|x - Px\|_2^2 + \|Px - y\|_2^2.$$

Assim, $\|x - y\|_2^2 \geq \|x - Px\|_2^2 \forall y \in M$ e então,

$$\min_{y \in M} \|x - y\|_2 \geq \|x - Px\|_2.$$

Agora, note que o mínimo é atingido tomando $y = Px$, donde segue o resultado. \square

Se definirmos $y^* = Px$, onde P é um projetor ortogonal, podemos reformular o resultado do teorema acima através das condições de (1.4) da seguinte maneira: Dados $x \in \mathbb{R}^n$ um subespaço M , temos que

$$\min_{y \in M} \|x - y\|_2 = \|x - y^*\|_2 \Leftrightarrow y^* \in M \text{ e } x - y^* \perp M.$$

1.3 Métodos de Projeção

Seja A uma matriz $n \times n$ e $b \in \mathbb{R}^n$. Os métodos de projeção extraem soluções aproximadas para o sistema linear

$$Ax = b \tag{1.6}$$

em um determinado subespaço K de \mathbb{R}^n com $\dim K = m$.

Para extrairmos uma solução aproximada de (1.6) no subespaço K , teremos m incógnitas a determinar, visto que x será escrito como combinação linear de m vetores da base de K . Assim, devemos impor m restrições para podermos extrair uma solução aproximada em K . Vamos fazer isso impondo que $b - Ax$ seja ortogonal a m vetores LI, que definem um subespaço R com $\dim R = m$.

O método de projeção pode ser ortogonal (quando $R = K$) ou oblíquo (quando $R \neq K$). Tendo em vista a Proposição 1.4, vamos considerar daqui em diante somente os métodos de projeção tais que $K \cap R^\perp = \{0\}$.

Dado um chute inicial $x_0 \in \mathbb{R}^n$ para a solução de (1.6), vamos buscar uma solução aproximada \tilde{x} de (1.6) dada por

$$\begin{cases} \tilde{x} \in x_0 + K = \{z \in \mathbb{R}^n ; z = x_0 + \delta, \delta \in K\} \\ b - A\tilde{x} \perp R. \end{cases} \tag{1.7}$$

Se definirmos o resíduo inicial como $r_0 = b - Ax_0$, então teremos $\tilde{x} = x_0 + \delta$,

$\delta \in K$ e $r_0 - A\delta \perp R$. Isto permite expressar a solução aproximada \tilde{x} na forma

$$\begin{cases} \tilde{x} = x_0 + \delta, \delta \in K \\ \langle r_0 - A\delta, w \rangle = 0 \quad \forall w \in R. \end{cases} \quad (1.8)$$

No caso particular em que $x_0 = 0$, temos que $\tilde{x} = x_0 + \delta = \delta$ e

$$r_0 - A\delta = b - Ax_0 - A\tilde{x} = b - A\tilde{x}.$$

Assim, (1.8) pode ser reformulado como encontrar $\tilde{x} \in K$ tal que $\langle b - A\tilde{x}, w \rangle = 0 \quad \forall w \in R$, isto é, encontrar $\tilde{x} \in K$ tal que $b - A\tilde{x} \perp R$. Por (1.3), isto é equivalente a encontrar $\tilde{x} \in K$ tal que $Q_K^R(b - A\tilde{x}) = 0$, sendo Q_K^R um projetor definido por

$$\begin{cases} Q_K^R x \in K \\ x - Q_K^R x \perp R. \end{cases}$$

Dizemos que K é um conjunto invariante sobre A se $x \in K \implies Ax \in K$. A proposição a seguir mostra que neste caso a solução aproximada pelo método de projeção coincide (sob condições adicionais) com a solução exata do problema (1.8).

Proposição 1.7 *Assuma que K é invariante sobre A , $x_0 = 0$ e $b \in K$. Então, a solução aproximada obtida de qualquer método de projeção (obliquo ou ortogonal) em K é exata.*

Demonstração. A solução aproximada $\tilde{x} \in K$ satisfaz $Q_K^R(b - A\tilde{x}) = 0$. Como $b \in K$, segue que $Q_K^R b = b$. Como K é invariante sobre A , segue que $A\tilde{x} \in K$, donde $Q_K^R A\tilde{x} = A\tilde{x}$. Assim,

$$0 = Q_K^R(b - A\tilde{x}) = Q_K^R b - Q_K^R A\tilde{x} = b - A\tilde{x}.$$

Portanto, $A\tilde{x} = b$, isto é, \tilde{x} é solução exata. □

1.3.1 Representação Matricial dos Métodos de Projeção

Sejam $v_1, v_2, \dots, v_m \in \mathbb{R}^n$ uma base de K e $w_1, w_2, \dots, w_m \in \mathbb{R}^n$ uma base de R . Vamos denotar respectivamente por V e W as matrizes cujas colunas são formadas por estas bases:

$$V = [v_1, v_2, \dots, v_m]_{n \times m}, \quad W = [w_1, w_2, \dots, w_m]_{n \times m}. \quad (1.9)$$

A seguir vamos expressar o vetor \tilde{x} (1.8) gerado por um método de projeção em termos das matrizes V e W

Proposição 1.8 *Seja \tilde{x} dado por (1.8) e $y \in \mathbb{R}^n$ tal que $\tilde{x} = x_0 + Vy$. Temos que $W^T AVy = W^T r_0$*

Demonstração. De fato, temos da condição de ortogonalidade que

$$\begin{aligned}\langle r_0 - AVy, w \rangle &= 0 & \forall w \in R \\ w^T (r_0 - AVy) &= 0 & \forall w \in R \\ w^T r_0 &= w^T AVy & \forall w \in R.\end{aligned}$$

Em particular, para os vetores da base de R , temos:

$$\begin{cases} w_1^T r_0 = w_1^T AVy \\ w_2^T r_0 = w_2^T AVy \\ \vdots \\ w_m^T r_0 = w_m^T AVy \end{cases} \Rightarrow \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_m^T \end{bmatrix} r_0 = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_m^T \end{bmatrix} AVy.$$

Na forma matricial, temos $W^T r_0 = W^T AVy$. □

Assim, se $W^T AV$ for não singular, podemos escrever a aproximação $\tilde{x} = x_0 + Vy$ como

$$\tilde{x} = x_0 + V (W^T AV)^{-1} W^T r_0. \quad (1.10)$$

Vejamos dois casos em que $W^T AV$ é não singular na proposição a seguir:

Proposição 1.9 *Sejam A uma matriz $n \times n$ e K, R dois subespaços de \mathbb{R}^n de dimensão m satisfazendo uma das seguintes condições:*

- (i) *A é positiva definida e $R = K$;*
- (ii) *A é não singular e $R = AK$.*

Então, a matriz $B = W^T AV$ é não singular, sendo V e W dadas por (1.9), e $\{v_1, v_2, \dots, v_m\}$ e $\{w_1, w_2, \dots, w_m\}$ bases quaisquer de K e R , respectivamente.

Demonstração.

(i) Como $R = K$, temos que existe G não singular tal que $W = VG$. Assim,

$$B = W^T AV = (VG)^T AV = G^T V^T AV.$$

Mostremos que $V^T AV$ é positiva definida. De fato, como A é positiva definida, temos que $x^T Ax > 0 \forall x \neq 0$. Assim, $x^T V^T AVx = (Vx)^T A(Vx) > 0$ para todo $Vx \neq 0$. Mas V é não singular, donde $Vx = 0 \Leftrightarrow x = 0$. Assim, $x^T V^T AVx > 0 \forall x \neq 0$. Logo, $V^T AV$ é positiva definida, e portanto, não singular. Daí, segue que B é não singular, já que é produto de matrizes não singulares.

(ii) Como $R = AK$, temos que existe G não singular $m \times m$ tal que $W = AVG$. Assim,

$$B = W^T AV = (AVG)^T AV = G^T A^T V^T AV = G^T (AV)^T AV.$$

Mostremos agora que $(AV)^T AV$ é não singular, e daí segue o resultado. Suponha que $(AV)^T AV$ seja singular. Logo, existe $x \neq 0$ tal que $(AV)^T AVx = 0$ e então

$$(AV)^T AVx = 0 \Rightarrow x^T (AV)^T AVx = 0 \Rightarrow \|AVx\|^2 = 0 \Rightarrow AVx = 0, \text{ com } x \neq 0.$$

Como A é não singular, segue que $Vx = 0$ e $x \neq 0$, o que contradiz a não singularidade de V . Logo, $(AV)^T AV$ é não singular. \square

1.4 Métodos de Projeção Unidimensionais

Vamos considerar o caso particular do método de projeção em uma dimensão ($m = 1$), ou seja, $K = \text{span}\{v\}$ e $R = \text{span}\{w\}$. Seja $x_0 \in \mathbb{R}^n$ o chute inicial e $r_0 = b - Ax_0$ o vetor resíduo inicial. Temos $\tilde{x} = x_0 + \delta$ com $\delta = \alpha v$, $\alpha \in \mathbb{R}$, enquanto a condição $r_0 - A\delta \perp z \forall z \in R$ nos dá:

$$\begin{aligned} \langle r_0 - A\alpha\delta, z \rangle &= 0, \quad \forall z \in R \\ \langle r_0, z \rangle - \alpha \langle Av, z \rangle &= 0 \quad \forall z \in R. \end{aligned}$$

Mas dado $z \in R$, temos que $z = \beta w$ com $\beta \in \mathbb{R}$, donde segue que

$$\alpha = \frac{\langle r_0, \beta w \rangle}{\langle Av, \beta w \rangle} = \frac{\langle r_0, w \rangle}{\langle Av, w \rangle}.$$

Na forma iterativa, definimos $K = \text{span}\{v_k\}$, $R = \text{span}\{w_k\}$ e

$$x_{k+1} = x_k + \alpha_k v_k, \quad \alpha_k = \frac{\langle r_k, w_k \rangle}{\langle Av_k, w_k \rangle}, \quad (1.11)$$

com $r_k = b - Ax_k$ para $k = 1, 2, \dots$. Note que a condição (1.7) corresponde a $\langle r_{k+1}, v_k \rangle = 0$.

Tendo em vista a Proposição 1.9, vamos apresentar métodos de projeção definidos por $R = K$ (ou seja, $w_k = v_k$) e $R = AK$ (ou seja, $w_k = Av_k$):

- **Método de Máxima Descida:** o algoritmo de máxima descida consiste em tomar em cada etapa $v_k = r_k$ e $w_k = r_k$. Com esta escolha, temos que $\langle r_{k+1}, r_k \rangle = 0$, ou seja, o resíduo da iteração atual é ortogonal ao resíduo da iteração anterior. Além disso, pode-se

mostrar que a solução aproximada x_{k+1} em (1.11) minimiza a forma quadrática

$$f(\tilde{x}) = \frac{1}{2} \|\tilde{x} - x\|_A^2 = \frac{1}{2} \langle A(\tilde{x} - x), \tilde{x} - x \rangle$$

sobre todos os vetores da forma $\tilde{x} = x_k + \alpha r_k$ e além disso, temos $r_k = -\nabla f(x_k)$, que corresponde à direção de máxima descida.

Algoritmo 1 Método de Máxima Descida

1: **Para** $k = 0, 1, \dots$, **faça**

2: $r_k = b - Ax_k$;

3: $\alpha_k = \langle r_k, r_k \rangle / \langle Ar_k, r_k \rangle$;

4: $x_{k+1} = x_k + \alpha_k r_k$;

5: **Fim**

• **Iteração MR:** analogamente ao Método de Máxima Descida, a iteração MR (denominada *Orthomin(1)* em [15]) também utiliza $v_k = r_k$, porém escolhe $w_k = Ar_k$. Para este método, a condição (1.7) corresponde a $\langle r_{k+1}, Ar_k \rangle = 0$, ou seja, r_{k+1} e r_k são vetores A -conjugados. Pode-se mostrar que cada passo da iteração MR minimiza o resíduo $f(\tilde{x}) = \|b - A\tilde{x}\|_2$ sobre todos os vetores da forma $\tilde{x} = x_k + \alpha r_k$.

Algoritmo 2 Iteração MR

Para $k = 0, 2, \dots$, **faça**

$r_k = b - Ax_k$;

$\alpha_k = \langle r_k, r_k \rangle / \langle Ar_k, Ar_k \rangle$;

$x_{k+1} = x_k + \alpha_k r_k$;

Fim

A convergência dos dois métodos é descrita pelos seguintes resultados:

Teorema 1.10 *Seja A simétrica positiva definida. Para todo $k \geq 0$, o vetor-erro $e_k = x_* - x_k$ satisfazem a seguinte relação:*

$$\|e_{k+1}\|_A \leq \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \|e_k\|_A$$

e o método de máxima descida converge para qualquer valor inicial x_0 .

Demonstração. vide [27, Teor. 5.9].

Teorema 1.11 *Seja A uma matriz positiva definida (mas não necessariamente simétrica) e sejam $\mu = \lambda_{\min}((A + A^T)/2)$ e $\sigma = \|A\|_2$. Os vetores residuais do algoritmo MR satisfazem a relação*

$$\|r_{k+1}\|_2 \leq \left(1 - \frac{\mu^2}{\sigma^2}\right)^{\frac{1}{2}} \|r_k\|_2$$

e o algoritmo converge para qualquer vetor inicial x_0 .

Demonstração. vide [27, Teor. 5.10].

Capítulo 2

Método de Arnoldi

Vimos no final do capítulo anterior uma forma de se escolher espaços K e L de dimensão 1 para métodos de projeção. Estas escolhas foram motivadas por problemas de minimização de formas quadráticas associadas a medidas do erro da solução aproximada \tilde{x} . Neste capítulo veremos como construir espaços K e L de dimensão maior que 1 e introduzir um método que ortogonaliza estes espaços, tornando mais estável e eficiente o cálculo da aproximação \tilde{x} .

2.1 Subespaços de Krylov

Sejam $A \in \mathbb{R}^{n \times n}$ e $0 \neq v \in \mathbb{R}^n$. O subespaço $K_m = K_m(A, v)$ de \mathbb{R}^n gerado pelos produtos sucessivos de A por v ,

$$K_m = \text{span} \{v, Av, A^2v, \dots, A^{m-1}v\}, \quad (2.1)$$

é dito *subespaço de Krylov de ordem m* .

Note que K_m é o subespaço de \mathbb{R}^n de todos os vetores que podem ser escritos como $x = p(A)v$, onde p é um polinômio de grau no máximo $m - 1$.

Veremos a seguir duas proposições que relacionam o grau do *polinômio minimal* de um vetor v com um subespaço de Krylov. O polinômio minimal de um vetor v é o polinômio mônico de menor grau tal que $p(A)v = 0$. O grau do polinômio minimal de v com respeito a A será chamado *grau de v com respeito a A* .

Proposição 2.1 *Seja μ o grau de v com respeito a A . Então, K_μ é invariante sobre A e $K_m = K_\mu \forall m \geq \mu$.*

Demonstração. Mostremos que K_μ é invariante sobre A . Seja $p(x) = \beta_0 + \beta_1x + \dots + \beta_{\mu-1}x^{\mu-1} + x^\mu$ o polinômio minimal de v com respeito a A . Então, $p(A)v = 0$, logo

$$\begin{aligned} p(A)v &= \beta_0v + \beta_1Av + \dots + \beta_{\mu-1}A^{\mu-1}v + A^\mu v = 0 \\ A^\mu v &= -\beta_0v - \beta_1Av - \dots - \beta_{\mu-1}A^{\mu-1}v. \end{aligned} \quad (2.2)$$

Agora, seja $x = \alpha_0 v + \alpha_1 Av + \dots + \alpha_{\mu-1} A^{\mu-1} v \in K_\mu$. Segue de (2.2) que

$$\begin{aligned} Ax &= \alpha_0 Av + \alpha_1 A^2 v + \dots + \alpha_{\mu-1} A^\mu v \\ &= \alpha_0 Av + \alpha_1 A^2 v + \dots + \alpha_{\mu-1} (-\beta_0 v - \beta_1 Av - \dots - \beta_{\mu-1} A^{\mu-1} v) \\ &= (-\alpha_{\mu-1} \beta_0) v + (\alpha_0 - \alpha_{\mu-1} \beta_1) Av + \dots + (-\alpha_{\mu-1} \beta_{\mu-1}) A^{\mu-1} v \in K_\mu. \end{aligned}$$

Portanto, K_μ é invariante sobre A . Mostremos agora que $K_m = K_\mu \forall m \geq \mu$. Como $\mu \leq m$, temos $K_\mu \subseteq K_m$. Mostremos que $K_m \subseteq K_\mu \forall m \geq \mu$ por indução.

Para $m = \mu$, temos $K_m \subseteq K_\mu$ trivialmente. Suponha que $K_{\mu+j} \subseteq K_\mu$ e mostremos que $K_{\mu+j+1} \subseteq K_\mu$. De fato, dado $x \in K_{\mu+j+1}$ arbitrário, temos que

$$x = \sum_{i=1}^{\mu+j} \alpha_i A^i v = \sum_{i=1}^{\mu+j-1} \alpha_i A^i v + \alpha_{\mu+j} A^{\mu+j} v = \sum_{i=1}^{\mu+j-1} \alpha_i A^i v + \alpha_{\mu+j} A (A^{\mu+j-1} v).$$

Note que $A^i v \in K_{\mu+j} \subseteq K_\mu$ para todo $i = 1, \dots, \mu + j - 1$. Além disso, $A^{\mu+j-1} v \in K_{\mu+j} \subseteq K_\mu$, e como K_μ é invariante sobre A , segue que $A(A^{\mu+j-1} v) \in K_\mu$. Logo, $x \in K_\mu$ para $x \in K_{\mu+j+1}$ arbitrário, ou seja, $K_{\mu+j+1} \subseteq K_\mu$. \square

Proposição 2.2 *O subespaço de Krylov K_m é de dimensão m se, e somente se, o grau μ de v com respeito a A não é menor do que m , ou seja,*

$$\dim K_m = m \Leftrightarrow \text{grau}(v) \geq m.$$

Além disso, $\dim K_m = \min \{m, \mu\}$.

Demonstração. Por definição, os vetores $v, Av, \dots, A^{m-1}v$ geram o espaço K_m . Para que estes vetores formem uma base de K_m , a combinação linear $\alpha_0 v + \alpha_1 Av + \dots + \alpha_{m-1} A^{m-1} v$ deve ser nula se, e somente se, os escalares $\alpha_i, i = 0, \dots, m-1$, são todos nulos. Isto é equivalente a dizer que o único polinômio com grau menor ou igual a $m-1$ tal que $p(A)v = 0$ é o polinômio nulo, ou seja, o grau μ do polinômio minimal não pode ser menor ou igual a $m-1$.

Vejam agora que $\dim K_m = \min \{m, \mu\}$. De fato, se $\mu \geq m$, então $\dim K_m = m = \min \{m, \mu\}$ pelo que acabamos de provar. Caso $\mu = \text{grau}(v) < m$, então, pela Proposição 2.1, segue que $K_m = K_\mu$, donde $\dim K_m = \mu = \min \{m, \mu\}$. \square

2.2 Método de Arnoldi

O método de Arnoldi é um método de projeção ortogonal em K_m para uso em geral de matrizes não hermitianas. O algoritmo gera uma base ortogonal para o subespaço de Krylov K_m pelo processo de ortogonalização de Gram-Schmidt.

Algoritmo 3 Método de Arnoldi

Escolha um vetor v_1 tal que $\|v_1\|_2 = 1$;
Para $j = 1, 2, \dots, m$ **faça**
 Calcule $h_{ij} = \langle Av_j, v_i \rangle$ para $i = 1, 2, \dots, j$;
 Calcule $w_j = Av_j - \sum_{i=1}^j h_{ij}v_i$;
 $h_{j+1,j} = \|w_j\|_2$;
 Se $h_{j+1,j} = 0$, páre;
 $v_{j+1} = w_j/h_{j+1,j}$;
Fim

Proposição 2.3 *Assuma que o algoritmo de Arnoldi não pára antes do passo m , ou seja, $h_{j+1,j} \neq 0$ para $1 \leq j < m$ na linha 5 do algoritmo. Então, os vetores v_1, v_2, \dots, v_m formam uma base ortonormal para o subespaço de Krylov*

$$K_m = \text{span} \{v_1, Av_1, \dots, A^{m-1}v_1\}.$$

Demonstração. Por construção, os vetores v_1, v_2, \dots, v_m são ortonormais, já que o algoritmo utiliza o processo de Gram-Schmidt para determiná-los. Assim, basta mostrarmos que eles geram K_m . Para tanto, mostremos que cada vetor v_j é da forma $q_{j-1}(A)v_1$, onde q_{j-1} é um polinômio de grau $j-1$. A prova será feita por indução em j .

Se $j = 1$, então $v_1 = q_0(A)v_1$, com $q_0(t) = 1$.

Assuma agora que o resultado é válido para todos os inteiros menores ou iguais a j e mostremos que vale para $j+1$. Temos que

$$v_{j+1}h_{j+1,j} = w_j = Av_j - \sum_{i=1}^j h_{ij}v_i = Aq_{j-1}(A)v_1 - \sum_{i=1}^j h_{ij}q_{i-1}(A)v_1,$$

logo

$$v_{j+1} = q_j(A)v_1, \quad q_j(t) = \frac{1}{h_{j+1,j}} \left(tq_{j-1}(t) - \sum_{i=1}^j h_{ij}q_{i-1}(t) \right).$$

□

Proposição 2.4 *Sejam V_m a matriz $n \times m$ com vetores coluna v_1, \dots, v_m ; \overline{H}_m a matriz $(m+1) \times m$ Hessenberg cujas entradas não nulas h_{ij} são definidas pelo algoritmo de Arnoldi e H_m a matriz obtida de \overline{H}_m deletando a última linha. Então, valem as seguintes relações:*

$$AV_m = V_m H_m + w_m e_m^T, \quad (2.3)$$

$$AV_m = V_{m+1} \overline{H}_m, \quad (2.4)$$

$$V_m^T AV_m = H_m. \quad (2.5)$$

Demonstração.

- $AV_m = V_m H_m + w_m e_m^T$:

Das linhas 4 e 7 do algoritmo de Arnoldi temos, para $j = 1, 2, \dots, m$,

$$Av_j = w_j + \sum_{k=1}^j h_{kj} v_k = v_{j+1} h_{j+1,j} + \sum_{k=1}^j h_{kj} v_k. \quad (2.6)$$

Sejam

$$B = V_m = [v_1 \ v_2 \ \dots \ v_m]_{n \times m}, \quad C = AV_m = [Av_1 \ Av_2 \ \dots \ Av_m]_{n \times m} \quad (2.7)$$

e $[v_k]_i$ a posição i do vetor v_k . Então

$$C_{ij} = [Av_j]_i = [v_{j+1}]_i h_{j+1,j} + \sum_{k=1}^j h_{kj} [v_k]_i.$$

Se $j < m$:

$$C_{ij} = [Av_j]_i = [v_{j+1}]_i h_{j+1,j} + \sum_{k=1}^j h_{kj} [v_k]_i = \sum_{k=1}^{j+1} [v_k]_i h_{kj} = \sum_{k=1}^{j+1} B_{ik} h_{kj} = \sum_{k=1}^m B_{ik} h_{kj}, \quad (2.8)$$

sendo que a última igualdade em (2.8) ocorre porque $h_{kj} = 0$ para $j+1 < k \leq m$.

Se $j = m$:

$$C_{im} = [Av_m]_i = \sum_{k=1}^m [v_k]_i h_{km} + h_{m+1,m} [v_{m+1}]_i = \sum_{k=1}^m B_{ik} h_{km} + [w_m]_i.$$

A última igualdade acima segue da linha 7 do algoritmo.

Para escrevermos uma única expressão para C_{ij} , com $j = 1, \dots, m$, defina a matriz

$$Z_{ij} = \begin{cases} 0, & \text{se } 1 \leq i \leq m \text{ e } 1 \leq j \leq m-1 \\ [w_m]_i & \text{se } j = m, \end{cases}$$

ou, equivalentemente, $Z = w_m e_m^T$. Então, para $j = 1, \dots, m$ temos

$$C_{ij} = \sum_{k=1}^m B_{ik} h_{km} + Z_{ij},$$

ou na forma matricial, $C = V_m H_m + Z = V_m H_m + w_m e_m^T$.

- $AV_m = V_{m+1} \bar{H}_m$:

Por (2.6), temos que

$$Av_j = \sum_{k=1}^{j+1} h_{kj} v_k, \quad j = 1, 2, \dots, m. \quad (2.9)$$

Seja C dado por (2.7), $D = V_{m+1} = [v_1 \ v_2 \ \dots \ v_{m+1}]_{n \times m+1}$ e $[v_k]_i$ a posição i do vetor v_k . Segue de (2.9) que

$$C_{ij} = [Av_j]_i = \sum_{k=1}^{j+1} h_{kj} [v_k]_i = \sum_{k=1}^{m+1} h_{kj} [v_k]_i.$$

A última igualdade da expressão acima ocorre porque para $1 \leq j < m$ e $j+1 < k \leq m+1$, temos que $h_{kj} = 0$, já que \bar{H}_m é Hessenberg. Assim,

$$C_{ij} = \sum_{k=1}^{m+1} D_{ik} h_{kj} \Rightarrow AV_m = C = V_{m+1} \bar{H}_m.$$

- $V_m^T AV_m = H_m$:

Multiplicando ambos os lados de (2.3) (que já foi demonstrada) por V_m^T , temos

$$V_m^T AV_m = V_m^T V_m H_m + V_m^T w_m e_m^T.$$

Pela ortonormalidade dos vetores v_i 's, segue que $V_m^T V_m = I$. Além disso, pela linha 7 do algoritmo de Arnoldi temos

$$V_m^T w_m e_m^T = V_m^T v_{m+1} h_{m+1,m} e_m^T = h_{m+1,m} (v_m^T v_{m+1}) = h_{m+1,m} \cdot 0 = 0.$$

Assim, $V_m^T AV_m = H_m$. □

Proposição 2.5 *O algoritmo de Arnoldi quebra no passo j (ou seja, $\|w_j\|_2 = h_{j+1,j} = 0$ na linha 5 do algoritmo) se, e somente se, o polinômio minimal de v_1 com respeito a A tem grau j .*

Demonstração. (\Leftarrow) Suponha que o grau μ do polinômio minimal de v_1 é j e mostremos que $w_j = 0$. De fato, suponha $w_j \neq 0$. Daí, v_{j+1} pode ser definido e então K_{j+1} tem dimensão $j+1$. Assim, pela Proposição 2.2, $\mu \geq j+1$, o que contradiz $\mu = j$. Logo, $w_j = 0$ e o algoritmo quebra no passo j .

(\Rightarrow) Suponha $w_j = 0$. Então, $\dim K_{j+1} \neq j+1$ e da Proposição 2.2 segue que $\text{grau}(v_1) < j+1$, ou seja, $\mu = \text{grau}(v_1) \leq j$. Por outro lado, pela proposição 2.3, os vetores v_1, v_2, \dots, v_j formam uma base para K_j , donde $\dim K_j = j$. Novamente pela Proposição 2.2, temos $\mu \geq j$. Logo, $\mu = \text{grau}(v_1) = j$. □

2.2.1 Métodos de Projeção Baseados no Método de Arnoldi

Dado um chute inicial x_0 para solução do sistema linear $Ax = b$, e dadas as matrizes V_m e H_m geradas pelo Método de Arnoldi, vamos considerar métodos de projeção que utilizam o espaço $K = K_m(A, r_0)$.

Note que os vetores-coluna da matriz V_n definem uma base para o espaço K neste caso. Assim, podemos representar os vetores $\tilde{x} \in x_0 + K_m(A, r_0)$ na forma $\tilde{x} = x_0 + V_m y$, com $y \in \mathbb{R}^m$.

Analogamente à Seção 1.4, vamos apresentar dois métodos de projeção definidos pelas escolhas $L = K$ e $L = AK$, para as quais, sob algumas condições, se garante que a solução aproximada \tilde{x} pode ser escrita na forma (1.10).

• **Método de Ortogonalização Completa (FOM):** este método de projeção toma $L = K = K_m(A, r_0)$, onde $r_0 = b - Ax_0$. De acordo com (1.7), temos uma solução aproximada $\tilde{x} \in x_0 + K_m(A, r_0)$ tal que $b - A\tilde{x} \perp K_m(A, r_0)$.

Se $v_1 = r_0 / \|r_0\|_2$ no método de Arnoldi e $\beta = \|r_0\|_2$, então

$$V_m^T r_0 = V_m^T (\beta v_1) = \beta V_m^T (v_1) = \beta e_1. \quad (2.10)$$

Na última igualdade da expressão acima usamos o fato dos vetores v_1, \dots, v_m serem ortonormais. Agora, assumindo A positiva definida, temos por (1.10), (2.5) e (2.10) que a solução aproximada no subespaço $x_0 + K_m(A, r_0)$ é dada por

$$\tilde{x} = x_0 + V_m (V_m^T A V_m)^{-1} V_m^T r_0 = x_0 + V_m (H_m^{-1}) \beta e_1,$$

ou, equivalentemente:

$$\begin{cases} \tilde{x} = x_0 + V_m y_m, \\ y_m = H_m^{-1} (\beta e_1). \end{cases}$$

• **Método de Resíduo Mínimo Generalizado (GMRES):** neste método tomamos $K = K_m$ e $L = AK_m$, sendo K_m o subespaço de Krylov com $v_1 = r_0 / \|r_0\|_2$.

Assim, buscamos $\tilde{x} \in x_0 + K_m(A, r_0)$ tal que $b - A\tilde{x} \perp AK_m(A, r_0)$, ou seja, buscamos $\tilde{y} \in \mathbb{R}^m$ tal que

$$\langle b - A\tilde{x}, AV_m y \rangle = \langle b - A(x_0 + V_m \tilde{y}), AV_m y \rangle = 0 \quad \forall y \in \mathbb{R}^m. \quad (2.11)$$

A condição (2.11) corresponde a encontrar $\tilde{y} \in \mathbb{R}^m$ que minimiza o funcional

$$J(y) = \|b - Ax\|_2 = \|b - A(x_0 + V_m y)\|_2. \quad (2.12)$$

Algoritmo 4 Método de Ortogonalização Completa (FOM)

```

1:  $r_0 = b - Ax_0$ ;  $\beta = \|r_0\|_2$ ;  $v_1 = r_0/\beta$ ;
2: Para  $j = 1, 2, \dots, m$  faça
3:    $w_j = Av_j$ ;
4:   Para  $i = 1, 2, \dots, j$  faça
5:      $h_{ij} = \langle w_j, v_i \rangle$ ;
6:    $w_j = w_j - h_{ij}v_i$ ;
7:   Fim
8:    $h_{j+1,j} = \|w_j\|_2$ ;
9:   Se  $h_{j+1,j} = 0$  então
10:     Faça  $m = j$  e vá para linha 15;
11:   Senão
12:      $v_{j+1} = w_j/h_{j+1,j}$ ;
13:   Fim
14: Fim
15:  $H_m = (h_{ij})$ ,  $1 \leq i, j \leq m$ ;
16: Resolva o sistema linear  $H_m y_m = \beta e_1$ ;
17:  $\tilde{x} = x_0 + V_m y_m$ ;

```

Por (2.4), temos

$$\begin{aligned} b - Ax &= b - A(x_0 + V_m y) = r_0 - AV_m y \\ &= \beta v_1 - V_{m+1} \bar{H}_m y = V_{m+1} (\beta e_1 - \bar{H}_m y). \end{aligned}$$

Como os vetores coluna de V_{m+1} são ortonormais, então $V_{m+1}^T V_{m+1} = I$ e daí

$$\begin{aligned} J(y)^2 &= \|b - Ax\|_2^2 \\ &= \|V_{m+1} (\beta e_1 - \bar{H}_m y)\|_2^2 \\ &= (V_{m+1} (\beta e_1 - \bar{H}_m y))^T (V_{m+1} (\beta e_1 - \bar{H}_m y)) \\ &= (\beta e_1 - \bar{H}_m y)^T V_{m+1}^T V_{m+1} (\beta e_1 - \bar{H}_m y) \\ &= (\beta e_1 - \bar{H}_m y)^T (\beta e_1 - \bar{H}_m y) \\ &= \|\beta e_1 - \bar{H}_m y\|_2^2, \end{aligned}$$

assim, a aproximação GMRES é dada por $\tilde{x} = x_0 + V_m y_m$, sendo que y_m minimiza a função $J(y) = \|\beta e_1 - \bar{H}_m y\|_2$.

No caso em que a matriz A é diagonalizável e tem espectro $\sigma(A)$, a convergência dos métodos FOM e GMRES pode ser expressa em termos do parâmetro

$$\varepsilon^{(m)} = \min_{p \in P_m; p(0)=1} \max_{\lambda \in \sigma(A)} |p(\lambda)|.$$

Teorema 2.6 *Seja A uma matriz diagonalizável com autovetores u_1, \dots, u_n , e sejam $\alpha_1, \dots, \alpha_n$ as coordenadas do vetor x solução do sistema $Ax = b$ com respeito à base $\{u_1, \dots, u_n\}$, ou*

Algoritmo 5 Método GMRES

```

1:  $r_0 = b - Ax_0$ ;  $\beta = \|r_0\|_2$ ;  $v_1 = r_0/\beta$ ;
2: Para  $j = 1, 2, \dots, m$  faça
3:    $w_j = Av_j$ ;
4:   Para  $i = 1, \dots, j$  faça
5:      $h_{ij} = \langle w_i, v_i \rangle$ ;
6:      $w_j = w_j - h_{ij}v_i$ ;
7:   Fim
8:    $h_{j+1,j} = \|w_j\|_2$ ;
9:   Se  $h_{j+1,j} = 0$  então
10:    Faça  $m = j$  e vá para linha 15;
11:   Senão
12:      $v_{j+1} = w_j/h_{j+1,j}$ ;
13:   Fim
14: Fim
15:  $\bar{H}_m = (h_{ij})$ ,  $1 \leq i \leq m+1$  e  $1 \leq j \leq m$ ;
16:  $y_m = \min_y \|\beta e_1 - \bar{H}_m y\|_2$ ;
17:  $\tilde{x} = x_0 + V_m y_m$ ;

```

seja,

$$x = \alpha_1 u_1 + \dots + \alpha_n u_n.$$

Seja $\alpha = \alpha_1 + \dots + \alpha_n$. A solução \tilde{x} gerada pelo algoritmo FOM satisfaz

$$\|\tilde{x} - x\|_2 \leq \alpha \sqrt{1 + \gamma_m^2 \kappa_m^2} \varepsilon^{(m)},$$

sendo $\gamma_m^2 = \|PA - PAP\|_2$, $\kappa_m = \|(PAP)^{-1}\|_2$ e P o projetor ortogonal em $K_m(A, r_0)$.

Demonstração. vide [26, p. 115].

Teorema 2.7 *Seja* $A = XDX^{-1}$ *uma matriz diagonalizável. O resíduo* $\tilde{r} = b - A\tilde{x}$ *da solução* \tilde{x} *gerada pelo algoritmo GMRES satisfaz*

$$\|\tilde{r}\|_2 \leq \kappa(X) \|r_0\|_2 \varepsilon^{(m)}, \quad \kappa(X) = \|X\|_2 \|(X)^{-1}\|_2.$$

Demonstração. vide [28, Prop. 4].

No caso particular em que a matriz A é positiva definida e seu espectro está contido em um círculo com centro em $C > 0$ e raio $R < C$, temos que $\varepsilon^{(m)} \leq (R/C)^m$ [28, Teor. 5].

Embora as estimativas apresentadas acima possam ser usadas para matrizes não-normais, elas podem não indicar precisamente a taxa de convergência dos métodos [15, 29].

2.3 Algoritmo de Lanczos Simétrico

O algoritmo de Lanczos simétrico pode ser visto como uma simplificação do método de Arnoldi para o caso particular em que a matriz é simétrica. O que veremos a seguir é que quando tomamos A simétrica, a matriz Hessenberg torna-se então uma matriz tridiagonal simétrica.

Proposição 2.8 *Assuma que o método de Arnoldi é aplicado em uma matriz simétrica A . Então, os coeficientes h_{ij} gerados pela matriz são tais que*

$$\begin{cases} h_{ij} = 0, & 1 \leq i < j - 1 \\ h_{j,j+1} = h_{j+1,j}, & j = 1, 2, \dots, m. \end{cases}$$

Em outras palavras, a matriz H_m obtida do processo de Arnoldi é tridiagonal e simétrica.

Demonstração. Como $A^T = A$, segue que $(H_m)^T = (V_m^T A V_m)^T = V_m^T A V_m = H_m$, ou seja, H_m é simétrica. Como H_m também é Hessenberg, ou seja, $h_{ij} = 0$ para $i > j + 1$, segue que $h_{ji} = h_{ij} = 0$ para $i > j + 1$. Assim, $h_{ij} = 0$ para $|i - j| > 1$, ou seja, H_m é uma matriz tridiagonal. \square

Para descrever o algoritmo de Lanczos, usaremos a seguinte notação:

$$\alpha_j = h_{jj} \text{ e } \beta_j = h_{j-1,j} = h_{j,j-1}$$

e a matriz resultante de H_m é a matriz tridiagonal T_m dada por

$$T_n = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \alpha_{m-1} & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}.$$

Os métodos FOM e GMRES, baseados no método de Arnoldi, possuem versões equivalentes deduzidas do algoritmo de Lanczos simétrico: os métodos de gradientes conjugados (CG) e o método de resíduos conjugados (CR, ou Orthomin(2) em [15]), respectivamente (vide [27, Sec. 6.7-6.8]).

Algoritmo 6 Algoritmo de Lanczos Simétrico

- 1: Escolha um vetor v_1 tal que $\|v_1\|_2 = 1$. Tome $\beta_1 = 0$ e $v_0 = 0$;
 - 2: **Para** $j = 1, 2, \dots, m$ **faça**
 - 3: $w_j = Av_j - \beta_j v_{j-1}$;
 - 4: $\alpha_j = \langle w_j, v_j \rangle$;
 - 5: $\beta_{j+1} = \|w_j\|_2$;
 - 6: **Se** $\beta_{j+1} = 0$ **então**
 - 7: Páre;
 - 8: **Senão**
 - 9: $v_{j+1} = \frac{w_j}{\beta_{j+1}}$;
 - 10: **Fim**
 - 11: **Fim**
-

Capítulo 3

Processo de Biortogonalização de Lanczos

Os métodos de projeção descritos na Seção 2.2.1 têm uma elevada demanda por memória em sistemas lineares de grande porte, pois devemos armazenar a matriz de Hessenberg H_m . Uma alternativa para reduzir esta demanda é considerar $m \ll n$ e utilizar as versões com *restart* dos algoritmos [27]. Nestas versões, calculamos \tilde{x} , redefinimos $x_0 \leftarrow \tilde{x}$, analogamente a (1.11), e repetimos este processo até que a solução \tilde{x} seja satisfatória. Entretanto, este processo costuma convergir lentamente [13].

Vimos no final do capítulo anterior que a matriz de Hessenberg gerada pelo método de Arnoldi se reduz a uma matriz tridiagonal no caso em que A é uma matriz simétrica, logo o problema de demanda por memória não se aplica a este caso.

O algoritmo de Biortogonalização de Lanczos é uma extensão do algoritmo de Lanczos simétrico para matrizes não simétricas que também resulta em uma matriz tridiagonal, mas que não impõe a ortogonalidade entre os vetores-coluna da matriz V_m . Ao invés desta propriedade, impõe-se a biortogonalidade com respeito a um outro conjunto de vetores associado ao espaço de Krylov $K_m(A^T, w_1)$.

3.1 Descrição do Algoritmo

Dados dois vetores iniciais $v_1 \in \mathbb{R}^N$ e $w_1 \in \mathbb{R}^N$ não nulos tais que $\langle v_1, w_1 \rangle \neq 0$, o método de biortogonalização de Lanczos gera duas sequências de vetores $\{v_n\}_{n=1}^m$ e $\{w_n\}_{n=1}^m$ tais que

$$\text{span} \{v_1, v_2, \dots, v_m\} = K_m(A, v_1), \quad (3.1)$$

$$\text{span} \{w_1, w_2, \dots, w_m\} = K_m(A^T, w_1), \quad (3.2)$$

e

$$\langle v_i, w_j \rangle = \begin{cases} \delta_i \neq 0 & \text{se } i = j \\ 0 & \text{se } i \neq j. \end{cases} \quad (3.3)$$

A construção desses vetores v_n e w_n é baseada nas seguintes fórmulas de recursão com três termos:

$$v_{n+1} = Av_n - \alpha_n v_n - \beta_n v_{n-1}, \quad (3.4)$$

$$w_{n+1} = A^T w_n - \tilde{\alpha}_n w_n - \tilde{\beta}_n w_{n-1}, \quad (3.5)$$

sendo que as constantes α_n , β_n , $\tilde{\alpha}_n$ e $\tilde{\beta}_n$ são escolhidas impondo que a condição de biortogonalidade (3.3) seja válida. Para $n = 1$, tomaremos $v_0 = w_0 = 0$.

Vamos determinar as constantes α_n e β_n em (3.4). Segue da condição (3.3) que $\langle v_{j+1}, w_j \rangle = 0$, logo

$$\begin{aligned} 0 = \langle v_{j+1}, w_j \rangle &= \langle Av_j - \alpha_j v_j - \beta_j v_{j-1}, w_j \rangle \\ &= \langle Av_j, w_j \rangle - \alpha_j \langle v_j, w_j \rangle - \beta_j \underbrace{\langle v_{j-1}, w_j \rangle}_0, \end{aligned}$$

ou seja, $\langle Av_j, w_j \rangle - \alpha_j \delta_j = 0$. Como $\delta_j \neq 0$, segue que

$$\alpha_j = \frac{\langle Av_j, w_j \rangle}{\delta_j}. \quad (3.6)$$

Agora, levando-se em conta que $\langle v_{j+1}, w_i \rangle = 0$ para $i < j$, temos:

$$\begin{aligned} 0 = \langle v_{j+1}, w_i \rangle &= \langle Av_j, w_i \rangle - \alpha_j \underbrace{\langle v_j, w_i \rangle}_0 - \beta_j \langle v_{j-1}, w_i \rangle \\ &= \langle v_j, A^T w_i \rangle - \beta_j \langle v_{j-1}, w_i \rangle \\ &= \langle v_j, w_{i+1} + \tilde{\alpha}_i w_i + \tilde{\beta}_i w_{i-1} \rangle - \beta_j \langle v_{j-1}, w_i \rangle \\ &= \langle v_j, w_{i+1} \rangle + \tilde{\alpha}_i \underbrace{\langle v_j, w_i \rangle}_0 + \tilde{\beta}_i \underbrace{\langle v_j, w_{i-1} \rangle}_0 - \beta_j \langle v_{j-1}, w_i \rangle \\ &= \langle v_j, w_{i+1} \rangle - \beta_j \langle v_{j-1}, w_i \rangle. \end{aligned}$$

Se $i < j - 1$, então todos os produtos internos na expressão acima são nulos. Se $i = j - 1$, então teremos:

$$0 = \langle v_j, w_j \rangle - \beta_j \langle v_{j-1}, w_{j-1} \rangle = \delta_j - \beta_j \delta_{j-1}.$$

Como $\delta_{j-1} \neq 0$, segue que

$$\beta_j = \frac{\delta_j}{\delta_{j-1}}. \quad (3.7)$$

Por raciocínio análogo ao anterior, determinaremos agora $\tilde{\alpha}_n$ e $\tilde{\beta}_n$ em (3.5). Temos

novamente de (3.3) que $\langle v_i, w_{j+1} \rangle = 0$ para $i \leq j$. Para $i = j$, temos:

$$\begin{aligned} 0 = \langle v_j, w_{j+1} \rangle &= \langle v_j, A^T w_j - \tilde{\alpha}_j w_j - \beta_j w_{j-1} \rangle \\ &= \langle v_j, A^T w_j \rangle - \tilde{\alpha}_j \langle v_j, w_j \rangle - \underbrace{\tilde{\beta}_j \langle v_j, w_{j-1} \rangle}_0, \end{aligned}$$

de modo que $\langle v_j, A^T w_j \rangle - \tilde{\alpha}_j \delta_j = 0$. Como $\delta_j \neq 0$, segue que

$$\tilde{\alpha}_j = \frac{\langle v_j, A^T w_j \rangle}{\delta_j} = \frac{\langle Av_j, w_j \rangle}{\delta_j}. \quad (3.8)$$

Agora, tomando $i < j$, obtemos

$$\begin{aligned} 0 = \langle v_i, w_{j+1} \rangle &= \langle v_i, A^T w_j \rangle - \tilde{\alpha}_i \underbrace{\langle v_i, w_j \rangle}_0 - \tilde{\beta}_i \langle v_i, w_{j-1} \rangle \\ &= \langle Av_i, w_j \rangle - \tilde{\beta}_i \langle v_i, w_{j-1} \rangle \\ &= \langle v_{i+1} + \alpha_i v_i + \beta_i v_{i-1}, w_j \rangle - \tilde{\beta}_i \langle v_i, w_{j-1} \rangle \\ &= \langle v_{i+1}, w_j \rangle + \alpha_i \underbrace{\langle v_i, w_j \rangle}_0 + \beta_i \underbrace{\langle v_{i-1}, w_j \rangle}_0 - \tilde{\beta}_i \langle v_i, w_{j-1} \rangle \\ &= \langle v_{i+1}, w_j \rangle - \tilde{\beta}_i \langle v_i, w_{j-1} \rangle. \end{aligned}$$

Se $i < j - 1$, então os produtos internos na expressão acima são nulos. Se $i = j - 1$, então teremos:

$$0 = \langle v_j, w_j \rangle - \tilde{\beta}_j \langle v_{j-1}, w_{j-1} \rangle = \delta_j - \tilde{\beta}_j \delta_{j-1}.$$

Como $\delta_{j-1} \neq 0$, segue que

$$\tilde{\beta}_j = \frac{\delta_j}{\delta_{j-1}}. \quad (3.9)$$

Por (3.7), (3.6), (3.8) e (3.9), segue que

$$\alpha_n = \tilde{\alpha}_n = \frac{\langle Av_n, w_n \rangle}{\delta_n} \text{ e } \beta_n = \tilde{\beta}_n = \frac{\delta_n}{\delta_{n-1}}.$$

Algoritmo 7 Algoritmo de Biortogonalização de Lanczos

- 1: Escolha dois vetores $v_1 \in \mathbb{R}^N$ e $w_1 \in \mathbb{R}^N$ tais que $\langle v_1, w_1 \rangle \neq 0$
 - 2: Sejam $\delta_0 = 1$ e $w_0 = v_0 = 0$;
 - 3: **Para** $n = 1, 2, \dots, N$ **faça**
 - 4: $\delta_n = \langle v_n, w_n \rangle$. Se $\delta_n = 0$, páre.
 - 5: $\alpha_n = \langle Av_n, w_n \rangle / \delta_n$;
 - 6: $\beta_n = \delta_n / \delta_{n-1}$;
 - 7: $v_{n+1} = Av_n - \alpha_n v_n - \beta_n v_{n-1}$;
 - 8: $w_{n+1} = A^T w_n - \alpha_n w_n - \beta_n w_{n-1}$;
 - 9: **Fim**
-

Se $\delta_n \neq 0$ para $1 \leq n \leq m$ na linha 4 do algoritmo, então os vetores v_i e w_i ($i = 1, \dots, m$) gerados pelo algoritmo satisfazem (3.3) por construção, ou seja, formam um sistema biortogonal. Daí vem o nome para esse algoritmo.

Os vetores v_1, v_2, \dots, v_m gerados pelo algoritmo de Lanczos são chamados *vetores de Lanczos à direita* e w_1, w_2, \dots, w_m , *vetores de Lanczos à esquerda*. Na proposição a seguir veremos que esses vetores satisfazem (3.1)-(3.2) e considerando a matriz tridiagonal T_m dada por

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & \cdots & 0 \\ 1 & \alpha_2 & \beta_3 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & 1 & \alpha_{m-1} & \beta_m \\ 0 & \cdots & 0 & 1 & \alpha_m \end{pmatrix}, \quad (3.10)$$

mostraremos que valem algumas relações similares às que mostramos para o Método de Arnoldi na Proposição 2.4.

Proposição 3.1 *Se $\delta_n \neq 0$ para $1 \leq n \leq m$ na linha 4 do algoritmo, então os vetores v_i , $i = 1, \dots, m$ formam uma base de $K_m(A, v_1)$ e os vetores w_i , $i = 1, \dots, m$ formam uma base de $K_m(A^T, w_1)$.*

Além disso, sendo V_m a matriz $n \times m$ com vetores coluna v_1, \dots, v_m ; W_m a matriz $n \times m$ com vetores coluna w_1, \dots, w_m ; T_m a matriz tridiagonal definida em (3.10) e $D^{(m)} = \text{diag}(\delta_1, \delta_2, \dots, \delta_m)$, valem as seguintes relações:

$$AV_m = V_m T_m + v_{m+1} e_m^T, \quad (3.11)$$

$$A^T W_m = W_m T_m^T + w_{m+1} e_m^T, \quad (3.12)$$

$$W_m^T AV_m = D^{(m)} T_m. \quad (3.13)$$

Demonstração. Mostremos que $\{v_1, v_2, \dots, v_m\}$ é base de $K_m(A, v_1)$. De fato, sejam $\alpha_1, \alpha_2, \dots, \alpha_m$ constantes tais que

$$\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_m v_m = 0.$$

Tomando o produto interno em ambos os lados por w_i ($1 \leq i \leq m$), encontramos

$$\begin{aligned} \langle \alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_m v_m, w_i \rangle &= 0 \\ \alpha_1 \underbrace{\langle v_1, w_i \rangle}_0 + \alpha_2 \underbrace{\langle v_2, w_i \rangle}_0 + \dots + \alpha_i \underbrace{\langle v_i, w_i \rangle}_{\delta_i} + \dots + \alpha_m \underbrace{\langle v_m, w_i \rangle}_0 &= 0 \\ \alpha_i \delta_i &= 0, \end{aligned}$$

e como $\delta_i \neq 0$, temos $\alpha_i = 0$ para $i = 1, \dots, m$. Logo, $\{v_1, v_2, \dots, v_m\}$ é LI, e como

$\dim K_m(A, v_1) \leq m$, resta mostrarmos que $v_n \in K_m(A, v_1) \forall n = 1, \dots, m$. Para tanto, mostraremos que $v_n \in K_n(A, v_1) \forall n$ e daí, como $K_n(A, v_1) \subseteq K_m(A, v_1) \forall n \leq m$, segue que $v_n \in K_m(A, v_1) \forall n \leq m$.

A prova será feita por indução. Para $n = 1$, temos $v_1 \in K_1(A, v_1)$ trivialmente. Supondo que $v_i \in K_i(A, v_1)$ para todo $i \leq n$, mostremos que $v_{n+1} \in K_{n+1}(A, v_1)$. Como $v_n \in K_n(A, v_1)$, temos que

$$\begin{aligned} v_n &= \alpha_0 v_1 + \alpha_1 A v_1 + \dots + \alpha_{n-1} A^{n-1} v_1 \\ A v_n &= \alpha_0 A v_1 + \alpha_1 A^2 v_1 + \dots + \alpha_{n-1} A^n v_1 \in K_{n+1}(A, v_1). \end{aligned} \quad (3.14)$$

Além disso,

$$\alpha_n v_n \in K_n(A, v_1) \subseteq K_{n+1}(A, v_1), \quad (3.15)$$

$$\beta_n v_{n-1} \in K_{n-1}(A, v_1) \subseteq K_{n+1}(A, v_1). \quad (3.16)$$

Segue de (3.4) que $v_{n+1} \in K_{n+1}(A, v_1)$.

Mostremos agora as relações (3.11)-(3.13):

- $AV_m = V_m T_m + v_{m+1} e_m^T$:

Da linha 7 do algoritmo, temos:

$$A v_n = v_{n+1} + \alpha_n v_n + \beta_n v_{n-1}, \quad n = 1, 2, \dots, m.$$

Note que os elementos da matriz T_m podem ser escritos da seguinte forma:

$$t_{ij} = \begin{cases} \alpha_j, & \text{se } i = j \\ \beta_j & \text{se } i = j - 1 \\ 1 & \text{se } i = j + 1 \\ 0, & \text{se } i < j - 1 \text{ ou } i > j + 1. \end{cases}$$

Assim, voltando na expressão acima, encontramos

$$A v_j = \begin{cases} \sum_{k=1}^m v_k t_{kj}, & j < m \\ \sum_{k=1}^m v_k t_{kj} + v_{j+1}, & j = m. \end{cases}$$

Sejam $C = AV_m$, $B = V_m$ e $[v_k]_i$ a posição i do vetor v_k .

Se $j < m$:

$$C_{ij} = [Av_j]_i = \sum_{k=1}^m [v_k]_i t_{kj} = \sum_{k=1}^m B_{ik} t_{kj}. \quad (3.17)$$

Se $j = m$:

$$C_{im} = [Av_m]_i = \sum_{k=1}^m [v_k]_i t_{km} + [v_{m+1}]_i = \sum_{k=1}^m B_{ik} t_{km} + [v_{m+1}]_i. \quad (3.18)$$

Definindo

$$Z_{ij} = \begin{cases} 0, & \text{se } 1 \leq i \leq m \text{ e } 1 \leq j \leq m-1 \\ [v_{m+1}]_i & \text{se } j = m, \end{cases}$$

podemos escrever (3.17)-(3.18) como sendo

$$C_{ij} = \sum_{k=1}^m B_{ik} t_{kj} + Z_{ij}, \quad i, j = 1, \dots, m,$$

ou na forma matricial, $C = V_m T_m + Z = V_m T_m + v_{m+1} e_m^T$.

- $A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T$:

Temos de (3.5) que

$$A^T w_j = w_{j+1} + \alpha_j w_j + w_{j-1}, \quad j = 1, 2, \dots, m.$$

Escrevendo os elementos da matriz T_m^T como

$$t_{ij} = \begin{cases} \alpha_j, & \text{se } i = j \\ 1 & \text{se } i = j - 1 \\ \beta_{j+1} & \text{se } i = j + 1 \\ 0, & \text{se } i < j - 1 \text{ ou } i > j + 1, \end{cases}$$

a demonstração da relação (3.13) torna-se análoga à anterior.

- $W_m^T A V_m = D^{(m)} T_m$:

Para mostrarmos essa relação, partiremos da relação (3.13), que já foi demonstrada. Multiplicando ambos os lados de (3.13) por W_m^T , temos:

$$W_m^T A V_m = W_m^T V_m T_m + W_m^T v_{m+1} e_m^T.$$

Da condição de biortogonalidade (3.3), segue que $W_m^T V_m = D^{(m)} = \text{diag}(\delta_1, \delta_2, \dots, \delta_n)$.

Voltando na equação acima e usando as linhas 5 e 7 do algoritmo, temos:

$$\begin{aligned}
 W_m^T A V_m &= D^{(m)} T_m + W_m^T v_{m+1} e_m^T \\
 &= D^{(m)} T_m + W_m^T (A v_m - \alpha_m v_m - \beta_m v_{m-1}) e_m^T \\
 &= D^{(m)} T_m + (W_m^T A v_m) e_m^T - \alpha_m (W_m^T v_m) e_m^T - \beta_m (W_m^T v_{m-1}) e_m^T \\
 &= D^{(m)} T_m + w_m^T A v_m - \alpha_m w_m^T A v_m - w_m^T v_{m-1} \\
 &= D^{(m)} T_m + \alpha_m \delta_m - \alpha_m \delta_m - 0 = D^{(m)} T_m.
 \end{aligned}$$

□

A seguir, apresentamos um exemplo em que aplicamos o Algoritmo de Biortogonalização de Lanczos.

Exemplo 3.2 Sejam $A = \begin{bmatrix} 3 & 2 & 10 & 6 & 2 & 9 \\ 4 & 4 & 10 & 2 & 6 & 2 \\ 6 & 10 & 6 & 9 & 8 & 1 \\ 7 & 5 & 1 & 7 & 10 & 8 \\ 6 & 9 & 4 & 9 & 3 & 3 \\ 7 & 2 & 6 & 3 & 4 & 7 \end{bmatrix}$, $e v_1 = w_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}^T$.

Ao implementarmos o algoritmo de Biortogonalização de Lanczos, obtemos os seguintes resultados:

```

delta1 = 1.00000      delta4 = 5390.9
delta2 = 200.52      delta5 = -8.7818e+006
delta3 = 4345.6      delta6 = -1.6075e+007
    
```

V6 =

```

0.1048    10.323    -22.685    -171.91    7209.8    -1591.6
0.2097     4.0826     21.39     453.22    -18636    -438.71
0.3145     4.8659     51.18    -31.975    1460.6    -1083.4
0.4193     4.1816    -55.935    -170.46    6077.9     7118.9
0.5241    -2.8972     39.97    -214.74    9853.6    -4723.2
0.629     -5.8877    -24.957     186.15    -7983.2     143.28
    
```

W6 =

```

0.10483    10.742    -35.654    -84.261    3274.8     2539.6
0.20966     6.7033    -14.8     308.76    -11777    -2695.5
0.31449     2.979     74.843    -391.62    15251     2389.6
    
```

0.41931	2.2947	-7.533	178.09	-7059.9	-3215.9
0.52414	-1.6392	-15.395	348.78	-13786	-2490.9
0.62897	-5.678	-8.694	-302.44	11949	3500.1

Como $\delta_n \neq 0$ para $1 \leq n \leq 6$, pela Proposição (3.1) segue que os vetores colunas de V_6 formam uma base para $K_5(A, v_1)$ e os vetores colunas de W_6 formam uma base para $K_5(A, v_1)$.

Ao calcularmos a matriz $W_6^T AV_6$, obtemos uma matriz não singular dada por

Columns 1 through 4:

26.527	200.52	2.1032e-012	6.5938e-011
200.52	1083.2	4345.6	4.7839e-010
1.5916e-012	4345.6	8727.1	5390.9
4.718e-011	3.3901e-010	5390.9	2.1598e+005
-2.6284e-010	-1.5862e-009	-2.7358e-009	-8.7818e+006
5.7614e-008	4.3205e-007	2.7847e-007	-3.1749e-006

Columns 5 and 6:

-4.6566e-010	7.155e-008
-3.376e-009	5.1893e-007
-4.191e-009	2.665e-007
-8.7818e+006	-3.4204e-006
3.5952e+008	-1.6075e+007
-1.6075e+007	4.922e+007

Veremos a seguir que em geral é necessário que $W_m^T AV_m$ seja não singular em um dos métodos de projeção propostos para a solução de sistema lineares da forma $Ax = b$. Além disso, sabendo que $D^{(6)} = \text{diag}(\delta_1, \delta_2, \dots, \delta_6)$ vamos utilizar a Proposição 3.1 para obter a matriz tridiagonal T_6 :

Columns 1 through 4:

2.6527e+001	2.0052e+002	2.1032e-012	6.5938e-011
1.0000e+000	5.4019e+000	2.1671e+001	2.3857e-012
3.6626e-016	1.0000e+000	2.0082e+000	1.2405e+000
8.7518e-015	6.2886e-014	1.0000e+000	4.0064e+001

2.9931e-017	1.8062e-016	3.1153e-016	1.0000e+000
-3.5841e-015	-2.6877e-014	-1.7323e-014	1.9750e-013

Columns 5 and 6:

-4.6566e-010	7.1550e-008
-1.6836e-011	2.5879e-009
-9.6441e-013	6.1327e-011
-1.6290e+003	-6.3448e-010
-4.0940e+001	1.8305e+000
1.0000e+000	-3.0619e+000

Para fins de comparação, a matriz T_6 segundo a definição (3.10) é dada por

26.527	200.52	0	0	0	0
1	5.4019	21.671	0	0	0
0	1	2.0082	1.2405	0	0
0	0	1	40.064	-1629	0
0	0	0	1	-40.94	1.8305
0	0	0	0	1	-3.0619

3.1.1 Métodos de Projeção Baseados na Biortogonalização de Lanczos

Dado um chute inicial x_0 para solução do sistema linear $Ax = b$, considere um método de projeção que toma $K = K_m(A, v_1)$ e $L = K_m(A^T, w_1)$. Este método nos dá uma solução aproximada \tilde{x} de forma que $\tilde{x} \in x_0 + K$ e $b - A\tilde{x} \perp L$. Assumindo que a matriz $W_m^T A V_m$ é não singular, temos de (1.10) que \tilde{x} é dada por:

$$\tilde{x} = x_0 + V_m (W_m^T A V_m)^{-1} W_m^T r_0, \quad (3.19)$$

sendo $r_0 = b - Ax_0$ é o resíduo inicial.

Tomando $\beta = \|r_0\|$ e $v_1 = r_0/\|r_0\|$, e usando a condição de biortogonalidade (3.3), temos

$$W_m^T r_0 = W_m^T (\beta v_1) = \beta W_m^T (v_1) = \beta \delta_1 e_1. \quad (3.20)$$

Por (3.19), (3.13) e (3.20), a solução aproximada no subespaço $x_0 + K_m(A, v_1)$ é dada por

$$\tilde{x} = x_0 + V_m (D^{(m)} T_m)^{-1} (\beta \delta_1 e_1),$$

ou, equivalentemente,

$$\begin{cases} \tilde{x} = x_0 + V_m y_m, \\ y_m = (D^{(m)} T_m)^{-1} (\beta \delta_1 e_1). \end{cases} \quad (3.21)$$

A implementação do cálculo das equações (3.21) baseadas na decomposição LU do produto $D^{(m)} T_m$ (vide [27, Sec. 7.3.1]) conduz ao método de gradientes biconjugados (BCG).

Por outro lado, em analogia com o GMRES, vamos considerar $K = K_m(A, v_1)$ e $L = AK_m(A, v_1)$. Neste caso, buscamos $\tilde{x} = x_0 + V_m \tilde{y}$ tal que $b - A\tilde{x} \perp AK_m(A, v_1)$, o que corresponde a encontrar $\tilde{y} \in \mathbb{R}^m$ que minimiza o funcional $J(y) = \|b - Ax\|_2$.

Tomando novamente $v_1 = r_0/\beta$, segue da Proposição 3.1 que

$$\begin{aligned} b - Ax &= r_0 - AV_m y \\ &= \beta v_1 - V_{m+1} \bar{T}_m y = V_{m+1} (\beta e_1 - \bar{T}_m y), \end{aligned}$$

sendo \bar{T}_m a matriz $m+1 \times m$ tal que $V_m T_m + v_{m+1} e_m^T = V_{m+1} \bar{T}_m$. Entretanto, temos em geral que $V_{m+1}^T V_{m+1} \neq I$, ou seja

$$J(y)^2 = \|b - Ax\|_2^2 = \|V_{m+1} (\beta e_1 - \bar{T}_m y)\|_2^2 \neq \|\beta e_1 - \bar{T}_m y\|_2^2.$$

Podemos entretanto abrir mão de minimizar $J(y)$ e escolher o vetor \tilde{y} que minimiza o funcional “quase” resíduo

$$\tilde{J}(y) = \|\beta e_1 - \bar{T}_m y\|_2. \quad (3.22)$$

Esta é a filosofia do método de resíduo quase-minimal (QMR). A vantagem deste método sobre o método BCG é que não precisamos assumir que a matriz $W_m^T AV_m$ seja não-singular. Além disso, a matriz $T \bar{T}_m$ tem posto completo [11], de modo que o sistema de equações normais associado ao problema de minimização de (3.22),

$$\bar{T}_m^T \bar{T}_m \tilde{y} = \beta \bar{T}_m^T e_1,$$

possui solução única.

Convém observar que, quando a matriz $W_m^T AV_m$ é mal-condicionada, o método BGC apresenta um padrão instável de convergência [11]. De fato, diversos trabalhos relatam oscilações na curva de convergência deste método (vide, por exemplo, [13]).

3.2 Quebras no Algoritmo

Dizemos que o Algoritmo 7 *quebra* quando $\delta_n = 0$ na linha 4. Neste caso, o coeficiente α_n envolve uma divisão por zero, impossibilitando o cálculo dos novos vetores

v_{n+1} e w_{n+1} .

Podemos também ter valores para δ_n muito próximos de zeros, que chamamos de *quase-quebras* (near-breakdowns). O que ocorre nestes casos é que para obtermos as constantes α_n e β_n , aparecem divisões por quantidades muito pequenas e depois de alguns passos, o efeito acumulativo dessas divisões pode introduzir erros de arredondamento grandes.

Seja L o menor inteiro tal que $\delta_{L+1} = \langle v_{L+1}, w_{L+1} \rangle = 0$ na linha 4 no algoritmo. Note que $L \leq N$, já que $\dim \mathbb{R}^N = N$.

Temos que $\langle v_{L+1}, w_{L+1} \rangle = 0$ pode ocorrer de duas maneiras diferentes:

- Caso 1 (*quebra favorável*): $v_{L+1} = 0$ ou $w_{L+1} = 0$.

Neste caso, ou os vetores de Lanczos à direita geram um subespaço A -invariante ou os vetores de Lanczos à esquerda geram um subespaço A^T -invariante:

Se $v_{L+1} = 0$, temos de (3.4) que $Av_L = \alpha_L v_L + \beta_L v_{L-1} \in \text{span} \{v_1, v_2, \dots, v_L\}$, ou seja, $\text{span} \{v_1, v_2, \dots, v_L\} = K_L(v_1, A)$ é invariante sobre A . Neste caso, pela Proposição 1.7, a solução aproximada do sistema linear $Ax = b$ é exata tomando como chute inicial $x_0 = 0$ e $v_1 = b / \|b\|$.

Se $w_{L+1} = 0$, temos de (3.5) que $A^T w_L = \alpha_L w_L + \delta_L w_{L-1} \in \text{span} \{w_1, w_2, \dots, w_L\}$, ou seja, $\text{span} \{w_1, w_2, \dots, w_L\}$ é invariante sobre A^T e novamente a solução do sistema linear $A^T x = b$ é exata tomando como chute inicial $x_0 = 0$ e $v_1 = b / \|b\|$.

- Caso 2 (*quebra séria*): $v_{L+1} \neq 0$ e $w_{L+1} \neq 0$.

Neste caso, os vetores de Lanczos v_1, v_2, \dots, v_L não geram um subespaço A -invariante e w_1, w_2, \dots, w_L não geram um subespaço A^T -invariante de \mathbb{R}^N .

De fato. Temos que (3.3) é satisfeita para $1 \leq i, j \leq L$. Além disso, se $v_{L+1} \neq 0$, é possível mostrarmos que $\langle v_{L+1}, w_i \rangle = 0$ para $i \leq L$ por raciocínio análogo ao feito para encontrarmos as constantes α_n e β_n do algoritmo, mas desta vez substituindo essas constantes no cálculo. Utilizando esses resultados, mostremos, primeiramente, que se $v_{L+1} \neq 0$ e $w_{L+1} \neq 0$, então $\{v_1, v_2, \dots, v_{L+1}\}$ e $\{w_1, w_2, \dots, w_{L+1}\}$ são conjuntos linearmente independentes (LI).

Sejam $\alpha_1, \alpha_2, \dots, \alpha_{L+1}$ escalares tais que $\alpha_1 v_1 + \alpha_2 v_2 + \dots + \alpha_{L+1} v_{L+1} = 0$. Fazendo o produto interno em ambos os lados por w_i , $i = 1, \dots, L$, temos

$$\alpha_1 w_i^T v_1 + \dots + \alpha_i w_i^T v_i + \dots + \alpha_{L+1} w_i^T v_{L+1} = 0 \Rightarrow \alpha_i \delta_i = 0 \Rightarrow \alpha_i = 0, \quad 1 \leq i \leq L.$$

Assim, $\alpha_1 v_1 + \dots + \alpha_L v_L + \alpha_{L+1} v_{L+1} = 0 \Rightarrow \alpha_{L+1} v_{L+1} = 0$. Como $v_{L+1} \neq 0$, segue que $\alpha_{L+1} = 0$ e portanto, $\{v_1, v_2, \dots, v_{L+1}\}$ é LI. De forma análoga, mostra-se que $\{w_1, w_2, \dots, w_{L+1}\}$ é LI.

Temos de (3.4) que $Av_L = v_{L+1} + \alpha_L v_L + \beta_L v_{L-1}$. Como $\{v_1, v_2, \dots, v_{L+1}\}$ é LI, segue que $v_{L+1} \notin \text{span}\{v_1, v_2, \dots, v_L\}$ e então, $Av_L \notin \text{span}\{v_1, v_2, \dots, v_L\}$, isto é, v_1, v_2, \dots, v_L não geram um subespaço A -invariante. De forma análoga, mostra-se que w_1, w_2, \dots, w_L não geram um subespaço A^T -invariante de \mathbb{R}^N .

A seguir, vamos denotar por $L_r = L_r(v_1, A)$ e $L_l = L_l(w_1, A^T)$ o grau de v_1 com respeito a A e o grau de w_1 com respeito a A^T , respectivamente.

Lema 3.3 *Seja $\delta_{L+1} = \langle v_{L+1}, w_{L+1} \rangle = 0$ uma quebra favorável no algoritmo de Biortogonalização de Lanczos. Valem as seguintes relações:*

$$\begin{aligned} v_{L+1} = 0 &\Rightarrow L = L_r, \\ w_{L+1} = 0 &\Rightarrow L = L_l. \end{aligned}$$

Demonstração. Como o algoritmo não quebra antes do passo $L + 1$, pela Proposição 3.1, temos que $\{v_1, v_2, \dots, v_L\}$ formam uma base para $K_L(A, v_1)$, com $v_j \in K_j(A, v_1)$, $1 \leq j \leq L$, conforme visto na demonstração dessa proposição.

Suponha que ocorra uma quebra favorável no algoritmo, com $v_{L+1} = 0$. Então, $\{v_1, \dots, v_L\}$ é A -invariante. Assim, $A^L v_1 \in \text{span}\{v_1, \dots, v_L\}$. Como $v_j \in K_j(A, v_1)$ para $1 \leq j \leq L$, temos

$$A^L v_1 = \sum_{i=0}^{L-1} \alpha_i A^i v_1 \Rightarrow \left(A^L - \sum_{i=0}^{L-1} \alpha_i A^i \right) v_1 = 0,$$

ou seja,

$$p(A)v_1 = 0, \quad p(x) = x^L - \sum_{i=0}^{L-1} \alpha_i x^i.$$

Como p é mônico e $p(A)v_1 = 0$, segue que o grau do polinômio minimal de v_1 com respeito a A é no máximo L , ou seja, $L_r \leq L$. Por outro lado, como $\{v_1, v_2, \dots, v_L\}$ formam uma base para $K_L(A, v_1)$, temos $\dim K_L(A, v_1) = L$ e então, pela Proposição 2.2, segue que $L_r \geq L$. Portanto, $L_r = L$. De forma análoga, mostra-se que se a quebra for favorável, com $w_{L+1} = 0$, então $L = L_l$. \square

Lema 3.4 *Valem as seguintes relações:*

$$\begin{aligned} v_{L+1} = 0 \text{ e } w_{L+1} \neq 0 &\Rightarrow L_l > L_r, \\ v_{L+1} \neq 0 \text{ e } w_{L+1} = 0 &\Rightarrow L_l < L_r. \end{aligned}$$

Demonstração. Suponha $v_{L+1} = 0$ e $w_{L+1} \neq 0$. Como $v_{L+1} = 0$, pela proposição anterior temos que $L = L_r$. Vimos também que se $w_{L+1} \neq 0$, então $\{w_1, w_2, \dots, w_{L+1}\}$ é LI, donde $\dim K_N(A^T, w_1) \geq L + 1$. Por outro lado, pela Proposição 2.2, $\dim K_N(A^T, w_1) =$

$\min \{N, L_l\} \leq L_l$. Assim,

$$L_l \geq \dim K_N(A^T, w_1) \geq L + 1 = L_r + 1 > L_r.$$

Analogamente, mostra-se que se $v_{L+1} \neq 0$ e $w_{L+1} = 0$, então $L_l < L_r$. □

Proposição 3.5 *Se ocorre uma quebra favorável no algoritmo de Biortogonalização de Lanczos, então*

$$L = L_* = \min \{L_l, L_r\}.$$

Se a quebra for séria, então

$$L < L_* = \min \{L_l, L_r\}.$$

Demonstração. Se ocorre uma quebra favorável no algoritmo de Biortogonalização de Lanczos, então $\langle v_{L+1}, w_{L+1} \rangle = 0$ pode ocorrer de duas maneiras: ou ambos os vetores são nulos ou apenas um deles é nulo. Se ocorrer o primeiro caso, então, pelo Lema 3.3, temos $L_l = L_r$, donde $L = L_* = \min \{L_l, L_r\}$. Por outro lado, se ocorrer o segundo caso, temos duas possibilidades:

- $v_{L+1} = 0$ e $w_{L+1} \neq 0$: Neste caso, pelo Lema 3.3, temos $L = L_r$ e pelo Lema 3.4, $L_l > L_r$. Assim, $L_* = \min \{L_l, L_r\} = L_r = L$.
- $v_{L+1} \neq 0$ e $w_{L+1} = 0$: Neste caso, pelo Lema 3.3, temos $L = L_l$ e Lema 3.4, $L_l < L_r$. Assim, $L_* = \min \{L_l, L_r\} = L_l = L$.

Portanto, na quebra favorável ocorre $L = L_* = \min \{L_l, L_r\}$. Suponha agora que ocorra uma quebra séria, ou seja, $v_{L+1} \neq 0$ e $w_{L+1} \neq 0$. Vimos que neste caso, $\{v_1, \dots, v_{L+1}\}$ e $\{w_1, \dots, w_{L+1}\}$ são LI. Além disso, pela demonstração da Proposição 3.1, temos que

$$v_j \in K_L(A, v_1) \subset K_{L+1}(A, v_1) \quad \forall 1 \leq j \leq L,$$

já que $K_L(A, v_1) \subseteq K_{L+1}(A, v_1)$. Obtendo para v_L expressões semelhantes a (3.14)-(3.15), segue de (3.4) que $v_{L+1} \in K_{L+1}$. Assim, $\{v_1, \dots, v_{L+1}\}$ é base para $K_{L+1}(A, v_1)$. Logo, $\dim K_{L+1}(A, v_1) = L + 1$ e pela Proposição 2.2, segue que $L_r \geq L + 1 > L$.

Pode-se mostrar de forma análoga que $\{w_1, \dots, w_{L+1}\}$ é base para $K_{L+1}(A^T, v_1)$, de modo que $\dim K_{L+1}(A^T, v_1) = L + 1 \Rightarrow L_l \geq L + 1 > L$. Assim,

$$\begin{cases} L_r \geq L + 1 > L \\ L_l \geq L + 1 > L, \end{cases}$$

ou seja, $L < L_* = \min \{L_l, L_r\}$. □

As quebras sérias, ou no caso da aritmética de precisão finita, as *quase-quebras* sérias, colocaram o algoritmo de Lanczos não simétrico em descrédito [12]. No entanto, pelo processo *look ahead*, que veremos adiante, é possível “pular” a iteração em que o algoritmo quebra.

Capítulo 4

Algoritmo de Lanczos com Estratégia *Look Ahead*

Para descrevermos a idéia básica do algoritmo Lanczos *look ahead*, usaremos uma conexão entre o processo de Lanczos e os polinômios ortogonais formais (FOPs).

Da definição de subespaço de Krylov (2.1), note que podemos escrever:

$$\begin{aligned}K_n(v_1, A) &= \{\phi(A)v_1/\phi \in P_{n-1}\}, \\K_n(w_1, A^T) &= \{\phi(A^T)w_1/\phi \in P_{n-1}\}.\end{aligned}$$

Isto nos diz que os vetores de um subespaço de Krylov K_n estão associados a polinômios de grau no máximo $n - 1$. Veremos na proposição a seguir que, em particular, os vetores de Lanczos v_n e w_n do algoritmo podem ser escritos em termos de um mesmo polinômio de grau $n - 1$.

Proposição 4.1 *Sejam v_n e w_n vetores gerados pelo algoritmo de biortogonalização de Lanczos. Existe um único polinômio mônico $\phi_{n-1} \in P_{n-1}$ tal que*

$$v_n = \phi_{n-1}(A)v_1 \text{ e } w_n = \phi_{n-1}(A^T)w_1. \quad (4.1)$$

Demonstração. A prova será feita por indução. Para $n = 1$, temos $v_1 = \phi_0(A)v_1$, com $\phi_0(A) = 1$.

Suponha que $v_i = \phi_{i-1}(A)v_1$ para $i = 2, 3, \dots, n - 1$, com ϕ_{i-1} mônico de grau $i - 1$. Mostremos que $v_n = \phi_{n-1}(A)v_1$. Temos de (3.4) que

$$\begin{aligned}v_n &= Av_{n-1} - \alpha_{n-1}v_{n-1} - \beta_{n-1}v_{n-2} \\&= A(\phi_{n-2}(A)v_1) - \alpha_{n-1}(\phi_{n-2}(A)v_1) - \beta_{n-1}(\phi_{n-3}(A)v_1) \\&= \phi_{n-1}(A)v_1 - \alpha_{n-1}(\phi_{n-2}(A)v_1) - \beta_{n-1}(\phi_{n-3}(A)v_1) \\&= \tilde{\phi}_{n-1}(A)v_1.\end{aligned}$$

Note que $\tilde{\phi}_{n-1}(A)v_1$ é mônico de grau $n-1$, já que $\phi_{n-2}(A)v_1$ é um polinômio mônico de grau $n-2$ e quando multiplicado por A , continua mônico e passa a ter grau $n-1$. Analogamente, mostra-se que $w_n = \phi_{n-1}(A^T)w_1$.

Mostremos agora a unicidade de $\phi_{n-1} \in P_{n-1}$. Suponha que ϕ_{n-1} não seja único. Então, $v_n = \phi_{n-1}(A)v_1$ e $v_n = \tilde{\phi}_{n-1}(A)v_1$, com ϕ_{n-1} e $\tilde{\phi}_{n-1}$ mônicos. Assim,

$$\begin{aligned} v_n &= \phi_{n-1}(A)v_1 = \alpha_0 v_1 + \alpha_1 A v_1 + \dots + \alpha_{n-2} A^{n-2} v_1 + A^{n-1} v_1, \\ v_n &= \tilde{\phi}_{n-1}(A)v_1 = \beta_0 v_1 + \beta_1 A v_1 + \dots + \beta_{n-2} A^{n-2} v_1 + A^{n-1} v_1. \end{aligned}$$

Igualando as expressões acima, obtemos:

$$\begin{aligned} v_n &= \alpha_0 v_1 + \dots + \alpha_{n-2} A^{n-2} v_1 + A^{n-1} v_1 = \beta_0 v_1 + \dots + \beta_{n-2} A^{n-2} v_1 + A^{n-1} v_1 \\ \Rightarrow & (\alpha_0 - \beta_0) v_1 + (\alpha_1 - \beta_1) A v_1 + \dots + (\alpha_{n-2} - \beta_{n-2}) A^{n-2} v_1 = 0. \end{aligned} \quad (4.2)$$

Por outro lado, como o algoritmo não quebra antes do passo $L+1$, segue da Proposição 3.1 que $\{v_1, \dots, v_L\}$ formam base de $K_L(v_1, A) = \text{span}\{v_1, Av_1, \dots, A^{L-1}v_1\}$, assim, $\dim K_L(v_1, A) = L$ e então, $\{v_1, Av_1, \dots, A^{L-1}v_1\}$ é LI.

Em particular, $\{v_1, Av_1, \dots, A^{n-2}v_1\}$ é LI, logo temos de (4.2) que $\alpha_j - \beta_j = 0$, $0 \leq j \leq n-2$, isto é, $\phi_{n-1} = \tilde{\phi}$. \square

Note que a proposição acima vale para todos os vetores gerados pelo algoritmo, o que inclui os vetores v_{L+1} e w_{L+1} , que não são vetores de Lanczos.

Considere a seguinte forma bilinear

$$\langle \Phi, \psi \rangle = (\Phi(A^T)w_1)^T (\psi(A)v_1) = w_1^T \Phi(A)\psi(A)v_1. \quad (4.3)$$

A seguir, mostraremos que a partir dessa forma bilinear, podemos reescrever a condição de biortogonalidade (3.3) em função dos polinômios mônicos ϕ_i associados aos vetores v_i segundo a relação (4.1). Para tanto, enunciaremos primeiro um resultado que será usado na proposição a seguir.

Lema 4.2 *Sejam ϕ_k , $0 \leq k \leq j$ polinômios mônicos de grau k . O conjunto $\{\phi_0, \dots, \phi_j\}$ é uma base para P_j .*

Demonstração. Como $\dim P_j = j+1$, basta mostrarmos que $\{\phi_0, \dots, \phi_j\}$ é LI.

Se $j=0$, temos $\phi_0 = 1$ e então ϕ_0 é base para P_0 . Suponha $j \neq 0$ e sejam $\alpha_0, \alpha_1, \dots, \alpha_j$ tais que

$$f(x) = \alpha_0 \phi_0(x) + \alpha_1 \phi_1(x) + \dots + \alpha_j \phi_j(x) = 0.$$

Como ϕ_k é mônico para $0 \leq k \leq j$, temos que o coeficiente do termo x^k é 1. Assim,

$$f(x) = 0 \Rightarrow f'(x) = 0 \Rightarrow \tilde{\phi}_{j-2}(x) + j\alpha_j x^{j-1} = 0, \text{ com } \tilde{\phi}_{j-2} \in P_{j-2}.$$

Como $j \neq 0$, então $\alpha_j = 0$. Assim,

$$f(x) = \alpha_0 \phi_0(x) + \alpha_1 \phi_1(x) + \dots + \alpha_{j-1} \phi_{j-1}(x) = 0.$$

Repetindo recursivamente esse processo j vezes, obtemos $\alpha_k = 0$, $0 \leq k \leq j$ e portanto, $\{\phi_0, \dots, \phi_j\}$ é LI. \square

Proposição 4.3 *A condição de biortogonalidade*

$$\langle v_j, w_i \rangle = \begin{cases} \delta_i \neq 0, & i = j \\ 0, & i \neq j \end{cases}$$

pode ser reescrita em termos dos polinômios ϕ_i definidos em (4.1) da seguinte forma:

$$\langle \phi_{i-1}, \phi \rangle = 0, \quad \forall \phi \in P_{i-2}, \quad (4.4)$$

$$\langle \phi_{i-1}, \phi_{i-1} \rangle = \delta_i \neq 0, \quad 1 \leq i, j \leq n. \quad (4.5)$$

Demonstração. De fato, segue da Proposição 4.1 e de (4.1) que existe um polinômio mônico $\phi_{i-1} \in P_{i-1}$ tal que

$$w_i^T v_j = (\phi_{i-1}(A^T) w_1)^T (\phi_{j-1}(A) v_1) = \langle \phi_{i-1}, \phi_{j-1} \rangle, \quad 1 \leq i, j \leq n.$$

Assim, a condição de biortogonalidade equivale a

$$\langle \phi_{i-1}, \phi_{j-1} \rangle = \begin{cases} \delta_i \neq 0 & \text{se } i = j \\ 0 & \text{se } i \neq j. \end{cases}$$

Portanto, se $i = j$, vale (4.5). Mostremos que $\langle \phi_{i-1}, \phi_{j-1} \rangle = 0$ para $i \neq j$ se, e somente se, $\langle \phi_{i-1}, \phi \rangle = 0 \quad \forall \phi \in P_{i-2}$.

(\Rightarrow) Suponha $\langle \phi_{i-1}, \phi_{j-1} \rangle = 0$, com $i \neq j$, e seja $\phi \in P_{i-2}$ arbitrário. Pelo Lema 4.2, existem escalares $\alpha_0, \alpha_1, \dots, \alpha_{i-2}$ tais que $\phi = \alpha_0 \phi_0 + \alpha_1 \phi_1 + \dots + \alpha_{i-2} \phi_{i-2}$. Então,

$$\begin{aligned} \langle \phi_{i-1}, \phi \rangle &= \langle \phi_{i-1}, \alpha_0 \phi_0 + \alpha_1 \phi_1 + \dots + \alpha_{i-2} \phi_{i-2} \rangle \\ &= \alpha_0 \langle \phi_{i-1}, \phi_0 \rangle + \dots + \alpha_{i-2} \langle \phi_{i-1}, \phi_{i-2} \rangle = 0. \end{aligned}$$

Portanto, $\langle \phi_{i-1}, \phi \rangle = 0 \quad \forall \phi \in P_{i-2}$.

(\Leftarrow) Suponha que $\langle \phi_{i-1}, \phi \rangle = 0 \quad \forall \phi \in P_{i-2}$ para $1 \leq i, j \leq n$, com $i \neq j$.

Se $j < i$, então $j-1 < i-1 \leq i-2$ implica $\phi_{j-1} \in P_{i-2}$. Assim, $\langle \phi_{i-1}, \phi_{j-1} \rangle = 0$ por hipótese.

Se $j > i$, então, pelo caso anterior, $\langle \phi_{j-1}, \phi_{i-1} \rangle = 0$. Como $\langle \phi_{j-1}, \phi_{i-1} \rangle = \langle \phi_{i-1}, \phi_{j-1} \rangle$, segue que $\langle \phi_{i-1}, \phi_{j-1} \rangle = 0$. \square

Definição 4.4 Um polinômio não nulo $\phi_{n-1} \in P_{n-1}$ de grau exato $n - 1$ que cumpre (4.4), isto é, $\langle \phi_{n-1}, \phi \rangle = 0 \quad \forall \phi \in P_{n-2}$, é dito um polinômio ortogonal formal (**FOP**) de grau $n - 1$ com respeito a forma bilinear (4.3).

Note que se $n = 1$ em (4.4), temos $\langle \phi_0, \phi \rangle = 0 \quad \forall \phi \in P_{-1}$. E então, por não existir polinômio de grau negativo, a condição (4.4) é vazia para $n = 1$. Assim, por vacuidade, qualquer $\phi_0 = \gamma_0 \neq 0$ é um FOP de grau 0.

Temos também que os polinômios mônicos únicos $\phi_0, \phi_1, \dots, \phi_{L-1}$ associados aos vetores de Lanczos segundo a Proposição 4.1 são todos FOPs, já que pela Proposição 4.3 esses polinômios satisfazem (4.4).

Na proposição a seguir, veremos uma condição necessária e suficiente para que um polinômio satisfaça a condição (4.4). Em seguida, enunciaremos um corolário que nos dá uma condição para analisarmos quando um polinômio ϕ_{n-1} de grau exato $n - 1$ é FOP.

Proposição 4.5 $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1\lambda + \gamma_2\lambda^2 + \dots + \gamma_{n-1}\lambda^{n-1}$ satisfaz (4.4) se, e somente se, os coeficientes $\gamma_0, \gamma_1, \dots, \gamma_{n-2}$ são solução do sistema linear

$$\begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{n-2} \\ m_1 & \ddots & & \ddots & \vdots \\ m_2 & & \ddots & & \vdots \\ \vdots & \ddots & & & m_{2n-5} \\ m_{n-2} & \cdots & \cdots & m_{2n-5} & m_{2n-4} \end{bmatrix} \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_{n-2} \end{bmatrix} = -\gamma_{n-1} \begin{bmatrix} m_{n-1} \\ m_n \\ m_{n+1} \\ \vdots \\ m_{2n-3} \end{bmatrix}, \quad (4.6)$$

$$m_j = w_1^T A^j v_1 = \langle 1, \lambda^j \rangle, \quad j = 0, 1, \dots \quad (4.7)$$

Demonstração. (\Rightarrow) Suponha que $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1\lambda + \gamma_2\lambda^2 + \dots + \gamma_{n-1}\lambda^{n-1}$ satisfaz (4.4). Usando (4.3), temos

$$\begin{aligned} \langle \phi_{n-1}, \phi \rangle &= 0 \quad \forall \phi \in P_{n-2} \\ w_1^T \phi_{n-1}(A) \phi(A) v_1 &= 0 \quad \forall \phi \in P_{n-2} \\ w_1^T \left(\sum_{j=0}^{n-1} \gamma_j A^j \right) \phi(A) v_1 &= 0 \quad \forall \phi \in P_{n-2}. \end{aligned}$$

Em particular, para $\phi(A) = A^i$, $0 \leq i \leq n-2$,

$$\begin{aligned} \sum_{j=0}^{n-1} \gamma_j w_1^T A^j A^i v_1 &= 0 \\ \gamma_{n-1} w_1^T A^{n-1+i} v_1 + \sum_{j=0}^{n-2} \gamma_j w_1^T A^{j+i} v_1 &= 0 \\ \sum_{j=0}^{n-2} (w_1^T A^{j+i} v_1) \gamma_j &= -(w_1^T A^{n-1+i} v_1) \gamma_{n-1} \\ \sum_{j=0}^{n-2} (m_{j+i}) \gamma_j &= -\gamma_{n-1} (m_{n-1+i}), \quad 0 \leq i \leq n-2, \end{aligned}$$

com $m_k = w_1^T A^k v_1$, $k = 0, 1, 2, \dots$. Portanto, os coeficientes $\gamma_0, \gamma_1, \dots, \gamma_{n-2}$ são uma solução do sistema linear (4.6).

(\Leftarrow) Suponha que os coeficientes $\gamma_0, \gamma_1, \dots, \gamma_{n-2}$ são solução do sistema linear (4.6), ou seja,

$$\sum_{j=0}^{n-1} \gamma_j w_1^T A^j A^i v_1 = 0, \quad 0 \leq i \leq n-2.$$

Mostremos que $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \dots + \gamma_{n-1} \lambda^{n-1}$ satisfaz (4.4). De fato, dado $\phi(A) \in P_{n-2}$,

$$\exists \alpha_0, \alpha_1, \dots, \alpha_{n-2} \in \mathbb{R} \ ; \ \phi(A) = \sum_{i=0}^{n-2} \alpha_i A^i.$$

Assim,

$$\langle \phi_{n-1}, \phi \rangle = w_1^T \phi_{n-1}(A) \phi(A) v_1 = w_1^T \sum_{j=0}^{n-1} \gamma_j A^j \sum_{i=0}^{n-2} \alpha_i A^i v_1 = \sum_{i=0}^{n-2} \alpha_i \underbrace{\left(\sum_{j=0}^{n-1} \gamma_j w_1^T A^j A^i v_1 \right)}_0 = 0$$

Portanto, $\langle \phi_{n-1}, \phi \rangle = 0 \ \forall \phi \in P_{n-2}$. □

Corolário 4.6 *Seja $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1 \lambda + \gamma_2 \lambda^2 + \dots + \gamma_{n-1} \lambda^{n-1}$, com $\gamma^{n-1} \neq 0$. Então, ϕ_{n-1} é FOP de grau $n-1$ se, e somente se, os coeficientes $\gamma_0, \gamma_1, \dots, \gamma_{n-2}$ são solução do sistema linear (4.6)*

Demonstração.

(\Rightarrow) Suponha que ϕ_{n-1} seja FOP de grau $n-1$. Então, pela Definição (4.4), ϕ_{n-1} satisfaz (4.4) e daí, da Proposição 4.5 segue que os coeficientes $\gamma_0, \gamma_1, \dots, \gamma_{n-2}$ são solução do sistema linear (4.6).

(\Leftarrow) Suponha que os coeficientes satisfazem o sistema linear (4.6). Então, pela Proposição 4.5 temos que ϕ_{n-1} satisfaz (4.4). Como $\gamma^{n-1} \neq 0$ por hipótese, segue que

ϕ_{n-1} é FOP. □

Do Corolário 4.6, temos que se o sistema (4.6) é impossível, então não existem FOPs. Nos referimos a esse caso como *deficiente*. Não estudaremos esse caso neste trabalho.

A matriz coeficiente que aparece no sistema linear (4.6) da proposição 4.5 é chamada de *matriz de Hankel*. Dizemos que uma matriz A de ordem $n + 1$ é de Hankel se for da forma

$$A = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots & a_n \\ a_1 & a_2 & \cdots & a_n & a_{n+1} \\ a_2 & a_3 & \ddots & & a_{n+2} \\ \vdots & \vdots & & \ddots & \vdots \\ a_n & a_{n+1} & a_{n+2} & \cdots & a_{2n} \end{bmatrix} .$$

O termo geral da matriz de Hankel é dado por $a_{ij} = a_{i+j-2}$ para alguma sequência $a_0, a_1, \dots, a_{2n-1}, a_{2n}$. As entradas de A são constantes ao longo das diagonais perpendiculares à diagonal principal.

4.1 FOPs Regulares e Singulares

Nesta seção, introduziremos o conceito de FOPs regulares e singulares e em seguida, veremos algumas de suas propriedades que nos permitem fazer uma conexão com os momentos de quebras do Processo de Biortogonalização de Lanczos. Tais propriedades nos ajudarão a entender o processo de Biortogonalização de Lanczos *look ahead*, processo tal que nos permite calcular vetores de Lanczos, mesmo quando existem tais quebras.

Considere o conjunto V_n definido por

$$V_n = \{ \phi \in P_{n-1} \mid \langle \phi, \psi \rangle = 0 \quad \forall \psi \in P_{n-2} \} .$$

Proposição 4.7 V_n é um subespaço vetorial do espaço P_n dos polinômios com coeficientes reais, de grau menor ou igual a n .

Demonstração. Primeiramente, note que $V_n \neq \emptyset$, pois $0 \in V_n$. Sejam $\phi_1, \phi_2 \in V_n$ e $\psi \in P_{n-2}$. Temos que

$$\langle \phi_1 + \phi_2, \psi \rangle = \langle \phi_1, \psi \rangle + \langle \phi_2, \psi \rangle = 0 + 0 = 0 \quad \forall \psi \in P_{n-2} .$$

Por outro lado, dado $\alpha \in \mathbb{R}$ e $\phi \in V_n$,

$$\langle \alpha\phi, \psi \rangle = \alpha \langle \phi, \psi \rangle = \alpha \cdot 0 = 0 \quad \forall \psi \in P_{n-2},$$

Logo, V_n é espaço vetorial. □

Definição 4.8 V_n é dito regular se $\dim V_n = 1$ e $V_n \cap P_{n-2} = \{0\}$, e dito singular caso contrário.

Note que se V_n é regular, então se $\phi, \tilde{\phi} \in V_n$, existe $k \in \mathbb{R}$ tal que $\phi = k\tilde{\phi}$.

Proposição 4.9 V_n é regular se, e somente se, a matriz coeficiente M do sistema linear (4.6) é não singular.

Demonstração.

(\Leftarrow) Suponha M não singular e seja $\phi \in V_n \cap P_{n-2}$. Pela Proposição 4.5, temos que existem u e γ_{n-1} tais que o sistema linear (4.6) é dado por

$$Mu = -\gamma_{n-1}v. \tag{4.8}$$

Como $\phi \in P_{n-2}$, temos que $\gamma_{n-1} = 0$, logo $Mu = 0$. Mas M é não singular, donde $u = 0$ e portanto $\phi = 0$. Assim, $V_n \cap P_{n-2} = \{0\}$. Resta mostrarmos que $\dim V_n = 1$.

Seja $\tilde{\phi} \in V_n$. Então, existem \tilde{u} e $\tilde{\gamma}_{n-1}$ tais que $M\tilde{u} = -\tilde{\gamma}_{n-1}v$.

Se $\phi = 0$, então $\phi = k\tilde{\phi}$, com $k = 0$. Se $\phi \neq 0$, então $\gamma_{n-1} \neq 0$ pelo que mostramos acima. Assim, usando (4.8), temos

$$M\tilde{u} = -\tilde{\gamma}_{n-1}v = -\tilde{\gamma}_{n-1} \frac{\gamma_{n-1}}{\gamma_{n-1}}v = k(-\gamma_{n-1}v) = kMu = M(ku), \tag{4.9}$$

com $k = \tilde{\gamma}_{n-1}/\gamma_{n-1}$. Aplicando a inversa de M em ambos os lados de (4.9), obtemos $\tilde{u} = ku$. Além disso, note que $\tilde{\gamma}_{n-1} = \tilde{\gamma}_{n-1}\gamma_{n-1}/\gamma_{n-1} = k\gamma_{n-1}$. Pelo Corolário 4.6,

$$\tilde{\phi}(\lambda) = \sum_{j=0}^{n-1} \tilde{\gamma}_j \lambda^j = \sum_{j=0}^{n-1} k\gamma_j \lambda^j = k\phi(\lambda),$$

de modo que $\dim V_n = 1$.

(\Rightarrow) Suponha V_n regular e assuma por absurdo que M é singular. Então, existe $u_0 = \begin{bmatrix} \gamma_0^0 & \gamma_1^0 & \cdots & \gamma_{n-2}^0 \end{bmatrix}^T \neq 0$ tal que $M(u_0) = 0$. Seja $\phi \in V_n$, $\phi \neq 0$ (note que ϕ existe, já que $\dim V_n \neq 0$). Como $\phi \neq 0$ e V_n é regular, existem $u = \begin{bmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{n-2} \end{bmatrix}^T$ e $\gamma_{n-1} \neq 0$ tais que $Mu = -\gamma_{n-1}v$. Por outro lado,

$$M(u + u_0) = Mu + Mu_0 = Mu = -\gamma_{n-1}v.$$

Portanto, pela Proposição (4.6), temos que $\tilde{\phi}(\lambda) = (\gamma_0 + \gamma_0^0) + (\gamma_1 + \gamma_1^0)\lambda + \dots + (\gamma_{n-2} + \gamma_{n-2}^0)\lambda^{n-2} + (\gamma_n - 1)\lambda^{n-1}$ é FOP de grau $n-1$. Logo, $\tilde{\phi} \in V_n$. Como V_n é regular, temos que $\tilde{\phi} = k\phi$ e então,

$$\begin{cases} \gamma_0 + \gamma_0^0 = k\gamma_0 \\ \gamma_1 + \gamma_1^0 = k\gamma_1 \\ \vdots \\ \gamma_{n-2} + \gamma_{n-2}^0 = k\gamma_{n-2} \\ \gamma_{n-1} = k\gamma_{n-1}. \end{cases}$$

Da última igualdade acima, segue que $k = 1$. Substituindo $k = 1$ nas equações anteriores, temos que $\gamma_j^0 = 0$ ($0 \leq j \leq n-1$), o que contradiz $u_0 \neq 0$. Logo, M é não singular. \square

Definição 4.10 *Seja $\phi_{n-1} \in V_n$, $\phi_{n-1} \neq 0$. Dizemos que ϕ_{n-1} é **FOP regular** de grau $n-1$ se V_n for regular e **FOP singular** se V_n for singular.*

Vimos que polinômios de grau 0 não nulos sempre são FOPs de grau 0, uma vez que a condição (4.4) é vazia para $n = 1$. Agora, com a definição acima, note também que FOPs de grau zero são sempre regulares, uma vez que V_1 é formado apenas por polinômios constantes e então, dados $\phi_0, \tilde{\phi}_0 \in V_1$, existe $k \in \mathbb{R}$ tal que $\phi_0 = k\tilde{\phi}_0$. Assim, $\dim V_1 = 1$. Além disso, $V_n \cap P_{n-2} = \{0\}$ é válido para $n = 1$ por vacuidade.

Observe ainda que FOPs de mesmo grau são todos ou regulares ou singulares. Assim, se ϕ_n é FOP regular de grau n , e $\tilde{\phi}_n$ é FOP de grau n , então $\tilde{\phi}_n$ também é regular, e portanto existe $k \in \mathbb{R}$ tal que $\tilde{\phi}_n = k\phi_n$. Isto nos mostra que FOPs regulares de mesmo grau são polinômios múltiplos. Com isso, se quisermos a unicidade de um FOP regular de grau n , basta impormos que o coeficiente do termo de grau n seja um valor fixo. Em particular, se impormos que um FOP regular seja mônico, temos que ele é único.

Outro resultado que temos a respeito de FOPs regulares segue trivialmente da Proposição 4.9: Um FOP de grau n é regular se, e somente se, a matriz M do sistema linear (4.6), de dimensão n , for não singular. Assim, se a matriz M for singular, não existirá FOP regular de grau n . Na proposição a seguir veremos outra maneira de verificar a existência de FOP regular de grau n . No entanto, para utilizá-la, precisaremos saber da existência de FOP regular de grau $n-1$.

Proposição 4.11 *Seja ϕ_{n-1} FOP regular de grau $n-1$. Então, existe FOP regular de grau n se, e somente se, $\langle \phi_{n-1}, \phi_{n-1} \rangle \neq 0$.*

Demonstração. Seja $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1\lambda + \dots + \gamma_{n-1}\lambda^{n-1}$ FOP regular de grau $n-1$. Então, $\gamma_{n-1} \neq 0$ e pelo Corolário 4.6, temos

$$\tilde{M}u = -\gamma_{n-1}m, \tag{4.10}$$

com $u = [\gamma_0 \ \gamma_1 \ \cdots \ \gamma_{n-2}]^T$, $m = [m_{n-1} \ m_n \ \cdots \ m_{2n-3}]^T$, sendo os coeficientes m_j dados por (4.7), e \tilde{M} a matriz não singular

$$\tilde{M} = \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_{n-2} \\ m_1 & \ddots & & \ddots & \vdots \\ m_2 & & \ddots & & \vdots \\ \vdots & \ddots & & & m_{2n-5} \\ m_{n-2} & \cdots & \cdots & m_{2n-5} & m_{2n-4} \end{bmatrix}.$$

Como $\gamma_{n-1} \neq 0$, temos que

$$m = -\frac{1}{\gamma_{n-1}} \tilde{M}u. \quad (4.11)$$

(\Rightarrow) Suponha que exista FOP regular $\phi_n(\lambda) = \alpha_0 + \alpha_1\lambda + \dots + \alpha_n\lambda^n$ de grau n . Então, pela Proposição 4.5, temos

$$M\alpha = -\alpha_n n,$$

com $\alpha = [\alpha_0 \ \alpha_1 \ \cdots \ \alpha_{n-1}]^T$, $n = [m_n \ m_{n+1} \ \cdots \ m_{2n-1}]^T$ e M é a matriz não singular dada por

$$M = \begin{bmatrix} m_0 & m_1 & \cdots & m_{n-2} & m_{n-1} \\ m_1 & \ddots & \ddots & \vdots & \vdots \\ \vdots & \ddots & & \vdots & \vdots \\ m_{n-2} & \cdots & \cdots & m_{2n-4} & m_{2n-3} \\ m_{n-1} & m_n & \cdots & m_{2n-3} & m_{2n-2} \end{bmatrix}.$$

Reescrevendo a matriz M acima em função dos termos do sistema linear (4.10) e usando (4.11), temos:

$$M = \begin{bmatrix} \tilde{M} & m \\ m^T & m_{2n-2} \end{bmatrix} = \begin{bmatrix} \tilde{M} & -\frac{1}{\gamma_{n-1}} \tilde{M}u \\ -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T & m_{2n-2} \end{bmatrix}.$$

Suponha por absurdo que $\langle \phi_{n-1}, \phi_{n-1} \rangle = 0$. Então

$$\begin{aligned} 0 &= \langle \phi_{n-1}, \phi_{n-1} \rangle = w_1^T \phi_{n-1}(A) \phi_{n-1}(A) v_1 = w_1^T \sum_{j=0}^{n-1} \gamma_j A^j \sum_{i=0}^{n-1} \gamma_i A^i v_1 \\ &= \sum_{j=1}^n \sum_{i=1}^n \gamma_{j-1} w_1^T A^{j-1} A^{i-1} v_1 \gamma_{i-1} = \sum_{j=1}^n \sum_{i=1}^n \gamma_{j-1} m_{(j-1)+(i-1)} \gamma_{i-1} = v^T M v, \end{aligned}$$

com $v = \begin{bmatrix} \gamma_0 & \cdots & \gamma_{n-2} & \gamma_{n-1} \end{bmatrix}^T = \begin{bmatrix} u \\ \gamma_{n-1} \end{bmatrix}$. Note que $v \neq 0$, já que $\gamma_{n-1} \neq 0$.

Assim,

$$\begin{aligned} 0 = v^T M v &= \begin{bmatrix} u^T & \gamma_{n-1} \end{bmatrix} \begin{bmatrix} \tilde{M} & -\frac{1}{\gamma_{n-1}} \tilde{M}u \\ -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T & m_{2n-2} \end{bmatrix} \begin{bmatrix} u \\ \gamma_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} u^T & \gamma_{n-1} \end{bmatrix} \begin{bmatrix} \tilde{M}u - \tilde{M}u \\ -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T u + m_{2n-2} \gamma_{n-1} \end{bmatrix} \\ &= \begin{bmatrix} u^T & \gamma_{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ k \end{bmatrix}, \text{ com } k = -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T u + m_{2n-2} \gamma_{n-1} \\ &= u^T \cdot 0 + \gamma_{n-1} \cdot k \\ &= \gamma_{n-1} \cdot k. \end{aligned}$$

Como $\gamma_{n-1} \neq 0$, segue que $k = 0$. Com isso, $Mv = \begin{bmatrix} 0 \\ k \end{bmatrix} = 0$, com $v \neq 0$. Logo, M é singular, o que é absurdo. Portanto, $\langle \phi_{n-1}, \phi_{n-1} \rangle \neq 0$.

(\Leftarrow) Mostremos que se $\langle \phi_{n-1}, \phi_{n-1} \rangle \neq 0$, então existe FOP regular de grau n .

Vimos na demonstração da recíproca que $\langle \phi_{n-1}, \phi_{n-1} \rangle = v^T M v$, sendo que

$$M = \begin{bmatrix} \tilde{M} & -\frac{1}{\gamma_{n-1}} \tilde{M}u \\ -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T & m_{2n-2} \end{bmatrix} \text{ e } v = \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{n-1} \end{bmatrix} = \begin{bmatrix} u \\ \gamma_{n-1} \end{bmatrix}.$$

Além disso, vimos que

$$v^T M v = \gamma_{n-1} \cdot k, \quad k = -\frac{1}{\gamma_{n-1}} (\tilde{M}u)^T u + m_{2n-2} \gamma_{n-1}.$$

Assim, usando o fato de \tilde{M} ser simétrica, temos

$$\langle \phi_{n-1}, \phi_{n-1} \rangle \neq 0 \Rightarrow -u^T \tilde{M}u + (\gamma_{n-1})^2 m_{2n-2} \neq 0. \quad (4.12)$$

Como ϕ_{n-1} é FOP regular, a matriz \tilde{M} é não singular. Usando o fato de \tilde{M}^{-1} também ser simétrica e levando-se em conta (4.10), podemos escrever

$$u^T \tilde{M}u = (\tilde{M}^{-1} \tilde{M}u)^T \tilde{M}u = (\tilde{M}u)^T \tilde{M}^{-T} (\tilde{M}u) = (-\gamma_{n-1} m)^T \tilde{M}^{-1} (-\gamma_{n-1} m). \quad (4.13)$$

Substituindo (4.13) em (4.12), encontramos

$$\begin{aligned} -(\gamma_{n-1})^2 m^T \tilde{M}^{-1} m + (\gamma_{n-1})^2 m_{2n-2} &\neq 0 \\ (\gamma_{n-1})^2 \left(-m^T \tilde{M}^{-1} m + m_{2n-2} \right) &\neq 0 \\ -m^T \tilde{M}^{-1} m + m_{2n-2} &\neq 0. \end{aligned} \quad (4.14)$$

Por outro lado, podemos escrever a matriz M como um produto de matrizes de forma a facilitar o cálculo do determinante de M :

$$M = \begin{bmatrix} \tilde{M} & m \\ m^T & m_{2n-2} \end{bmatrix} = \begin{bmatrix} \tilde{M} & 0 \\ m^T & 1 \end{bmatrix} \begin{bmatrix} I & \tilde{M}^{-1} m \\ 0 & m_{2n-2} - m^T \tilde{M}^{-1} m \end{bmatrix}.$$

Usando o teorema de Laplace para o cálculo de determinantes, encontramos:

$$\begin{aligned} \det(M) &= \det \left(\begin{bmatrix} \tilde{M} & 0 \\ m^T & 1 \end{bmatrix} \right) \det \left(\begin{bmatrix} I & \tilde{M}^{-1} m \\ 0 & m_{2n-2} - m^T \tilde{M}^{-1} m \end{bmatrix} \right) \\ &= \left(m_{2n-2} - m^T \tilde{M}^{-1} m \right) \det(\tilde{M}). \end{aligned}$$

Usando (4.14) e o fato de \tilde{M} ser não singular, segue que $\det M \neq 0$, donde M é não singular. Esse resultado nos mostra que se existir um FOP de grau n , então ele será regular. Basta agora garantirmos a existência.

Como M é não singular, tome $\phi_n(\lambda) = \beta_0 + \beta_1 \lambda + \dots + \beta_{n-1} \lambda^{n-1} + \lambda^n$, com $\beta_i = -M^{-1} m_{i+n}$ para $1 \leq i \leq n-1$. Note que ϕ_n é solução do sistema linear (4.6) e então, do Corolário 4.6 segue que ϕ_n é FOP. \square

Lembremos que tomamos L como sendo o menor inteiro tal que $w_{L+1}^T v_{L+1} = 0$, ou seja, $L+1$ é o primeiro passo tal que ocorre uma quebra no algoritmo. Caso tal quebra seja séria, isto é, $v_{L+1} \neq 0$ e $w_{L+1} \neq 0$, então, por 4.3, temos que

$$0 = w_{L+1}^T v_{L+1} = (\phi_L(A^T) w_1)^T \phi_L(A) v_1 = \langle \phi_L, \phi_L \rangle \Rightarrow \langle \phi_L, \phi_L \rangle = 0. \quad (4.15)$$

Como os polinômios contantes não nulos são FOPs regulares de grau 0, usando a Proposição 4.11 recursivamente a partir de $n = 1$, temos que todo FOP de grau n , com $n \leq L$, é regular. Como ϕ_L é FOP, por (4.15) e pela Proposição 4.11 segue que não existe FOP regular ϕ_{L+1} de grau $L+1$. Assim, $L+1$ é primeiro índice para o qual não existe FOP regular.

Além disso, vimos no algoritmo do Processo de Biortogonalização de Lanczos que, quando ocorre a quebra no passo $L+1$, isto é, $w_{L+1}^T v_{L+1} = 0$, o algoritmo pára e os últimos vetores calculados são v_{L+1} e w_{L+1} . Assim, quando temos a não existência do FOP de grau $L+1$, temos também a impossibilidade de calcular o vetor de índice $L+2$ no algoritmo.

Note ainda que embora sejam calculados os vetores v_{L+1} e w_{L+1} , tais vetores não são considerados de Lanczos, pois junto aos demais não formam um sistema biortogonal, visto que $\delta_{L+1} = 0$. No entanto, veremos a partir da próxima seção a implementação de uma técnica (estratégia look-ahead) que sana este problema da biortogonalização, tornando tais vetores de Lanczos.

Proposição 4.12 *Ocorre uma quebra séria no algoritmo de Biortogonalização de Lanczos se, e somente se, existe $n < L_* + 1$ tal que não existe FOP regular de grau n .*

Demonstração. (\Leftarrow) Sem perda de generalidade, seja n o menor natural tal que não exista FOP regular de grau n . Então, existe FOP regular de grau $n - 1$ e podemos tomar em particular, o FOP ϕ_{n-1} mônico associado ao vetor v_{n-1} . Assim, podemos usar a Proposição 4.11, donde segue que $\langle \phi_{n-1}, \phi_{n-1} \rangle = 0$. Por (4.3) e pela Proposição 4.1, temos $\langle \phi_{n-1}, \phi_{n-1} \rangle = 0 \Rightarrow \langle v_n, w_n \rangle = 0$. Note ainda da Proposição 4.11 que n é o menor natural tal que isso acontece. Logo, n é o passo em que ocorre a primeira quebra no algoritmo, isto é, $n = L + 1$. Resta mostrarmos que tal quebra é seria, isto é $v_{L+1} \neq 0$ e $w_{L+1} \neq 0$. Temos que $n < L_* + 1 \Rightarrow L < L_*$. Assim, $L \neq L_*$ e então, da Proposição (3.5), segue o resultado.

(\Rightarrow) Segue da Proposição (3.5).

□

4.2 Lanczos *Look Ahead*

A idéia do algoritmo Lanczos *look ahead* será estendermos o algoritmo de Biortogonalização de Lanczos, calculando vetores mesmo após a quebra. Para tanto, nos passos n em que existirem FOPs regulares de para grau n , vamos obter vetores de Lanczos v_{n+1} e w_{n+1} , que serão chamados de vetores regulares, e nos passos em que não existirem FOPs regulares, calcularemos vetores que completem a base dos subespaços de Krylov $K_m(A, v_1)$ e $K_m(A^T, w_1)$.

Note que verificar a existência de um FOP regular de grau n é bem simples. Basta verificarmos se a matriz coeficiente M do sistema linear (4.6), de dimensão n , é não singular. Para tanto, basta verificarmos se o menor valor singular (em aritmética exata) de M é positivo, isto é, $\sigma_{\min}(M) > 0$. Se isso ocorrer, então teremos que $\phi_n(\lambda) = \gamma_0 + \gamma_1\lambda + \dots + \gamma_n\lambda^n$, com $\gamma_i = M^{-1} - \gamma_n m_{i+n}$ para $0 \leq i \leq n - 3$ e γ_n qualquer, é FOP regular. Em particular, tomando $\gamma_n = 1$, temos um FOP regular mônico de grau n .

A idéia do processo *look ahead* será feita em analogia com o que vimos até agora de propriedades de FOPs. Considere, primeiramente, o subconjunto maximal de índices

$$\{n_1, n_2, \dots, n_J\} \subseteq \{1, 2, \dots, L_*\}; \quad n_1 = 1 < n_2 < \dots < n_J \leq L_* \quad (4.16)$$

tais que, para cada $j = 1, 2, \dots, J$, existe um FOP regular mônico ϕ_{n_j-1} . Note que $n_1 = 1$, pois $\phi_{n_1} = 1$ é FOP mônico regular.

Se $n_j < L_*$, o algoritmo de Lanczos Look Ahead é incapaz de gerar as bases para $K_m(v_1, A)$ e $K_m(w_1, A^T)$. Este caso é denominado quebra incurável [12].

Aqui, vamos assumir o resultado de que três FOPs regulares consecutivos $\phi_{n_{j-1}-1}$, ϕ_{n_j-1} , $\phi_{n_{j+1}-1}$ são conectados via três termos recorrentes (Para detalhes, ver [12] e referências lá citadas). Como consequência, fazendo analogia com (4.1), podemos tomar

$$v_{n_j} = \phi_{n_j-1}(A)v_1 \text{ e } w_{n_j} = \phi_{n_j-1}(A^T)w_1, \quad (4.17)$$

obtendo assim duas sequências de vetores $\{v_{n_j}\}_{j=1}^J$ e $\{w_{n_j}\}_{j=1}^J$ que são calculadas por três termos recorrentes. Esses vetores serão chamados de vetores regulares, uma vez que a eles associamos FOPs regulares. Note que os vetores iniciais v_1 e w_1 do algoritmo são sempre regulares, visto que $v_1 = \phi_{n_1}(A)v_1$ e $w_1 = \phi_{n_1}(A^T)w_1$, sendo que $\phi_{n_1} = 1$ é um FOP mônico regular.

No algoritmo de Biortogonalização de Lanczos, vimos que os vetores de Lanczos gerados pelo algoritmo formavam um sistema biortogonal. Veremos na proposição como fica essa propriedade para os vetores regulares:

Proposição 4.13 *Os vetores regulares v_{n_j} e w_{n_j} , $j = 1, 2, \dots, J$ satisfazem a seguinte relação:*

$$w_{n_j}^T v = w^T v_{n_j} = 0 \quad \forall v \in K_{n_j-1}(v_1, A), \quad w \in K_{n_j-1}(w_1, A^T).$$

Demonstração. Mostremos que $w_{n_j}^T v = 0 \forall v \in K_{n_j-1}(v_1, A)$:

Temos que $w_{n_j} = \phi_{n_j-1}(A^T)w_1$, com ϕ_{n_j-1} FOP regular. Dado $v \in K_{n_j-1}(A, v_1)$, temos que $v = \alpha_0 v_1 + \alpha_1 A v_1 + \dots + \alpha_{n_j-2} A^{n_j-2} v_1 = \tilde{\phi}(A)v_1$ para algum $\tilde{\phi} \in P_{n_j-2}$. Assim,

$$w_{n_j}^T v = (\phi_{n_j-1}(A^T)w_1)^T (\tilde{\phi}(A)v_1) = \langle \phi_{n_j-1}, \tilde{\phi} \rangle.$$

Como ϕ_{n_j-1} é FOP, segue da Definição 4.4 que $\langle \phi_{n_j-1}, \tilde{\phi} \rangle = 0$. Portanto, $w_{n_j}^T v = 0$ para todo $v \in K_{n_j-1}(v_1, A)$. De forma análoga, mostra-se que $w^T v_{n_j} = 0$ para todo $w \in K_{n_j-1}(w_1, A^T)$. \square

Voltando à Proposição 4.11, note que se garantirmos a existência de FOP de grau $n_j - 1$, então a não existência de FOP de grau n_j equivale a $\langle \phi_{n_j-1}, \phi_{n_j-1} \rangle = 0$. Por (4.3) e (4.17), isto equivale a $\langle v_{n_j}, w_{n_j} \rangle = 0$, isto é, ocorre uma quebra no algoritmo no passo n_j . Note também que se tivermos o menor valor singular da matriz coeficiente M do sistema linear (4.6) próximo de zero, esta matriz estará próxima de ser uma matriz singular e daí, podemos dizer que estamos próximos de não termos FOP regular. Novamente por (4.11), teríamos uma quase-quebra.

Para evitarmos essas quase-quebras, não vamos considerar os FOPs regulares quando a matriz do sistema (4.6) for próxima de não singular. Com isso, geramos duas subsequências de vetores regulares $\{v_{n_{j_k}}\}_{k=1}^K$ e $\{w_{n_{j_k}}\}_{k=1}^K$, sendo que $\{n_{j_k}\}_{k=1}^K \subseteq \{n_j\}_{j=1}^J$ é um subconjunto de (4.16). Note que $j_1 = 1$, já que v_1 e w_1 são sempre vetores regulares. Para não sobrecarregar as notações, daqui em diante vamos denotar n_{j_k} por n_k , de modo que as subsequências mencionadas acima serão denotadas por $\{v_{n_k}\}_{k=1}^K$ e $\{w_{n_k}\}_{k=1}^K$.

Como nosso objetivo é gerar vetores que formam bases para os subespaços de Krylov $K_n(v_1, A)$ e $K_n(w_1, A^T)$, precisamos acrescentar vetores às subsequências $\{v_{n_k}\}_{k=1}^K$ e $\{w_{n_k}\}_{k=1}^K$ de forma a completar bases para tais subespaços.

Para garantirmos (3.1)-(3.2), para $n = n_{k-1}, \dots, n_k - 1$, com $k = 2, 3, \dots, K$, acrescentaremos os vetores às subsequências $\{v_{n_k}\}_{k=1}^K$ e $\{w_{n_k}\}_{k=1}^K$, respectivamente:

$$v_n \in K_n(v_1, A) - K_{n-1}(v_1, A) \text{ e } w_n \in K_n(w_1, A^T) - K_{n-1}(w_1, A^T).$$

Tais vetores serão chamados de vetores internos.

Vamos considerar os vetores regulares e internos agrupados em blocos e vamos impor a condição de biortogonalidade entre tais blocos. Para cada $k = 1, 2, 3, \dots, K$, considere os k -ésimos blocos dados por:

$$V_k = \begin{bmatrix} v_{n_k} & v_{n_{k+1}} & \cdots & v_{n_{k+1}-1} \end{bmatrix} \text{ e } W_k = \begin{bmatrix} w_{n_k} & w_{n_{k+1}} & \cdots & w_{n_{k+1}-1} \end{bmatrix}.$$

Note que o primeiro vetor de cada um dos k -ésimos blocos acima é regular e os demais são os vetores internos que foram acrescentados entre dois vetores regulares. Os vetores internos serão escolhidos de forma que blocos com índice k diferentes ainda sejam biortogonais um com o outro. Mais adiante, detalharemos como os vetores internos e regulares são escolhidos no algoritmo *look ahead*.

Veremos agora um exemplo (apresentado em [24]) de como ficam esses blocos num caso em que surge quebra no algoritmo de Biortogonalização de Lanczos:

Exemplo 4.14 *Sejam* $A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$, e $v_1 = w_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}^T$.

Vamos verificar, primeiramente, os índices para os quais existem FOPs regulares. Sabemos que um FOP regular de grau n existe se, e somente se, a matriz coeficiente do sistema linear (4.6), de dimensão n , é não singular. Como os termos dessa matriz são dados pelos coeficientes $m_j = w_1^T A^j v_1$, $j = 0, 1, \dots$, basta calcularmos tais termos a partir de A e dos vetores iniciais v_1 e w_1 . A partir desse cálculo, obtivemos matrizes coeficientes

M de dimensões $n = 1, 2, \dots$. Para cada uma dessas matrizes, calculamos o menor valor singular e obtivemos os seguintes resultados:

dim M	$\sigma_{min}(M)$	Singularidade
1	9.1000e+001	não singular
2	2.0585e+000	não singular
3	9.2199e-002	não singular
4	4.5765e-015	\approx singular
5	8.1798e-002	não singular
6	9.0000e+000	não singular
≥ 7	≈ 0	singular

Note que quando $\dim M = 4$, o menor valor singular de M é muito próximo de zero. Temos, então, que a matriz M está próxima de ser singular e estamos no caso que comentamos de quase-quebra. Assim, não consideramos FOP regular de grau 4. Como aos vetores regulares v_n e w_n associamos FOPs regulares de grau $n - 1$, segue que v_5 não será regular.

Com isso, teríamos cinco vetores regulares: v_1, v_2, v_3, v_4 e v_6 ; e apenas um vetor interno: v_5 . Assim, os blocos de vetores seriam:

$$V_1 = \begin{bmatrix} v_1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} v_2 \end{bmatrix}, \quad V_3 = \begin{bmatrix} v_3 \end{bmatrix}, \quad V_4 = \begin{bmatrix} v_4 & v_5 \end{bmatrix}, \quad \text{e} \quad V_5 = \begin{bmatrix} v_6 \end{bmatrix}.$$

Veremos, agora, o que acontece ao implementarmos o algoritmo de Biortogonalização de Lanczos neste exemplo. Abaixo, segue alguns dados gerados pelo algoritmo. Para a exibição das matrizes V_n e W_n , consideramos os termos com módulo menor que 10^{-15} como nulos. No entanto, tais valores não foram truncados durante os cálculos.

```
delta1 = 1.00000          delta4 = 3.0066e-016
delta2 = 0.038763        delta5 = -5.2052e+014
delta3 = -0.0073945
```

$V_n =$

```
0.10483  0.54142  -0.78768  0.62897  8.2758e+014  -8.4442e+014
0.20966  -0.07027  0.63485  -1.25794  -1.6552e+015  1.4074e+015
0.31449  -0.05299  -0.00588  0.62897  8.2758e+014  -8.4442e+014
0.41931  -0.03571  -0.01763  0.00000  8.2298e-001  1.7188e+000
0.52414  -0.01843  -0.02939  0.00000  6.8304e-001  1.6250e+000
0.62897  -0.00115  -0.04115  0.00000  8.9003e-001  1.5000e+000
```

$W_n =$

```
0.10483  0.12211  -0.04115  0.00000  8.7191e-001  1.5000e+000
```

0.20966	0.13939	-0.05290	0.00000	8.9017e-001	1.5000e+000
0.31449	0.15667	-0.06466	0.00000	6.2640e-001	1.0000e+000
0.41931	0.17395	-0.07642	-0.62897	-8.2758e+014	7.0369e+014
0.52414	0.19123	-0.71714	1.25794	1.6552e+015	-1.4074e+015
0.62897	-0.42047	0.70539	-0.62897	-8.2758e+014	8.4442e+014

Note que quando geramos os vetores v_4 e w_4 que aparecem nas colunas 4 das matrizes acima, temos $\delta_4 = \langle v_4, w_4 \rangle = 3.0066e-016$ muito próximo de zero. Assim, temos uma quase-quebra no quarto passo do algoritmo. Caso desconsideremos essa quebra e continuemos o algoritmo, o que percebemos é que no quarto passo, α_4 aparecerá dividido por δ_4 . Tal operação gera erros de arredondamento, pois estamos efetuando uma divisão por um número muito próximo de zero. Veja, por exemplo, que nas coordenadas dos vetores v_5 e w_5 nas quintas colunas das matrizes acima aparecem valores como $8.2758e+014$ e $1.6552e+015$ respectivamente, que são valores absurdamente grandes. Além disso, note também que os valores de δ_n , com $n \geq 4$ são ou muito próximos de zero ou muito grandes. Isto nos mostra como o algoritmo fica instável a partir da primeira quase-quebra que ocorre. Assim, devemos considerar a quebra no quarto passo e então, o Algoritmo de Biortogonalização de Lanczos pára e gera apenas seis vetores de Lanczos: v_1, v_2, v_3, w_1, w_2 e w_3 .

Na prática, por razões de estabilidade dos cálculos em versão finita, convém normalizar os vetores de Lanczos a direita e a esquerda [20, 12]. Vamos apenas ilustrar essa questão com esse mesmo exemplo apresentado acima. Primeiramente, mostremos as alterações que ocorrem no algoritmo de Biortogonalização de Lanczos ao exigirmos que os vetores v_n e w_n sejam normalizados assim que calculados, e em seguida, implementaremos esse novo algoritmo nesse exemplo. Chamaremos esse algoritmo de *Lanczos Normalizado* e trataremos dele na próxima seção.

4.2.1 Algoritmo de Lanczos Normalizado

O que faremos nessa seção será apresentarmos uma nova versão do Algoritmo de Biortogonalização de Lanczos, impondo agora a condição de que os vetores v_n e w_n gerados pelo algoritmo sejam normalizados assim que calculados.

Primeiramente, vamos redefinir os vetores em (3.4)-(3.5) como sendo

$$\begin{aligned} \tilde{v}_{n+1} &= Av_n - \alpha_n v_n - \beta_n v_{n-1}, \\ \tilde{w}_{n+1} &= A^T w_n - \tilde{\alpha}_n w_n - \tilde{\beta}_n w_{n-1}. \end{aligned} \tag{4.18}$$

Em seguida, calculamos

$$v_n = \frac{\tilde{v}_n}{\|\tilde{v}_n\|} \quad \text{e} \quad w_n = \frac{\tilde{w}_n}{\|\tilde{w}_n\|}.$$

Como fizemos no algoritmo de Biortogonalização de Lanczos, impondo que (3.3)

seja válido, vamos determinar as constantes α_n , β_n , $\tilde{\alpha}_n$ e $\tilde{\beta}_n$ em (4.18).

Por (3.3), temos que $\langle v_{j+1}, w_i \rangle = 0$ para $i = j$. Daí, temos:

$$0 = \langle v_{j+1}, w_j \rangle = \left\langle \frac{\tilde{v}_{j+1}}{\|\tilde{w}_j\|} \frac{\tilde{v}_{j+1}}{\|\tilde{w}_j\|} \right\rangle = \frac{\langle \tilde{v}_{j+1}, \tilde{w}_j \rangle}{\|\tilde{v}_{j+1}\| \|\tilde{w}_j\|},$$

o que garante que $\langle \tilde{v}_{j+1}, \tilde{w}_j \rangle = 0$. Assim,

$$\begin{aligned} \langle Av_j - \alpha_j v_j - \beta_j v_{j-1}, w_j \rangle &= 0 \\ \langle Av_j, w_j \rangle - \alpha_j \langle v_j, w_j \rangle - \beta_j \underbrace{\langle v_{j-1}, w_j \rangle}_0 &= 0 \\ \langle Av_j, w_j \rangle - \alpha_j \delta_j &= 0 \Rightarrow \alpha_j = \frac{\langle Av_j, w_j \rangle}{\delta_j}. \end{aligned}$$

Por outro lado, por (3.3), temos que $\langle v_i, w_{j+1} \rangle = 0$ para $i = j$. Por raciocínio análogo ao anterior, obtemos:

$$\tilde{\alpha}_j = \frac{\langle v_j, A^T w_j \rangle}{\delta_j} = \frac{\langle Av_j, w_j \rangle}{\delta_j} = \alpha_j.$$

Para determinarmos β_n , usaremos $\langle v_{j+1}, w_i \rangle = 0$ para $i < j$. Daí, temos por contas análogas as feitas no processo de Biortogonalização de Lanczos que:

$$\langle v_{j+1}, w_i \rangle = 0 \Rightarrow \langle \tilde{v}_{j+1}, w_i \rangle = 0 \Rightarrow \langle v_j, \tilde{w}_{i+1} \rangle - \beta_j \langle v_{j-1}, w_i \rangle = 0.$$

Na última equação, se $i < j - 1$, os produtos internos são todos nulos. Caso $i = j - 1$, temos:

$$\begin{aligned} 0 &= \langle v_j, \tilde{w}_j \rangle - \beta_j \langle v_{j-1}, w_{j-1} \rangle \\ &= \langle v_j, A^T w_{j-1} \rangle - \tilde{\alpha}_{j-1} \underbrace{\langle v_j, w_{j-1} \rangle}_0 - \tilde{\beta}_{j-1} \underbrace{\langle v_j, w_{j-2} \rangle}_0 \beta_j \langle v_{j-1}, w_{j-1} \rangle. \end{aligned}$$

Portanto, $\beta_j = \langle Av_j, w_{j-1} \rangle / \delta_{j-1}$. Por raciocínio análogo ao anterior, usando agora $\langle v_i, w_{j+1} \rangle = 0$ para $i \leq j$, obtemos $\tilde{\beta}_j = \langle Av_{j-1}, w_j \rangle / \delta_{j-1}$.

Comparando as constantes que aparecem nos termos recorrentes do Algoritmo de Biortogonalização de Lanczos e de Lanczos Normalizado, percebemos que a constante α_n permanece a mesma em ambos os casos. Além disso, a constante $\beta_n = \delta_n / \delta_{n-1}$ do algoritmo sem a normalização pode ser reescrita como $\beta_n = \langle Av_n, w_{n-1} \rangle / \delta_{n-1}$, que é justamente a constante β_n que aparece no algoritmo com normalização. Vejamos como isso acontece:

Por (3.4), temos $Av_n = v_{n+1} + \alpha_n v_n + \beta_n v_{n-1}$. Fazendo o produto interno em

ambos os lados com w_{n-1} , obtemos:

$$\begin{aligned} \langle Av_n, w_{n-1} \rangle &= \langle v_{n+1} + \alpha_n v_n + \beta_n v_{n-1}, w_{n-1} \rangle \\ &= \underbrace{\langle v_{n+1}, w_{n-1} \rangle}_0 + \alpha_n \underbrace{\langle v_n, w_{n-1} \rangle}_0 + \beta_n \underbrace{\langle v_{n-1}, w_{n-1} \rangle}_{\delta_{n-1}} \\ \Rightarrow \beta_n &= \frac{\langle Av_n, w_{n-1} \rangle}{\delta_{n-1}}. \end{aligned}$$

Assim, vale a seguinte relação no algoritmo de Biortogonalização de Lanczos:

$$\beta_n = \frac{\delta_n}{\delta_{n-1}} = \frac{\langle Av_n, w_{n-1} \rangle}{\delta_{n-1}} \Rightarrow \delta_n = \langle Av_n, w_{n-1} \rangle.$$

Aqui, é natural nos perguntarmos se essa mesma relação continua válida ao normalizarmos os vetores, isto é, será que também podemos escrever $\beta_n = \delta_n / \delta_{n-1}$ no algoritmo de Lanczos Normalizado? Para respondermos a essa pergunta, vamos escrever $\langle Av_n, w_{n-1} \rangle$ em função de δ_n . Temos que:

$$\begin{aligned} \langle Av_n, w_{n-1} \rangle &= \langle v_n, A^T w_{n-1} \rangle = \langle v_n, \tilde{w}_n + \tilde{\alpha}_{n-1} w_{n-1} + \tilde{\beta}_{n-1} w_{n-2} \rangle \\ &= \langle v_n, \tilde{w}_n \rangle + \tilde{\alpha}_{n-1} \underbrace{\langle v_n, w_{n-1} \rangle}_0 + \tilde{\beta}_{n-1} \underbrace{\langle v_n, w_{n-2} \rangle}_0 = \langle v_n, w_n \|\tilde{w}_n\| \rangle \\ &= \|\tilde{w}_n\| \langle v_n, w_n \rangle = \|\tilde{w}_n\| \delta_n. \end{aligned}$$

Como, em geral, temos $\|\tilde{w}_n\| \neq 1$ no algoritmo Normalizado, segue que não podemos escrever $\beta_n = \delta_n / \delta_{n-1}$ neste algoritmo. De forma análoga, mostra-se que $\langle Av_{n-1}, w_n \rangle = \|\tilde{v}_n\| \delta_n$. Assim, esta relação também não é válida para $\tilde{\beta}_n$.

Uma vez obtidas as constantes α_n , β_n e $\tilde{\beta}_n$, podemos escrever o Algoritmo de Lanczos Normalizado:

Algoritmo 8 Algoritmo de Lanczos Normalizado

- 1: Escolha dois vetores $\tilde{v}_1 \in \mathbb{R}^N$ e $\tilde{w}_1 \in \mathbb{R}^N$ tais que $\langle \tilde{v}_1, \tilde{w}_1 \rangle \neq 0$
 - 2: Sejam $\delta_0 = 1$ e $w_0 = v_0 \equiv 0$
 - 3: $v_1 = \tilde{v}_1 / \|\tilde{v}_1\|$;
 - 4: $w_1 = \tilde{w}_1 / \|\tilde{w}_1\|$;
 - 5: **Para** $n = 1, 2, \dots, m$ **faça**
 - 6: $\delta_n = \langle v_n, w_n \rangle$. Se $\delta_n = 0$, páre.
 - 7: $\alpha_n = \langle Av_n, w_n \rangle / \delta_n$;
 - 8: $\beta_n = \langle Av_n, w_{n-1} \rangle / \delta_{n-1}$;
 - 9: $\tilde{\beta}_n = \langle Av_{n-1}, w_n \rangle / \delta_{n-1}$;
 - 10: $\tilde{v}_{n+1} = Av_n - \alpha_n v_n - \beta_n v_{n-1}$;
 - 11: $\tilde{w}_{n+1} = A^T w_n - \alpha_n w_n - \tilde{\beta}_n w_{n-1}$;
 - 12: $v_{n+1} = \tilde{v}_{n+1} / \|\tilde{v}_{n+1}\|$;
 - 13: $w_{n+1} = \tilde{w}_{n+1} / \|\tilde{w}_{n+1}\|$;
 - 14: **Fim**
-

De forma análoga ao algoritmo de Biortogonalização de Lanczos, se o algoritmo de Lanczos simétrico não quebra antes do passo $m + 1$, ou seja, se $\delta_n \neq 0$ para $1 \leq n \leq m$ na linha 6 do algoritmo, então os vetores v_i , $i = 1, \dots, m$ e w_j , $j = 1, \dots, m$ gerados pelo algoritmo satisfazem (3.3) por construção, ou seja, formam um sistema biortogonal. Além disso, os vetores v_i e w_i , $i = 1, \dots, m$, formam bases de $K_m(A, v_1)$ e $K_m(A^T, w_1)$ respectivamente.

Voltemos agora ao Exemplo (4.14) e vamos implementar o algoritmo de Lanczos Normalizado para compararmos com os resultados obtidos da Biortogonalização de Lanczos. Note que, neste caso, temos $\tilde{v}_1 = \tilde{w}_1 = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}^T$ e $v_1 = \tilde{v}_1 / \|\tilde{v}_1\|$. Abaixo, seguem os resultados obtidos. Para efeito de visualização, nas matrizes V_n e W_n , consideramos os termos com módulo menor que 10^{-15} como nulos. No entanto, tais valores não foram truncados durante os cálculos.

```
delta1 = 1.00000
delta2 = 0.12814
delta3 = -0.0072044
delta4 = 9.5497e-017
delta5 = -9.5497e-017
```

$V_n =$

```
0.10483  0.98440 -0.77749  0.40825  0.40825  0.40825
0.20966 -0.12776  0.62664 -0.81650 -0.81650 -0.81650
0.31449 -0.09635 -0.00580  0.40825  0.40825  0.40825
0.41931 -0.06493 -0.01741  0.00000  0.00000  0.00000
0.52414 -0.03351 -0.02901  0.00000  0.00000  0.00000
0.62897 -0.00209 -0.04062  0.00000  0.00000  0.00000
```

$W_n =$

```
0.10483  0.22201 -0.04062  0.00000  0.00000  0.00000
0.20966  0.25343 -0.05222  0.00000  0.00000  0.00000
0.31449  0.28485 -0.06382  0.00000  0.00000  0.00000
0.41931  0.31627 -0.07543 -0.40825 -0.40825 -0.35898
0.52414  0.34768 -0.70787  0.81650  0.81650  0.86155
0.62897 -0.76448  0.69626 -0.40825 -0.40825 -0.35898
```

Observe que $\delta_4 = 9.5497e-017$ é muito próximo de zero, o que nos indica uma quase-quebra no algoritmo. Mas, diferente do que ocorreu ao implementarmos o algoritmo de Biortogonalização de Lanczos neste exemplo, os vetores passam a ficar mais comporta-

dos, não aparecendo coordenadas nos próximos vetores muito grandes, como ocorria com v_5 e w_5 no caso anterior.

Note ainda que os vetores v_4 e v_5 são iguais, logo linearmente dependentes. O mesmo ocorre com w_4 e w_5 . Assim, caso ignorássemos as quebras, assumindo que os vetores $\{v_1, \dots, v_5\}$ e $\{w_1, \dots, w_5\}$ permanecem gerando os espaços $K_5(A, v_1)$ e $K_5(A^T, w_1)$, então estes espaços teriam dimensão 4. Por outro lado, sejam

$$B = \begin{bmatrix} v_1 & Av_1 & A^2v_1 & A^3v_1 & A^4v_1 \end{bmatrix}, \quad C = \begin{bmatrix} w_1 & A^T w_1 & (A^T)^2 w_1 & (A^T)^3 w_1 & (A^T)^4 w_1 \end{bmatrix}.$$

Temos que $B = (1/\|\tilde{v}_1\|)\tilde{B}$ e $C = (1/\|\tilde{w}_1\|)\tilde{C}$, sendo que

$$\tilde{B} = \begin{bmatrix} \tilde{v}_1 & A\tilde{v}_1 & A^2\tilde{v}_1 & A^3\tilde{v}_1 & A^4\tilde{v}_1 \end{bmatrix}, \quad \tilde{C} = \begin{bmatrix} \tilde{w}_1 & A^T\tilde{w}_1 & (A^T)^2\tilde{w}_1 & (A^T)^3\tilde{w}_1 & (A^T)^4\tilde{w}_1 \end{bmatrix}.$$

As matrizes \tilde{B} e \tilde{C} são facilmente obtidas, de modo que encontramos

$$B = \frac{1}{\|\tilde{v}_1\|} \begin{bmatrix} 1 & 6 & 5 & 4 & 3 \\ 2 & 1 & 6 & 5 & 4 \\ 3 & 2 & 1 & 6 & 5 \\ 4 & 3 & 2 & 1 & 6 \\ 5 & 4 & 3 & 2 & 1 \\ 6 & 5 & 4 & 3 & 2 \end{bmatrix} \quad \text{e} \quad C = \frac{1}{\|\tilde{w}_1\|} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 1 \\ 4 & 5 & 6 & 1 & 2 \\ 5 & 6 & 1 & 2 & 3 \\ 6 & 1 & 2 & 3 & 4 \end{bmatrix}.$$

Calculando o posto das matrizes B_1 e C_1 , obtemos posto 5. Assim, o posto das matrizes B e C também são 5, donde $\dim K_5(A, v_1) = \dim K_5(A^T, w_1) = 5$, de modo que os vetores $\{v_1, \dots, v_5\}$ e $\{w_1, \dots, w_5\}$ não geram $K_5(A, v_1)$ e $K_5(A^T, w_1)$. Portanto, não podemos deixar de considerar as quebras sérias no algoritmo.

4.2.2 Algoritmo Sequencial

Nesta seção, veremos uma versão do algoritmo de Lanczos *look ahead* denominada *algoritmo sequencial* [12]. Vamos mostrar como os vetores regulares e internos são calculados e a condição que será imposta para o algoritmo calcular um ou outro tipo de vetor. O algoritmo é denominado sequencial porque um vetor (regular ou interno) é calculado por vez, em cada passo n do algoritmo.

No passo n , o algoritmo já terá construído os blocos V_1, W_1, \dots, V_{k-1} e W_{k-1} , e estará construindo o bloco

$$V_k = \begin{bmatrix} v_{n_k} & v_{n_k+1} & \cdots & v_n \end{bmatrix} \quad \text{e} \quad W_k = \begin{bmatrix} w_{n_k} & w_{n_k+1} & \cdots & w_n \end{bmatrix}.$$

Os primeiros vetores v_{n_k} e w_{n_k} de cada bloco são os vetores regulares, e os demais,

são internos. Os blocos são construídos de forma que

$$W_i^T V_j = \begin{cases} 0, & \text{se } i \neq j \\ \delta_i & \text{se } i = j \end{cases} \quad i, j = 1, 2, \dots, k, \quad (4.19)$$

sendo que $\delta_1, \delta_2, \dots, \delta_{k-1}$ são não singulares. Entretanto, a matriz $\delta_k = W_k^T V_k$ pode ser singular. Neste caso, os blocos V_k e W_k são considerados incompletos e os vetores v_{n+1} e w_{n+1} serão adicionados a estes blocos. Se $\delta_k = W_k^T V_k$ for não singular, dizemos que os blocos V_k e W_k são completos, e novos blocos V_{k+1} e W_{k+1} são iniciados com os vetores regulares $v_{n_{k+1}}$ e $w_{n_{k+1}}$, sendo $n_{k+1} = n + 1$.

Para o cálculo dos vetores regulares, uma maneira de determiná-los seria através dos FOPs regulares. Vimos que uma vez que exista FOP regular de grau n , isto é, a matriz coeficiente do sistema linear (4.6), de dimensão n , é não singular, conseguimos associar ao vetor v_n e w_n um FOP regular mônico ϕ_{n-1} de grau $n - 1$, sendo

$$v_{n_k} = \phi_{n-1}(A) v_1 \text{ e } w_{n_k} = \phi_{n-1}(A^T) w_1.$$

Vimos também que para determinarmos ϕ_{n-1} , pela proposição 4.5, basta tomarmos $\phi_{n-1}(\lambda) = \gamma_0 + \gamma_1 \lambda + \dots + \lambda_{n-1}$, com $\gamma_i = M^{-1} - m_{i+n-1}$ para $0 \leq i \leq n - 2$.

No entanto, calcularmos os vetores mônicos associados aos FOPs regulares é computacionalmente caro, pois para cada FOP regular, precisamos calcular os momentos associados ao sistema (4.6) e a matriz inversa de M . Assim, em vez de calcularmos os FOPs mônicos, vamos calcular vetores regulares de forma análoga a que fizemos nos dois algoritmos anteriores: vamos impor que a condição (4.19) seja satisfeita. Além disso, no algoritmo, vamos optar por normalizarmos os vetores, pois como vimos na seção anterior, o algoritmo passa a ficar mais estável com a normalização dos vetores de Lanczos.

Para o cálculo do vetor regular $v_{n_{k+1}}$ do bloco V_{k+1} , tomaremos $v_{n_{k+1}}$ da forma

$$v_{n_{k+1}} = A v_{n_k} + y_k + y_{k-1}, \quad y_k \in \{V_k\} \text{ e } y_{k-1} \in \{V_{k-1}\},$$

sendo que $\{V_k\}$ e $\{V_{k-1}\}$ são os espaços gerados pelos vetores colunas de V_k e V_{k-1} , respectivamente. Como $y_k \in \{V_k\}$, com $V_k = [v_{n_k} \ v_{n_{k+1}} \ \dots \ v_{n_{k+1}-1}]$, temos

$$\begin{aligned} y_k &= x_1 v_{n_k} + x_2 v_{n_{k+1}} + \dots + x_{n_{k+1}-n_k} v_{n_{k+1}-1} \\ &= x_1 \begin{bmatrix} v_{n_k}^1 \\ v_{n_k}^2 \\ \vdots \\ v_{n_k}^n \end{bmatrix} + x_2 \begin{bmatrix} v_{n_{k+1}}^1 \\ v_{n_{k+1}}^2 \\ \vdots \\ v_{n_{k+1}}^n \end{bmatrix} + \dots + x_{n_{k+1}-n_k} \begin{bmatrix} v_{n_{k+1}-1}^1 \\ v_{n_{k+1}-1}^2 \\ \vdots \\ v_{n_{k+1}-1}^n \end{bmatrix} \\ &= \begin{bmatrix} x_1 v_{n_k}^1 + v_{n_{k+1}}^1 + \dots + x_{n_{k+1}-n_k} v_{n_{k+1}-1}^1 \\ x_1 v_{n_k}^2 + v_{n_{k+1}}^2 + \dots + x_{n_{k+1}-n_k} v_{n_{k+1}-1}^2 \\ \vdots \\ x_1 v_{n_k}^n + v_{n_{k+1}}^n + \dots + x_{n_{k+1}-n_k} v_{n_{k+1}-1}^n \end{bmatrix}, \end{aligned}$$

ou seja, $y_k = V_k x_k$ com $x_k = [x_1 \ x_2 \ \cdots \ x_{n_{k+1}-n_k}]$. De forma análoga, temos que se $y_{k-1} \in \{V_{k-1}\}$, então $y_{k-1} = V_{k-1} x_{k-1}$, com $x_{k-1} = [y_1 \ y_2 \ \cdots \ y_{n_k-n_{k-1}}]$. Assim, $v_{n_{k+1}}$ fica escrito na forma

$$v_{n_{k+1}} = Av_{n_k} + V_k x_k + V_{k-1} x_{k-1}. \quad (4.20)$$

Agora, precisamos determinar os vetores x_k e x_{k-1} . Vamos determinar esses vetores impondo a condição (4.19).

Por (4.19), temos $W_i^T V_{j+1} = 0$ para todo $i \leq j$. Assim, para $i = j = k$, temos

$$W_k^T V_{k+1} = 0 \Rightarrow W_k^T v_{n_{k+1}} = 0.$$

Fazendo então o produto interno em ambos os lados de (4.20) por W_k^T , temos

$$\begin{aligned} 0 &= W_k^T v_{n_{k+1}} = W_k^T Av_{n_k} + W_k^T V_k x_k + \underbrace{W_k^T V_{k-1} x_{k-1}}_0 \\ &\Rightarrow 0 = W_k^T Av_{n_k} + \delta_k x_k \Rightarrow x_k = -\delta_k^{-1} W_k^T Av_{n_k}. \end{aligned}$$

Resta agora determinarmos x_{k-1} . Como $W_i^T V_{j+1} = 0 \ \forall i \leq j$, tomando em particular $j = k$ e $i = j - 1 = k - 1$, temos

$$W_{k-1}^T V_{k+1} = 0 \Rightarrow W_{k-1}^T v_{n_{k+1}} = 0.$$

Fazendo agora o produto interno em ambos os lados de (4.20) por W_{k-1}^T , temos

$$\begin{aligned} 0 &= W_{k-1}^T v_{n_{k+1}} = W_{k-1}^T Av_{n_k} + \underbrace{W_{k-1}^T V_k x_k}_0 + W_{k-1}^T V_{k-1} x_{k-1} \\ &\Rightarrow 0 = W_{k-1}^T Av_{n_k} + \delta_{k-1} x_{k-1} \Rightarrow x_{k-1} = -\delta_{k-1}^{-1} W_{k-1}^T Av_{n_k}. \end{aligned}$$

Portanto, o vetor regular $v_{n_{k+1}}$ do bloco V_{k+1} é dado por

$$v_{n_{k+1}} = Av_{n_k} - V_k \delta_k^{-1} W_k^T Av_{n_k} - V_{k-1} \delta_{k-1}^{-1} W_{k-1}^T Av_{n_k}. \quad (4.21)$$

De forma análoga ao caso anterior, tomamos o vetor regular $w_{n_{k+1}}$ do bloco W_{k+1} da forma

$$w_{n_{k+1}} = A^T w_{n_k} + W_k y_k + V_{k-1} y_{k-1}. \quad (4.22)$$

Para determinarmos y_k e y_{k-1} , como no caso anterior, impomos a condição (4.19).

Por (4.19), temos $W_{j+1}^T V_i = 0 \ \forall i \leq j$.

Para $i = j = k$, temos:

$$\begin{aligned}
 W_{k+1}^T V_k = 0 &\Rightarrow \begin{bmatrix} w_{n_{k+1}}^T \\ w_{n_{k+1}+1}^T \\ \vdots \\ w_{n_{k+2}-1}^T \end{bmatrix} \begin{bmatrix} v_{n_k} & v_{n_{k+1}} & \cdots & v_{n_{k+1}-1} \end{bmatrix} = 0 \\
 &\Rightarrow w_{n_{k+1}}^T v_{n_k} = w_{n_{k+1}}^T v_{n_{k+1}} = \cdots = w_{n_{k+1}}^T v_{n_{k+1}-1} = 0 \\
 &\Rightarrow v_{n_k}^T w_{n_{k+1}} = v_{n_{k+1}}^T w_{n_{k+1}} = \cdots = v_{n_{k+1}-1}^T w_{n_{k+1}} = 0 \\
 &\Rightarrow V_k^T w_{n_{k+1}} = 0.
 \end{aligned}$$

Assim, fazendo o produto interno em ambos os lados de (4.22) por V_k^T , temos

$$\begin{aligned}
 0 &= V_k^T w_{n_{k+1}} = V_k^T A^T w_{n_k} + V_k^T W_k y_k + \underbrace{V_k^T V_{k-1}}_0 y_{k-1} \\
 &\Rightarrow 0 = V_k^T A^T w_{n_k} + \delta_k y_k \Rightarrow y_k = -\delta_k^{-T} V_k^T A^T w_{n_k}.
 \end{aligned}$$

Como $W_{j+1}^T V_i = 0 \forall i \leq j$, tomando em particular $j = k$ e $i = k - 1$, obtemos

$$\begin{aligned}
 W_{k+1}^T V_{k-1} = 0 &\Rightarrow w_{n_{k+1}}^T v_{n_{k-1}} = w_{n_{k+1}}^T v_{n_{k-1}+1} = \cdots = w_{n_{k+1}}^T v_{n_{k-1}} = 0 \\
 &\Rightarrow v_{n_{k-1}}^T w_{n_{k+1}} = v_{n_{k-1}+1}^T w_{n_{k+1}} = \cdots = v_{n_{k-1}}^T w_{n_{k+1}} = 0 \\
 &\Rightarrow V_{k-1}^T w_{n_{k+1}} = 0.
 \end{aligned}$$

Fazendo então o produto interno em ambos os lados de (4.22) por V_{k-1}^T , obtemos

$$y_{k-1} = -\delta_{k-1}^{-T} V_{k-1}^T A^T w_{n_k}.$$

Portanto, o vetor regular $w_{n_{k+1}}$ do bloco W_{k+1} é dado por:

$$w_{n_{k+1}} = A^T w_{n_k} - W_k \delta_k^{-T} V_k^T A^T w_{n_k} - W_{k-1} \delta_{k-1}^{-T} V_{k-1}^T A^T w_{n_k}. \quad (4.23)$$

Para o cálculo dos vetores internos, queremos que esses vetores sejam escolhidos de forma a completarmos base para os subespaços de Krylov $K_n(v_1, A)$ e $K_n(w_1, A^T)$. Além disso, vamos tomar um vetor v_{n_k+i} do bloco V_k e w_{n_k+i} do bloco W_k da forma:

$$v_{n_k+i} = A v_{n_k+i-1} + V_{k-1} \tilde{x}_{k-1}, \quad (4.24)$$

$$w_{n_k+i} = A^T w_{n_k+i-1} + W_{k-1} \tilde{y}_{k-1}. \quad (4.25)$$

Para determinarmos \tilde{x}_{k-1} e \tilde{y}_{k-1} , impomos novamente a condição (4.19). Por

contas análogas ao caso anterior, obtemos:

$$\begin{cases} W_{k-1}^T v_{n_{k+i}} = 0, \\ V_{k-1}^T w_{n_{k+i}} = 0, \end{cases}$$

e então, fazendo o produto interno em ambos os lados de (4.24) por W_{k-1}^T e em (4.25) por V_{k-1}^T , obtemos:

$$\begin{cases} \tilde{x}_{k-1} = \delta_{k-1}^{-1} W_{k-1}^T A v_{n_{k+i-1}}, \\ \tilde{y}_{k-1} = \delta_{k-1}^{-1} V_{k-1}^T A^T w_{n_{k+i-1}}. \end{cases}$$

Portanto, os vetores internos são dados por:

$$\begin{cases} v_{n_{k+i}} = A v_{n_{k+i-1}} + V_{k-1} \delta_{k-1}^{-1} W_{k-1}^T A v_{n_{k+i-1}}, \\ w_{n_{k+i}} = A^T w_{n_{k+i-1}} + W_{k-1} \delta_{k-1}^{-1} V_{k-1}^T A^T w_{n_{k+i-1}}. \end{cases} \quad (4.26)$$

O que podemos nos perguntar agora é como decidir se vamos calcular no algoritmo um vetor regular ou um vetor interno. Para responder a essa pergunta, devemos voltar a condição (4.19).

Ao impormos a condição (4.19) para o algoritmo de Lanczos *look ahead*, exigimos que para calcularmos um vetor regular $v_{n_{k+1}}$, isto é, iniciarmos um novo bloco V_{k+1} , devemos ter $\delta_k = W_k^T V_k$ não singular. Caso δ_k seja singular, o bloco V_k é dito incompleto e devemos adicionar vetores internos a ele.

Assim, em cada passo do algoritmo, vamos verificar se δ_k é singular. Em caso positivo, calcularemos vetores regulares, iniciando um novo bloco. Do contrário, calcularemos vetores internos, que serão adicionados ao último bloco construído.

Para verificarmos se δ_k é não singular (em aritmética exata), basta verificarmos se o menor valor singular de δ_k é positivo, isto é, $\sigma_{\min}(\delta_k) > 0$. Para tanto, faremos a decomposição *SVD* de δ_k e verificaremos se o menor valor da diagonal da matriz S é positivo. Para evitar matrizes δ_k próximas de serem singulares, escolhemos um escalar de tolerância tol e exigimos que $(\delta_k) > tol$. Vamos escolher a constante sugerida por Parlett, Taylor e Liu [24]: $tol = \epsilon^{1/3}$

Agora, voltemos ao exemplo (4.14). Vimos que ao implementarmos tanto o algoritmo de Biortogonalização de Lanczos quanto o Algoritmo de Lanczos Normalizado neste exemplo, houve quebra do algoritmo no quarto passo, sendo gerados apenas 3 vetores de Lanczos a direita e 3 vetores de Lanczos a esquerda. A vantagem de normalizarmos os vetores foi que ganhamos maior estabilidade nos resultados obtidos.

Vimos também que pela análise dos determinantes das matrizes coeficientes do sistema linear (4.6), de diferentes dimensões, é possível verificar quando existe ou não FOP regular de um certo grau, o que nos permitiu analisar quantos vetores regulares e internos o algoritmo de Lanczos com *look ahead* geraria. Foi previsto que seriam construídos os seguintes blocos:

Algoritmo 9 Algoritmo de Lanczos *Look Ahead*

- 1: Escolha dois vetores $\tilde{v}_1 \in \mathbb{R}^N$ e $\tilde{w}_1 \in \mathbb{R}^N$ tais que $\langle \tilde{v}_1, \tilde{w}_1 \rangle \neq 0$
 - 2: $v_1 = \tilde{v}_1 / \|\tilde{v}_1\|$, $w_1 = \tilde{w}_1 / \|\tilde{w}_1\|$;
 - 3: Sejam $n_1 = 1$, $k = 1$, $v_0 = w_0 = 0$, $V_0 = W_0 = 0$, $\delta_0 = 1$;
 - 4: Sejam $V_1 = v_1$, $W_1 = w_1$, $\delta_1 = W_1^T V_1$;
 - 5: Seja $tol = \epsilon^{\frac{1}{3}}$
 - 6: **Para** $n = 1, 2, \dots, m$ **faça**
 - 7: Calcule *SVD* de δ_l ;
 - 8: **Se** $\min(\text{diag}(S)) > tol$ **então**
 - 9: $v_n = Av_n - V_k \delta_k^{-1} W_k^T Av_n - V_{k-1} \delta_{k-1}^{-1} W_{k-1}^T Av_n$;
 - 10: $w_n = A^T w_n - W_k \delta_k^{-T} V_k^T A^T w_n - W_{k-1} \delta_{k-1}^{-T} V_{k-1}^T A^T w_n$;
 - 11: Tome $n_{k+1} = n + 1$, $k = k + 1$, $V_k = W_k = \emptyset$
 - 12: **Senão**
 - 13: $\tilde{v}_{n+1} = Av_n - V_{k-1} \delta_{k-1}^{-1} W_{k-1}^T Av_n$;
 - 14: $\tilde{w}_{n+1} = A^T w_n + W_{k-1} \delta_{k-1}^{-1} V_{k-1}^T A^T w_n$;
 - 15: **Fim**
 - 16: Calcule $\rho_{n+1} = \|\tilde{v}_{n+1}\|$ e $\xi_{n+1} = \|\tilde{w}_{n+1}\|$;
 - 17: Se $\rho_{n+1} = 0$ | $\xi_{n+1} = 0$, quebre;
 - 18: Senão, tome:
 - 19: $v_{n+1} = \tilde{v}_{n+1} / \rho_{n+1}$, $w_{n+1} = \tilde{w}_{n+1} / \xi_{n+1}$;
 - 20: $V_k = [V_k \ v_{n+1}]$, $W_k = [W_k \ w_{n+1}]$, $\delta_k = W_k^T V_k$
 - 21: **Fim**
-

$$V_1 = \begin{bmatrix} v_1 \end{bmatrix}, \quad V_2 = \begin{bmatrix} v_2 \end{bmatrix}, \quad V_3 = \begin{bmatrix} v_3 \end{bmatrix}, \quad V_4 = \begin{bmatrix} v_4 & v_5 \end{bmatrix}, \quad \text{e} \quad V_5 = \begin{bmatrix} v_6 \end{bmatrix}$$

sendo apenas v_5 um vetor interno.

Vejam os resultados que obtemos com a implementação do algoritmo com *look ahead*. Primeiramente, apresentamos as matrizes δ_n geradas pelo algoritmo. Apenas para melhor visualização, consideramos os termos com módulo menor que 10^{-15} como nulos.

```

delta1 = 1.00000          delta5 =
delta2 = 0.12814         0.00000  -0.16667
delta3 = -0.0072044     -0.16667  0.66667
delta4 = 0               delta6 = 0.0067568
    
```

Note que no quarto passo, temos $\delta_4 = 0$ e então v_5 é calculado como vetor interno, como esperávamos. Como δ_5 é não singular, v_6 é calculado como vetor regular, iniciando um novo bloco. Abaixo seguem os blocos construídos pelo algoritmo.

V1 =	V2 =	V3 =	V4 =	V5 =
0.10483	0.9844038	-0.7774945	0.40825	-0.00000
0.20966	-0.1277631	0.6266374	-0.81650	0.40825
0.31449	-0.0963459	-0.0058022	0.40825	-0.81650
				0.738272
				-0.671156
				0.033558

Primeiramente, vamos analisar os menores valores singulares das matrizes coeficientes do sistema linear (4.6), como fizemos no Exemplo (4.14). Assim, a partir da análise da existência ou não de FOPs regulares de grau n , podemos prever os vetores que serão calculados como regulares e os que serão internos.

dim M	$\sigma_{min}(M)$	Singularidade
1	3.8500e+002	não singular
2	2.6551e+000	não singular
3	7.5066e-002	não singular
4	1.9847e-014	\approx singular
5	6.9682e-017	\approx singular
6	5.6472e-015	\approx singular
7	9.4156e-015	\approx singular
8	1.2111e-014	\approx singular
9	5.5126e-002	não singular
10	2.5000e+001	não singular
≥ 11	≈ 0	singular

Como as matrizes coeficiente de dimensões 4 a 8 são próximas de serem singulares, consideramos que não existem FOPs regulares de tais graus, e portanto, não calculamos vetores regulares de índices 5 a 9. Assim, se implementarmos o algoritmo de Lanczos *look ahead*, calcularemos vetores internos do passo 4 ao 8 (v_5, v_6, \dots, v_9) e vetores regulares nos demais passos. Note ainda que os vetores internos ocorrem sequencialmente, o que nos mostra que estarão todos em um mesmo bloco.

Observe também que caso implementássemos o algoritmo de Biortogonalização de Lanczos ou de Lanczos Normalizado, teríamos quebra em ambos os algoritmos no quarto passo, pois δ_4 é aproximadamente zero. Assim, seriam gerados apenas três vetores de Lanczos a direita (v_1, v_2 e v_3) e três a esquerda (w_1, w_2 e w_3). Com isso, vemos a relevância do Algoritmo com *look ahead*, que continua a calcular vetores de Lanczos mesmo com a quebra.

Ao implementarmos o algoritmo de Lanczos *look ahead*, obtivemos os seguintes valores¹ para $\delta_k = W_k^T V_k$:

```

delta1 = 1.000000
delta2 = 0.055939
delta3 = -0.0021604
delta4 = 0
delta5 =
    0  0
    0  0
    
```

¹Para efeito de visualização, consideramos os termos com módulo menor que 10^{-15} como nulos.

delta6 =

```

0  0  0
0  0  0
0  0  0

```

delta7 =

```

0.00000  0.00000  0.00000  0.00000
0.00000  0.00000  0.00000  0.00000
0.00000  0.00000  0.00000 -0.16667
0.00000  0.00000 -0.16667  0.66667

```

delta8 =

```

0.00000  0.00000  0.00000  0.00000  0.00000
0.00000  0.00000  0.00000  0.00000 -0.16667
0.00000  0.00000  0.00000 -0.16667  0.66667
0.00000  0.00000 -0.16667  0.66667 -1.00000
0.00000 -0.16667  0.66667 -1.00000  0.66667

```

delta9 =

```

0.00000  0.00000  0.00000  0.00000  0.00000 -0.16667
0.00000  0.00000  0.00000  0.00000 -0.16667  0.66667
0.00000  0.00000  0.00000 -0.16667  0.66667 -1.00000
0.00000  0.00000 -0.16667  0.66667 -1.00000  0.66667
0.00000 -0.16667  0.66667 -1.00000  0.66667 -0.16667
-0.16667  0.66667 -1.00000  0.66667 -0.16667  0.00000

```

delta10 = 0.0016502

Ao calcularmos os determinantes das matrizes δ_k , $k = 1, \dots, 10$, obtivemos:

k	$\det \delta_k$	Singularidade	k	$\det \delta_k$	Singularidade
1	1.000000	não singular	6	1.9576e-051	\approx singular
2	0.055939	não singular	7	8.2935e-036	\approx singular
3	-0.0021604	não singular	8	3.9525e-020	\approx singular
4	5.1225e-017	\approx singular	9	-2.1433e-005	não singular
5	-2.9856e-034	\approx singular	10	0.0016502	não singular

Assim, vemos que serão construídos vetores internos v_n , para $n = 5, 6, \dots, 9$, como esperávamos pelo cálculo dos menores valores singulares.

A condição de biortogonalidade (4.19) é visualizada pela matriz $D = W_n^T V_n$:

D =

Columns 1 through 6:

1.00000	0.00000	-0.00000	-0.00000	0.00000	0.00000
0.00000	0.05590	-0.00000	0.00000	0.00000	0.00000
-0.00000	0.00000	-0.00220	-0.00000	-0.00000	-0.00000
0.00000	-0.00000	0.00000	0.00000	-0.00000	0.00000
0.00000	-0.00000	0.00000	-0.00000	0.00000	-0.00000
0.00000	-0.00000	-0.00000	0.00000	-0.00000	0.00000
0.00000	-0.00000	0.00000	-0.00000	0.00000	-0.16670
0.00000	-0.00000	0.00000	0.00000	-0.16670	0.66670
0.00000	-0.00000	0.00000	-0.16670	0.66670	-1.00000
0.00000	-0.00000	0.00000	0.00000	0.00000	-0.00000

Columns 7 through 10:

0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000
0.00000	-0.00000	0.00000	-0.00000
-0.00000	0.00000	-0.16670	-0.00000
0.00000	-0.16670	0.66670	-0.00000
-0.16670	0.66670	-1.00000	0.00000
0.66670	-1.00000	0.66670	0.00000
-1.00000	0.66670	-0.16670	-0.00000
0.66670	-0.16670	0.00000	0.00000
0.00000	0.00000	0.00000	0.00170

A matriz D é uma matriz diagonal em blocos, como podemos melhor visualizar na Figura 4.2.2, que nos mostra a estrutura dessa matriz. Os blocos que aparecem na diagonal correspondem as matrizes $W_k^T V_k$, com $k = 1, \dots, 5$, são todos não singulares, o que nos mostra que a condição (4.19) é válida.

Note que durante o algoritmo, esses blocos δ_k podem ser temporariamente singulares. Enquanto um desses blocos for singular, será calculado um vetor interno no passo seguinte e o bloco crescerá em uma linha e uma coluna, até que vire não singular. Nesse momento, o bloco estará completo e iniciam-se novos blocos V_{k+1} e W_{k+1} e um novo bloco $\delta_{k+1} = W_{k+1}^T V_{k+1}$.

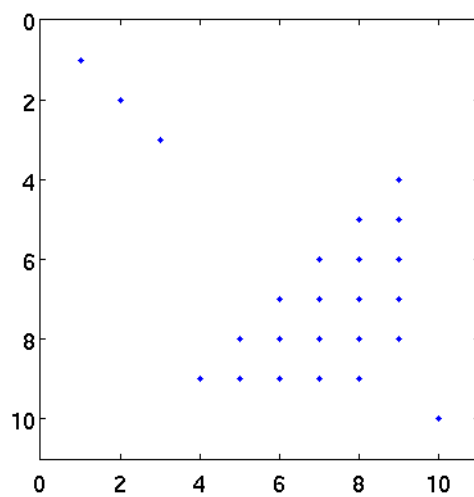


Figura 4.1: Estrutura da Matriz $D = W_n^T V_n$

Conclusão

Neste trabalho, estudamos alguns métodos de projeção baseados em subespaços de Krylov, tendo como principal enfoque o método de Lanczos não simétrico com estratégia *look ahead*.

Vimos que a estratégia *look ahead* nos permitiu superar quase-quebras no algoritmo de Biortogonalização de Lanczos, continuando o cálculo de vetores Lanczos a partir da construção de vetores internos e regulares. Nos exemplos numéricos, este resultado ficou claro: ao utilizarmos o Algoritmo de Biortogonalização de Lanczos em um exemplo particular, houve quebra no quarto passo, sendo gerados apenas 3 vetores Lanczos à direita e 3 à esquerda, enquanto no algoritmo com *look ahead* foram gerados n vetores Lanczos à direita e n à esquerda, sendo $n = \dim A$.

A condição $\sigma_{\min}(\delta_k) > tol$ foi utilizada no algoritmo Look Ahead como critério para construção de vetor regular, sendo $tol = \epsilon^{1/3}$ a constante sugerida por Parlett, Taylor e Liu [24]. Convém observar que Gutknecht mostra que apenas essa condição não é suficiente para garantirmos vetores Lanczos linearmente independentes (Ver Figura 4.1 em [17]) e apresenta mais duas condições para a construção de vetores regulares. Um estudo detalhado acrescentando tais condições seria uma proposta para trabalhos futuros.

Outra proposta para um trabalho futuro seria apresentarmos a demonstração da biortogonalidade entre os blocos construídos no algoritmo de Lanczos *look ahead* [14]. Neste trabalho, assumimos a condição de biortogonalidade entre os blocos e utilizamos casos particulares desta condição para determinarmos vetores que apareciam nas expressões dos vetores regulares e internos do algoritmo *look ahead*. Especificamente, utilizamos a biortogonalidade do vetor regular com os dois blocos calculados anteriormente e a biortogonalidade do vetor interno com apenas um bloco anterior.

Além das propostas acima apresentadas, podemos acrescentar ainda o estudo de aplicações do Algoritmo de Lanczos com *look ahead* a soluções de sistemas lineares [18], em particular o caso *QMR* [13]; técnicas de preconditionamento [18] e a tentativa de estender os métodos de Lanczos não simétricos para o caso da aritmética complexa [10].

Referências Bibliográficas

- [1] E. H. Ayachour. Avoiding look-ahead in the Lanczos method and Padé approximation. *Appl. Math.*, 26(1):33–62, 1999.
- [2] C. Baheux. New implementations of Lanczos method. *J. Comput. Appl. Math.*, 57(1-2):3–15, 1995.
- [3] C. Brezinski and M. Redivo Zaglia. Breakdowns in the computation of orthogonal polynomials. In *Nonlinear numerical methods and rational approximation*, pages 49–59. Kluwer, Dordrecht, 1994.
- [4] C. Brezinski and M. Redivo-Zaglia. A look-ahead strategy for the implementation of some old and new extrapolation methods. *Numer. Algorithms*, 11(1-4):35–55, 1996.
- [5] C. Brezinski, M. Redivo Zaglia, and H. Sadok. Avoiding breakdown and near-breakdown in Lanczos type algorithms. *Numer. Algorithms*, 1(3):261–284, 1991.
- [6] C. Brezinski, M. Redivo Zaglia, and H. Sadok. A breakdown-free Lanczos type algorithm for solving linear systems. *Numer. Math.*, 63(1):29–38, 1992.
- [7] C. Brezinski and H. Sadok. Lanczos-type algorithms for solving systems of linear equations. *Appl. Numer. Math.*, 11(6):443–473, 1993.
- [8] A. Draux. Formal orthogonal polynomials revisited. Applications. *Numer. Algorithms*, 11(1-4):143–158, 1996.
- [9] R. Fletcher. Conjugate gradient methods for indefinite systems. In *Numerical analysis (Proc 6th Biennial Dundee Conf., Univ. Dundee, Dundee, 1975)*, pages 73–89. Lecture Notes in Math., Vol. 506. Springer, Berlin, 1976.
- [10] R. W. Freund. Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices. *SIAM J. Sci. Statist. Comput.*, 13(1):425–448, 1992.
- [11] R. W. Freund. The look-ahead Lanczos process for nonsymmetric matrices and its applications. In J. D. Brown, editor, *Proceedings of the Cornelius Lanczos International Centenary Conference*, pages 33–47, Philadelphia, PA, 1994. SIAM.

- [12] R. W. Freund, M. H. Gutknecht, and N. M. Nachtigal. An implementation of the look-ahead Lanczos algorithm for non-Hermitian matrices. *SIAM J. Sci. Comput.*, 14(1):137–158, 1993.
- [13] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numer. Math.*, 60(3):315–339, 1991.
- [14] R. W. Freund and H. Y. Zha. A look-ahead algorithm for the solution of general Hankel systems. *Numer. Math.*, 64(3):295–321, 1993.
- [15] A. Greenbaum. *Iterative methods for solving linear systems*. SIAM, Philadelphia, PA, 1997.
- [16] M. H. Gutknecht. A completed theory of the unsymmetric Lanczos process and related algorithms. I. *SIAM J. Matrix Anal. Appl.*, 13(2):594–639, 1992.
- [17] M. H. Gutknecht. A completed theory of the unsymmetric Lanczos process and related algorithms. I. *SIAM J. Matrix Anal. Appl.*, 13(2):594–639, 1992.
- [18] M. H. Gutknecht. Lanczos-type solvers for nonsymmetric linear systems of equations. *Acta Numer.*, 6:271–397, 1997.
- [19] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, 49:409–436 (1953), 1952.
- [20] N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, Philadelphia, PA, 1996.
- [21] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Research Nat. Bur. Standards*, 45:255–282, 1950.
- [22] C. Lanczos. Solution of systems of linear equations by minimized-iterations. *J. Research Nat. Bur. Standards*, 49:33–53, 1952.
- [23] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix. *Math. Comp.*, 31(137):148–162, 1977.
- [24] B. N. Parlett, D. R. Taylor, and Z. A. Liu. A look-ahead Lanczos algorithm for unsymmetric matrices. *Math. Comp.*, 44(169):105–124, 1985.
- [25] V. Puzyrev, J. Koldan, J. de la Puente, G. Houzeaux, M. Vázquez, and J. M. Cela. A parallel finite-element method for three-dimensional controlled-source electromagnetic forward modelling. *Geophysical Journal International*, 193(2):678–693, 2013.

- [26] Y. Saad. Krylov subspace methods for solving large unsymmetric linear systems. *Math. Comp.*, 37(155):105–126, 1981.
- [27] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.
- [28] Y. Saad and M. H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.
- [29] Y. Saad and H. A. van der Vorst. Iterative solution of linear systems in the 20th century. *J. Comput. Appl. Math.*, 123(1-2):1–33, 2000.
- [30] C. H. Tong and Q. Ye. A linear system solver based on a modified Krylov subspace method for breakdown recovery. *Numer. Algorithms*, 12(1-2):233–251, 1996.
- [31] H. A. van der Vorst. Bi-CGSTAB: a fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 13(2):631–644, 1992.
- [32] Q. Ye. A breakdown-free variation of the nonsymmetric Lanczos algorithms. *Math. Comp.*, 62(205):179–207, 1994.