Universidade Estadual de Campinas

Projeto 2

Análise Numérica

RA	*Nome*
118122	Matheus Marques
156231	Leonardo Uchoa
155738	Hugo Calegari

Junho, 2016

Sumário

1 Introdução

Este trabalho visa discutir e reforçar tópicos importantes da disciplina de Análise Numérica e suas diversas aplicações. Através de definições dos teoremas e de rigorosas demonstrações, de aplicações cotidianas e ilustrações, foram discutidos os temas da Decomposição SVD, de Schur e Espectral. Será também estudado e discutido a Equação e da Lei de Inércia de Sylvester, visando mostrar sua importância para aspectos teóricos, principalmente, que servem como base em várias questões teóricas, mas que também desaguam em inúmeras aplicações.

Adicionalmente, passaremos a conhecer mais profundamente, estendendo o conhecimento adquirido em sala de aula, o Algorítmo QR e a Decomposição QR (que são fundamentalmente diferentes), trabalhando, respectivamente, questões relacionadas à (novamente) autovalores e a aplicações que podem ser vista em, por exemplo, estatística - ao debruçarmos sobre ferramenta de Quadrados Mínimos. Por fim, será estudado métodos para aproximação de solução de sistemas não-lineares, utilizando os métodos do Ponto Fixo, de Newton e de Broyden, cujas aplicações serão em exercícios treino e no ramo de estatística.

2 A Decomposição em Valores Singulares

2.1 Interpretação Geométrica 1

Foi escolhido começar com a intuição geométrica do problema pois alguns tópicos da demonstração da decomposição utilizam argumentos geométricos. Portanto é preferível, primeiramente, compreender a construção básica e o contexto em que será explicada a demonstração (esta primera abordagem geométrica segue [TB97]).

A Decomposição em Valores Singulares (ou Singular Value Decomposition, "SVD") usa como motivação o fato de que a imagem de uma esfera unitária S sobre o efeito de uma matriz mxn é uma híper-elípse, como a figura 1 a seguir nos mostra.

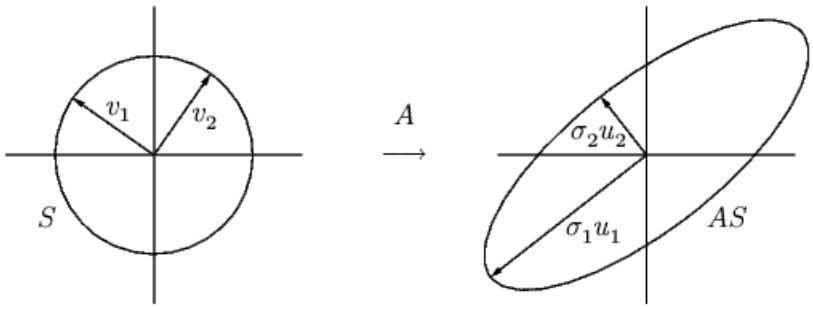

Figura 1: SVD de uma matriz $2x2$

O efeito de A em S pode ser claramente percebido como os atos de "esticar"e rotacionar S. Os fatores que determinam a magnitude do ato de esticar sobre S, com direções $v_1, \ldots, v_m$, são $\sigma_1, \ldots, \sigma_m$ (cujos alguns podem vir a ser 0) e as direções são $u_1, \ldots, u_m$. Ou seja, gostaríamos que o efeito de A em v_i fosse $\sigma_i u_i$,

$$Av_i = \sigma_i v_i \quad \equiv \quad AV = U\Sigma \quad \equiv \quad A = U\Sigma V^*.$$

onde,

$$\begin{aligned} U \in \mathbb{C}^{mxm} &\quad \text{é unitária,} \\ V \in \mathbb{C}^{nxn} &\quad \text{é unitária,} \\ \Sigma \in \mathbb{R}^{mxn} &\quad \text{é diagonal.} \end{aligned}$$

Portanto, a primeira vista, podemos encarar o SVD como uma decomposição para encontrar quem são os vetores que geram AS e seus respectivos fatores de escala, o que caracteriza encontrar a elípse descrita pela figura 1.

2.2 Decomposição em Valores Singulares

Teorema 2.1. ***(Decomposição em Valores Singulares)*** *Se $A \in \mathbb{R}^{mxn}$, então existem matriz ortogonais*

$$U = [u_1, \ldots, u_m] \in \mathbb{R}^{mxm} \quad e \quad V = [v_1, \ldots, v_n] \in \mathbb{R}^{nxn}$$

tal que

$$U^T AV = \Sigma \in \mathbb{R}^{mxn} \quad , \quad p = min\{m, n\}$$

onde $\Sigma = diag(\sigma_1, \ldots, \sigma_p) \in \mathbb{R}^{mxn}$ e $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$

Demonstração. A demonstração do resultado segue as propostas de [TB97] e de [Dem97], considerando a idéia principal do primeiro autor e detalhes de aprofundamento do segundo autor.

A prova será divida em duas partes, as demonstrações de existência e unicidade, que têm a característica de serem construtivas. Primeiramente começamos com a de existência, cuja idéia é usar indução nas dimensões m e n da matriz A, assumindo que SVD vale para $(m-1)x(n-1)$ e provando para mxn. Escolhemos v tal que $||v||_2 = 1$ e $||A||_2 = ||Av||_2 \geq 0$, o que é possível pois, já que $||A||_2$ é a norma do operador, ou seja

$$||A||_2 = \max_{x \neq 0} \frac{||Ax||}{||x||} = \max_{||v||=1} ||Av||_2.$$

Adicionalmente, escolha $u = \frac{Av}{||Av||_2}$ um vetor unitário e $\tilde{U}$, $\tilde{V}$ de forma que $U = [u, \tilde{U}] \in \mathbb{R}^{mxm}$ e $V = [v, \tilde{V}] \in \mathbb{R}^{nxn}$ sejam ortogonais. Portanto

$$U^*AV = \begin{bmatrix} u^* \\ \tilde{U}^* \end{bmatrix} A \begin{bmatrix} v & \tilde{V}^* \end{bmatrix} = \begin{bmatrix} u^*Av & u^*A\tilde{V} \\ \tilde{U}^*Av & \tilde{U}^*A\tilde{V}^* \end{bmatrix}.$$

Para o caso de $m = n = 1$, ao lembrarmos que $u = \frac{Av}{||Av||_2}$ e que u é ortogonal (mais especificamente, que matrizes ortogonais são invariantes à norma euclidiana), obtemos

$$u^*Av = \frac{(Av)^*Av}{||Av||_2} = \frac{||Av||_2^2}{||Av||_2} = ||Av||_2 = ||A||_2 = \sigma$$

e também $\tilde{U}^*Av = \tilde{U}^*u||Av||_2 = \langle \tilde{U}, u\rangle ||Av||_2 = 0$ (u e $\tilde{U}^*$ são ortogonais, pois são colunas de U e U é ortogonal). Adicionalmente, devemos ter que $u^*A\tilde{V} = 0$ pois, caso contrário,

$$\sigma = ||A||_2 = ||U^*AV||_2 \geq |||[1, 0, \ldots, 0]U^*AV||_2 = ||\sigma[u^*A\tilde{V}]|| > \sigma$$

, o que é uma contradição[1]. Logo

$$U^*AV = \begin{bmatrix} \sigma & 0 \\ a & \tilde{U}^*A\tilde{V} \end{bmatrix} = \begin{bmatrix} \sigma & 0 \\ 0 & \tilde{A} \end{bmatrix}.$$

Podemos agora aplicar a indução em $\tilde{A}$ para obter $\tilde{A} = U_1\Sigma_1V_1^*$, onde U_1 $in\mathbb{R}^{(n-1)x(n-1)}$, $\Sigma_1 \in \mathbb{R}^{(n-1)x(n-1)}$ e $V_1 \in \mathbb{R}^{(n-1)x(n-1)}$. Então

$$U^*AV = \begin{bmatrix} \sigma & 0 \\ 0 & U_1\Sigma_1V_1^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix} \begin{bmatrix} \sigma & 0 \\ 0 & \sigma_1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}^*.$$

ou equivalentemente,

$$A = \left(U \begin{bmatrix} 1 & 0 \\ 0 & U_1 \end{bmatrix}\right) \begin{bmatrix} \sigma & 0 \\ 0 & \sigma_1 \end{bmatrix} \left(V \begin{bmatrix} 1 & 0 \\ 0 & V_1 \end{bmatrix}\right)^*,$$

que é a decomposição desejada.

Agora vamos partir para a unicidade, que utiliza amplamente a interpretação geométrica que a decomposição fornece. Se os comprimentos dos semieixos da híper-elípse são distintos, então os os próprios semieixos serão determinados pela geometria. Algebricamente, a argumentação é baseada nessa interpretação. Primeiramente, note que σ_1 é unicamente determinado de forma que $\sigma = ||A||_2$, como visto na demonstração anterior. Agora, suponha que além de v, existe outro vetor linearmente independente w com $||w||_2 = 1$ e $||Aw||_2 = 1$. Defina, também, um vetor unitário v_1, ortogonal a v, como combinação linear de v e w,

$$v_1 = \frac{w - (v^*w)v}{||w - (v^*w)v||_2}.$$

Já que $||A||_2 = \sigma$, $||Av_1||_2 \geq \sigma$; mas a desigualdade deveria ser uma igualdade pois, como $w = vc + v_1s$ para constantes c e s (escolhidas de forma que $|c|^2 + |s|^2 = 1$), teríamos $||Aw||_2 \geq \sigma$. Este vetor v_1 é o segundo vetor singular direito de A correspondente ao valor singular σ, que irá guiar ao vetor y (igual às últimas n-1 componentes de V_1^*v1) com $||y||_2 = 1$ e $||\tilde{A}y||_2 = \sigma$. Assim, concluímos que se o vetor singular v não for único, o valor singular σ também não será. Para completar a demonstração, já que v é unicamente determinado, o espaço ortogonal dos vetores v_j é unicamente definido e, portanto, os valores singulares σ_j também, o que completa a demonstração. □

Adicionalmente, vale ressaltar que em momento algum foi citada a restrição de que a matriz A deve ser definida-positiva e simétrica. Entretanto, se a matriz tiver estas boas propriedades, podemos obter resultados mais poderosos, a saber, as Decomposições SVD, de Schur e Espectral coincidem - ao fim da sequência de exercícios 1-2, o fato de que estas decomposições são fundamentalmente diferentes ficará claro; também será explicado o porquê de as fatorações coincidirem.

[1] foi utilizado que $||A||_2 = \max_{x\neq 0} \frac{||Ax||_2}{||x||_2} = \sqrt{\lambda_{\max}(A^*A)}$

2.3 Interpretação Geométrica 2

Agora que demonstramos o Teorema 1, temos mais experiência sobre matrizes e transformações lineares, o que permite um aprofundamento na questão geométria do SVD. Algebricamente, a missão do SVD é buscar mudanças de bases no domínio e na imagem (que geralmente são diferentes) da transformação linear

$$A : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$
$$A : x \longmapsto Ax = y.$$

de forma que a matriz se torne diagonal. Em outras palavras, encontrar um sistema de coordenadas para $\mathbb{R}^n$ ($span(\mathbb{R}^n) = \{v_1, \ldots, v_n\}$) e outro sistema de coordenadas para $\mathbb{R}^m$ ($span(\mathbb{R}^m) = \{u_1, \ldots, u_m\}$) de forma que A seja diagonal (Σ), *i.e*, que A mapeie o vetor $x = \sum_{i=1}^{n} \beta_i v_i$ em $y = Ax = \sum_{i=1}^{n} \sigma_i \beta_i u_i$.

Esta é a teoria algébrica por detrás do SVD. Entretanto, podemos dar um passo adiante e enxergar o sentido geométrico escondido nesta transformação linear. Assumindo que A é não-singular, seja S a esfera unitária em $\mathbb{R}^n$, $S = \{x \in \mathbb{R}^n : ||x||_2 = 1\}$ e $A \cdot S = \{Ax : x \in \mathbb{R}^n \text{ e } ||x||_2 = 1\}$ ser a imagem de S sobre A. Considere os seguintes fatos

1. $V^* S = S$
2. $w \in \Sigma S \Leftrightarrow ||\Sigma^2 w||_2^2 = 1$.

O primeiro fato leva em consideração que V é ortogonal e, portanto, mapeia vetores unitários em vetores unitários. Mais que isso : como visto em [Wat10], matrizes unitárias preservam ângulos e comprimentos, o que implica que a ação de V^* sobre S preserva completamente a estrutura de S, somente a rotacionando (pois seus vetores linha são ortonormais por construção). O segundo item nos mostra o efeito da matriz Σ na esfera unitária S, pois gostaríamos de saber o que resulta da aplicação de um conjunto de valores singulares à um conjunto de vetores-coordenada ortogonais entre sí. O resultado é que se

$$w \in \Sigma S \Leftrightarrow ||\Sigma^2 w||_2^2 = \sum_{i=1}^{n} \left(\frac{w_i}{\sigma_i} \right)^2 = 1,$$

que define um elipsóide com eixos principais $\sigma_i e_i$, onde e_i é a i-ésima coluna da matriz identidade, o que nos leva a concluir que a matriz está centrada na origem. Por fim, ao multiplicar cada $w = \Sigma v$ por U, temos uma rotação da elípse se forma que cada coluna e_i se torna uma coluna u_i, ou seja, U é uma matriz de rotação e Σ a matriz de "esticamento". A figura 2, que se encontra em [Dem97], nos mostra os efeitos de cada matriz na construção da elípse.

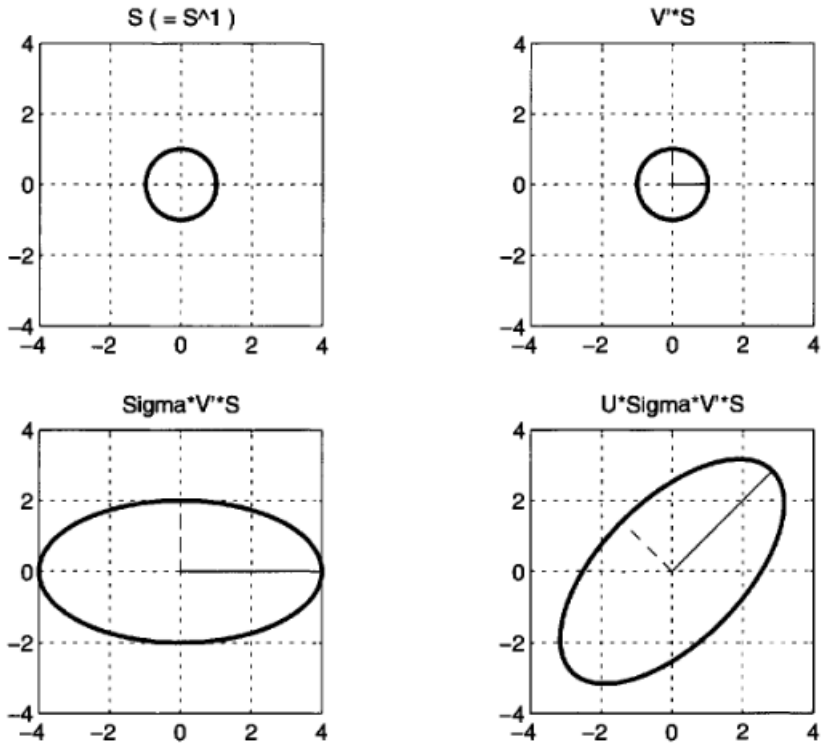

Figura 2: Processo de construção da elípse Ax

Como ilustrado, o processo consistem em, primeiramente, aplicarmos V^* em S para rotacionar a esfera, utilizar uma transformação de escala com Σ para alterar o formato da esfera e outra rotação com U para encontrar as direções exatas.

2.4 Compressão de Imagens Usando SVD

É de interesse estudar o caso de compressão de imagens coloridas utilizando técnicas de SVD. Para isso, foram utilizadas as dicas mencionadas, além de figuras coloridas.

A técnica de decomposição em valores singulares é útil aqui pois ela remove detalhes desnecessários, mantendo informação importante. Isso resulta em uma compressão do arquivo, reduzindo as especificações de armazenamento.

Como imagens digitais são armazenadas como matrizes, em escala de cinza, o número associado varia em um grau que vai de branco ao preto. Basicamente, a técnica minimiza a necessidade de números diferentes (ou escalas) através da minimização dos números em valores singulares que a matriz correspondente possui.

De forma geral, a seguinte relação é válida.

- Uma imagem M, sem nenhum tratamento, e de tamanho $m \times n$ pixels, requer $z_M = mn$ de espaço de armazenamento;

- Após a aplicação de SVD, a imagem toma até $z_M = k(m+n+1)$, onde k é o posto da matriz M.

Conclui-se que a aplicação de SVD reduz o posto da matriz, logo as especificações de armazenamento também são reduzidas.

Inicialmente, para aplicar essa técnica é necessário converter a imagem colorida inicialmente para uma escala de cinza. Isso foi realizado utilizando a imagem do tigre e do gato. Nosso caso é aplicar a técnica para uma imagem colorida, diretamente.

Computadores armazenam imagens coloridas em escalas de Vermelho, Verde e Azul (RGB). Logo, quando a técnica de SVD é aplicada em imagens coloridas, é necessária a aplicação em cada uma das imagens separadamente.

Através da função disponibilizada no anexo, vamos interpretar duas imagens com dimensões distintas. Antes, vamos introduzir como o cálculo será feito, seguindo o que foi recomendado na referência.

O valor máximo do posto que resulta em ganhos na compressão pode ser calculada através de:

$$k < \frac{mn}{m+n+1}$$

Logo, k sendo o posto deve ser menor do que esse valor.

A seguir, foram feitas duas implementações para imagens de dimensões diferentes. Note as diferenças entre os postos e as quantidades de armazenamento. Os valores de k foram escolhidos especificamente para refletir a percentagem de compressão.

2.4.1 Natureza

Posto	Armazenamento	Percentagem do Original
253	113850	100%
81	57024	50.09%
40	28160	24.73%
20	14080	12.37%

Tabela 1: Tabela Comparativa Para Natureza

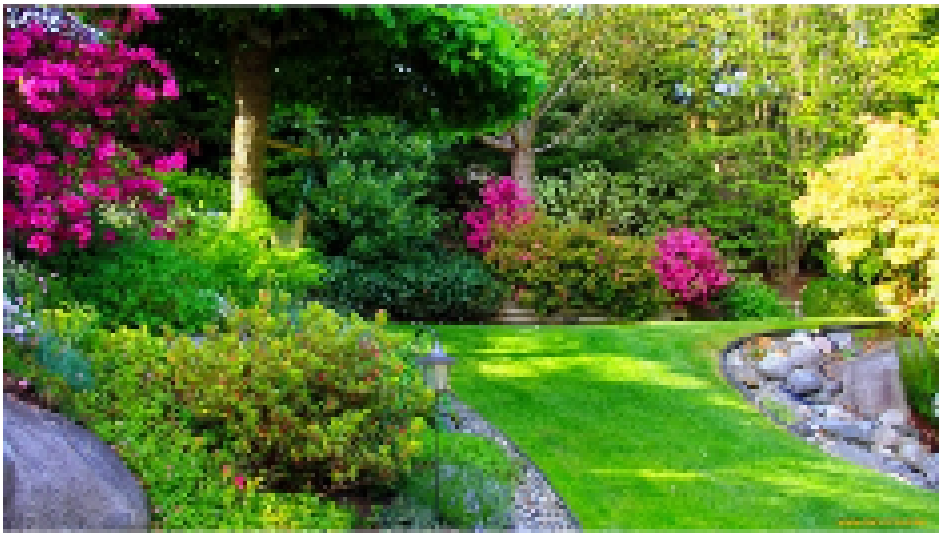

Figura 3: Imagem de Natureza Original

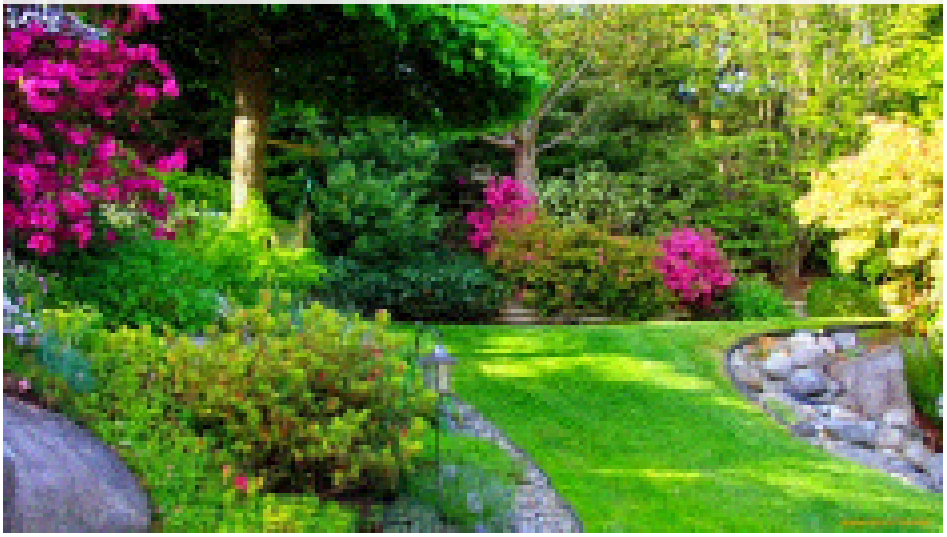

Figura 4: Imagem de Natureza com compressão a 50

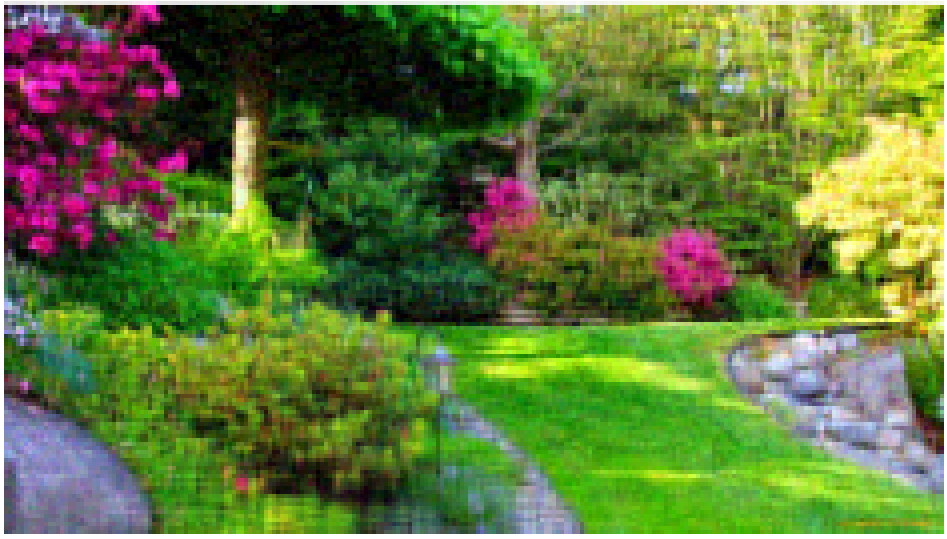

Figura 5: Imagem de Natureza com compressão a 25

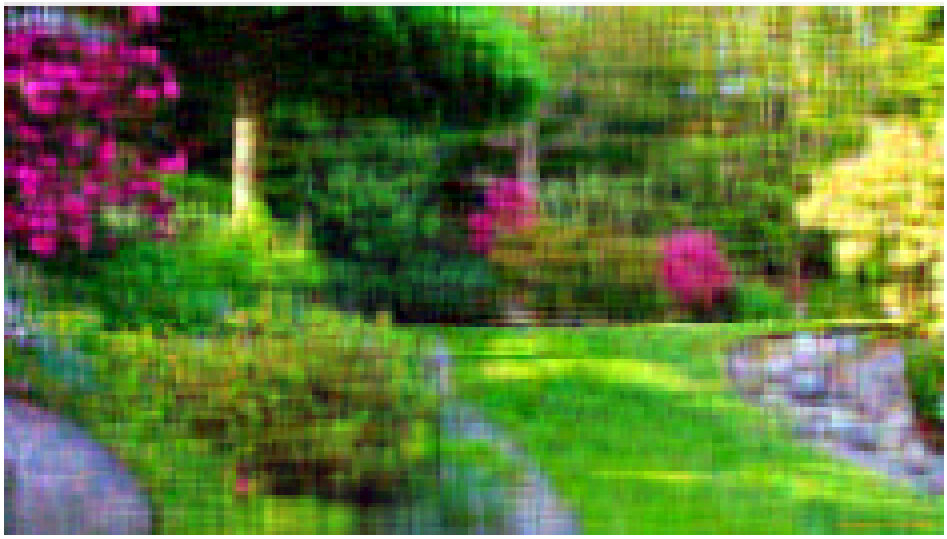

Figura 6: Imagem de Natureza com compressão a 12.5

2.4.2 Praia

Posto	Armazenamento	Percentagem do Original
2592	10077696	100%
777	5035737	49.97%
389	2521109	25.02%
194	1257314	12.48%

Tabela 2: Tabela Comparativa Para Praia

Figura 7: Imagem de Praia Original

Figura 8: Imagem de Praia com compressão a 50

Figura 9: Imagem de Praia com compressão a 25

Figura 10: Imagem de Praia com compressão a 12,5

2.4.3 Conclusão

Quanto menor forem as dimensões da imagem (a figura natureza, por exemplo), mais visível fica a distorção. Uma imagem com maior quantidade de pixels consegue manter uma maior fluidez mesmo após uma compressão pesada.

3 Teorema de Schur e Teorema Espectral

O teorema espectral é, reconhecidamente, um dos resultados mais conhecidos em Álgebra Linear, nos trazendo, consigo, as noções de autovalor e autovetor, que têm alto valor teórico e deságuam em diversas aplicações - como em estatística, física, engenharía e matemáticas puras e aplicadas. Por ser uma ferramente altamente explorada, seu entendimento é feito necessário, o que justifica o estudo de sua base teórica.

Não obstante, temos a Decomposição de Schur, que pode ser entendida como uma extensão do Teorema Espectral -pois pode ser aplicada a qualquer matriz quadrada- e que tem papel fundamental no Algorítmo QR (ver *e.g* [Wat02], páginas 356-358), onde o problema a ser atacado é a busca de autovalores. Desta forma, vamos agora estudar os teoremas e seus demonstrações.

3.1 Teorema de Schur

Teorema 3.1. ***Decomposição de Schur*** *Seja $A \in \mathbb{C}^{nxn}$. Então existem $U \in \mathbb{C}^{nxn}$, unitária, e $T \in \mathbb{C}^{nxn}$, triangular superior, tal que $T = U^*AU$.*

Demonstração. A demonstração do resultado é semelhante à prova da Decomposição SVD, logo, novamente iremos fazer uma indução sobre as dimensões da matriz A, n. Para o caso $n = 1$, temos que $A = a \in \mathbb{R}, \quad U = u \in \mathbb{R}, \quad T = t \in \mathbb{R}$, o que implica que $a = utu^{-1} = t$. Sendo $A \in \mathbb{C}^{kxk}$, λ um autovalor de A com autovetor ν associado, escolhido de forma que $||\nu||_2 = 1$ e $U_1 = [\nu, W]$, onde $U \in \mathbb{C}^{kxk}$ é uma matriz ortogonal (cujos vetores coluna são, por exemplo, uma base de $\mathbb{C}^{kxk}$ com ν sendo a primeira coluna) e $W \in \mathbb{C}^{kx(k-1)}$ é uma submatriz de U, o que implica que $\langle W, \nu \rangle = 0$. Definindo $A_1 = U_1^T A U_1$, vamos à indução propriamente dita :

$$A_1 = \begin{bmatrix} \nu^* \\ W^* \end{bmatrix} A \begin{bmatrix} \nu & W \end{bmatrix} = \begin{bmatrix} \nu^* A\nu & \nu^* AW \\ W^* A\nu & W^* AW \end{bmatrix} \overset{A\lambda = \lambda\nu}{=} \begin{bmatrix} \nu^* \lambda\nu & \nu^* AW \\ W^* \lambda\nu & W^* AW \end{bmatrix} \underset{\langle W,\nu\rangle = 0}{\overset{\nu^*\nu = 1}{=}} \begin{bmatrix} \lambda & \nu^* AW \\ 0 & W^* AW \end{bmatrix};$$

$$A_1 = \begin{bmatrix} \lambda & w^* \\ 0 & \hat{A} \end{bmatrix}.$$

onde $w = W^* A^* \nu$ e $\hat{A} = W^* AW$. Nesta parte, utilizamos a hipótese indutiva de que a Decomposição de Schur vale para $A \in \mathbb{C}^{kxk}$, nos garantindo a existência de $\hat{T} \in \mathbb{C}^{(k-1)x(k-1)}$, que é tringular superior (hipótese do teorema) com $\hat{T} = \hat{U}_2^* \hat{A} \hat{U}_2$, $\hat{U}_2 \in \mathbb{C}^{(k-1)x(k-1)}$ unitária. Para fazer o passo indutivo, vamos partir do caso $n = k$ e utilizar a hipótese indutiva, como planejado, mas para isso, devemos primeiro definir a matriz U_2 para fazer a indução. Sendo

$$U_2 = \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2 \end{bmatrix},$$

temos que

$$U_2^* A_1 U_2 = \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2^* \end{bmatrix} A_1 \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2^* \end{bmatrix} \begin{bmatrix} \lambda & w^* \\ 0 & \hat{A} \end{bmatrix} \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2 \end{bmatrix} = \begin{bmatrix} \lambda & w^* \\ 0 & \hat{U}_2^* \hat{A} \end{bmatrix} \begin{bmatrix} 1 & 0^{tr} \\ 0 & \hat{U}_2 \end{bmatrix} =$$

$$= \begin{bmatrix} \lambda & w^* \hat{U}_2 \\ 0 & \hat{U}_2^* \hat{A} \hat{U}_2 \end{bmatrix} = \begin{bmatrix} \lambda & w^* \hat{U}_2 \\ 0 & \hat{T} \end{bmatrix} = T.$$

Por fim, $T = U_2^* A_1 U_2 = U_2^* U_1^* A U_1 U_2 = (U_1 U_2)^* A U_1 U_2$. Se definirmos $U = U_1 U_2$, então $T = U^* A U$, como gostaríamos.

Vale ressaltar que, ao fim da demonstração, fica claro a sua similaridade com a prova da Decomposição SVD, pois os passos principais da indução são essencialmente os mesmos.

□

3.2 Decomposição Espectral

Teorema 3.2. ***Teorema Espectral para Matrizes Reais e Simétricas*** *Seja $A \in \mathbb{R}^{nxn}$ uma matriz simétrica, então existem $U \in \mathbb{R}^{nxn}$, ortogonal, e $D \in \mathbb{R}^{nxn}$, diagonal, tal que $D = U^T A U$.*

Demonstração. A demonstração deste resultado é a mesma que a do Teorema 2, exceto pelo fato de que A é real e simétrica. Logo, novamente iremos fazer indução sob n, a começar por $n = 1$, que é exatamente igual à feita no Teorema 2 (a única diferença é que U é ortogonal real, ou seja $U^T = U^{-1}$). Para o caso geral, novamente assumimos que o resultado é válido para o caso de $n = k - 1$ e provamos para $n = k$.

Seguindo as idéias das demonstrações anteriores, sabemos que $\lambda \in \mathbb{R}$ (λ real, pois A é simétrica e real) é um autovalor de A, com autovetor $\nu \in \mathbb{R}^k$ associado, escolhido de forma que $||\nu||_2 = 1$. Adicionalmente, considere U_1 uma matriz ortogonal com a primeira coluna sendo ν e as outras como um complemento para uma base ortogonal do $\mathbb{R}^k$, *i.e*, $U_1 = [\nu, W]$, onde $W \in \mathbb{R}^{kx(k-1)}$ é uma submatriz de U.

Agora, vamos definir $A_1 = U_1^T A U_1$, e ver como ficará a estrutura de A_1.

$$A_1 = \begin{bmatrix} \nu^t \\ W^t \end{bmatrix} A \begin{bmatrix} \nu & W \end{bmatrix} = \begin{bmatrix} \nu^t A\nu & \nu^t AW \\ W^t A\nu & W^t AW \end{bmatrix} \overset{A\lambda=\lambda\nu}{=} \begin{bmatrix} \nu^t \lambda\nu & (W^t A^t \nu)^t \\ W^t \lambda\nu & W^t AW \end{bmatrix} \underset{\langle W,\nu\rangle=0}{\overset{\nu^t\nu=1}{=}} \begin{bmatrix} \lambda & (W^t A^t \nu)^t \\ 0 & W^t AW \end{bmatrix};$$

$$\begin{bmatrix} \lambda & (W^t A^t \nu)^t \\ 0 & W^t AW \end{bmatrix} \overset{A=A^t}{=} \begin{bmatrix} \lambda & (W^t A\nu)^t \\ 0 & W^t AW \end{bmatrix} = \begin{bmatrix} \lambda & (W^t \lambda\nu)^t \\ 0 & W^t AW \end{bmatrix} = \begin{bmatrix} \lambda & 0^t \\ 0 & W^t AW \end{bmatrix} = \begin{bmatrix} \lambda & 0^t \\ 0 & \hat{A} \end{bmatrix}.$$

de forma que $\hat{A} = W^t AW \in \mathbb{R}^{(k-1)x(k-1)}$. Ao utilizar a hipótese de indução, somos garantidos as matrizes $\hat{U}_2$, ortogonal, e $\hat{D}$, diagonal, de ordens $k-1$ tais que $\hat{D} = \hat{U}_2^t \hat{A} \hat{U}_2$. Entretanto, para realizarmos a indução, partindo de $U_2^t A_1 U_2$, precisamos primeiro definir U_2 :

$$U_2 = \begin{bmatrix} 1 & 0^t \\ 0 & \hat{U}_2 \end{bmatrix}.$$

Logo,

$$U_2^t A_1 U_2 = \begin{bmatrix} 1 & 0^t \\ 0 & \hat{U}_2^t \end{bmatrix} A_1 \begin{bmatrix} 1 & 0^t \\ 0 & \hat{U}_2 \end{bmatrix} = \begin{bmatrix} 1 & 0^t \\ 0 & \hat{U}_2^t \end{bmatrix} \begin{bmatrix} \lambda & 0^t \\ 0 & \hat{A}^t \end{bmatrix} \begin{bmatrix} 1 & 0^t \\ 0 & \hat{U}_2 \end{bmatrix} = \begin{bmatrix} \lambda & 0^t \\ 0 & \hat{U}_2^t \hat{A} \hat{U}_2 \end{bmatrix} \underset{\text{indutivo}}{\overset{\text{passo}}{=}} \begin{bmatrix} \lambda & 0^t \\ 0 & \hat{D} \end{bmatrix} = D,$$

e portanto $D = U_2^t A_1 U_2 = U_2^t U_1^t A U_1 U_2 = (U_1 U_2)^t A U_1 U_2$. Se definirmos $U = U_1 U_2$, juntamos todas partes da demonstração para concluir que $D = U^t AU$ e finalizar a prova do teorema.

Entretanto, o teorema pode ser extendido, garantido também que as colunas da matriz U são os autovetores de A e os elementos da diagonal de A são os autovalores de A. Sua demonstração utiliza o Teorema 8.2.1, em [Pul15] (página 495), com a modificação que as colunas de U são vetores ortogonais dois a dois e, portanto, linearmente independentes (como pode ser visto em [JB80], página 225), garantindo a validade da extensão do teorema.

□

4 Considerações Adicionais a Respeito das Fatorações: teorias, interpretações e aplicações

Ao fim desta demonstração, é aberto um espaço para uma importante discussão : a relação entre a

1. Decomposição SVD,
2. Decomposição de Schur,
3. Decomposição Espectral.

Como pôde ser visto, ao longos das três provas, a essência de suas demonstrações são muito similares, o que não é por acaso. Ao observar os enunciados e, atentamente, às suas condições, nota-se que : se começarmos da decomposição 3 até a 1, as hipóteses vão se tornando mais gerais e mais abrangentes. A ligação entre as decomposições está em A, a transformação $x \longmapsto Ax = y$. Se a matriz for simétrica e positiva definida, então, como citado anteriormente, as decomposições 3, 2 e 1 coincidem. O caso de simetria implica que $\sigma_i = |\lambda_i|$, que $\nu_i = sinal(\lambda_i)u_i$, onde $sinal(0) = 1$, (ver *e.g*, [Dem97] páginas 110-111, teorema 3.3) e que $\lambda_i \in \mathbb{R}$ (ver *e.g*, [Wat10] página 339, corolário 5.4.13). Se for simétrica e também definida positiva, as decomposições coincidem pois, neste caso, $U = V$, e consequentemente $A = U\Sigma U^T$.

A Decomposição SVD também tem alta importância no estudo do teorema do posto e da nulidade, nos ajudando a complementar o entendimento sobre mapeamentos lineares, domínio, imagem e postos de matrizes. Uma boa ilustração deste resultado pode ser encontrando no site do Professor Gilbert Strang (famoso escritor de livros relacionados a Álgebra Linear), do Instituto de Tecnologia de Massachusetts, (vide [oT], na capítulo 7 : "Matemática Aplicada e A^TA") e em [Wat02], páginas 263 e 264. A saber, seja $r = rank(A)$ (lembrando que $rank(A) = rank(A^T)$), a ilustração nos diz que

$$\begin{aligned}
\mathcal{R}(A) &= span\{u_1, \ldots, u_r\} \\
\mathcal{N}(A) &= span\{v_{r+1}, \ldots, v_m\} \\
\mathcal{R}(A^T) &= span\{v_1, \ldots, v_r\} \\
\mathcal{N}(A^T) &= span\{u_{r+1}, \ldots, u_n\}
\end{aligned}$$

implicando que $\mathcal{R}(A^T) = \mathcal{N}(A)^\perp$ e que $\mathcal{R}(A) = \mathcal{N}(A)^\perp$, onde $\mathcal{R}$ é a imagem da transformação (do inglês, "range") e $\mathcal{N}$ é o espaço nulo da transformação.

Um caso de aplicação muito interessante de aplicação de Decomposições em Valores Singulares e Autovalores surge na estatística, no ramo de simulações. A geração de variáveis aleatórias com distribuição normal pode ser feita utilizando a já vista **Fatoração de Cholesky**, pois a famosa matriz de variância de covariância tem a propriedade de ser definida positiva e simétrica. Entretanto, há casos onde esta matriz é degenerada (em outras palavras, a variação inerente do problema está descrita em um espaço de dimensão mais baixo logo, alguns de seus valores singulares serão zero), forçando a matriz a perder tais propriedades e portanto não permitindo uma inteligente aplicação da fatoração de Cholesky. A decomposição SVD pode então ser uma útil abordagem para este problema. Problemas de instabilidade numérica também não são raros, cabendo uma abordagem utilizando a Decomposição Espectral (utilizando, por exemplo o Algorítmo de Autovalores de Jacobi), que pode reduzir problemas de estabilidade. Mais informações sobre este problema podem ser encontradas no artigo [JW06], e com motivações em [Wik]. Aplicações também podem ser encontradas em [Wic02] e em [eYS14](este último tem conexão com o tópico "Subespaço de Krylov", visto no decorrer deste curso de MS512).

5 Equação de Sylvester e Lei da Inércia de Sylvester

5.1 Equação de Sylvester

A equação linear matricial: **(1)** $AX - XB = C$, em que $A \in \mathbb{R}^{mXm}$, $B \in \mathbb{R}^{nXn}$ e $C \in \mathbb{R}^{mXn}$ são dados e $X \in \mathbb{R}^{mXn}$ uma matriz a ser determinada, é chamada de equação de Sylvester. Esta estrutura de equação matricial tem única solução somente se não houver autovalores em comum entre as matrizes A e B.

Tal equação é de grande interesse, pois ela pode ser incluída ou vista em vários casos especiais de problemas importantes como:

$\triangleright$ sistema linear $Ax = x$;

$\triangleright$ inversão de matrizes $AX = I$;

$\triangleright$ autovetor correspondente a um dado autovalor $(A - bI)x = 0$;

$\triangleright$ comutatividade de matrizes $AX - XA = 0$.

Uma outra representação de **(1)** é a seguinte:

(2) $(I_n \otimes A - B^T \otimes I_m)vec(X) = vec(C)$, onde $A \otimes B$ é por definição dado pela seguinte forma $a_{ij}B$, ou seja, o produto de Kronecker e o operador vec "empilha"as colunas de uma matriz em um vetor longo (as colunas compõem, assim, um vetor coluna). De forma mais didática, o produto de Kronecker pode ser escrito da seguinte forma, considerando-se as matrizes $A \in \mathbb{R}^{mXm}$ e $B \in \mathbb{R}^{nXn}$:

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1m}B \\ a_{21}B & a_{22}B & \dots & a_{2m}B \\ \vdots & \dots & \dots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mm}B \end{pmatrix}$$

A matriz de coeficiente $(I_n \otimes A - B^T \otimes I_m)$ em **(2)** tem dimensão dada por $mnXmn$ e uma estrutura especial. Para exemplificarmos, vamos considerar o caso em que $n = 3$. Teremos, assim, as seguintes dimensões de matrizes: $A \in \mathbb{R}^{mXm}$, $B \in \mathbb{R}^{3X3}$, $C \in \mathbb{R}^{mX3}$ e $X \in \mathbb{R}^{mX3}$.

$I_3 \otimes A$, de acordo com a definição é dada por:

$$I_3 \otimes A = \begin{pmatrix} i_{11}A & i_{12}A & i_{13}A \\ i_{21}A & i_{22}A & i_{23}A \\ i_{31}A & i_{32}A & i_{33}A \end{pmatrix}$$

E uma vez que os elementos da diagonal da matriz identidade são unitários e fora desta temos elementos nulos, chegamos no seguinte resultado:

$$I_3 \otimes A = \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{pmatrix}$$

Para $B^T \otimes I_m$ temos a seguinte notação (lembrando que temos de transpor a matriz B, segue que B^T):

$$B^T = \begin{pmatrix} b_{11} & b_{21} & b_{31} \\ b_{12} & b_{22} & b_{32} \\ b_{13} & b_{23} & b_{33} \end{pmatrix}$$

E então $B^T \otimes I_m$ é:

$$B^T \otimes I_m = \begin{pmatrix} b_{11}I_m & b_{21}I_m & b_{31}I_m \\ b_{12}I_m & b_{22}I_m & b_{32}I_m \\ b_{13}I_m & b_{23}I_m & b_{33}I_m \end{pmatrix}$$

Consequentemente, a $(I_3 \otimes A - B^T \otimes I_m)$ resulta em:

$$(I_3 \otimes A - B^T \otimes I_m) = \begin{pmatrix} A - b_{11}I_m & -b_{21}I_m & -b_{31}I_m \\ -b_{12}I_m & A - b_{22}I_m & -b_{32}I_m \\ -b_{13}I_m & -b_{23}I_m & A - b_{33}I_m \end{pmatrix}$$

Como observamos anteriormente, a matriz de coeficiente é altamente estruturada matematicamente, porém, não é fácil de se obter vantagens a partir desta. Assim, o objetivo de pesquisas tem sido a análise e a solução direta da forma **(1)**. Das várias fórmulas que têm sido obtidas para tal solução, podemos destacar: se $\int_0^\infty exp^{At} exp^{Bt} dt$ existir, então menos esta integral é uma solução para a equação de Sylvester.

Com o objetivo de mostrar essa teoria mais presente em nossos estudos, podemos observar a equação de Sylvester na diagonalização de matrizes em blocos. Por exemplo, vamos supor que queiramos encontrar uma transformação similar tal que o elemento (1,2) da matriz A de bloco a seguir seja nulo.

$$A = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

Para isso pode-se determinar a seguinte transformação:

$$\begin{pmatrix} I & -X \\ 0 & X \end{pmatrix}^{-1} \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix} \begin{pmatrix} I & -X \\ 0 & X \end{pmatrix} = \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}$$

Tal igualdade é válida, se e somente se, $A_{11}X - XA_{22} = A_{12}$. Assim, a diagonalização em bloco de matrizes é reduzida a resolver a equação de Sylvester, no qual é possível resolver, se e somente se, os autovalores de A_{11} e de A_{22} forem distintos.

A equação de Sylvester tem muitas variantes e casos especiais, como a equação de Lyapunov $AX + XA^* = C$, no qual $*$ significa a matriz transposta conjugada e a equação de discreta de Sylvester $X + AXB = C$. Além disso, a equação de Sylvester foi generalizada para vários termos com coeficientes matriciais em ambos os lados, que nos permite escrever:

(3) $\sum_{i=1}^{k} A_i X B_i = C$.

Para $k \leq 2$ e $m = n$ esta equação pode ser resolvida em $O(n^3)$ operações. Para $k > 2$, nenhum algoritmo de $O(n^3)$ operações é conhecido e métodos numéricos eficientes permanece como um problema em aberto. O sistema **(3)** aparece em discretização de elementos finitos estocásticos de equações diferenciais com entradas aleatórias. As matrizes A_i e B_i são grandes e esparsas, e dependendo de propriedades estatísticas das entradas aleatórias, k pode ser arbitrariamente grande. Em pesquisas atuais, soluções iterativas eficientes e precondicionadores têm sido desenvolvidos para tais sistemas.

▷ Podemos observar a importância da determinação numérica de autovalores. Os métodos vistos em aula demonstram sua fundamental importância e aplicação para a construção de uma nova base e visão teórica.

▷ Conseguimos notar a importância da estrutura de matrizes em blocos, referente a diagonalização de matrizes. No curso de $MS512$ utilizamos este conceito principalmente na fatoração QR

para a problemas de quadrados mínimos. Isto nos evidência a necessidade de conceitos sólidos e claros, além da interdependência dos conceitos estudados.

5.2 Lei da inércia de Sylvester

É sabido que a inércia de uma matriz Hermitiana é um vetor de inteiros (ν, ζ, π) tal que ν é o número de autovalores negativos, ζ é o número de autovalores nulos e π é o número de autovalores positivos. A lei da inércia de Sylvester nos diz que para qualquer matriz Hermitiana A e uma matriz não singular X a inércia de A é a mesma do que X^*AX. Esta transformação é chamada de congruente, portanto a lei de Sylvester estabelece que o número de autovalores negativos, nulos e positivos não muda sob transformações congruentes.

Consideremos um exemplo no qual queiramos determinar os autovalores de matrizes tridiagonais Hermitianas T. Suponhamos que queremos o k-ésimo menor autovalor de T. Seja $N(x)$ o número de autovalores que sejam menores do que x. Nós precisamos determinar o ponto onde $N(x)$ pula de $k-1$ para k. É factível fazer isto pelo método da bissecção se pudermos computar facilmente $N(x)$. Vamos supor que podemos fatorar $T - xI = LDL^*$, no qual D é uma matriz diagonal e L é uma matriz inferior bidiagonal. Esta fatoração pode ser realizada em $O(n)$ operações e a lei da inércia de Sylvester nos diz que $T - xI$ e D têm a mesma inércia, então o número de elementos negativos da diagonal de D são iguais ao número de autovalores de $T - xI$ que são menores do que zero, no qual é o número de autovalores de T que são menores do que x, isto é, $N(x)$.

Como não existe nenhum pivotiamento na fatoração pode-se pensar que esta aproximação seria numericamente instável (que seria de fato caso nosso objetivo fosse resolver o sistema linear com tal fatoração), mas no sentido de determinarmos a diagonal de D pode ser mostrado que é perfeitamente estável.

A magnitude dos autovalores depois da transformação de congruência foi mostrado por Ostrowski como segue: $\lambda_k(X^*AX) = \theta_k\lambda_k A$, onde $\lambda_n(X^*X) \leq \theta_k \leq \lambda_1(X^*X)$, onde os autovalores $\lambda_n, \ldots, \lambda_1$ estão ordenados. Este resultado é útil para o desenvolvimento de pertubações mínimas de uma matrix que mudou sua inércia de alguma maneira.

$\triangleright$ Uma vez determinado todos os autovalores da matriz T e quisermos saber o $N(x)$ basta calcularmos $T - xI$ que é equivalente à transformação LDL^*.

$\triangleright$ No curso de $MS512$ (assim como no curso de $MS211$) tivemos a noção dos problemas causados por perturbações em um sistema linear. Além disso, esudamos o número de condição de uma matriz na qual nos permite também a compreensão de problemas de instabilidade numérica.

6 Métodos Iterativos para Solução de Sistemas de Equações Não Lineares

No mundo contemporâneo, dada a evolução tecnológica e científica atual, a humanidade se depara com problemas crescentemente mais difíceis, criando a necessidade de se contornar tais problemas com técnicas mais avançadas. Nesse contexto surgiram os sistemas de equações não lineares, que surgem com elevada frequência em aplicações não triviais. Como exemplos, temos as equações diferenciais, que têm origem na física e em engenharias, e a equação de quadrados mínimos não lineares, na estatística.

A maneira habitual de se atacar sistemas de equações não lineares é, ao invés de resolvê-las analiticamente, utilizar técnicas de aproximação de suas soluções, de forma que a solução esteja

suficientemente próxima da verdade. Tais aproximações serão o foco de estudo deste exercício, cujo ponto central é o Método do Ponto Fixo.

6.1 Método do Ponto Fixo

O objetivo de estudarmos aproximações para equações não lineares é para se obter a solução do sistema

$$f_1(x_1, x_2, \ldots, x_n) = 0 \tag{1}$$
$$f_2(x_1, x_2, \ldots, x_n) = 0 \tag{2}$$
$$\cdots \tag{3}$$
$$f_n(x_1, x_2, \ldots, x_n) = 0. \tag{4}$$

onde as funções f_i não são lineares em seus argumentos. A ideia a ser utilizada no método do ponto fixo é representar este sistema com a função F, que mapeia $\mathbb{R}^n$ em $\mathbb{R}^n$ como

$$\mathbf{F}(x_1, x_2, \ldots, x_n) = (f_1(x_1, x_2, \ldots, x_n), \ldots, f_n(x_1, x_2, \ldots, x_n)),$$

tal que

$$\mathbf{F}(\mathbf{x}) = \mathbf{0},$$

e em seguida utilizar um método iterativo que convirja para a solução exata, seguindo algum critério de parada. Primeiramente começamos por definir o que é um ponto fixo.

Definição 1. (Ponto Fixo) A função G de $D \in \mathbb{R}^n$ em $\mathbb{R}^n$ tem um ponto fixo em $\mathbf{p} \in D$ se $\mathbf{G}(\mathbf{p}) = \mathbf{0}$.

Agora, seguimos ao teorema principal que irá nos dizer como realizar a iteração (este teorema pode ser encontrado em [eJDF13], p.601-602).

Teorema 6.1. ***(Ponto Fixo)*** *Seja $D = [(x_1, x_2, \ldots, x_n), \ldots, f_n)^t \quad | \quad a_i \leq x_i \leq b_i$, para cada $i = 1, 2, \ldots, n]$ para alguma coleção de constantes $a_1, a_2, \ldots, a_n$ e $b_1, b_2, \ldots, b_n$. Suponha que G é uma função contínua de $D \in \mathbb{R}^n$ em $\mathbb{R}^n$ com a característica de que $\boldsymbol{G}(\boldsymbol{x}) \in D$, sempre que $\boldsymbol{x} \in D$. Então $\boldsymbol{G}$ tem um ponto fixo em D.*

Suponha, adicionalmente, que todas as funções componentes de $\boldsymbol{G}$ tem derivadas parciais e a constante $K < 1$ exista tal que

$$\left| \frac{\partial g_i(x)}{\partial x_j} \right| \leq \frac{K}{n}, \quad \textit{sempre que } x \in D,$$

para cada $j = 1, 2, \ldots, n$ e cada função componente g_i. Então a sequência $\{x^{(k)}\}_{k=0}^{\infty}$, caracterizada por um $x^{(0)}$, escolhido arbitrariamente no conjunto D e gerada pela função de iteração

$$\boldsymbol{x}^{(k)} = \boldsymbol{G}(\boldsymbol{x}^{(k-1)}), \tag{5}$$

para cada $k \geq 1$, converge para o único ponto fixo $\boldsymbol{p} \in D$. Além disso,

$$||\boldsymbol{x}^{(k)} - \boldsymbol{p}||_\infty \leq \frac{K^t}{1 - K} ||\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(0)}||_\infty.$$

O teorema acima nos conta que a maneira com que o método deve ser aplicado. Ele nos diz que,

Método 1. ***Ponto Fixo****: Se* $\boldsymbol{F}(\boldsymbol{x})$ *for contínua em uma vizinhança do ponto fixo desejado e se as funções* $\boldsymbol{x} = \boldsymbol{x}(f_1, f_2, \ldots, f_n)$ *têm derivadas parciais nesta mesma vizinhançã, de forma que*

$$\left| \frac{\partial g_i(x)}{\partial x_j} \right| \leq \frac{K}{n}, \quad \textit{sempre que } x \in D,$$

,então o procedimento é realizar a operação, utilizando uma aproximação inicial $x^{(0)}$*,*

$$\boldsymbol{x}^{(k)} = \boldsymbol{G}(\boldsymbol{x}^{(k-1)}),$$

repetidas vezes, atualizando o resultado obtido no nível k *da iteração na função de iteração* $k+1$*.*

Entretanto, precisamos especificar um critério de parada para que o método não fique iterante infinitamente. Um critério razoável é $||\mathbf{x}^{(k-1)} - \mathbf{x}^{(k)}|| \leq \epsilon$, onde ϵ é a magnitude máxima que o erro pode assumir. Abaixo segue um pseudo-algorítmo para implementação do Método do Ponto Fixo para encontrar uma aproximação para $\mathbf{p} = G(\mathbf{p})$, dado uma aproximação inicial $\mathbf{p}_0$.

Algorítmo 1. ***Ponto Fixo***

ENTRADA *aproximação inicial* $\boldsymbol{p}_0 =$*,*
tolerância TOL
número máximo de iterações N

SAÍDA *solução* $\boldsymbol{p} = (x_1, \ldots, x_n)$ *ou aviso de falha (número de iterações excedido)*

Passo 1 *Defina* $k = 1$*.*

Passo 2 *Enquanto* $k \leq N$ *realize os passos de 3 a 5.*

Passo 3 *Obtenha* $G(\boldsymbol{p}_0)$ *e faça* $\boldsymbol{p} = G(\boldsymbol{p}_0)$

Passo 4 *Se* $||\boldsymbol{p} - \boldsymbol{p}_0|| < TOL$*, retorne* $\boldsymbol{p}$ *(procedimento foi concluído com sucesso)*
Pare.

Passo 5 *Faça* $\boldsymbol{p}_0 = \boldsymbol{p}$ *e*
$k = k + 1$*.*

SAÍDA *retorne a mensagem 'O método falhou após N iterações'*
Pare.

A sequência de figuras a seguir (baseadas em [eVLRL96]), esboça o funcionamento geométrico deste método numérico, para o caso univariado. As duas primeiras figuras ilustram o caso de convergência que, como podemos ver, reforça a ideia de que buscamos uma aproximação do ponto fixo p que, por 5, convém de ser a interseção entre a reta identidade e função $\mathbf{G(x)}$. Já as duas últimas figuras ilustram situações onde a sequência não converge, mostrando que, a cada iteração, o método se afasta o ponto fixo.

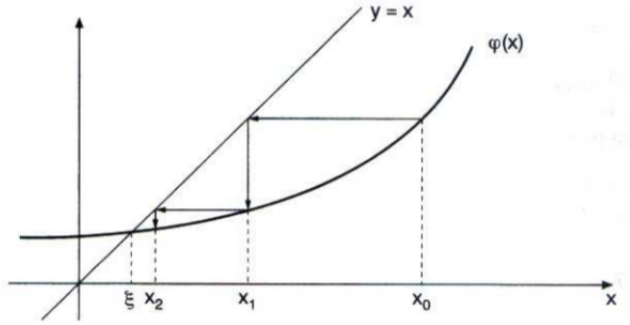

Figura 11: Interpretação geométrica; caso de convergência 1.

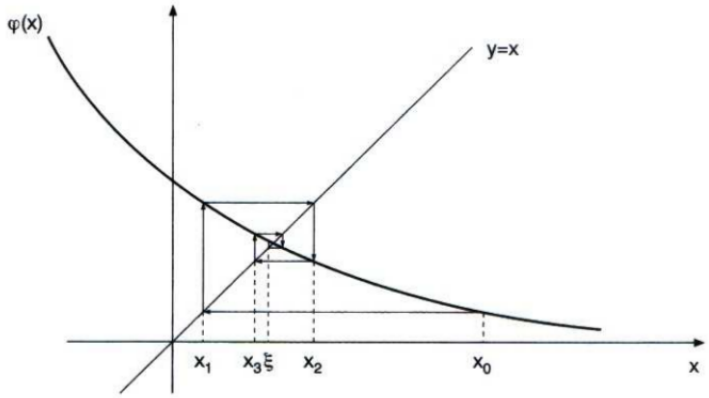

Figura 12: Interpretação geométrica; caso de convergência 2.

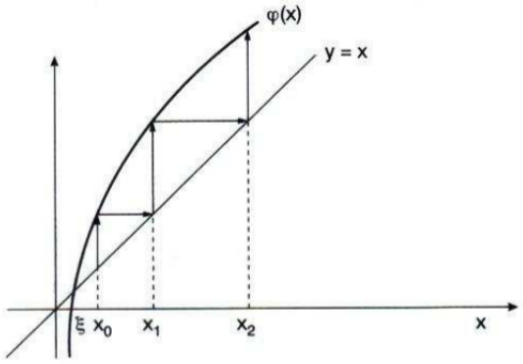

Figura 13: Interpretação geométrica; caso de divergência 1.

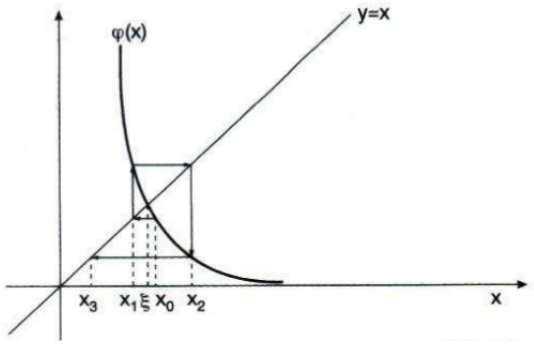

Figura 14: Interpretação geométrica; caso de divergência 2.

Para um leitor que, pela primeira vez, se depara com esta técnica, o que pode ser visto é que, primeiramente fornecemos um ponto x_0 para a função e, com esse ponto, descobrimos o ponto x_1 que fornece o mesmo valor de (neste caso φ para manter a concordância com a figura) $\varphi(x)$ em $y = x$, a reta identidade, e assim por diante. Uma beleza que pode ser vista neste método é que a sua implementação computacional é bem simples, pois basta uma função de laço ("loop") para que o algorítmo se inicie e um critério de parada que termine, o que esboça a simplicidade e fluidez do método. Todavia, como citado em [eJDF13] (página 607), não é usual que esta técnica seja bem sucedida. O que nos obriga a procurar outras formas de aproximar a solução do sistema não linear alvo. Mas o valor do método do Ponto Fixo não vem exclusivamente de sua implementação prática, mas sim de base teórica, que inspira muitos outros métodos - não só para resolver sistema de equações não lineares, mas também lineares. Como exemplo, os métodos de Gauss-Seidel, Jacobi e SOR, além do método dos Gradientes Conjugados. Todos estes tem uma ligação especial com a chamada "estrutura de ponto fixo".

6.2 Método de Newton

O método de Newton é uma tentativa de acelerar a convergência, em relação ao Método do Ponto Fixo. A tentativa é passar de uma convergência linear, para uma quadrática, diminuindo o número de iterações. Entretanto, o método continua mantendo uma estrutura de ponto fixo, pertencendo à família dos métodos estacionários.

Supomos, inicialmente um modelo do tipo $\mathbf{G}(\mathbf{x}) = \mathbf{x} - A(\mathbf{x})^{-1}\mathbf{F}(\mathbf{x})$, onde A não-singular é definido por suas entradas $a_{ij}(\mathbf{x}) : \mathbb{R}^n \longmapsto \mathbb{R}$, de forma que A deve ser escolhido a garantir uma convergência quadrática para a solução de $\mathbf{F}(\mathbf{x}) = \mathbf{0}$. Em seguida, utilizamos o teorema a seguir para descobrir que $A = J(\mathbf{x})$, sendo $J(\mathbf{x})$ a matriz Jacobiana, provinda do Cálculo Integral e Diferencial (ver, *e.g* [Apo69]).

Teorema 6.2. *Seja* $\boldsymbol{p}$ *a solução de* $\boldsymbol{G(x) = x}$*. Suponha que exista* $\delta > 0$ *tal que*

1. $\frac{\partial g_i}{\partial x_j}$ *é contínua em* $N_\delta = \{x : ||x - p|| < \delta\}$*, for para cada* $i = 1, 2, \ldots, n$ *e* $j = 1, 2, \ldots, n$*;*

2. $\frac{\partial^2 g_i(\boldsymbol{x})}{\partial x_j \partial x_k}$ *é contínua e* $|\frac{\partial^2 g_i(\boldsymbol{x})}{\partial x_j \partial x_k|} \leq M$ *para alguma constante* M*, sempre que* $\boldsymbol{x} \in N_\delta$ *para cada* $i = 1, 2, \ldots, n$*,* $j = 1, 2, \ldots, n$ *e* $k = 1, 2, \ldots, n$*;*

3. $\frac{\partial g_i(\boldsymbol{p})}{\partial x_k} = 0$ *para cada* $i = 1, 2, \ldots, n$ *e* $k = 1, 2, \ldots, n$.

Podemos, então, concluir que o número $\hat{\delta} \leq \delta$ *existe de forma que a sequência gerada por* $\boldsymbol{x}^{(k)} = \boldsymbol{G}(\boldsymbol{x}^{(k-1)})$ *converge quadraticamente para* $\boldsymbol{p}$ *para qualquer escolha de* $\boldsymbol{x}^{(0)}$ *satisfazendo* $||\boldsymbol{x}^{(0)} - \boldsymbol{p}||_\infty < \hat{\delta}$. *Mais ainda, vale a desigualdade*

$$||\boldsymbol{x}^{(k)} - \boldsymbol{p}||_\infty \leq \frac{n^2 M}{2} ||\boldsymbol{x}^{(k-1)} - \boldsymbol{p}||_\infty^2, \text{ para cada } k \geq 1.$$

Como planejado podemos, agora, utilizar o teorema anterior (que pode ser encontrado em [eJDF13], p. 608) para definir a função de iteração que descreve o Método multivariado de NewtonfR para sistemas não lineares :

Método 2. ***Newton Multivariado****: Se a matriz* $\boldsymbol{J}(\boldsymbol{x})$ *satisfazer as condições do teorema 3, então a função que gera sequência dada por* $\boldsymbol{x}^{(k)} = \boldsymbol{G}(\boldsymbol{x}^{(k-1)})$, *onde*

$$\boldsymbol{G}(\boldsymbol{x}) = \boldsymbol{x} - \boldsymbol{J(x)}^{-1} \boldsymbol{F}(\boldsymbol{x}),$$

definida como

$$\boldsymbol{x}^{(k)} = \boldsymbol{G}(\boldsymbol{x}^{(k-1)}) = [\boldsymbol{J}(\boldsymbol{x}^{(k-1)})]^{-1} \boldsymbol{F}(\boldsymbol{x}^{(k-1)})$$

é conhecida como Método de Newton para sistemas não lineares.

Ou seja, se o sistema linear associado $F'(\mathbf{x})s = F(\mathbf{x})$ (vindo de $\mathbf{G}(\mathbf{x}) = \mathbf{x} - \mathbf{J(x)}^{-1}\mathbf{F}(\mathbf{x})$), onde $F'(\mathbf{x}) = J(\mathbf{x})$ tiver solução x^* e se for tal que

- F' for Lipschitz contínua na vizinhança de $\mathbf{x}^*$
- $F'(\mathbf{x}^*)$ for não singular

a sequência dada pelo Método 1 converge. Um pseudo-algorítmo do Método de Newton é dado a seguir.

Algorítmo 2. ***Algorítmo de Newton***

ENTRADA *número* n *de equações e variáveis; aproximação inicial* $x = (x_1, \ldots, x_n)^t$, *tolerância* TOL; *número máximo de iterações* N.

SAIDA *solução aproximada* $x = (x_1, \ldots, x_n)^t$ *ou uma mensagem de que o número de iterações máximas foi excedido.*

Passo 1 *Defina* $k = 1$

Passo 2 *Enquanto* $(k \leq N)$ *faça os passos 3-7*

Passo 3 *Calcule* $F(x)$ *e* $J(x)$, *onde* $J(x)_{i,j} = \frac{\delta f_i(x)}{\delta x_j}$ *para* $1 \leq i, j \leq n$.

Passo 4 *Resolva o sistema linear de ordem* $n \times n$ *associado* $J(x)y = -F(x)$.

Passo 5 *Defina* $x = x + y$.

Passo 6 *Se $||y|| < TOL$, então SAIDA(x);*

Passo 7 *Defina $k = k + 1$.*

Passo 8 *SAIDA('Número máximo de iterações excedido');*

Ainda dois pontos importantes a serem citados sobre o método em questão. Primeiramente, apesar do método ter uma taxa de convergência quadrática (para o caso univariado, ver *e.g* [eVLRL96], p. 72-73), para o caso multivariado, a necessidade de se avaliar e inverter a matriz $J(\mathbf{x})$ a cada iteração é um processo bastante custoso, que pode tornar o método inviável computacionalmente. Segundo, a matriz $J(\mathbf{x})$ deve ser não-singular na vizinhaça do ponto fixo $\mathbf{p}$. Se acontecer da função ser próxima de singular, problemas de instabilidade numérica surgem, contando como mais uma -grande- dificuldade.

Para resolver este problema, foram criados métodos que contornem a necessidade de invertermos $J(\mathbf{x})$. Um exemplo de técnica é o chamado Método de Newton-Krylov, baseado nos Subspaços de Krylov. Este método utiliza uma estratégia de se resolver um sistema linear associado, de forma que a solução deste sistema já nos garanta $[\mathbf{J}(\mathbf{x}^{(k-1)})]^{-1}\mathbf{F}(\mathbf{x}^{(k-1)})$, removendo a dificuldade da inversão (para mais exemplos, ver *e.g* [Kel03]).

Um exemplo de aplicação (provavelmente o mais famoso, quando se trata do método de Newton) é dado no caso univariado ao se aproximar a $\sqrt{2}$, conhecido como o Método Babilônico. A técnica para se obter $\sqrt{r}$ pode ser expressa, dada uma aproximação inicial x_0, como

$$\begin{aligned} x_0 &\approx r \\ x_{n+1} &= \frac{1}{2}\left(x_n + \frac{r}{x_n}\right). \end{aligned}$$

Uma interpretação geométrica do método de Newton para o caso univariado é tida como : a partir de uma aproximação inicial x_0, traçamos a reta tangente T à curva S no ponto $\mathbf{G}(\mathbf{x}_0)$, e procuramos o ponto x_1 tal que $T \equiv 0$. Em seguida, avaliamos $\mathbf{G}(\mathbf{x}_1)$ e repetimos o processo. A seguir é apresentado o este processo iterativo para o caso univariado.

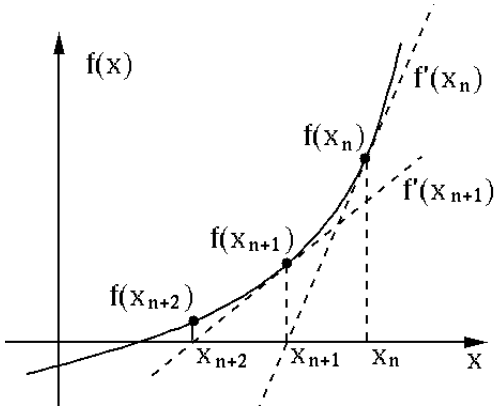

Figura 15: Método de Newton para o caso univariado

O método começa a partir de uma aproximação inicial x_n (pois o procedimento pode estar iterando há várias vezes) e calcula a função e sua derivada neste ponto para achar a reta tangente à curva no ponto x_n. Em seguida, encontra o ponto desta reta tangente que toca a abcissa, obtém o próximo ponto x_{n+1} e assim por diante. Mais ilustrações geométricas sobre o método podem ser encontradas em [eJDF13] ou [eVLRL96].

6.3 Método de Broyden

Como citado no tópico relacionado ao Método de Newton, uma dificuldade inerente da técnica é necessidade de, a cada iteração, computar a inversa da matriz jacobiana. Este problema pode ser parcialmente contornado ao resolvermos um sistema linear associado mas, como visto neste curso, a resolução de sistemas lineares nem sempre é uma tarefa trivial e que pode se tornar bastante custosa.

Dada a situação em que nos encontramos, métodos mais computacionalmente eficientes são desejados. Um opção é utilizar o Método de Broyden, que permite lidarmos com a resolução do sistema linear de uma maneira inteligente, de forma que nenhum sistema linear seja resolvido, resultando num problema onde só sejam avaliados produtos matriz-vetor. Sendo um método da classe quase-Newton, é de se esperar que mantenha as condições naturais do método original, ou seja, devemos partir das hipóteses

1. **F** é continuamente diferenciável em alguma conjunto aberto $D \subset \mathbb{R}^n$ e
2. Para um dado $x \in D$ e dado $s \neq 0$, temos $x^{(k+1)} = x^{(k)} + s$.

A maneira que esta abordagem lida com a dificuldade de inversão é proceder em modificar o Método de Newton de forma a utilizar aproximações para suas derivadas parciais, utilizando diferenças finitas para tal aproximação. Por esta razão, escolhemos B^+ como uma aproximação conveniente que não deixe de preservar as propriedades da matriz Jacobiana. À luz destes fatos, a formulação do Método de Broyden é dada pelas equações

$$x^{(k+1)} = x^{(k)} - B_k^{-1} F(x^{(k)}), k = 0, 1, 2, \ldots \tag{6}$$

$$y^{(k)} = F(x^{(k+1)}) - F(x^{(k)}), s^{(k)} = x^{(k+1)} - x^{(k)} \tag{7}$$

$$B_{k+1} = B_k + \frac{[y^{(k)} - B_k s^T] {s^{(k)}}^T}{{s^{(k)}}^T s^{(k)}}. \tag{8}$$

Entretanto, ainda não estamos livres de um sistema linear, este sendo

$$B_k s^k = -F(x^{(k)}).$$

Ao utilizar a fórmula de *Sherman-Morisson*, entretanto, conseguimos uma solução para este problema. Escolhendo $H_k = B_k^{-1}$ e $H_{k+1} = B_{k+1}^{-1}$ temos, finalmente o Método de Broyden.

Método 3. ***Broyden*** *Se F for continuamente diferenciável em algum conjunto aberto $D \subset \mathbb{R}^n$ e, para dados $x \in D$ e $s \neq 0$, temos $x^{(k+1)} = x^{(k)} + s$, o método de Broyden será dados pelo conjunto de equações (6),(7) e (8), se não utilizarmos o procedimento de atualização de Sherman-Morrison e*

$$x^{(k+1)} = x^{(k)} - H_k F(x^{(k)}), k = 0, 1, 2, \ldots \tag{9}$$

$$y^{(k)} = F(x^{(k+1)}) - F(x^{(k)}), s^{(k)} = x^{(k+1)} - x^{(k)} \tag{10}$$

$$H_{k+1} = H_k + \frac{[y^{(k)} - H_k s^T]{s^{(k)}}^T}{{s^{(k)}}^T s^{(k)}}. \tag{11}$$

se fizermos uso da equação de Sherman-Morrison.

O algorítmo foi, então, construído de maneira a contornar os problemas de inversão e solução de sistemas linear por iteração. Adicionalmente, ao implementarmos esta técnica desta maneira, somos capazes de reduzir o número de operações envolvidas em uma ordem de grandeza, tanto sobre a quantidade de avaliações de funções escalares (indo de $O(n^2)$para $O(n)$), quanto sobre a quantidade de operações aritméticas por iteração (obtido não só pela fórmula de Sherman-Morrison, mas também pelas atualizações (perturbações) de posto unitário da equação $B^+ = B + v\frac{1}{s^T s}s^T$), indo de $O(n^3)$ para $O(n^2)$.

Entretanto, não há só ganhos ao se empregar o método pois, sua taxa de convergência não é mais quadrática e sim superlinear, debilitando sua velocidade. Este fato pode ser facilmente verificado ao se observar a quantidade de operações por iteração, que continua sendo alto e bastante custoso, computacionalmente. Além disso, o Método de Broyden não auto-corretivo como o Método de Newton, o que implicar , em alguns casos, que a matriz B_k pode não convergir para a sua equivalente matriz Jacobiana.

Por fim, abaixo encontra-se um pseudo-algorítmo para o método de Broyden para se obter uma aproximação da solução do sistema não linear $\mathbf{F}(\mathbf{x}) = 0$, dada uma aproximação inicial $\mathbf{x}$.

Algorítmo 3. ***Algorítmo de Broyden***

ENTRADA *número n de equações e incógnitas; aproximação inicial* $\boldsymbol{x} = (x_1, \ldots, x_n)^t$*; tolerância TOL; número máximo de iterações N.*

SAÍDA *solução aproximada* $\boldsymbol{x} = (x_1, \ldots, x_n)^t$ *ou aviso de que a quantidade de iterações foi excedida.*

Passo 1 *Faça* $k = 1$ *e* $A_0 = J(\boldsymbol{x})$*, onde* $J(\boldsymbol{x})_{i,j} = \frac{\partial f_i}{\partial x_j}(\boldsymbol{x})$ *para* $1 \leq i$ *,* $j \leq n$*,*
defina, também, $\boldsymbol{v} = F(\boldsymbol{x})$*.*

Passo 2 *Obtenha* A_0^{-1} *(via método de escolha)e armazene o armazene em A.*

Passo 3 *Faça* $\boldsymbol{s} = -A\boldsymbol{v}$*;*
$\boldsymbol{x} = \boldsymbol{x} + \boldsymbol{s}$ *e*
$k = 2$*.*

Passo 4 *Enquanto* $k \leq N$ *realize os Passos de 5 a 13.*

Passo 5 *Faça* $\boldsymbol{w} = \boldsymbol{v}$ *para salvar* $\boldsymbol{v}$*,*
$\boldsymbol{v} = F(\boldsymbol{x})$*, (Note que* $\boldsymbol{v} = F(\boldsymbol{x}^{(k)})$*)*
$\boldsymbol{y} = \boldsymbol{v} - \boldsymbol{w}$*. (Note que* $\boldsymbol{y} = \boldsymbol{y}_k$*)*

Passo 6 *Faça* $\boldsymbol{z} = -A\boldsymbol{y}$*. (Note que* $\boldsymbol{z} = -A_{k-1}^{-1}\boldsymbol{y}_k$*)*

Passo 7 *Faça* $p = \boldsymbol{s}^t \boldsymbol{z}$. *(Note que* $p = \boldsymbol{s}_z^t A_{k-1}^{-1} \boldsymbol{y}_k$*)*

Passo 8 *Faça* $\boldsymbol{u}^t = \boldsymbol{s}^t A$.

Passo 9 *Faça* $A = A + \frac{1}{p}(\boldsymbol{s} + \boldsymbol{z})\boldsymbol{u}^t$. *(Note que* $A = A_k^{-1}$*)*

Passo 10 *Faça* $\boldsymbol{s} = -A\boldsymbol{v}$. *(Note que* $\boldsymbol{s} = -A_k^{-1} F(\boldsymbol{x}^{(k)})$*)*

Passo 11 *Faça* $\boldsymbol{x} + \boldsymbol{x} + \boldsymbol{s}$. *(Note que* $\boldsymbol{x} + \boldsymbol{x}^{(k+1)}$*)*

Passo 12 *Se* $||\boldsymbol{s}|| < TOL$*, então retorne* $\boldsymbol{x}$ *(O procedimento foi concluído com sucesso)*
Pare

Passo 13 *Faça* $k = k + 1$

Passo 14 *retorne ('Número máximo de iteraoes foi excedido')*
Pare

Vale a pena, antes de passarmos para a parte de aplicações, lembrar que, para o caso univariado, o Método de Broyden é popularmente conhecido como o Método da Secante. Mais informações podem ser encontradas em [eVLRL96] e em [eJDF13].

6.4 Aplicação dos Métodos de Newton e Broyden

É desejável colocar em prática os métodos de Newton e Broyden na resolução dos exercícios 7 e 11. Será aplicado, também, a fórmula de Sherman-Morrison.

6.4.1 Exercício 7

O exercício fornece o sistema não linear:

$$3x_1 - cos(x_2 x_3) - \frac{1}{3} = 0$$
$$x_1^2 - 625x_2^2 - \frac{1}{4} = 0$$
$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

que tem uma matriz Jacobiana singular na solução. É pedido a aplicação do método de Broyden com valor inicial de $\mathbf{x}^{(0)} = (1, 1 - 1)^t$. O algoritmo utilizado encontra-se no anexo.

A matriz Jacobiana, utilizada, foi:

$$J = \begin{bmatrix} 3 & x_3 sen(x_2 x_3) & x_2 sen(x_2 x_3) \\ 2x_1 & -2 * 625x_2 & 0 \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

No mesmo sistema, foi utilizado o algoritmo de Newton. A implementação também se encontra no anexo.

Também utilizou-se a fórmula de Sherman-Morrison, para eliminar a necessidade de inversão da matriz.

6.4.2 Exercício 11

O exercício 11 é relacionado com o exercício 13 da seção 8.1. Como introdução, aqui está o que foi pedido.

Quer-se determinar a relação entre o peso (W) em gramas de larvas da espécia *Pachysphinx modesta* e seu consumo de oxigênio (R) em mililitros por hora.

Foi assumido, através de conhecimento biológico prévio, que essa relação é da forma

$$R = bW^a$$

Em um primeiro momento, foi ajustado o seguinte modelo de regressão linear ordinária.

$$ln\, R = ln\, b + a\, ln\, W + \varepsilon$$

No relatório anexo, há também o cálculo do erro associado com a aproximação e a modificação do modelo adicionando o termo quadrático.

É de interesse, porém, ajustar o modelo sem a transformação. Para isso, é necessário encontrar as constantes a e b que minimizam

$$f(a,b) = \sum_{i=1}^{n} (R_i - bw_i^a)^2$$

Para isso, foi apresentado o seguinte sistema não-linear com incógnitas a e b.

$$\begin{aligned} \frac{\delta f}{\delta a} &= b^2 \sum_{i=1}^{n} w_i^{2a} log(w_i) - b \sum_{i=1}^{n} R_i w_i^a log(w_i) &= 0 \\ \frac{\delta f}{\delta b} &= b \sum_{i=1}^{n} w_i^{2a} - \sum_{i=1}^{n} R_i w_i^a &= 0 \end{aligned}$$

e

$$J = \begin{bmatrix} 2b^2 \sum_{i=1}^{n} w_i^{2a} log^2(w_i) - b \sum_{i=1}^{n} R_i w_i^a log^2(w_i) & b \sum_{i=1}^{n} w_i^{2a} log(w_i) - \sum_{i=1}^{n} R_i w_i^a log(w_i) \\ 2b \sum_{i=1}^{n} w_i^{2a} log(w_i) - \sum_{i=1}^{n} R_i w_i^a log(w_i) & \sum_{i=1}^{n} w_i^{2a} \end{bmatrix}$$

7 Argumentos para a compreensão da convergência do algoritmo QR

Com o objetivo de determinar o conjunto de todos os autovalores de uma matriz cheia, podemos utilizar o algoritmo QR. Dada uma matriz A de ordem n, este método consiste em construir uma sequência de matrizes, $A_1, A_2, ...$ como segue:

Fazemos $A_1 = A$. Posteriormente realizamos a fatoração QR da matriz A_1, ou seja, A_1 é reescrita como o produto Q_1R_1, isto é, $A_1 = Q_1R_1$ tal que Q_1 é ortogonal e R_1 é uma matriz triangular superior. No que segue, fazemos $A_2 = R_1Q_1$, que também pode ser fatorada como Q_2R_2. Assim, repete-se o processo, obtendo de maneira geral a seguinte estrutura $A_k = R_{k-1}Q_{k-1} = Q_kR_k$.

Este método é uma implementação coerente de iterações simultâneas, no qual por sua vez é uma extensão ou generalização do método das potências. Com isso, passaremos por questões fundamentais como o método das potências, iterações em subespaços, iterações simultâneas e por fim a análise de convergência do algoritmo QR.

7.1 Método das potências

O método das potências consiste em escolhido um vetor v e aplicando a matriz A pela esquerda deste vetor, obtemos a seguinte sequência: $v, Av, A^2v, A^3, ...$ Na prática é preciso redimensionar o vetor em cada passo de maneira ordenada para verificar e julgar se a sequência está convergindo. Assumindo uma estratégia razoável de escala (redimensionamento), a sequência de iterações, em geral, convergirá para um autovetor de A. Isto não é uma tarefa muito difícil de ser analisada. Suponhamos que A possua os seguintes autovalores $\lambda_1, \lambda_2, ..., \lambda_n$ com $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$. Vamos assumir por simplicidade que A é simples, isto é, A tem n autovetores linearmente independentes (LI) $v_1, v_2, ..., v_n$. A hipótese fundamental é aquela que estabelece que $\lambda_1 > \lambda_2$ (determinante para obtermos a noção de taxa de convergência). Vamos começar com um vetor v que pode ser expresso como combinação linear dos autovetores da matriz A como segue: $v = c_1v_1 + c_2v_2 + ... + c_nv_n$.

Sabemos que a definição de autovalores e autovetores é dada por $Av_i = \lambda_i v_i$. Multiplicando pela esquerda o vetor v definido anteriormente, temos:

$$Av = c_1Av_1 + c_2Av_2 + ... + c_nAv_n.$$

E assim, com o uso da definição, segue que:

$$Av = c_1\lambda_1v_1 + c_2\lambda_2v_2 + ... + c_n\lambda_nv_n.$$

Aplicando novamente a matriz A pela esquerda, temos $A^2v = c_1\lambda_1Av_1 + c_2\lambda_2Av_2 + ... + c_n\lambda_nAv_n$ que consequentemente chega-se a $A^2v = c_1\lambda_1^2v_1 + c_2\lambda_2^2v_2 + ... + c_n\lambda_n^2v_n$. De maneira geral, multiplicando A pela esquerda m vezes chegamos no seguinte resultado:

$$A^mv = c_1\lambda_1^mv_1 + c_2\lambda_2^mv_2 + ... + c_n\lambda_n^mv_n.$$

Se λ_1 fosse conhecido previamente, poderiamos redimensionar cada passo por este, isto é, obteríamos a seguinte igualdade:

$$\frac{A^mv}{(\lambda_1^m)} = c_1v_1 + c_2(\frac{\lambda_2}{\lambda_1})^mv_2 + ... + c_n(\frac{\lambda_n}{\lambda_1})^mv_n.$$

Com efeito, essa igualdade converge para o autovetor c_1v_1, considerando-se que c_1 é diferente de zero (a condição de $c_1 \neq 0$ é equivalente a $v \notin \langle v_2, ..., v_n\rangle$, isto é, v não pertence ao subespaço gerado pelos vetores $\langle v_2, ..., v_n\rangle$). A convergência é linear, com razão de convergência dado por aproximadamente $|\lambda_2/\lambda_1|$.

Esta estratégia de redimensionamento não pode ser obtida em problemas reais, mas a escolha exata da estratégia de redimensionamento não é muito importante. O fator de extrema relevância é a direção, não o tamanho aplicado nesta.

Outra estrutura teórica fundamental para a compreensão da convergência do método QR é a ideia de subespaços de iteração.

7.2 Iteração em subespaços

O autovetor v_1 é a representação do auto-espaço $\langle v_1\rangle$. Da mesma forma, a sequência

$$v, Av, A^2v, A^3v, ...,$$

deve ser vista como representativa de um espaço $\langle A^m v\rangle$, no qual todos os componentes desse espaço é gerado por este elemento.

O método visto anteriormente (método da potências) deve ser visto como um processo de iteração de subespaços. Primeiramente inicia-se com um espaço unidimensional $S = \langle v\rangle$ (definido anteriormente como combinação linear dos autovetores de A) escolhido. Então, realiza-se iterações sequenciais a partir desse espaço inicial, isto é, obtém-se a sequência:

1 $S, AS, A^2S, A^3S, ...$ Esta sequência converge para o autoespaço $T = \langle v_1\rangle$ no sentido de que o ângulo entre A^mS e T converge para zero.

De maneira geral, pode-se escolher um subespaço S de dimensão k e construir a sequência definida em **1**. Esta convergirá, em geral, para o subespaço invariante gerado pelos primeiros k vetores. Vamos continuar com a hipótese de que A é simples, ou seja, n autovetores linearmente independentes dados por $v_1, v_2, ..., v_n$. Sejam $T = \langle v_1, v_2, ..., v_k\rangle$, $U = \langle v_{k+1}, v_{k+2}, ..., v_n\rangle$ e vamos assumir que $|\lambda_k > \lambda_{k+1}|$. Ambos, T e U são invariantes sob A, e são definidos como espaços dominante e co-dominante respectivamente. Devemos notar que a sequência **1** em sua maioria converge para T.

Para discutir a convergência de subespaços nós definimos uma medida no conjunto de subespaços k-dimensionais dos C^n como:

$$d(S,T) = \sup_{s\in S;||s||_2} (\inf ||s-t||_2),$$

tal que $||\bullet||_2$ é a norma euclidiana. O resultado principal para a convergência de subespaços é:

Teorema: sejam T e U espaços dominante e co-dominante, respectivamente, como os definidos acima e seja S um subespaço k-dimensional de C^n tal que $S \cap U = (0)$. Então, existe uma constante C tal que $d(A^mS, T) \leq C|\lambda_{k+1}/\lambda_k|^m$ para todo m. Assim $A^mS \to T$ linearmente com razão $|\lambda_{k+1}/\lambda_k|$.

Seja v um vetor não nulo pertencente ao subespaço S. Vamos mostrar que a iteração A^mv se torna cada vez mais próxima de T à medida em que m aumenta. Além disso, v deve ser escrito como $v = c_1v_1 + \cdots + c_kv_k + c_{k+1}v_{k+1} + \cdots + c_nv_n$, isto é, v é representado em termos dos componentes dos espaços T e U. Desde que $v \notin U$, pelo menos um dos coeficientes $c_1, ..., c_k$ deve ser não nulo. Multiplicando sucessivamente o vetor v por A, temos:

$$Av = c_1Av_1 + \cdots + c_kAv_k + c_{k+1}Av_{k+1} + \cdots + c_nAv_n$$

que por definição de autovalor e autovetor chegamos em:

$$Av = c_1\lambda_1v_1 + \cdots + c_k\lambda_kv_k + c_{k+1}\lambda_{k+1}v_{k+1} + \cdots + c_n\lambda_nv_n.$$

Repetindo este processo m vezes e multiplicando por $\frac{1}{\lambda_k^m}$ chegamos no seguinte resultado:

$$A^mv/\lambda_k^m = c_1(\lambda_1/\lambda_k)^mv_1+\cdots+c_{k-1}(\lambda_{k-1}/\lambda_k)^mv_{k-1}+c_k\lambda_k+c_{k+1}(\lambda_{k+1}/\lambda_k)^mv_{k+1}+\cdots+c_n(\lambda_n/\lambda_k)^mv_n$$

Conseguimos perceber que os coeficientes não nulos dos componentes de T aumentam, ou pelo menos não diminuem à medida em que o valor de m aumenta. Ao mesmo tempo os coeficientes dos componentes de U tendem a zero linearmente com taxa dada por $|\lambda_{k+1}/\lambda_k|$. Cada sequência A^mv converge para T na mesma taxa, assim o limite de A^mS converge para T.

Subespaços invariantes são de interesse para obtermos autovalores, pois eles permitem reduzir o problema. De fato, seja $Q = [Q_1 Q_2]$ uma matriz unitária cujas primeiras k colunas (Q_1) formam uma base ortonormal para o subespaço invariante T. Com isso

$$Q^* A Q = \begin{pmatrix} Q_1^* A Q_1 & Q_1^* A Q_2 \\ Q_2^* A Q_1 & Q_2^* A Q_2 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}$$

tal que $Q_2^* A Q_1 = 0$, pois T é invariante.

Assim, o problema de autovalores para a matriz A foi dividido em dois subproblemas menores de autovalores A_{11} e A_{22}. Na prática nunca conseguimos obter um subespaço invariante. Apesar disso, tem-se um subespaço $A^m S$ tal que $d(A^m S, T)$ é pequena.

Seja $P = [P_1 P_2]$ uma matriz unitária tal que as primeiras k colunas de P_1 gerem $A^m S$, e seja:

$$P^* A P = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

Como $A^m S \to T$, seria de se esperar que B_{21} deve convergir para zero na mesma taxa. A volta disto também é válida, isto é, se $B_{21} \to 0$ as colunas geradoras de P_1 se aproximam de um subespaço invariante de A na mesma taxa.

7.3 Iterações simultâneas

Este método é o significado prático de iterações em subespaços. Seja S um subespaço k-dimensional de C^n tal que $S \cap U = 0$. Assim S não contém vetores nulos de A, desde que todos os vetores nulos estejam em U. A partir do **Teorema** visto anteriormente fica claro que $A^m S \cap U = 0$ para todo m e portanto $A^m S$ não possui vetores nulos. Vamos considerar que os vetores $q_1^0, \ldots, q_k^0$ formam uma base de S. Assim, claramente podemos notar que: $Aq_1^0, \ldots, Aq_k^0$ gera AS. Eles são LI também, porque S não tem vetores nulos e portanto formam uma base de S. Da mesma forma $A^m(q_1^0), \ldots, A^m(q_k^0)$ forma uma base de $A^m S$, $m = 2, 3, 4, ...$ Assim em teoria, pode-se iterar sobre uma base de S para obter bases para $AS, A^2 S, A^3 S, \ldots$. Embora essa estrutura seja interessante, existem razões do porque não é aconselhável realizar tal procedimento:

A: os vetores terão que ser redimensionados em ordem com o objetivo de evitar problemas de overflow e underflow.

B: cada sequência $q_i^0, A(q_i^0), A^2(q_i^0), \ldots$ independentemente converge para o subespaço dominante $\langle v_1 \rangle$. Para m suficientemente grande os vetores $A^m(q_1^0), \ldots, A^m(q_k^0)$ tomam quase a mesma direção, isto é, temos uma base mal condicionada. Isto pode ser evitado, mudando cada base obtida em cada passo por uma base bem condicionada no mesmo subespaço. Provavelmente a maneira mais eficiente de fazer isso seja a orto-normalização. Assim, o procedimento de iterações simultâneas é recomendado os procedimentos:

C: dado $q_1^m, \ldots, q_k^m$ uma base ortonormal de $A^m S$, calcula-se $Aq_1^m, \ldots, Aq_k^m$

D: ortonormaliza-se $Aq_1^m, \ldots, Aq_k^m$ da esquerda para a direita chegando em $q_1^{m+1}, \ldots, q_k^{m+1}$, que é uma base ortonormal de $A^{m+1} S$.

O procedimento de iterações simultâneas tem uma propriedade importante em relação à iteração em subespaços de baixa dimensão, sem custo extra. Seja

$$S_i = \langle q_1^0, \ldots, q_i^0 \rangle, i = 1, \ldots, k.$$

Então, $AS_i = \langle Aq_1^0, \ldots, Aq_i^0 \rangle = \langle q_1^1, \ldots, q_i^1 \rangle$ para todo i, desde que o procedimento de ortonormalização preserve o subespaço. Em geral temos

$$A^m S_i = \langle q_1^m, \ldots, q_i^m \rangle, i = 1, \ldots, k.$$

Com isso, iterações simultâneas procuram não somente subespaços de dimensão k, mas também subespaços de dimensões $1, 2, \ldots, k-1$.

De maneira simplificada, vamos verificar o que acontece quando iterações simultâneas é aplicado a um conjunto completo de vetores ortonormais $q_1^0, q_2^0, \ldots, q_n^0$. Para $k = 1, 2, \ldots, n-1$ sejam

$$S_k = \langle q_1^0, q_2^0, \ldots, q_k^0 \rangle, T_k = \langle v_1, \ldots, v_k \rangle, U_k = \langle v_{k+1}, \ldots, v_n \rangle$$

e vamos assumir que $S_k \cap U_k = (0)$ e $|\lambda_k| > |\lambda_{k+1}|$. Assim, $A^m S_k \to T_k$ linearmente quando $m \to \infty$. No que diz respeito a base, isto significa que $q_1^m, \ldots, q_n^m$ convergirá para uma base ortonormal $q_1, q_2, \ldots, q_n$ tal que os primeiros k vetores geram um subespaço invariante T_k.

7.4 O algoritmo QR

Teorema: seja A uma matriz complexa de ordem n. Então existe uma matriz unitária Q e uma matriz triangular superior R tal que $A = QR$. Se A é não singular, então a matriz R deve ser escolhida de tal forma que as entradas da diagonal principal sejam positivas. Neste caso Q e R são unicamente determinados.

Não somente a existência é garantida, mas também Q e R podem ser determinadas por um algoritmo estável com um custo de $\frac{2}{3}n^3$ multiplicações. A decomposição QR é uma realização de orto-normalização de Gram-Schmidt. De fato, vamos supor que a matriz A é não singular, $a_1, \ldots, a_n$ representem suas colunas e $q_1, \ldots, q_n$ sejam as colunas de Q. Então, temos $a_1 = q_1 r_{11}$, $a_2 = q_1 r_{12} + q_1 r_{22}$ e em geral segue que:

$$a_k = q_1 r_{1k} + q_2 r_{2k} + \cdots + q_k r_{kk},$$

$r_{kk} > 0$ e $k = 1, 2, \ldots, n$.

Assim, $\langle a_1 \rangle = \langle q_1 \rangle$, $\langle a_1, a_2 \rangle = \langle q_1, q_2 \rangle$, e em geral $\langle a_1, a_2, \ldots, a_k \rangle = \langle q_1, q_2, \ldots, q_k \rangle$, $k = 1, \ldots, n$. Isto é, as colunas de Q ortonormalizam as colunas de A.

Com o auxílio da decomposição QR, nós vamos expressar iterações simultâneas em forma matricial como segue: seja $\hat{Q}_m$ a matriz cujas colunas são $q_1^m, q_2^m, \ldots, q_n^m$ (assim como foi definido na parte de iterações simultâneas). Se fizermos $D_{m+1} = A\hat{Q}_m$, então as colunas de D_{m+1} são $Aq_1^m, Aq_2^m, \ldots, Aq_n^m$. Estas colunas podem ser ortonormalizadas pela decomposição QR como $D_{m+1} = \hat{Q}_{m+1} R_{m+1}$. Assim, pode-se escrever: **1**

$$D_{m+1} = A\hat{Q}_m, D_{m+1} = \hat{Q}_m R_{m+1}.$$

Uma maneira de verificar se há convergência depois de m passos é realizar transformações similares:

2 $A_m = \hat{Q}_m^* A \hat{Q}_m$ e observar se a matriz A_m converge para uma matriz triangular superior.

Vamos supor que começamos com $\hat{Q}_0 = I$. Isto é, começamos com a base $e_1, \ldots, e_n$ de vetores unitários padrões. Assim, $D_1 = A$ e $A = D_1 = \hat{Q}_1 R_1$. Fazendo $Q_1 = \hat{Q}_1$ nós temos $A = Q_1 R_1$. Encontrando A_1 como uma matriz não triangular superior, tomamos mais um passo (agora temos duas matrizes A e A_1 que podem ser vistas como operadores lineares em sistemas de coordenadas

diferentes). Continuando a operação em A, calculando $D_2 = A\hat{Q}_1$ e $D_2 = \hat{Q}_2 R_2$, ou podemos realizar operações equivalentes em A_1. Um vetor no qual é representado por v no sistema de coordenadas de A é representado por $\hat{Q}_1^* v$ no sistema A_1. Portanto, os vetores $q_1^1, \ldots, q_n^1$ no sistema A se tornam $e_1, \ldots, e_n$ no sistema A_1. Assim equação $D_2 = A\hat{Q}_1$ é equivalente a $A_1 = A_1 I$, e a decomposição QR $D_2 = \hat{Q}_2 R_2$ é equivalente a decomposição QR de A_1 $A_1 = Q_2 R_2$ (o R_2 é o mesmo nestas decomposições anteriores devido à unicidade da decomposição QR).

Se tivéssemos escolhido operar com a matriz A_1 poderíamos verificar a convergência calculando $A_2 = \hat{Q}_2 A_1 Q_2 = R_2 Q_2$. A equação $\hat{Q}_2 = Q_1 Q_2$ garante que A_2 é a mesma que aquela em **2**. Nós podemos continuar esse processo para produzir uma sequência de matrizes A_m, onde $A_{m-1} = Q_m R_m$, $A_m = R_m Q_m$ **3**.

Este é o algoritmo QR, e como nós já vimos, é equivalente com as iterações simultâneas. A matriz **3**(A_m) é a mesma que aquela apresentada em **2**. R_m de **3** é o mesmo que aquele de **1**, e o Q_m de **3** está relacionado com o $\hat{Q}_m$ de **1** como:

4 $\hat{Q}_m = Q_1 Q_2 \ldots Q_m$.

Q_m é a mudança de coordenada no passo m, enquanto que $\hat{Q}_m$ é a mudança acumulada de coordenadas depois de m passos.

Tendo estabelecido que QR é iteração simultânea começando com os vetores $e_1, \ldots, e_n$, nós podemos concluir que a sequência A_m gerada por QR converge para uma forma triangular (ou pelo menos triangular em bloco), desde que as condições de subespaços fornecidas:

$$\langle e_1, \ldots, e_k \rangle \cap \langle v_{k+1}, \ldots, v_n \rangle = (0), k = 1, \ldots, n-1$$

sejam satisfeitas.

7.5 Observações

7.5.1 Método das potências e suas extensões

Vamos assumir que estamos lidando com uma matriz $A \in C^{nxn}$ que é semi-simples, isto é, que possui n autovetores LI, associados com n autovalores, respectivamente. Além disso, consideremos que $\lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n$ e que q pode ser escrito como combinação linear dos autovetores de A, ou seja:

$$q = c_1 v_1 + c_2 v_2 + \cdots + c_n v_n.$$

Multiplicando pela esquerda o veto q, obtemos:

$$Aq = c_1 A v_1 + c_2 A v_2 + \cdots + c_n A v_n.$$

E pela definição de autovalores e autovetores $Av_i = \lambda_i v_i$, obtemos a seguinte igualdade:

$$Aq = c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \cdots + c_n \lambda_n v_n.$$

Aplicando A j vezes seguidamente à esquerda do vetor q, chegamos no seguinte resultado:

$$A^j q = c_1 \lambda_1^j v_1 + c_2 \lambda_2^j v_2 + \cdots + c_n \lambda_n^j v_n.$$

Desta última igualdade podemos colocar λ_1 em evidência e assim:

$$A^j q = \lambda_1^j (c_1 v_1 + c_2 (\lambda_2/\lambda_1)^j v_2 + \cdots + c_n (\lambda_n/\lambda_1)^j v_n).$$

Com a igualdade encontrada anteriormente e considerando o fator λ_1^j irrelevante, notamos que o componente $c_1 v_1$ se mantém fixo e os outros componentes tendem a zero com taxa $|\lambda_2/\lambda_1|^j$. Assim, observamos a convergência para um múltiplo do vetor v_1.

A cada multiplicação realizada anteriormente para chegarmos na forma $A^j q$, isto é, cada $q, Aq, A^2q, \ldots$, nada mais é do que a representação unidimensional de um subespaço que cada componente gera.

Com isso, podemos reescrever e assim reinterpretar $q, Aq, A^2q, \ldots$ como uma sequência de subespaços unidimensionais $S, AS, A^2S, A^3S, \ldots$ que convergem para o auto-espaço que é gerado po v_1.

Assim, conseguimos verificar que o método das potências é um método de iteração de subespaços (conceito descrito anteriormente).

Neste ponto é vantajoso realizar uma pequena generalização. Seja m um número pequeno tal como $1, 2, 4, 6$, escolha m shifts $\rho_1, \ldots, \rho_m \in C$, e faça

$$p(A) = (A - \rho_1 I)(A - \rho_2 I) \ldots (A - \rho_m I).$$

$p(A)$ tem os mesmos autovetores de A, e autovalores correspondentes $p(\lambda_1), \ldots, p(\lambda_n)$. Vamos ordenar tais valores como $|p(\lambda_1)| \geq |p(\lambda_2)| \geq \ldots |p(\lambda_n)|$. Então se $|p(\lambda_1)| > |p(\lambda_2)|$, os subespaços de iterações dados por:

$$S, p(A)S, p(A)^2S, p(A)^3S, \ldots$$

impulsionados por $p(A)$ (para quase toda a escolha de um subespaço inicial S) convergem para o subespaço invariante gerado por $v_1, \ldots, v_k$ linearmente com razão $|p(\lambda_{k+1})/p(\lambda_k)|$.

A vantagem é dupla ao se realizar esta generalização. Primeiro, os shifts realizados (essenciais) permitem rápida convergência. Segundo, é um preparativo para as iterações QR.

Proposição: Para $A \in C^{nxn}$, existe uma matriz unitária $Q \in C^{nxn}$ tal que $Qe_1 = \beta x$ para algum $\beta \neq 0$ e $B = Q^* AQ$ é uma matriz superior de Hessenberg. Q e B podem ser calculados diretamente em aproximadamente $\frac{16}{3}n^3$ flops. Q é o produto de $n-1$ reflexões.

7.5.2 Matrizes superiores de Hessenberg

A chave para a eficiência é trabalhar com matrizes de Hessenberg. Uma matriz H é superior de Hessenberg se $h_{ij} = 0$ sempre que $i > j + 1$. Devido ao fato de matrizes de Hessenberg serem próximas da forma triangular, elas são baratas de se trabalhar.

Toda matriz $A \in C^{nxn}$ é unitariamente similar à forma superior de Hessenberg. A transformação similar pode ser realizada por uma sequência de $n-1$ reflexões (transformações de Householder) a um custo de $O(n^3)$ flops.

7.5.3 Subespaços de Krylov

O subespaço de Krylov está intimamente relacionado com as matrizes de Hessenerg. Dado $x \neq 0$, o j-ésimo subespaço de Krylov associado com A e x, denotado por $K_j(A, x) = span\{x, Ax, A^2x, \ldots, A^{j-1}x\}$ (interessante notar a estrutura que definimos anteriormente em relação ao produto em sequência da matriz A e o vetor q, e o subespaço agora definido).

Sejam $e_1, \ldots, e_n$ as colunas da matriz identidade nxn, como o usual. Uma matriz superior de Hessenberg é propriamente superior se $h_{ij} \neq 0$ sempre que $i = j + 1$.

Lema 1: se H é propriamente superior de Hessenberg, então $K_j(A, e_1) = span\{e_1, e_2, ..., e_j\}$, $j = 1, ..., n$.

Lema 2: se $x = p(A)e_1$, então $p(A)K_j(A, e_1) = K_j(A, x)$, $j = 1, ..., n$.

A ligação entre o subespaço de Krylov e matrizes propriamente superiores de Hessenberg: em transformações similares para obtermos a forma superior de Hessenberg, as colunas principais da matriz de transformação geram subespaços de Krylov.

Proposição: suponha que $B = Q^{-1}AQ$ e B seja propriamente superior de Hessenberg. Seja $q_1, ..., q_n$ as colunas de Q. Então,

$$span\{q_1, ..., q_n\} = K_j(A, q_1).$$

7.5.4 Aspectos de convergência

Os passos dados na fatoração QR são efetivamente subespaços de iteração dirigido por uma matriz fixa: $p(A) = \alpha(A - \rho_1 I) \dots (A - \rho_m I)$, tal que $\rho_1, ..., \rho_m$ são shifts determinados previamente.

Devido a mudança do sistema de coordenada em cada passo, esta versão de subespaços de iteração mantém o subespaço fixo e muda a matriz em cada passo.

Vamos supor que os autovalores são: $|p(\lambda_1)| \geq |p(\lambda_2)| \geq \cdots \geq |p(\lambda_n)|$.

Para cada j no qual $|p(\lambda_j)| > |p(\lambda_{j+1})|$, o subespaço $span\{e_1, ..., e_j\}$ fica cada vez mais perto de um subespaço invariante de A_k, à medida em que k aumenta. $span\{e_1, ..., e_j\}$ é invariante em relação a A_k, se e somente se,

$$A_K = \begin{pmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{pmatrix}, \quad A_{11} \in C^{jxj}$$

nós inferimos que a convergência do subespaço de iteração implicará em convergência de A_k para a matriz de bloco triangular dada na forma acima. Isto acontece não só para uma escolha de j, mas pra todos os valores de j no qual $|p(\lambda_j)| > |p(\lambda_{j+1})|$.

Agora consideremos a situação na qual não houve convergência, mas muito perto de convergir. Então, nós temos:

$$A_k = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

onde A_{21} tem uma entrada não nula, na qual é pequena. Os autovalores de A_{22} não são $\lambda_{n-m+1}, ..., \lambda_n$, mas eles estão perto. Neste ponto faz sentido utilizarmos os m autovalores de A_{22} como novos shifts $\rho_1, .., \rho_m$ para iterações subsequentes. O novo $p(z) = (z - \rho_1)(z - \rho_2) \dots (z - \rho_m)$ terá $p(\lambda_{n-m+1}), ..., p(\lambda_n)$ muito pequenos, porque cada um desses λ_j é bem próximo de um dos ρ_i. Por outro lado, nenhum dos $p(\lambda_1), ..., p(\lambda_{n-m})$ será pequeno, por que nenhum dos λ_j é próximo o suficiente de qualquer um dos shifts. Assim, a razão $|p(\lambda_{n-m+1})/p(\lambda_{n-m})|$ vai ser muito mais pequena do que 1, e a convergência será acelerada. Depois de mais algumas iterações, nós teremos aproximações muito melhores para $\lambda_{n-m+1}, ..., \lambda_n$, e nós podemos usar estes como novos shifts, acelerando ainda mais a convergência.

8 Conclusão

8.1 Sobre as Decomposições

Como visto ao longo das demonstrações das decomposições, a fatoração SVD pode ser vista como uma extensão da fatoração de Schur que, por sua vez, é uma extensão da decomposição espectral

no caso de simetria, no campo dos reais. Adicionalmente, vimos que a ligação entre elas está na matriz A, que é um mapeamento linear levando x em $Ax = y$, e que este mapeamento tem como implicação geométrica o fato de que A transforma uma híper-esfera S em uma híper-elípse Ax. Vimos também as implicações teóricas desta decomposição : se a matriz A for definida positiva e simétrica, então as fatorações coincidem e os valores singulares se tornam autovalores. Este fato, além de esboçar a belíssima entre as decomposições, é muito útil de ponto de vista téorico, assim como as implicações algébricas da Fatoração SVD, que podem ser encontradas na página 14.

Para finalizar, estas fatorações não têm somente aplicações teóricas. Como exemplo, temos o processo de compressão de imagens, que tem como objetivo reduzir o tamanho da imagem sem reduzir drasticamente sua qualidade, assunto explorado nas páginas de número 5 até 11. Esta aplicação tem papel muito importante atualmente por causa do advento do celular, pois se tornou rápido e fácil, hoje, tirarmos fotografias (para dar mais peso à importância da técnica, vale relembrar que programas como Snapchat usam fotografias como sua principal fonte de renda).

8.2 Sobre os Sistemas Não Lineares

Sobre sistemas de equações não lineares, pode-se observar que muitas técnicas são provindas de problemas, em geral, complicados e que muitas vezes não tem solução analítica desenvolvidas (ver, *e.g*, [Kel03], seção "Equação-H de Chandrasekhar", que tem origem, por exemplo, na transferência de radiação e seção "Equações de Ornstein-Zernike", com origens na física-estatística), servindo de motivação para buscarmos a abordagem numérica para aproximação de soluções . Um exemplo não tão complexo quanto geralmente é, mas que é bastante representativo da classe a que o problema pertence é a aproximação numérica da solução do sistema de equações diferenciais na página 27.

Também foi percebido que existem muitos métodos derivados do Método de Newton, sendo derivados a partir de modificações desta técnica. O Método de Broyden, por exemplo, utiliza as mesmas suposições que o de Newton, mas utiliza métodos numéricos de aproximação de derivadas (mais especificamente, o diferenças finitas) para contornar a grande quantidade de flops envolvidos à avaliação e inversão da matriz Jacobiana, sem que a matriz resultante da aproximação perca as propriedades não *ad. hoc.* de $J(\mathbf{x})$. Entretanto, apesar de conseguirmos evitar uma grande quantidade de operações, a quantidade total de flops se mantem é elevada e ainda temos que sua taxa de convergência cai, passando de quadrática (ao se utilizar Newton) para superlinear. Neste caso, porém, trocar taxa de convergência por uma redução dramática no número de operações ainda se configura como uma vantagem, o que faz o Método de Broyden ser relevante para aplicações.

8.3 Sobre a Equação de Sylvester e sua lei de Inércia e o algoritmo QR

Inspirados pelas questões propostas no projeto conseguimos averiguar o rigor matemático necessário para construção e consolidação das teorias estudadas. Conseguimos estabelecer as conexões com o conteúdo apresentado em sala de aula, assim como observar a amplitude de aplicações de conceitos. Mais do que isso, diagnosticamos que com operações fundamentais, como soma e multiplicação de matrizes, pode-se obter teorias muito ricas.

Por exemplo, no estudo da Equação de Sylvester pudemos entrar em contato com a teoria de processos estocásticos, relacionado com as entradas aleatórias das matrizes e o processo de diagonalização de matrizes por blocos. Além disso, tal equação aparece em teoria de controle e redução de modelos. Estes assuntos estão associados com matrizes suficientemente grandes, esparsas e com posto pequeno (número de colunas ou linhas linearmente independentes).

Ademais, estudamos também a Lei da Inércia de Sylvester, no qual nos possibilita calcular os autovalores de uma matriz por uma transformação chamada de transformação congruente. Esta lei pode ser aplicada em matrizes Hermitianas tri-diagonais com o objetivo de se calcular os autovalores desta.

No que diz respeito ao algoritmo QR pudemos revisitar questões importantes de álgebra linear como resolução de sistemas lineares (assim como na Equação de Sylvester), conceitos de espaços e subespaços vetoriais, operadores lineares entre outros.

O algoritmo QR é de fácil compreensão ao construir sequências de matrizes e nestas aplicar a fatoração QR. O nosso principal suporte teórico com grande peso foi a teoria desenvolvida baseada na iteração de subespaços e iterações simultâneas. Nas iterações em subespaços temos, de forma simplificada, que uma sequência dada pelo produto de matriz vetor converge para o subespaço dominante e invariante. Porém, na prática, tem-se a distância entre os subespaços como uma maneira de determinar a convergência destes. Em relação as iterações simultâneas é a aplicação das iterações em subespaços conjuntamente com o processo de orto-normalização.

Referências

[Apo69] Tom M. Apostol. *Calculus*, volume 2. John Wiley and Sons, second edition edition, 1969.

[Dem97] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1st edition, September 1997.

[eJDF13] L. Burden e J. Douglas Faires. *Análise Numérica*. Cengage Learning, tradução da 8ª edição norte-americana edition, 2013.

[eVLRL96] M. A. G. Ruggiero e V. L. R Lopes. *Cáculo Numérico, Aspéctos Teóricos e Computacionais*. Pearson Education, 2 edition, 1996.

[eYS14] Abd-Krim Seghouane e Yousef Saad. Prewhitening high dimensional fmri data sets without eigendecomposition. *Neural Computation*, Maio 2014.

[GL96] G.H.Golub and C.F.van Loan. *Matrix Computations*. The Johns Hopkins University Press., 3 edition, October 1996.

[Hig02] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. SIAM, segunda edition, Agosto 2002.

[Hig04] Nicholas J. Higham. Sylvester's influence on applied mathematics. *Mathematics Today, 50(4)*, pages 202–206, Agosto 2004.

[JB80] V. L. Figueiredo H. G. Wetzler J.S Boldrini, S. I. R. Costa. *Algebra Linear*. Harbra, 3 edition, 1980.

[JW06] Chunlei Liu Jin Wang. Generating multivariate mixture of normal distributions using a modified cholesky decomposition. *Proceedings of the 2006 Winter Simulation Conference*, 2006.

[Kel03] C.T Kelley. *Solving Nonlinear Equations with Newton's Method*. SIAM, 2003.

[Mol04] Cleve Moler. *Numerical Computing with MATLAB*. The MathWorks, eletronic edition, 2004. http://www.mathworks.com/moler/index_ncm.html.

[oT] Massachusetts Institute of Technology. Svd decomposition. http://math.mit.edu/~gs/dela/.

[Pul15] P. Pulino. *Algebra Linear e suas Aplicações: Notas de Aula*. IMECC, UNICAMP, Janeiro 2015. http://www.ime.unicamp.br/~pulino/ALESA.

[TB97] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[Wat02] D. S. Watkins. *Fundamentals of Matrix Computations*. New Jersey: John Wiley and Sons, 2 e.d edition, 2002.

[Wat10] D. S. Watkins. *Fundamentals of Matrix Computations*. New Jersey: John Wiley and Sons, 3 e.d edition, 2010.

[Wic02] Rick Wicklin. Use the cholesky transformation to correlate and uncorrelate variables, Fevereiro 2002. http://blogs.sas.com/content/iml/2012/02/08/use-the-cholesky-transformation-to-correlate-and-uncorrelate-variables.html.

[Wik] Wikipedia. Multivariate normal distribution, drawing values from the distribution. https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Drawing_values_from_the_distribution.