UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE MATEMÁTICA

ALISSON LUCAS DE SOUZA

UMA ABORDAGEM CONTÍNUA DO PROBLEMA DA GEOMETRIA DE DISTÂNCIAS PARA RECONSTRUÇÃO E ALINHAMENTO DE PROTEÍNAS

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2017

ALISSON LUCAS DE SOUZA

UMA ABORDAGEM CONTÍNUA DO PROBLEMA DA GEOMETRIA DE DISTÂNCIAS PARA RECONSTRUÇÃO E ALINHAMENTO DE PROTEÍNAS

Trabalho de Conclusão de Curso apresentado ao Departamento Acadêmico de Matemática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do grau de "Licenciado em Matemática".

Orientador: Dr. André Luís Machado Martinez

CORNÉLIO PROCÓPIO

2017

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação
Departamento de Matemática
Curso de Licenciatura em Matemática

UTFPR
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

FOLHA DE APROVAÇÃO

BANCA EXAMINADORA

André Luís Machado Martinez
(orientador)

Glaucia Maria Bressan

Elenice Weber Stiegelmeier

"A Folha de Aprovação assinada encontra-se na Coordenação do Curso"

AGRADECIMENTOS

Teço meus sinceros agradecimentos às pessoas especiais que participaram dessa minha caminhada na gradução: aos meus familiares (em especial ao meu pai Airton e minha irmã Jacqueline, sem os quais eu nem teria todas as oportunidades que tive), aos meus amigos (Lucas, desde muito antes do curso, Sérgio, desde o começo, Luiz Otávio e Wendell, muito presentes, por terem sempre me apoiado; e, especialmente, Marcelo, por ter me indicado desde a iniciação científica até o tema deste trabalho), aos meus professores (Débora Albanez, por ter me acolhido sempre, Anderson Paião, por ter me ensinado muito mais do que eu imaginava, Thiago Pinguello, por ter sempre me encorajado a ser cada vez mais, Michele Valentino, por sempre me apoiar e acreditar em mim), e, finalmente, ao meu orientador André Martinez por ter aceitado me ensinar desde o início, sempre me apoiando e me aconselhando, além de me ensinar que o principal é ser **humano**.

Melhor que existir, é ser único! (Anderson Paião)

RESUMO

SOUZA, Alisson Lucas. UMA ABORDAGEM CONTÍNUA DO PROBLEMA DA GEOMETRIA DE DISTÂNCIAS PARA RECONSTRUÇÃO E ALINHAMENTO DE PROTEÍNAS. 138 f. Trabalho de Conclusão de Curso – Departamento Acadêmico de Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Neste trabalho estudamos conceitos matemáticos preliminares de Álgebra Linear, Hipercomplexos e Otimização, relacionados ao Problema da Geometria de Distâncias (PGD). A partir destes conceitos, abordamos o PGD para reconstrução de proteínas por meio da aplicação de métodos de otimização contínua. O trabalho, ainda, contempla os estudos do problema *Procrustes* para alinhamento molecular. Apresentamos testes numéricos da resolução do PGD utilizando os métodos de otimização não-linear BFGS e Gauss-Newton, com dados retirados do *Protein Data Bank*, e discutimos os resultados obtidos.

Palavras-chave: Problema da Geometria de Distâncias, Otimização Contínua, Alinhamento Molecular, Proteínas.

ABSTRACT

SOUZA, Alisson Lucas. A CONTINUOUS APPROACH OF THE DISTANCE GEOMETRY PROBLEM FOR PROTEIN RECONSTRUCTION AND ALIGNMENT. 138 f. Trabalho de Conclusão de Curso – Departamento Acadêmico de Matemática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

In this work we study preliminary mathematical concepts of Linear Algebra, Hypercomplex and Optimization, related to the Distance Geometry Problem (DGP). From these concepts, we approach PGD for protein reconstruction through the application of continuous optimization methods. The work also contemplates the studies of the problem Procrustes for molecular alignment. We apresent numerical tests of the resolution of DGP using the non-linear optimization methods BFGS and Gauss-Newton, with data taken from the Protein Data Bank, and we discuss the obtained results.

Keywords: Distance Geometry Problem, Continuous Optimization, Molecular Alignment, Proteins.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A Geometria de Distâncias (GD) é uma área atualmente consolidada, que possui a Matemática e a Computação como áreas fundamentais em seu alicerce. A GD surgiu em 1928, quando Menger Menger (1928) caracterizou vários conceitos geométricos utilizando a ideia de distância. Entretanto, apenas com os resultados de Blumenthal Blumenthal (1953) que o tema se tornou, de fato, uma nova área do conhecimento.

Em seus primórdios, a grande questão da GD era encontrar condições necessárias e suficientes para decidir se uma dada matriz D é uma Matriz de Distâncias Euclidianas, ou seja, se é uma matriz simétrica tal que existe um número inteiro positivo K e um conjunto de pontos em $\mathbb{R}^K$, onde as distâncias Euclidianas entre esses pontos são iguais às entradas da matriz D (LIMA, 2012). Esse resultado foi alcançado por Schoenberg Schoenberg (1935) e discutiremos sobre ele mais a frente do trabalho.

Atualmente o principal problema da GD, denominado como Problema da Geometria de Distâncias (PGD), consiste em determinar um conjunto de pontos em um dado espaço topológico, cujas distâncias, entre alguns deles, são conhecidas.

O conceito de distância é amplo e está presente em diversas áreas do conhecimento, então, consequentemente, a GD possui diversas aplicações, tais como: robótica, redes sem-fio, biologia molecular, entre outras (LIBERTI et al., 2014). A aplicação da GD na Biologia Molecular, em específico no cálculo estrutural de Proteínas, é o objetivo principal do presente trabalho.

As proteínas formam uma importante classe das biomoléculas. Elas são como uma chave para que os sistemas biológicos exerçam certas funções, porém para compreendê-las juntamente com suas funções é necessário encontrar suas estruturas tridimensionais. Existem duas principais abordagens experimentais para a determinação de uma estrutura proteica: a primeira é a Cristalografia em Raios-X e a segunda é a Espectroscopia de Ressonância Magnética Nuclear (NMR, do inglês *Nuclear Magnetic Ressonance*) (LIMA, 2012).

Na Espectroscopia NMR são detectados valores das distâncias entre certos pares de

átomos de uma dada proteína. Com isso, o problema relacionado é, então, encontrar as coordenadas tridimensionais dos átomos dado o conjunto de distâncias intra-átomos. Este problema é conhecidio como o Problema da Geometria de Distâncias Moleculares (PGDM), que nada mais é um PGD considerando os átomos de uma proteína como pontos no $\mathbb{R}^3$.

Suponha que temos uma proteína com n átomos. Seja $\{x_i \,|\, i = 1,2,\ldots,n\}$ o conjunto das coordenadas tridimensionais desses átomos, a saber $x_i = [x_{i,1}, x_{i,2}, x_{i,3}]^T$, onde $x_{i,1}$, $x_{i,2}$ e $x_{i,3}$ são, respectivamente, a primeira, a segunda e a terceira coordenadas do átomo x_i. Se as coordenadas x_i, $i = 1,2,\ldots,n$ são conhecidas, então as distâncias d_{ij} entre os átomos i e j podem ser facilmente calculadas como $d_{ij} = \|x_j - x_i\|$. Por outro lado, se as distâncias d_{ij} entre os átomos i e j são dadas, então as coordenadas $x_1, x_2, \ldots, x_n$ dos átomos também podem ser obtidas com base nas distâncias d_{ij}, porém este cálculo não é tão simples.

Podemos, então, definir o PGDM como: encontre as coordenadas tridimensionais dos átomos $x_1, x_2, \ldots, x_n$ de modo que

$$\|x_i - x_j\| = d_{ij}, \text{ para } (i,j) \in S,$$

onde S é um subconjunto de pares de átomos dos quais as distâncias são conhecidas.

Existem diversas formas de resolver este problema e, primordialmente, existem duas abordagens que englobam essas resoluções: na primeira, pesquisadores formulam um problema de Otimização Não-Linear e empregam métodos contínuos para obterem uma solução aproximada (LIMA, 2012; ALFAKIH et al., 1999; GLUNT et al., 1993; TROSSET, 2000); na segunda, o problema é resolvido utilizando técnicas de otimização discreta e teoria de grafos (LAVOR et al., 2011; LIBERTI et al., 2011). Neste trabalho, abordaremos o PGDM de forma contínua, de modo que utilizaremos métodos clássicos de otimização não-linear tanto para uma formulação com funções de várias variáveis, quanto com sistemas de equações não-lineares.

Além da reconstrução de uma estrutura proteica, um problema é o de alinhamento de proteínas, que no nosso trabalho em específico, consistirá em sobrepor a estrutura da proteína dada originalmente (de um Banco de Dados) com a estrutura gerada pela solução do PGDM. Esse problema de sobrepor duas estruturas é conhecido na literatura como *Procrustes*, que busca minimizar o valor do RMSD (do inglês, *Root Mean Square Deviation*) dessas duas estruturas, de modo que, quanto mais parecidas são, menor é o valor do RMSD (LIMA, 2012). Utilizaremos duas abordagens para resolver esse problema de alinhamento molecular, uma que utiliza decomposição em valores singulares e outra que considera quatérnios unitários.

Nosso trabalho está dividido do seguinte modo: no Capítulo 2, faremos um estudo

preliminar de quatérnios. Este Capítulo, juntamente com o Apêndice A, tem como objetivo fazer com que o leitor se situe no contexto dos resultados mais avançados que serão tratados nos Capítulos a seguir.

No Capítulo 3, abordaremos os conceitos elementares de Otimização, apresentando as condições de Otimalidade e os métodos clássicos de Otimização e de resolução de sistemas de equações. Esses métodos são essenciais para a nossa abordagem de resolução do Problema da Geometria de Distâncias.

No Capítulo 4, apresentaremos uma revisão dos conceitos básicos de Biologia relacionados a proteína, suas representações e, como foco principal, como funcionam as técnicas experimentais de determinação das estruturas proteicas. As proteínas, caracterizadas neste trabalho como objetos de aplicação, serão introduzidas de forma sucinta, para que o trabalho possa ser compreendido sem dificuldades pelos leitores.

No Capítulo 5, estudamos inicialmente o problema de ajustes de distâncias genéricas entre pares de pontos do espaço tridimensional Euclidiano. Como aplicação deste problema, investigamos a tarefa de reconstruir a estrutura tridimensional de uma proteína dadas as distâncias entre seus átomos. Mostraremos por meio um teorema quais são as condições necessárias e suficientes para que um conjunto de números reais positivos sejam distâncias Euclidianas entre pontos em um determinado espaço. Discutiremos o problema *Procrustes* para alinhamento molecular e estudaremos duas abordagens da literatura para a resolução deste problema.

No Capítulo 6, realizaremos dois testes numéricos utilizando os métodos clássicos de Otimização Não-Linear introduzidos no Capítulo 4 para resolver o Problema da Geometria de Distâncias aplicado em reconstrução de Proteínas. Em cada teste realizado, também, faremos uma comparação da estrutura original da proteína com a estrutura proteica da solução gerada pela resolução do PGD associado por meio do problema *Procrustes* e discutiremos o alinhamento das proteínas em cada caso. E, por fim, apresentamos as conclusões obtidas do trabalho e as perspectivas futuras no Capítulo 7.

2 CONCEITOS PRELIMINARES

Para a revisão teórica dos conceitos matemáticos utilizados neste trabalho, escrevemos dois textos: o presente neste capítulo, cujo conteúdo é uma introdução da álgebra hipercomplexa dos Quatérnios, e o texto complementar no Apêndice A, cujo conteúdo contempla os tópicos de Álgebra Linear, normas e fatoração de matrizes. As notações, definições e os resultados, trabalhados nesse capítulo, podem ser encontrados em Sehnal (2008), Coutsias et al. (2004), Milies (2004) e Hamilton (1844).

2.1 QUATÉRNIOS

Durante um período de intensa atividade na direção de uma crescente abstração que um notável matemático, Sir William Rowan Hamilton (1805-1865), concebeu a fundamentação definitiva dos números complexos como pares ordenados de números reais.

Sua reformulação, da teoria dos números complexos, parte de uma observação muito simples: Hamilton nota que a expressão $a+bi$ não denota uma soma genuína do mesmo tipo que $2+3$ e afirma que o uso do sinal $+$ é um acidente histórico e, certamente, bi não pode ser adicionado a a. Assim, percebe que escrever um número complexo na forma $a+bi$ não é mais do que o par ordenado de números reais (a,b). A partir dessa observação, Hamilton desenvolve a teoria formal, definindo soma e produto de pares da forma a seguir:

$$(a,b)+(c,d) = (a+c, b+d),$$
$$(a,b)(c,d) = (ac-bd, ad+bc).$$

Hamilton, também, era um físico e percebia claramente as implicações de sua descoberta, pois, havia desenvolvido uma álgebra que permitia trabalhar com os vetores do plano. Isso o levou a considerar um problema que seria fundamental para a física da época: desenvolver uma álgebra de ternas que daria a linguagem para trabalhar com vetores do espaço.

Trabalhou durante dez anos neste problema antes de descobrir onde estava a dificul-

dade essencial. Uma carta a seu filho Archibald, de Outubro de 1843, revela sua obsessão com a questão:

> Toda manhã, quando descia para o café, teu irmão William Edwin e você mesmo costumavam perguntar-me "Bem, pai, você já pode multiplicar ternas?" A isso eu sempre me via obrigado a responder, com um triste balanço de cabeça, "Não, eu posso somá-las ou subtrá-las".(LIMA, 2012, p. 59).

Para compreender como poderia ser feita esta multiplicação, Hamilton escrevia suas ternas na forma $a+bi+cj$, por semelhança ao que era feito com os complexos e tentava desenvolver o produto $(a+bi+cj)(x+yi+zj)$ e representá-lo como um elemento da mesma forma. Esperava ainda que o cumprimento dos vetores fosse igual ao produto dos comprimentos, isto é, $a^2+b^2+c^2=x^2+y^2+z^2$ fato que chamou de lei dos módulos.

Para desenvolver o produto, assumiu naturalmente que $i^2=j^2=-1$, mas a dificuldade estava em determinar qual deveria ser o valor dos produtos ij e ji. Foi a tentativa de preservar a lei dos cossenos que lhe impôs finalmente a necessidade de trabalhar com uma dimensão a mais.

Numa carta a seu filho, Hamilton descreve como foi a descoberta final:

> Mas no dia 16 de Outubro de 1843 - que era uma segunda-feira e reunião do Conselho da Real Sociedade da Irlanda - eu ia andando para participar e presidir, e tua mãe andava comigo, ao longo do Royal Canal, ..., embora ela falasse comigo ocasionalmente, uma corrente subjacente de pensamento estava acontecendo na minha mente, que finalmente teve um resultado, cuja importância senti imediatamente. Pareceu como se um circuito elétrico tivesse se fechado; e soltou uma faísca, o heraldo de muitos anos vindouros de pensamento trabalhado dirigidos, por mim, se poupando, e de qualquer forma por parte dos outros, se eu vivesse o suficiente para comunicar minha descoberta. Nesse instante eu peguei uma libreta de anotações que ainda existe e fiz um registro naquela hora. Não pude resistir ao impulso - tão não filósofo quanto possa ser - de gravar com um canivete numa pedra na ponte de Brougham, quando a cruzamos, a fórmula fundamental dos símbolos i, j, k,
>
> $$i^2=j^2=k^2=ijk=-1,$$
>
> que contém a solução do Problema. (LIMA, 2012, p. 59).

Com a multiplicação definida dessa forma, o conjunto dos quatérnios constitui o primeiro exemplo de anel não-comutativo, com divisão. Hamilton reconheceu imediatamente a importância de sua descoberta, especialmente pelas implicações para o desenvolvimento da física. No dia seguinte, em 17 de Outubro de 1843, Hamilton escreveu a seu amigo John T. Graves comunicando-lhe seus resultados. A semente de novos desenvolvimentos havia sido

plantada: em Dezembro desse mesmo ano, Graves descobriu uma álgebra de dimensão 8, os octônios.

O conjunto dos quatérnios, denotado por $\mathbb{H}$ (em homenagem a Hamilton), foi uma descoberta incrível e tem diversas implicações na Matemática e na Física. Por analogia com os números complexos sendo representado como a soma de uma parte real e uma imaginária, $a+bi$, um quatérnio também pode ser escrito como uma combinação linear.

Definição 2.1.1 *Um quatérnio $a \in \mathbb{H}$ é da forma*

$$a = a_0 + a_x i + a_y j + a_z k,$$

com a parte real $a_0 \in \mathbb{R}$ e a parte imaginária $a_x, a_y, a_z \in \mathbb{R}$ onde

$$i^2 = j^2 = k^2 = ijk = -1$$

e

$$\begin{aligned} ij = k, \quad jk = i, \quad ki = j, \\ ji = -k, \quad kj = -i, \quad ik = -j. \end{aligned}$$

Levando em consideração a intenção de Hamilton, ao representar um número complexo na forma vetorial, também podemos representar um quatérnio a como um vetor em $\mathbb{R}^4$:

$$a = (a_0, a_x, a_y, a_z) \in \mathbb{R}^4.$$

Assim, como fizemos com as matrizes e vetores, vamos agora apresentar as operações elementares dos quatérnios. Para isso, considere os quatérnios $a = a_0 + a_x i + a_y j + a_z k$ e $b = b_0 + b_x i + b_y j + b_z k$.

Definição 2.1.2 *A adição dos quatérnios a e b é dada por:*

$$a + b = (a_0 + b_0) + (a_x + b_x)i + (a_y + b_y)j + (a_z + b_z)k.$$

Definição 2.1.3 *A multiplicação entre um escalar $\alpha \in \mathbb{R}$ e um quatérnio é definida por:*

$$\alpha a = \alpha a_0 + \alpha a_x i + \alpha a_y j + \alpha a_z k.$$

Definição 2.1.4 *A multiplicação entre dois quatérnios é dada por:*

$$\begin{aligned} ab &= (a_0b_0 - a_xb_x - a_yb_y - a_zb_z) \\ &\quad + (a_0b_x + a_xb_0 + a_yb_z - a_zb_y)i \\ &\quad + (a_0b_y - a_xb_z + a_yb_0 + a_zb_x)j \\ &\quad + (a_0b_z + a_xb_y - a_yb_x - a_zb_0)k. \end{aligned} \tag{2.1.1}$$

Os quatérnios não são comutativos na multiplicação, ou seja, existem $x, y \in \mathbb{H}$ tais que $xy \neq yx$. Além disso, é conveniente expressar a multiplicação de dois quatérnios em termos de um produto de uma matriz $A \in \mathbb{R}^{4\times4}$ por um vetor. Pode-se escolher expressar o primeiro ou o segundo quatérnio em um produto como uma matriz como especificado a seguir:

$$ab = \begin{bmatrix} a_0 & -a_x & -a_y & -a_z \\ a_x & a_0 & -a_z & a_y \\ a_y & a_z & a_0 & -a_x \\ a_z & -a_y & a_x & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_x \\ b_y \\ b_z \end{bmatrix} = Ab$$

e

$$ba = \begin{bmatrix} a_0 & -a_x & -a_y & -a_z \\ a_x & a_0 & a_z & -a_y \\ a_y & -a_z & a_0 & a_x \\ a_z & a_y & -a_x & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_x \\ b_y \\ b_z \end{bmatrix} = \hat{A}b.$$

Note que $\hat{A}$ se difere de A na sub-matriz 3×3 inferior-direita (uma é a transposta da outra), que ilustra a não-comutatividade da multiplicação dos quatérnios.

Exemplo 2.1.5 *Sejam $a = 1 + 3i + 6j - 2k$ e $b = 4 + 2i - 3j + k$. Então, teremos as seguintes multiplicações:*

$$ab = \begin{bmatrix} 1 & -3 & -6 & 2 \\ 3 & 1 & 2 & 6 \\ 6 & -2 & 1 & -3 \\ -2 & -6 & 3 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 18 \\ 14 \\ 14 \\ -28 \end{bmatrix}$$

e

$$ab = \begin{bmatrix} 1 & -3 & -6 & 2 \\ 3 & 1 & -2 & -6 \\ 6 & 2 & 1 & 3 \\ -2 & 6 & -3 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ -3 \\ 1 \end{bmatrix} = \begin{bmatrix} 18 \\ 14 \\ 28 \\ 14 \end{bmatrix}.$$

Além disso, a soma dos quadrados dos elementos de cada coluna (ou linha) de A e $\hat{A}$ é

igual a

$$a_0^2 + a_x^2 + a_y^2 + a_z^2,$$

que é o produto escalar $a \cdot a$ (veja Definição 2.1.6).

As matrizes A e $\hat{A}$ são ortogonais (ortonormal, caso o quatérnio a possuir norma unitária, definida abaixo) e, portanto, o produto matricial $AA^T = (a \cdot a)I_4$ resulta numa matriz diagonal.

Definição 2.1.6 *O produto escalar de dois quatérnios é definido como a soma do produto dos componentes correspondentes:*

$$a \cdot b = a_0 b_0 + a_x b_x + a_y b_y + a_z b_z. \tag{2.1.2}$$

Definição 2.1.7 *O conjugado de um quatérnio, em alusão aos complexos, é definido como:*

$$a^* = a_0 - a_x i - a_y j - a_z k. \tag{2.1.3}$$

Observação 2.1.8 *A matriz em $\mathbb{R}^{4\times4}$ associada a uma multiplicação com um quatérnio conjugado é justamente a transposta da matriz associada a multiplicação do próprio quatérnio:*

$$a^* b = \begin{bmatrix} a_0 & a_x & a_y & a_z \\ -a_x & a_0 & a_z & -a_y \\ -a_y & -a_z & a_0 & a_x \\ -a_z & a_y & -a_x & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_x \\ b_y \\ b_z \end{bmatrix} = A^* b$$

e

$$ba = \begin{bmatrix} a_0 & a_x & a_y & a_z \\ -a_x & a_0 & -a_z & a_y \\ -a_y & a_z & a_0 & -a_x \\ -a_z & -a_y & a_x & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_x \\ b_y \\ b_z \end{bmatrix} = \hat{A}^* b.$$

Definição 2.1.9 *A norma de um quatérnio é definido como:*

$$\|a\| = \sqrt{a \cdot a} = \sqrt{aa^*} = \sqrt{a_0^2 + a_x^2 + a_y^2 + a_z^2}. \tag{2.1.4}$$

Observação 2.1.10 *Um quatérnio unitário é um quatérnio cuja norma é igual a 1.*

Como mencionado anteriormente, os quatérnios formam um anel não-comutativo com divisão. É um dos poucos exemplos de um anel não-comutativo que possui inverso multiplicativo.

Definição 2.1.11 *O inverso de um quatérnio não-nulo é definido por:*

$$a^{-1} = \frac{1}{a \cdot a^*} a^*.$$

Utilizando o produto dos quatérnios (2.1.1) e as definições de produto escalar (2.1.2), norma (2.1.4) e conjugado (2.1.3), as propriedades a seguir são válidas para todo $a, b, c \in \mathbb{H}$:

$$\begin{aligned}
(a^*)^* &= a. \\
(ab)^* &= b^* a^*. \\
(a+b)^* &= a^* + b^*. \\
\|a\| &= \|a^*\|. \\
\|ab\| &= \|a\|\|b\|. \\
(ab) \cdot (ac) &= (a \cdot a)(b \cdot c). \\
(ab) \cdot c &= a \cdot (cb^*).
\end{aligned} \tag{2.1.5}$$

Definição 2.1.12 *Um quatérnio $a \in \mathbb{H}$ é dito puramente imaginário quando $a_0 = 0$, isto é, quando $a = a_x i + a_y j + a_z k$.*

A seguir, apresentamos uma abordagem que considera um vetor tridimensional como sendo um quatérnio. Partindo da ideia que vetores tridimensionais podem ser representados por quatérnios puramente imaginários, seja $r = [r_x, r_y, r_z]^T$. Então o quatérnio correspondente é dado por:

$$r = r_x i + r_y j + r_z k.$$

Definição 2.1.13 *Uma rotação em um ângulo θ sobre um eixo representado por um vetor unitário $v = [v_x, v_y, v_z]^T$ pode ser representado por um quatérnio unitário*

$$q = \cos\left(\frac{\theta}{2}\right) + sen\left(\frac{\theta}{2}\right)(v_x i + v_y j + v_z k). \tag{2.1.6}$$

Agora seja $r = [r_x, r_y, r_z]^T$ um vetor em $\mathbb{R}^3$. A rotação do vetor r pelo ângulo θ sobre o eixo v é definido como

$$r' = qrq^*. \tag{2.1.7}$$

.

Exemplo 2.1.14 *A norma é preservada numa rotação por quatérnios. Em outras palavras, se $q \in \mathbb{H}$ é um vetor unitário que representa uma rotação e $r \in \mathbb{H}$ também é um vetor (isto é, um*

quatérnio puramente imaginário), então, sabendo que $\|q\| = \|q^*\| = 1$*, temos que:*

$$\|qrq^*\| = \|q\|\|r\|\|q^*\| = \|r\|.$$

Os conceitos estudados neste capítulo serão utilizados para o estudo do problema *Procrustes* no Capítulo 5, em particular sobre um método de sobreposição de estruturas que utiliza rotações.

3 CONCEITOS BÁSICOS DE OTIMIZAÇÃO

Nesse capítulo faremos uma complementação do estudo preliminar realizado até então, de modo que tenhamos, ao final deste, todos os pré-requisitos para trabalhar o Problema da Geometria de Distâncias sob a perspectiva clássica da Otimização Contínua. Neste estudo discutiremos os principais conceitos, resultados e métodos de Otimização necessários para situar o leitor no contexto da abordagem que faremos do PGD.

Por sua vez, as notações, definições e resultados discutidos neste Capítulo foram baseados em Ribeiro e Karas (2013), Luenberger e Ye (2015) e Nocedal e Wright (2006).

3.1 INTRODUÇÃO À OTIMIZAÇÃO

A Otimização é considerada uma área de pesquisa que possui alicerces na Matemática e na Computação. Seu objetivo principal é responder a pergunta "O que é melhor?", considerando problemas que podem ser modelados matematicamente com suas soluções dadas numericamente.

Antes de encontrar a solução do problema, é essencial moldá-lo matematicamente. Deste modo, dizemos que função objetivo é uma função que associa cada ponto do espaço de soluções a um número real.

Claramente à medida que o número de funções e variáveis necessárias para descrever um determinado problema aumentam, a dificuldade em determinar o conjunto de soluções deste problema acompanha esse aumento. Nesse contexto, surge a necessidade de desenvolver métodos matemáticos e computacionais que auxiliem o processo de Otimização.

A Otimização está muito atrelada a vários ramos que necessitam de tomada de decisões, seja em confiabilidade estrutural, economia, informática, medicina, biologia, logística, processos sísmicos, trasporte, entre outros. Quase sempre o objetivo é minimizar ou maximizar certa variável, como o custo ou o lucro de um determinado processo.

Mais formalmente, podemos dizer que a Otimização consiste em encontrar pontos de

mínimo e máximo de uma função real sobre um conjunto $\Omega \subset \mathbb{R}^n$. Isto pode ser expresso da seguinte forma:

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeito a } x \in \Omega, \end{aligned}$$

onde o conjunto Ω é definido por restrições de igualde e/ou desigualdade, ou seja

$$\Omega = \{x \in \mathbb{R}^n \mid c_\varepsilon(x) = 0,\ c_I(x) \leq 0\},$$

com $c_\varepsilon : \mathbb{R}^n \to \mathbb{R}^m$ e $c_I : \mathbb{R}^n \to \mathbb{R}^p$ funções quaisquer. Conforme as características do conjunto Ω e as propriedades da função objetivo, temos os diferentes problemas de Otimização.

Neste Capítulo estudaremos os problemas onde todas as funções utilizadas para definí-los são continuamente diferenciáveis e não-lineares, isto é, estudaremos o problema geral da *Programação Não-Linear* (PNL), em específico o problema irrestrito, quando $\Omega = \mathbb{R}^n$.

3.1.1 PROBLEMA DA OTIMIZAÇÃO

De forma geral, o problema de otimização é definido como

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeito a } x \in \Omega, \end{aligned}$$

onde $f : \mathbb{R}^n \to \mathbb{R}$ é uma função arbitrária e $\Omega \subset \mathbb{R}^n$ é um conjunto qualquer.

Definição 3.1.1 *Considere uma função $f : \mathbb{R}^n \to \mathbb{R}$ e $x^* \in \Omega \subset \mathbb{R}^n$. Dizemos que x^* é um minimizador local de f em Ω quando existe $\delta > 0$ tal que $f(x^*) \leq f(x)$ para todo $x \in B(x^*, \delta) \cap \Omega$, onde $B(x^*, \delta) = \{x \in \mathbb{R}^n \mid \|x - x^*\| < \delta\}$ é a bola aberta de centro x^* e raio δ. Caso $f(x^*) \leq f(x)$ para todo $x \in \Omega$, então x^* é dito minimizador global de f em Ω.*

Quando as desigualdades na Definição 3.1.1 forem estritas para $x \neq x^*$, dizemos que x^* é minimizador estrito de f. Se não for mencionado o conjunto Ω, significa que $\Omega = \mathbb{R}^n$ e, consequentemente, estamos trabalhando com um problema irrestrito.

Definição 3.1.2 *Considere uma função $f : \mathbb{R}^n \to \mathbb{R}$, um ponto $\bar{x} \in \mathbb{R}^n$ e uma direção $d \in \mathbb{R}^n \backslash \{0\}$. Dizemos que d é uma direção de descida para f, a partir de $\bar{x}$, quando existe $\delta > 0$ tal que $f(\bar{x} + td) < f(\bar{x})$, para todo $t \in (0, \delta)$.*

Uma classe de funções são as funções convexas, que possuem muitos resultados importantes.

Definição 3.1.3 *Uma função $f : S \subset \mathbb{R}^n \to \mathbb{R}$ é dita convexa se:*

- *Seu domínio $S \subset \mathbb{R}^n$ for convexo, isto é, se para quaisquer dois pontos $x, y \in S$ tivermos que $\alpha x + (1-\alpha)y \in S$ para todo $\alpha \in [0,1]$;*
- *Para quaisquer dois pontos $x, y \in S$ a seguinte propriedade for satisfeita:*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y), \text{ para todo } \alpha \in [0,1].$$

Teorema 3.1.4 *Quando $f : \mathbb{R}^n \to \mathbb{R}$ é convexa, qualquer minimizador local x^* é um minimizador global de f.*

Demonstração: Veja Nocedal e Wright (2006). ■

Para que as discussão sobre os tópicos de otimização fiquem da forma mais clara possível, definiremos e discutiremos sobre derivadas de funções de várias variáveis.

Definição 3.1.5 *Uma função $f : \mathbb{R}^n \to \mathbb{R}$ é dita de classe $\mathscr{C}$ (e denotamos $f \in \mathscr{C}$) se f é contínua, isto é, $\lim_{x \to a} f(x) = f(a)$. Se além disso f possuir as primeiras derivadas parciais contínuas, então f é de classe $\mathscr{C}^1$ (denotamos $f \in \mathscr{C}^1$). Genericamente, se f possuir as derivadas parciais de ordem p contínuas, então f é dita de classe $\mathscr{C}^p$ (e escrevemos $f \in \mathscr{C}^p$).*

Definição 3.1.6 *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathscr{C}^1$. O gradiente da função f é $\nabla f(x) \in \mathbb{R}^n$:*

$$\nabla f(x) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{bmatrix}.$$

Definição 3.1.7 *Seja $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe $\mathscr{C}^2$. A Hessiana da função f é $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$:*

$$\nabla^2 f(x) = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \cdots & \dfrac{\partial^2 f}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n x_1} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}.$$

Definição 3.1.8 *Considere uma função vetorial $F : \mathbb{R}^n \to \mathbb{R}^m$. Sua derivada, chamada de Jacobiana, é a matriz*

$$J_F = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}.$$

Note que a i-ésima linha da Jacobiana de F é o gradiente transposto da componente F_i. Em particular, para $m = 1$, temos $J_F = (\nabla F(x))^T$.

Teorema 3.1.9 *Se $\nabla f(\bar{x})^T d < 0$, então d é uma direção de descida para f, a partir de $\bar{x}$.*

Demonstração: Sabemos que

$$\nabla f(\bar{x})^T d = \frac{\partial f}{\partial d}(\bar{x}) = \lim_{t\to} \frac{f(\bar{x}+td) - f(\bar{x})}{t}.$$

Pela hipótese e pela preservação de sinal, existe $\delta > 0$ tal que

$$\frac{f(\bar{x}+td) - f(\bar{x})}{t} < 0,$$

para todo $t \in (-\delta, \delta)$, $t \neq 0$. Portanto, $f(\bar{x}+td) < f(\bar{x})$, para todo $t \in (0, \delta)$, o que completa a demonstração. ■

3.1.2 FÓRMULA DE TAYLOR

As aproximações de Taylor para uma função constituem uma das mais clássicas ferramentas em otimização, tanto no desenvolvimento da teoria quanto na construção dos algoritmos. Aparecem, por exemplo, nas demonstrações das condições de otimalidade de primeira e segunda ordem, que serão expostas no decorrer do trabalho, bem como na ideia do Método de Newton.

Trabalharemos aqui com aproximações de primeira e segunda ordem. As de ordem superior, apesar de serem mais precisas, deixam de ser convenientes pelo alto custo computacional para o cálculo das derivadas.

Uma relação importante das derivadas surge quando restringimos uma função definida em $\mathbb{R}^n$ aos pontos de um segmento de reta. Formalmente, dados $\bar{x}, d \in \mathbb{R}^n$ e $f : \mathbb{R}^n \to \mathbb{R}$, definimos $\varphi : I \subset \mathbb{R} \to \mathbb{R}$ por $\varphi(t) = f(\bar{x}+td)$. Vamos calcular as derivadas de φ, temos:

$$\varphi'(t) = \lim_{s\to 0} \frac{\varphi(t+s) - \varphi(t)}{s} = \frac{\partial f}{\partial d}(\bar{x}+td) = \nabla f(\bar{x}+td)^T d.$$

Para calcular φ'', note que $\sum_{j=1}^{n} d_j \frac{\partial f}{\partial x_j}(\bar{x}+td)$. Assim, lembrando que $\nabla^2 f(x)$ é simétrica, temos

$$\varphi''(t) = \sum_{j=1}^{n} d_j \nabla \frac{\partial f}{\partial x_j}(\bar{x}+td)^T d = d^T \nabla^2 f(\bar{x}+td)d.$$

Finalmente, vamos apresentar as Fórmulas de Taylor. As demonstrações podem ser encontradas em Lima (2013).

Teorema 3.1.10 *(Taylor de Primeira Ordem): Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função diferenciável e $\bar{x} \in \mathbb{R}^n$. Então, podemos escrever*

$$f(x) = f(\bar{x}) + \nabla f(x)^T (x-\bar{x}) + r(x),$$

com $\lim_{x\to\bar{x}} \frac{r(x)}{\|x-\bar{x}\|} = 0.$

O polinômio $p_1(x) = f(\bar{x}) + \nabla f(x)^T (x-\bar{x})$ é uma aproximação linear para f em torno do ponto $\bar{x}$ e é chamado polinômio de Taylor de ordem 1 da função. Dentre todos os polinômios de grau menor ou igual a 1, ele é o único que satisfaz

$$p(\bar{x}) = f(\bar{x}) \text{ e } \nabla p(\bar{x}) = \nabla f(\bar{x}).$$

O limite nulo no Teorema 3.1.10 significa que para x próximo de $\bar{x}$ o resto $r(x)$ é muito pequeno e vai para zero mais rápido que $\|x-\bar{x}\|$.

Agora, descreveremos uma função quadrática que aproxima uma dada função $f(x)$ na vizinhança de um ponto.

Teorema 3.1.11 *(Taylor de Segunda Ordem): Se $f : \mathbb{R}^n \to \mathbb{R}$ uma função duas vezes diferenciável e $\bar{x} \in \mathbb{R}^n$, então,*

$$f(x) = f(\bar{x}) + \nabla f(x)^T (x-\bar{x}) + \frac{1}{2}(x-\bar{x})^T \nabla^2 f(\bar{x})(x-\bar{x}) + r(x),$$

com $\lim_{x\to\bar{x}} \frac{r(x)}{\|x-\bar{x}\|^2} = 0.$

De forma análoga, o polinômio

$$p_2(x) = f(\bar{x}) + \nabla f(x)^T (x-\bar{x}) + \frac{1}{2}(x-\bar{x})^T \nabla^2 f(\bar{x})(x-\bar{x}),$$

é uma aproximação quadrática para f em torno do ponto $\bar{x}$ e é chamado de polinômio de Taylor de ordem 2 da função. Dentre todos os polinômios de grau menor ou igual a 2, ele é o único

que satisfaz

$$p(\bar{x}) = f(\bar{x}),\ \nabla p(\bar{x}) = \nabla f(\bar{x}) \text{ e } \nabla^2 p(\bar{x}) = \nabla^2 f(\bar{x}).$$

O limite nulo no Teorema 3.1.11 significa que para x próximo de $\bar{x}$, o resto $r(x)$ é muito pequeno e vai para zero muito mais rápido que $\|x-\bar{x}\|^2$.

Veremos a seguir outra fórmulação do polinômio de Taylor, na qual não supomos $d \to 0$ para estimar a diferença $f(\bar{x}+d) - f(\bar{x})$. Para a ordem 1, ele é exatamente o Teorema do Valor Médio. De modo geral, o chamamos de Taylor com resto de Lagrange. As demonstrações dos três seguintes Teoremas também podem ser encontradas em Lima (2013).

Teorema 3.1.12 *(Teorema do Valor Médio): Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função contínua e $\bar{x}, d \in \mathbb{R}^n$. Se f é diferenciável no segmento $(\bar{x}, \bar{x}+d)$, então existe $t \in (0,1)$ tal que*

$$f(\bar{x}+d) = f(\bar{x}) + \nabla f(\bar{x}+td)^T d.$$

Teorema 3.1.13 *(Taylor com Resto de Lagrange): Considere $f : \mathbb{R}^n \to \mathbb{R}$ uma função de classe C^1 e $\bar{x}, d \in \mathbb{R}^n$. Se f é duas vezes diferenciável no segmento $(\bar{x}, \bar{x}+d)$, então existe $t \in (0,1)$ tal que*

$$f(\bar{x}+d) = f(\bar{x}) + \nabla f(\bar{x}+td)^T d + \frac{1}{2} d^T \nabla^2 f(\bar{x}+td) d.$$

Para funções $f : \mathbb{R}^n \to \mathbb{R}^m$, com $m > 1$, não podemos garantir uma igualdade, como no Teorema 3.1.12. No entanto, ainda podemos obter informações sobre a variância da função a partir de uma estimativa da limitação da sua derivada. Isto é formalizado no seguinte Teorema.

Teorema 3.1.14 *(Desigualdade do Valor Médio): Considere $f : \mathbb{R}^n \to \mathbb{R}^m$ contínua e $\bar{x}, d \in \mathbb{R}^n$. Se f é diferenciável no segmento $(\bar{x}, \bar{x}+d)$ e $\|J_F\| \leq M$, para todo $x \in (\bar{x}, \bar{x}+d)$, então*

$$\|f(\bar{x}+d) - f(\bar{x})\| \leq M\|d\|.$$

3.2 CONDIÇÕES DE OTIMALIDADE

Vejamos agora as condições necessárias e suficientes para caracterizar um minimizador de um problema irrestrito.

Teorema 3.2.1 *(Condição Necessária de 1^a Ordem): Seja $f : \mathbb{R}^n \to \mathbb{R}$ diferenciável no ponto $x^* \in \mathbb{R}^n$. Se x^* é um minimizador local de f, então*

$$\nabla f(x^*) = 0. \tag{3.2.1}$$

Demonstração: Considere $d \in \mathbb{R}^n \backslash \{0\}$ arbitrário. Como x^* é minimizador local, então existe $\delta > 0$ tal que

$$f(x^*) \leq f(x^* + td), \tag{3.2.2}$$

para todo $t \in (0, \delta)$. Pela expansão de Taylor,

$$f(x^* + td) = f(x^*) + t\nabla f(x^*)^T d + r(t),$$

com $\lim_{t \to 0} \frac{r(t)}{t} = 0$. Usando (3.2.2) e dividindo por t, obtemos $0 \leq \nabla f(x^*)^T d + \frac{r(t)}{t}$. Passando o limite quando $t \to 0$, tiramos que $\nabla f(x^*)^T d \geq 0$. Se $\nabla f(x^*)$ não fosse nulo, poderíamos escolher $d = -\nabla f(x^*)$, resultando em

$$\|\nabla f(x^*)\|^2 = -\nabla f(x^*) d \leq 0,$$

o que é uma contradição. Logo, $\nabla f(x^*) = 0$. ■

Definição 3.2.2 *Um ponto $x^* \in \mathbb{R}^n$ que cumpre a condição* (3.2.1) *é dito ponto crítico ou estacionário da função f.*

Teorema 3.2.3 *(Condição Necessária de 2^a Ordem): Seja $f : \mathbb{R}^n \to \mathbb{R}$ duas vezes diferenciável no ponto $x^* \in \mathbb{R}^n$. Se x^* é um minimizador local de f, então a matriz Hessiana de f ponto x^* é semi-definida positiva, isto é,*

$$d^T \nabla^2 f(x^*) d \geq 0,$$

para todo $d \in \mathbb{R}^n$.

Demonstração: Considere $d \in \mathbb{R}^n \backslash \{0\}$. Por Taylor,

$$f(x^* + td) = f(x^* + t\nabla f(x^*)^T d + \frac{t^2}{2} d^T \nabla^2 f(x^*) d + r(t),$$

com $\lim_{t \to 0} \frac{r(t)}{t} = 0$. Como x^* é minimizador local, o Teorema 3.2.1 garante que $\nabla f(x^*) = 0$. Portanto, para t suficientemente pequeno,

$$0 \leq f(x^* + td) - f(x^*) = \frac{t^2}{2} d^T \nabla^2 f(x^*) d + r(t).$$

Dividindo por t^2 e passando o limite quando $t \to 0$, obtemos $d^T \nabla^2 f(x^*) d \geq 0$. ■

Teorema 3.2.4 *(Condição Suficiente de 2ª Ordem): Seja $f:\mathbb{R}^n\to\mathbb{R}$ duas vezes diferenciável no ponto $x^*\in\mathbb{R}^n$. Se x^* é um ponto estacionário da função f e $\nabla^2 f(x^*)$ é definida positiva, então x^* é um minimizador local estrito de f.*

Demonstração: Seja λ o menor autovalor de $\nabla^2 f(x^*)$. Como esta matriz é definida positiva, temos que $\lambda>0$. Além disso, pelo Lema A.2.10, $d^T\nabla^2 f(x^*)d\geq\lambda\|d\|^2$, para todo $d\in\mathbb{R}^n$. Por Taylor, já utilizando o fato de x^* ser um ponto estacionário, temos:

$$f(x^*+d)=f(x^*)+\frac{1}{2}d^T\nabla^2 f(x^*)d+r(d)\geq f(x^*)+\frac{1}{2}\lambda\|d\|^2+r(d),$$

onde $\lim_{d\to 0}\dfrac{r(d)}{\|d\|^2}=0$. Podemos escrever então

$$\frac{f(x^*+d)-f(x^*)}{\|d\|^2}\geq\frac{\lambda}{2}+\frac{r(d)}{\|d\|^2}.$$

Como $\lim_{d\to 0}\left[\dfrac{\lambda}{2}+\dfrac{r(d)}{\|d\|^2}\right]>0$, então existe $\delta>0$ tal que $\dfrac{\lambda}{2}+\dfrac{r(d)}{\|d\|^2}>0$, para todo $d\in B(0,\delta)\backslash\{0\}$, donde segue que $f(x^*+d)-f(x^*)>0$, para todo $d\in B(0,\delta)\backslash\{0\}$, ou equivalentemente,

$$f(x^*)<f(x^*),$$

para todo $x\in B(x^*,\delta)\backslash\{x^*\}$. ■

Para completar o Teorema 3.1.4 temos o seguinte resultado, cuja demonstração, também, se encontra em Nocedal e Wright (2006):

Teorema 3.2.5 *Quando $f:\mathbb{R}^n\to\mathbb{R}$ é convexa, qualquer ponto estacionário x^* é um minimizador global de f.*

Definição 3.2.6 *Um algoritmo é dito globalmente convergente quando para qualquer sequência de pontos (x^k) gerada pelo algoritmo e qualquer ponto de acumulação $\bar{x}$ de (x^k), temos que $\bar{x}$ é estacionário.*

3.3 PROBLEMAS DE MÍNIMOS-QUADRADOS

Em problemas de mínimos-quadrados, a função objetivo f possui a seguinte forma especial:

$$f(x)=\frac{1}{2}\sum_{j=1}^{m}r_j^2(x), \tag{3.3.1}$$

onde cada r_j é uma função suave (de classe C^n para todo n) de $\mathbb{R}^n$ em $\mathbb{R}$. Nos referimos a cada r_j como sendo um resíduo e assumimos $m \geq n$.

Os problemas de mínimos-quadrados possuem aplicação em diversas áreas de pesquisa e, possivelmente, é a maior fonte de problemas de otimização irrestrita. Para ver o porquê essa forma especial da f faz com que os problemas de mínimos-quadrados sejam mais simplificadas de se resolver do que os problemas gerais de minimização irrestrita, primeiro montamos os componentes individuais de (3.3.1) em um vetor residual $r : \mathbb{R}^n \to \mathbb{R}^m$, como a seguir

$$r(x) = [r_1(x), r_2(x), \ldots, r_m(x)]^T. \tag{3.3.2}$$

Podemos reescrever f como $f(x) = \frac{1}{2}\|r(x)\|_2^2$. As derivadas de $f(x)$ podem ser expressadas em termos da Jacobiana $J(x)$, a qual é uma matriz em $\mathbb{R}^{m\times n}$ das primeiras derivadas parciais dos resíduos, a qual iremos denotar agora como

$$J(x) = \begin{bmatrix} \nabla r_1(x)^T \\ \nabla r_2(x)^T \\ \vdots \\ \nabla r_m(x)^T \end{bmatrix}, \tag{3.3.3}$$

onde cada $\nabla r_j(x)$, $j = 1, 2, \ldots, m$, é o vetor gradiente de r_j. O gradiente e a Hessiana de f são dadas por:

$$\nabla f(x) = \sum_{j=1}^{m} r_j(x)\nabla r_j(x) = J(x)^T r(x), \tag{3.3.4}$$

$$\begin{aligned} \nabla^2 f(x) &= \sum_{j=1}^{m} \nabla r_j(x)\nabla r_j(x)^T + \sum_{j=1}^{m} r_j(x)\nabla^2 r_j(x) \\ &= J(x)^T J(x) + \sum_{j=1}^{m} r_j(x)\nabla^2 r_j(x). \end{aligned} \tag{3.3.5}$$

Em diversas aplicações, as primeiras derivadas parciais dos resíduos e, consequentemente a matriz Jacobiana, são relativamente fáceis e baratas de se calcular. Podemos assim obter o gradiente $\nabla f(x)$ como é escrito na fórmula (3.3.4). Utilizando $J(x)$, podemos também calcular o primeiro termo $J(x)^T J(x)$ da Hessiana $\nabla^2 f(x)$ sem envolver nenhuma derivada de segunda ordem das funções r_j. Esta disponibilidade de uma parte de $\nabla^2 f(x)$ ser "de graça" que é a característica distintiva dos problemas de mínimos-quadrados. Além disso, este termo $J(x)^T J(x)$ é frequentemente mais importante que o segundo termo somatório em (3.3.5), quer porque os resíduos r_j são quase afins perto da solução (isto é, o valor de $\nabla^2 r_j(x)$ é relativamente pequeno) quer por conta de pequenos resíduos (isto é, os $r_j(x)$ são relativamente pequenos).

3.3.1 PROBLEMAS DE MÍNIMOS-QUADRADOS LINEARES

Em aplicações de ajustes de dados, frequentemente são utilizados modelos $\Phi(x;t)$ que são funções lineares de x. Nesses casos, os resíduos $r_j(x)$ definidos por $r_j = \Phi(x;t) - y_j$, que representam as discrepâncias entre os dados observados e preditos para cada t_j, também são lineares, e o problema de minimizar

$$f(x) = \frac{1}{2}\sum_{j=1}\left[\Phi(x;t) - y_j\right]^2$$

é chamado de problema de mínimos-quadrados lineares. Podemos escrever o vetor residual como $r(x) = Jx - y$ para alguma matriz J e algum vetor y, ambos independentes de x, tal que a função objetivo se torna

$$f(x) = \frac{1}{2}\|Jx - y\|^2, \tag{3.3.6}$$

onde $y = r(0)$. Também teremos que

$$\nabla f(x) = J^T(Jx - y) \text{ e } \nabla^2 f(x) = J^T J.$$

Observação 3.3.1 *Note que o segundo termo em* $\nabla^2 f(x)$ *(veja* (3.3.5)*) desaparece, pois* $\nabla r_j = 0$*, para todo* $j = 1, 2, \ldots, m$*.*

É fácil ver que $f(x)$ em (3.3.6) é convexa, uma propriedade que não é necessariamente assegurada para o problema não-linear (3.3.1). O Teorema 3.2.5 nos assegura que qualquer ponto que satisfaz $\nabla f(x) = 0$ é minimizador global de f. Pontanto, x^* deve satisfazer o seguinte sistema de equações lineares:

$$J^T J x^* = J^T y. \tag{3.3.7}$$

Essas são conhecidas como as equações normais de (3.3.6). Existem diversas formas de resolver esse sistema linear de equações, tais como: fatoração Cholesky, fatoração QR, decomposição SVD e fatoração LU. Cada abordagem tem sua vantagem e desvantagem, as quais podem ser estudadas com mais profundidade em Nocedal e Wright (2006).

3.4 MÉTODOS DE OTIMIZAÇÃO IRRESTRITA

Vamos considerar o problema irrestrito

$$\begin{aligned} &\text{minimizar } f(x) \\ &\text{sujeito a } x \in \mathbb{R}^n. \end{aligned} \tag{3.4.1}$$

Estudaremos a seguir alguns métodos específicos de minimização do problema (3.4.1). Supõe-se que f é de classe $\mathscr{C}^2$.

3.4.1 MINIMIZAÇÃO UNIDIMENSIONAL

Dada uma função $f : \mathbb{R}^n \to \mathbb{R}$, um ponto $\bar{x} \in \mathbb{R}^n$ e uma direção de descida $d \in \mathbb{R}^n$, queremos encontrar $\bar{t} > 0$ tal que

$$f(\bar{x} + \bar{t}d) < f(\bar{x}).$$

Vejamos duas abordagens para este problema. A primeira consiste em fazer uma busca exata a partir do ponto $\bar{x}$ segundo a direção d. A segunda procura uma redução suficiente de f que seja de certo modo proporcional ao tamanho do passo.

3.4.1.1 BUSCA EXATA - MÉTODO DA SEÇÃO ÁUREA

Nosso objetivo consiste em minimizar f a partir do ponto $\bar{x}$ na direção d. Em outras palavras, temos que resolver o problema

$$\begin{array}{c} \text{minimizar} \quad f(\bar{x} + \bar{t}d) \\ \text{sujeito a} \quad t > 0. \end{array} \tag{3.4.2}$$

Este problema, em geral, é difícil de se resolver. Entretanto, para certas funções especiais, existem algoritmos para resolvê-lo. Veremos adiante tais funções, bem como um algoritmo. Antes, porém, vamos fazer um exemplo que pode ser resolvido de forma direta.

Exemplo 3.4.1 *Considere $f : \mathbb{R}^2 \to \mathbb{R}$ dada por*

$$f(x) = \frac{1}{2}(x_1 - 4)^2 + (x_2 - 2)^2,$$

e também $\bar{x} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ e $d = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$. Vamos fazer, então, a busca exata a patir de $\bar{x}$, na direção d.

Note primeiro que d é de fato uma direção de descida, pois

$$\nabla f(\bar{x})^T d = [-2 \ -2] \begin{bmatrix} 1 \\ 3 \end{bmatrix} = -8 < 0.$$

Para fazer a busca, considere

$$\varphi(t) = f(\bar{x} + td) = f(2 + t, 1 + 3t) = \frac{19t^2}{2} - 8t + 3,$$

cujo minimizador satisfaz $\varphi'(t) = 19t - 8 = 0$. *Assim,*

$$\bar{t} = \frac{8}{19} \; e \; \bar{x} + \bar{t}d = \begin{bmatrix} 2 \\ 1 \end{bmatrix} + \frac{8}{19} \begin{bmatrix} 1 \\ 3 \end{bmatrix} \approx \begin{bmatrix} 2,42 \\ 2,26 \end{bmatrix}.$$

No entanto, de forma geral, os problemas são complexos e requerem o auxílio de métodos numéricos para a obtenção da solução. Vamos definir a seguir uma função unimodal, para a qual existem métodos para minimizá-la.

Em seguida veremos o método da Seção Áurea, que encontra o ponto próximo de um minimizador com a precisão que se queira. Este algoritmo será aplicado para a função $\varphi : [0,\infty) \to \mathbb{R}$ definida por

$$\varphi(t) = f(\bar{x} + td).$$

Definição 3.4.2 *Uma função contínua* $\varphi : [0,\infty) \to \mathbb{R}$ *é dita unimodal quando admite um conjunto de minimizadores* $[t_1, t_2]$, *é estritamente crescente em* $[0, t_1]$ *e estritamente decrescente em* $[t_2, \infty)$.

Na Figura 1 temos duas funções unimodais. Na segunda, o intervalo de minimizadores é degenerado.

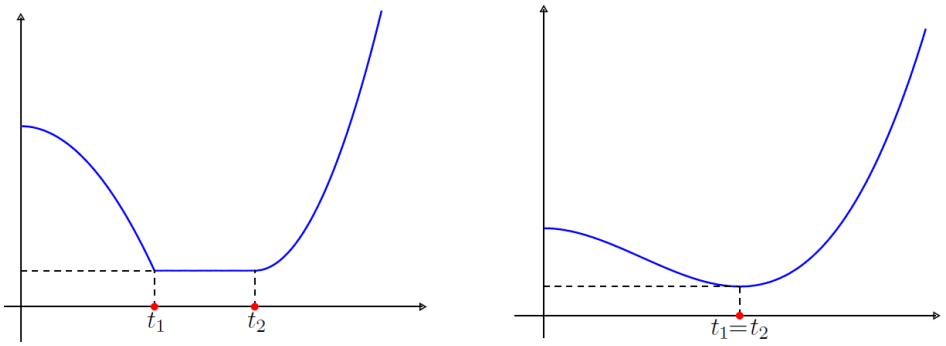


Figura 1: Exemplo de funções unimodais.

Fonte: (RIBEIRO; KARAS, 2013)

Para facilitar a descrição do método, vamos considerar a Figura 2. Suponha que um minimizador pertence ao intervalo $[a,b]$.

- Considere $a < u < v < b$ em $[0,\infty]$;
- Se $\varphi(u) < \varphi(v)$, então o trecho $[v,b]$ não pode conter um minimizador e pode ser descartado;

- Se $\varphi(u) \geq \varphi(v)$, então o trecho $[a,u]$ pode ser descartado;
- Particione o intervalo que sobrou e repita o processo.

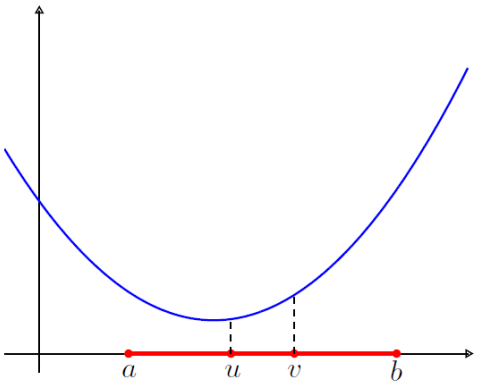


Figura 2: Método de Particionamento.
Fonte: (RIBEIRO; KARAS, 2013)

Vamos discutir agora como particionar o intervalo $[a,b]$. A obtenção deste intervalo, que deve conter um minimizador de φ, será tratada a seguir.

Uma estratégia natural é dividir o intervalo em três partes iguais, ou seja, definir

$$u = a + \frac{1}{3}(b-a) \text{ e } v = a + \frac{2}{3}(b-a).$$

Assim, descartamos $\frac{1}{3}$ do intervalo corrente a cada etapa. Entretanto, esta forma de particionar o intervalo tem uma desvantagem. Precisamos fazer duas novas avaliações de função por etapa, pois o ponto que sobrou, u ou v, não pode ser reaproveitado (veja a Figura 3).

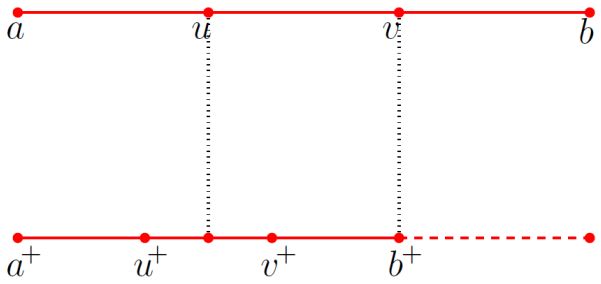


Figura 3: Partição do intervalo $[a,b]$ em três partes iguais.
Fonte: (RIBEIRO; KARAS, 2013)

Uma estratégia mais eficiente consiste em escolher os pontos u e v que dividem o segmento $[a,b]$ na razão áurea, de acordo com a Definição 3.4.3.

Definição 3.4.3 *Um ponto c divide o segmento $[a,b]$ na razão áurea quando a razão entre o maior dos segmentos e o segmento todo é igual à razão entre o menor e o maior dos segmentos. Tal razão é conhecida como o número de ouro e vale* $\dfrac{\sqrt{5}-1}{2} \approx 0,618$.

Desta forma, u e v devem satisfazer

$$\frac{b-u}{b-a} = \frac{u-a}{b-u} \text{ e } \frac{v-a}{b-a} = \frac{b-v}{v-a}.$$

Considerando θ_1 e θ_2 tais que,

$$u = a+\theta_1(b-a) \text{ e } v = a+\theta_2(b-a), \tag{3.4.3}$$

obtemos $1-\theta_1 = \dfrac{\theta_1}{1-\theta_1}$ e $\theta_2 = \dfrac{1-\theta_2}{\theta_2}$. Portanto,

$$\theta_1 = \frac{3-\sqrt{5}}{2} \approx 0,382 \text{ e } \theta_2 = \frac{\sqrt{5}-1}{2} \approx 0,618.$$

Note que

$$\theta_1+\theta_2 = 1 \text{ e } \theta_2^2 = \theta_1. \tag{3.4.4}$$

Uma das vantagens da divisão na razão áurea em relação à divisão em três partes iguais é que descartamos mais de 38% do intervalo ao invés de $33,33\%$. Outra vantagem, se refere a economia em avaliação de função.

No processo iterativo, a cada etapa descartamos o intervalo $[a,u]$ ou $[v,b]$, obtendo um novo segmento que deverá ser particionado novamente. Indicamos por $[a^+,b^+]$ o novo intervalo que será particionado pelos pontos u^+ e v^+.

Conforme veremos no próximo resultado, o ponto u é aproveitado na próxima etapa e passa a ser v^+ quando descartamos $[v,b]$. Assim, o valor da função $\varphi(u)$ é aproveitado para a próxima etapa.

Proposição 3.4.4 *No método da seção áurea, se $[v,b]$ é descartado então $v^+ = u$. Analogamente, se $[a,u]$ é descartado então $u^+ = v$.*

Demonstração: Como $[v,b]$ foi descartado, temos que $b^+ = v$ e $a^+ = a$. Portanto, usando (3.4.3), temos que

$$v^+ = a^+ + \theta_2(b^+ - a^+) = a+\theta_2(v-a).$$

Usando (3.4.3) e a relação (3.4.4), obtemos

$$v^+ = a + \theta_2^2(b-a) = a + \theta_1(b-a) = u,$$

provando a primeira afirmação. A outra afirmação se prova de forma análoga. ■

A Figura 4 ilustra essa propriedade.

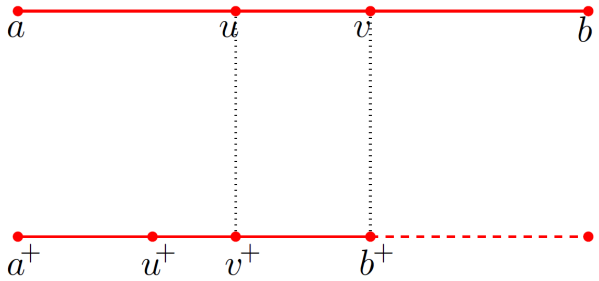


Figura 4: Partição do intervalo $[a,b]$ por Seção Áurea.
Fonte: (RIBEIRO; KARAS, 2013)

No Algoritmo 1 apresentamos o método da Seção Áurea, que tem duas fases. Na primeira, obtemos um intervalo $[a,b]$ que contém um minimizador de Φ. A ideia desta etapa é considerar um intervalo inicial $[0,2\rho]$, com $\rho > 0$, e ampliá-lo, deslocando para a direita, até que um crescimento de φ seja detectado. Na segunda fase, o intervalo $[a,b]$ é reduzido, por meio do descarte de subintervalos, até que reste um intervalo de tamanho suficiente para que uma precisão ε seja alcançada.

Algoritmo 1 ALGORITMO DA SEÇÃO ÁUREA.

1: **Dados de entrada** ($\varepsilon > 0, \rho > 0, \theta_1 = \frac{3-\sqrt{5}}{2}, \theta_2 = 1-\theta_1$).
2: **Dados de saída**($\bar{t}$: tamanho do passo por Seção Áurea).
3: Faça $a = 0, s = \rho, b = 2\rho$.
4: **Enquanto** $\varphi(b) < \varphi(s)$, faça
5: Faça $a = s, s = b, b = 2b$.
6: **Fim Enquanto**.
7: **Enquanto** $|b-a| > \varepsilon$, faça
8: **Se** $\varphi(u) < \varphi(v)$, **então**
9: Faça $b = v, v = u, u = a + \theta_1(b-a)$.
10: **Senão**
11: Faça $a = u, u = v, v = a + \theta_2(b-a)$.
12: **Fim Se**.
13: **Fim Enquanto**.
14: Faça $\bar{t} = \frac{u+v}{2}$.

Caso φ seja uma função unimodal, o Algoritmo 1 funciona perfeitamente e encontra uma aproximação para um minimizador dentro de uma tolerância dada. Caso a função não seja unimodal, o algoritmo pode não ser eficaz.

3.4.1.2 BUSCA INEXATA - CONDIÇÃO DE ARMIJO

Em muitas situações não convém aplicar a busca exata, ou porque φ não é unimodal, ou pelo alto custo computacional de se fazer uma busca exata a cada iteração. O método de Armijo procura uma boa redução da função ao longo da direção, sem tentar minimizá-la.

Considere, então, um ponto $\bar{x} \in \mathbb{R}^n$, uma direção de descida $d \in \mathbb{R}^n$ e $\eta \in (0,1)$. Basicamente, a regra de Armijo encontra $\bar{t} > 0$ tal que

$$f(\bar{x} + \bar{t}d) \leq f(\bar{x}) + \eta \bar{t} \nabla f(\bar{x})^T d. \tag{3.4.5}$$

A condição (3.4.5) significa que queremos mais que uma simples redução em f. Esta redução deve ser proporcional ao tamanho do passo. O próximo resultado garante que isto pode ser de fato obtido.

Teorema 3.4.5 *Considere uma função diferenciável $f : \mathbb{R}^n \to \mathbb{R}$, um ponto $\bar{x} \in \mathbb{R}^n$, uma direção*

de descida $d \in \mathbb{R}^n$ *e* $\eta \in (0,1)$. *Então existe* $\delta > 0$ *tal que*

$$f(\bar{x}+td) \leq f(\bar{x}) + \eta t \nabla f(\bar{x})^T d,$$

para todo $t \in [0,\delta)$.

Demonstração: Caso $\nabla f(\bar{x})^T d = 0$, o resultado segue da definição da direção de descida. Suponha então que $\nabla f(\bar{x})^T d < 0$. Assim, como $\eta < 1$, temos

$$\lim_{t\to 0} \frac{f(\bar{x}+td) - f(\bar{x})}{t} = \nabla f(\bar{x})^T d < \eta \nabla f(\bar{x})^T d.$$

Portanto, existe $\delta > 0$ tal que

$$\frac{f(\bar{x}+td) - f(\bar{x})}{t} < \eta \nabla f(\bar{x})^T d,$$

para todo $t \in (0,\delta)$. Isto implica que

$$f(\bar{x}+td) \leq f(\bar{x}) + \eta t \nabla f(\bar{x})^T d,$$

para todo $t \in [0,\delta)$, o que completa a demonstração. ■

A condição de Armijo pode ser interpretada de uma forma interessante. Considere a função $\varphi : [0,\infty) \to \mathbb{R}$ dada por

$$\varphi(t) = f(\bar{x}+td).$$

A aproximação de primeira ordem de φ em torno de $t = 0$, também chamada de modelo linear, é

$$p(t) = \varphi(0) + t\varphi'(0) = f(\bar{x}) + t\nabla f(\bar{x})^T d.$$

Assim, podemos reescrever a relação (3.4.5) como

$$\varphi(0) - \varphi(\bar{t}) = f(\bar{x}) - f(\bar{x}+\bar{t}d) \geq \eta \left(p(0) - p(\bar{t})\right).$$

Isto significa que procuramos um passo cuja redução na função objetivo seja pelo menos uma fração η da redução obtida no modelo linear. Veja a ilustração na Figura 5. Note também, nesta figura a reta dada por

$$q(t) = f(\bar{x}) + \eta t \nabla f(\bar{x})^T d.$$

A condição de Armijo é satisfeita para os pontos tais que φ está abaixo de q.

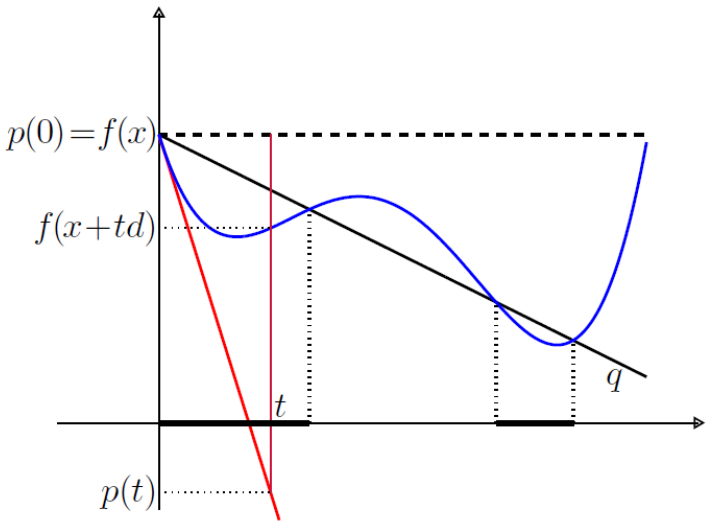


Figura 5: Interpretação Geométrica da Condição de Armijo.

Fonte: (RIBEIRO; KARAS, 2013)

Tanto do ponto de vista computacional quanto teórico, é importante que o tamanho do passo $\bar{t}$, satisfazendo (3.4.5), não seja muito pequeno. Uma maneira de garantir tal propriedade consiste em iniciar com $t = 1$ e, se necessário, reduzir t até que (3.4.5) seja satisfeita. Sintetizamos essa ideia no Algoritmo 2.

Algoritmo 2 MÉTODO DA BUSCA DE ARMIJO.

1: **Dados de entrada** ($\bar{x} \in \mathbb{R}^n, d \in \mathbb{R}^n, \gamma \in (0,1), \eta \in (0,1)$).
2: **Dados de saída**($\bar{t}$: tamanho do passo por Armijo).
3: Faça $t = 1$.
4: **Enquanto** $f(\bar{x}+td) > f(\bar{x}) + \eta t \, \nabla f(\bar{x})^T d$, faça
5: Faça $t = \gamma t$.
6: **Fim Enquanto**.

O método de Armijo não encontra um ponto próximo a um minimizador unidirecional, mas é muito eficiente. Para algoritmos bem projetados, faz um número muito pequeno de cálculos de função, sendo portanto muito rápido.

3.4.2 MÉTODO DO GRADIENTE

Uma das estratégias mais conhecidas de minimização irrestrita é o método clássico do gradiente, também chamado de Cauchy ou Método da Máxima Descida. Este método consiste em um processo iterativo que a cada etapa faz uma busca na direção oposta ao vetor gradiente da

função objetivo no ponto corrente. A justificativa desta escolha se baseia no fato de que, dentre as direções ao longo das quais f decresce, a direção oposta ao gradiente é a de decréscimo mais acentuado. De fato, se $d = -\nabla f(x)$ e $v \in \mathbb{R}^n$ é tal que $\|v\| = \|d\|$, então

$$\frac{\partial f}{\partial d}(x) = \nabla f(x)^T d = -\|\nabla f(x)\|^2 = -\|\nabla f(x)\| \, \|v\| \leq \nabla f(x)^T v = \frac{\partial f}{\partial v}(x).$$

No Algoritmo 3, deixamos em aberto a determinação do tamanho do passo. Dentre as diversas formas de busca existentes, podemos utilizar a busca exata (Algoritmo 1) ou inexata (Algoritmo 2).

Algoritmo 3 MÉTODO DO GRADIENTE.

1: **Dados de entrada**($x^0 \in \mathbb{R}^n$).
2: **Dados de saída**(x^*: mínimo aproximado).
3: Faça $k = 0$.
4: **Enquanto** $\nabla f(x^k) \neq 0$, faça
5: Defina $d^k = -\nabla f(x^k)$.
6: Obtenha $t_k > 0$: definindo $t_k = 1$, pelo Algoritmo 1 ou pelo Algoritmo 2.
7: Faça $x^{k+1} = x^k + t_k d^k$.
8: Faça $k = k + 1$.
9: **Fim Enquanto**.

A Figura 6 ilustra 4 iterações do Algoritmo 3 com a busca exata aplicado para minimizar uma função quadrática convexa. Esta figura sugere duas propriedades do algoritmo. Uma delas, formalizada no Lema 3.4.6, é o fato de duas direções consecutivas serem ortogonais. A outra propriedade se refere à convergência, descrita no Teorema 3.4.7.

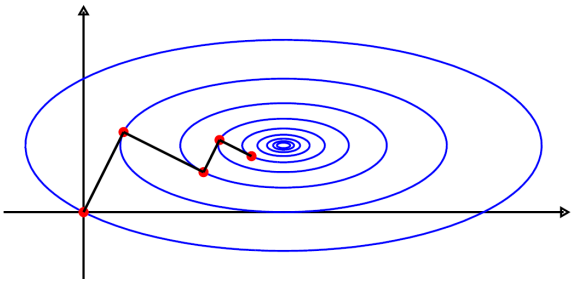


Figura 6: Iterações do Algoritmo do Gradiente.

Fonte: (RIBEIRO; KARAS, 2013)

Lema 3.4.6 *No Algoritmo 3, se t_k é obtido por uma minimização local de $f(x^k + td^k)$, então* $(d^{k+1})^T d^k = 0$.

Demonstração: Ver (RIBEIRO; KARAS, 2013). ■

Teorema 3.4.7 *O Algoritmo 3, com o tamanho do passo t_k calculado pela busca exata, é globalmente convergente, segundo a Definição 3.2.6. O mesmo resultado vale se utilizarmos a busca de Armijo para calcular t_k.*

Demonstração: Ver Ribeiro e Karas (2013). ■

3.4.3 MÉTODO DE NEWTON

O método de Newton é uma das ferramentas mais importantes em otimização. Tanto o algoritmo básico, chamado de Newton Puro, quanto suas variantes, que incorporam busca linear, são muito utilizados para resolver sistemas não-lineares e também para minimização de funções.

Considere uma função $f : \mathbb{R}^n \to \mathbb{R}$ de classe $\mathscr{C}^2$. Nosso objetivo consiste em encontrar um minimizador de f. De acordo com as condições necessárias de otimalidade, devemos resolver o sistema de n equações e n incógnitas dado por $\nabla f(x) = 0$.

Generalizando, considere $F : \mathbb{R}^n \to \mathbb{R}^n$ de classe $\mathscr{C}^1$ e o problema de resolver o sistema (normalmente não-linear)

$$F(x) = 0.$$

Como na maioria das vezes não conseguimos resolvê-lo de forma direta, os processos iterativos constituem a forma mais eficiente de lidar com tais situações.

A ideia é aproximar F por seu polinômio de Taylor de primeira ordem. Dada uma estimativa $\bar{x}$, considere o sistema linear

$$F(\bar{x}) + J_F(\bar{x})(x - \bar{x}) = 0, \tag{3.4.6}$$

onde J_F representa a matriz Jacobiana de F. Caso $J_F(\bar{x})$ seja inversível, sistema (3.4.6) pode ser resolvido, fornecendo

$$x^+ = \bar{x} - (J_F(\bar{x}))^{-1} F(\bar{x}).$$

Esta estimativa representa uma iteração do método de Newton para resolução de equações (ver Figura 7):

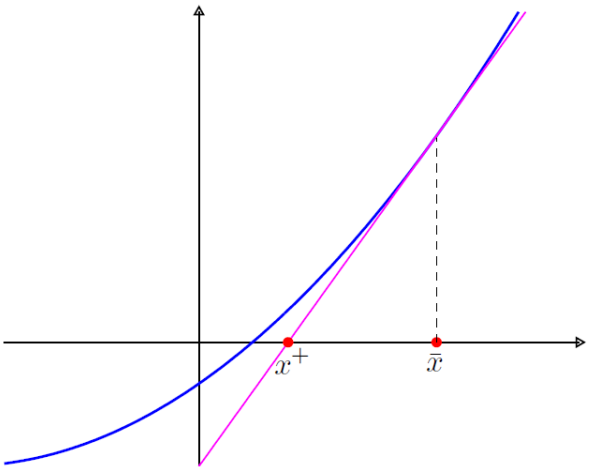


Figura 7: Uma iteração do Método de Newton.
Fonte: (RIBEIRO; KARAS, 2013)

Para o problema de minimizar f, aplicamos a estratégia acima para $F = \nabla f$, obtendo

$$x^+ = \bar{x} - \left(\nabla^2 f(\bar{x})\right)^{-1} \nabla f(\bar{x}). \tag{3.4.7}$$

Com base em (3.4.7) podemos formalizar o método de Newton para minimizar a função f. Basicamente, temos três variantes no algoritmo, o método "puro", onde não fazemos busca unidimensional e aceitamos o passo completo ($t_k = 1$, para todo $k \in \mathbb{N}$), e as outras duas fazem uso de busca (exata ou inexata).

Cabe ressaltar que, do ponto de vista computacional, o cálculo da direção d^k é feito resolvendo-se o sistema de equações lineares

$$\nabla^2 f(x^k) d = -\nabla f(x^k),$$

que tem um custo computacional menor do que o gasto para inverter uma matriz. Outra observação é que, diferentemente do que acontece no algoritmo do Gradiente, o passo de Newton pode não estar bem definido, caso a matriz Hessiana $\nabla^2 f(x^k)$ seja singular. Além disso, mesmo que o passo d^k seja calculado, esta direção pode não ser de descida. Entretanto, se $\nabla^2 f(x^k)$ é definida positiva, então o passo d^k está bem definido e é uma direção de descida.

Algoritmo 4 MÉTODO DE NEWTON.

1: **Dados de entrada**($x^0 \in \mathbb{R}^n$).
2: **Dados de saída**(x^*: mínimo aproximado).
3: Faça $k = 0$.
4: **Enquanto** $\nabla f(x^k) \neq 0$, faça
5: Defina $d^k = -\left(\nabla^2 f(x^k)\right)^{-1} \nabla f(x^k)$.
6: Obtenha $t_k > 0$: definindo $t_k = 1$, pelo Algoritmo 1 ou pelo Algoritmo 2.
7: Faça $x^{k+1} = x^k + t_k d^k$.
8: Faça $k = k + 1$.
9: **Fim Enquanto**.

O passo de Newton também pode ser obtido por uma abordagem diferente da que foi exposta acima. Para isto considere a aproximação de Taylor de segunda ordem de f, dada por

$$p(x) = f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T \nabla^2 f(x^k)(x - x^k).$$

Com o objetivo de minimizar $p(x)$, fazemos

$$\nabla f(x^k) + \nabla^2 f(x^k)(x - x^k) = \nabla p(x) = 0,$$

obtendo exatamente o passo d^k do Algoritmo 4. Desta forma, se $\nabla^2 f(x^k)$ é definida positiva, então o passo de Newton minimiza o modelo quadrático de f em torno de x^k. A Figura 8 ilustra essa abordagem que, para $n = 1$, mostra a função e o modelo, bem como os pontos x^k e x^{k+1}.

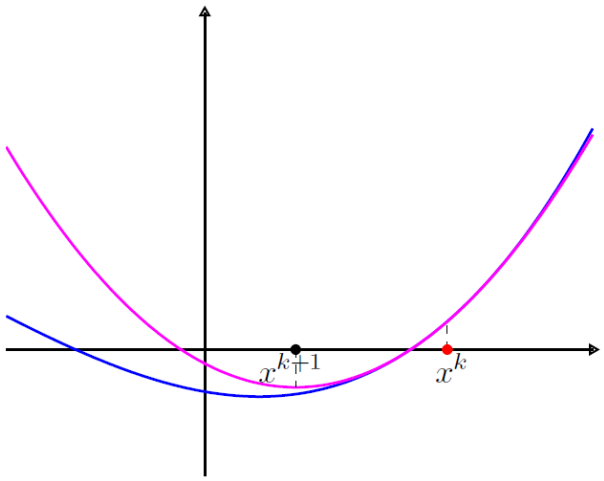


Figura 8: Uma iteração do Método de Newton pelo modelo quadrático.

Fonte: (RIBEIRO; KARAS, 2013)

Esta abordagem sugere que se o método de Newton for aplicado em uma função quadrática, então basta uma iteração para resolver o problema. De fato, considere uma função quadrática $f(x) = \frac{1}{2}x^T Ax + b^T x + c$. Dado $x^0 \in \mathbb{R}^n$, o passo obtido é

$$d^0 = -\left(\nabla^2 f(x^0)\right)^{-1} \nabla f(x^0) = -A^{-1}(Ax^0 + b) = -x^0 - A^{-1}b.$$

Portanto, o minimizador x^* é obtido em um só passo, pois

$$x^1 = x^0 + d^0 = -A^{-1}b = x^*.$$

Teorema 3.4.8 *Suponha que $\nabla^2 f(x)$ é definida positiva, para todo $x \in \mathbb{R}^n$. Então o Algoritmo 4, com o tamanho do passo t_k calculado pela busca exata, é globalmente convergente, segundo a Definição 3.2.6. O mesmo resultado vale se utilizarmos a busca de Armijo para calcular t_k.*

Agora considere $F : \mathbb{R}^n \to \mathbb{R}^n$ uma função de classe $\mathscr{C}^1$ e o problema

$$F(x) = 0.$$

Como vimos no início do texto, caso a matriz Jacobiana $J_F(\bar{x})$ seja inversível, o passo de Newton é dado por

$$d = -\left(J_F(\bar{x})\right)^{-1} F(\bar{x}).$$

Assim, o método pode ser sumarizado no Algoritmo 5.

Algoritmo 5 MÉTODO DE NEWTON PARA SISTEMA DE EQUAÇÕES.

1: **Dados de entrada**($x^0 \in \mathbb{R}^n$).
2: **Dados de saída**(x^*: mínimo aproximado).
3: Faça $k = 0$.
4: **Enquanto** $F(x^k) \neq 0$, faça
5: Defina $d^k = -\left(J_F(x^k)\right)^{-1} \nabla F(x^k)$.
6: Faça $x^{k+1} = x^k + d^k$.
7: Faça $k = k + 1$.
8: **Fim Enquanto**.

3.4.4 MÉTODOS QUASE-NEWTON

Veremos agora uma classe de métodos que estão entre Cauchy e Newton, no sentido de melhorar a performance em relação a Cauchy e ser computacionalmente mais baratos quando

comparados com Newton. A ideia é construir aproximações para a Hessiana da função objetivo ao longo das iterações.

3.4.4.1 O ALGORITMO BÁSICO

O processo iterativo que estudaremos para minimizar uma função $f:\mathbb{R}^n \to \mathbb{R}$ considera as direções de busca dadas por

$$d^k = -H_k \nabla f(x^k), \tag{3.4.8}$$

onde $H_k \in \mathbb{R}^{n\times n}$ é definida positiva. Tal expressão surge de modo natural quando pensamos, como no caso de Newton, em aproximar f por um modelo quadrático em torno de x^k. Entretanto, aqui consideramos:

$$m_k(d) = f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T B_k d,$$

onde $B_k \in \mathbb{R}^{n\times n}$ é uma matriz simétrica qualquer ao invés de $\nabla^2 f(x^k)$. Se B_k for definida positiva, o minimizador do modelo quadrático é dado por

$$-B_k^{-1} \nabla f(x^k).$$

Deste modo, obtemos (3.4.8) escolhendo $B_k = H_k^{-1}$. Mais formalmente, vamos trabalhar em cima do seguinte algoritmo básico.

Algoritmo 6 QUASE-NEWTON.

1: **Dados de entrada**($x^0 \in \mathbb{R}^n, H_0 \in \mathbb{R}^{n\times n} > 0$).
2: **Dados de saída**(x^*: mínimo aproximado).
3: Faça $k = 0$.
4: **Enquanto** $\nabla f(x^k) \neq 0$, faça
5: Defina $d^k = -H_k \nabla f(x^k)$.
6: Obtenha $t_k > 0$: pelo Algoritmo 1 ou pelo Algoritmo 2.
7: Faça $x^{k+1} = x^k + d^k$.
8: Determine $H_{k+1} > 0$.
9: Faça $k = k + 1$.
10: **Fim Enquanto**.

Note que se $H_k = I$, a direção de busca é a de Cauchy. Por outro lado, se $H_k = \left(\nabla^2 f(x^k)\right)^{-1}$, temos a direção de Newton.

Veremos adiante duas maneiras clássicas de atualizar H_k de modo que, ao longo das

iterações, as matrizes obtidas se aproximem da inversa de $\nabla^2 f(x^k)$. O objetivo dos métodos Quase-Newton é utilizar informações das derivadas de primeira ordem para obter a Hessiana de f.

Para entender uma condição que será imposta sobre as matrizes é instrutivo analisar o que ocorre no modo quadrático. Considere então a função

$$f(x) = \frac{1}{2}x^T Ax + b^T x + c, \tag{3.4.9}$$

com $A \in \mathbb{R}^{n\times n}$ definida positiva, $b \in \mathbb{R}^n$ e $c \in \mathbb{R}$. Dados $x^k, x^{k+1} \in \mathbb{R}^n$ e definindo $p^k = x^{k+1} - x^k$, temos

$$\nabla f(x^{k+1}) = \nabla f(x^k) + Ap^k, \tag{3.4.10}$$

que pode ser escrito como

$$q^k = Ap^k, \tag{3.4.11}$$

onde $q^k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

Assim, se obtemos $x^0, x^1, \ldots, x^n$, de modo que os passos $p^0, p^1, \ldots, p^{n-1}$ sejam linearmente independentes e conhecemos os gradientes $\nabla f(x^0), \nabla f(x^1), \ldots, \nabla f(x^n)$, então a inversa A^{-1} fica unicamente determinada, isto é, se uma matriz satisfaz

$$Hq^j = p^j, \tag{3.4.12}$$

para todo $j = 0, 1, 2, \ldots, n-1$, então $H = A^{-1}$. De fato, escrevendo $P = (p^0\ p^1\ \ldots\ p^{n-1})$ e $Q = (q^0\ q^1\ \ldots\ q^{n-1})$, temos por (3.4.11) e (3.4.12),

$$HAP = HQ = P,$$

donde segue que $HA = I$.

Em vista da relação (3.4.12) vamos impor que a matriz H_{k+1}, a ser determinada no Algoritmo 6, satisfaça a condição

$$H_{k+1}q^j = p^j, \tag{3.4.13}$$

para todo $j = 0, 1, \ldots, k$.

3.4.4.2 O MÉTODO DFP

Uma das formas mais conhecidas para a obtenção da matriz H_{k+1} foi proposta por Davidon, Fletcher e Powell. O método, referenciado por DFP, considera correções de posto 2 e tem várias propriedades interessantes, dentre as quais a positividade, cumprimento da relação

(3.4.13) e o fato de gerar direções conjugadas (para definição e propriedades sobre direções conjugadas, recomendamos (RIBEIRO; KARAS, 2013) e (LUENBERGER; YE, 2015)).

A fórmula para a nova matriz é dada por

$$H_{k+1} = H_k + \frac{p^k (p^k)^T}{(p^k)^T q^k} - \frac{H_k q^k (q^k)^T H_k}{(q^k)^T H_k q^k}, \tag{3.4.14}$$

onde $p^k = x^{k+1} - x^k$ e $q^k = \nabla f(x^{k+1}) - \nabla f(x^k)$.

Vamos agora apresentar as primeiras propriedades desta matriz. Naturalmente, a primeira coisa que devemos verificar é que a fórmula está bem definida, ou seja, que os denominadores não se anulam.

Lema 3.4.9 *Suponha que no Algoritmo 6 o tamanho do passo t_k é obtido por uma minimização local de $f(x^k + td^k)$ e que H_k é definida positiva. Então,*

$$(p^k)^T q^k > 0 \text{ e } (q^k)^T H_k q^k > 0.$$

Além disso, H_{k+1} calculada por (3.4.14) *é definida positiva.*

Demonstração: Como $t_k > 0$ é minimizador local de $\varphi(t) = f(x^k + td^k)$, temos

$$\nabla f(x^{k+1})^T p^k = t_k \nabla f(x^{k+1})^T d^k = t_k \varphi'(t_k) = 0.$$

Portanto,

$$(p^k)^T q^k = (p^k)^T (\nabla f(x^{k+1}) - \nabla f(x^k)) = t_k \nabla f(x^k)^T H_k \nabla f(x^k) > 0, \tag{3.4.15}$$

pois H_k é definida positiva e $\nabla f(x^k) \neq 0$. Em particular, temos $q^k \neq 0$, donde segue que $(q^k)^T H_k q^k > 0$. Para provar que H_{k+1} é definida positiva note que, dado $y \in \mathbb{R}^n \backslash \{0\}$,

$$y^T H_{k+1} y = y^T H_k y + \frac{(y^T p^k)^2}{(p^k)^T q^k} - \frac{(y^T H_k q^k)^2}{(q^k)^T H_k q^k}.$$

Por Lema, existe $Q \in \mathbb{R}^{n \times n}$ tal que $H_k = QQ^T$. Fazendo $u = Q^T y$ e $v = Q^T q^k$, temos que

$$u^T u = y^T H_k y,\ v^T v = (q^k)^T H_k q^k \text{ e } u^T v = y^T H_k q^k.$$

Desta forma, usando a desigualdade de Cauchy-Schwarz e (3.4.15), podemos concluir

que

$$y^T H_{k+1} y = \frac{(u^T u)(v^T v) - (u^T v)^2}{v^T v} + \frac{(y^T p^k)^2}{(p^k)^T q^k} \geq 0.$$

Basta verificar que esta soma não se anula. De fato, se a primeira parcela é nula, então existe $\gamma \neq 0$ tal que $u = \gamma v$, o que equivale a $y = \gamma q^k$. Assim,

$$y^T p^k = \gamma (p^k)^T q^k \neq 0,$$

completando a demonstração. ■

Este Lema é válido para funções gerais, não necessariamente quadráticas.

3.4.4.3 O MÉTODO BFGS

Outro método clássico para atualizar as matrizes no Algoritmo 6 é devido a Broyden, Fletcher, Goldfarb e Shanno (BFGS) e também tem boas propriedades teóricas como o método DFP. Além disso, o desempenho computacional do método BFGS é superior ao DFP, razão pela qual ele é amplamente utilizado em implementação de algoritmos para problemas de grande porte.

A ideia tem uma certa simetria com a do método DFP. Consiste em olhar para a relação (3.4.12), mas pensando em uma aproximação para a Hesiana, ao invés de sua inversa. Desta forma, motivados por (3.4.11), procuramos uma matriz B_{k+1} tal que

$$B_{k+1} p^j = q^j, \tag{3.4.16}$$

para todo $j = 0, 1, \ldots, k$.

Para simplificar a notação e entender melhor como obter essa nova matriz, vamos suprimir os índices dos elementos envolvidos. Desta forma, considere $B \in \mathbb{R}^{n \times n}$ definida positiva e $p, q \in \mathbb{R}^n$ tais que $p^T q > 0$. Queremos obter $B_+ \in \mathbb{R}^{n \times n}$ por uma correção de posto 2 na matriz B, de modo que $B_+ p = q$. Para isto, devem existir escalares $a, b \in \mathbb{R}$ e vetores $u, v \in \mathbb{R}^n$ tais que

$$q = B_+ p = (B + auu^T + bvv^T)p = Bp + a(u^T p)u + b(v^T p)v.$$

Uma escolha possível para satisfazer a condição é

$$a(u^T p)u = p \quad \text{e} \quad b(v^T p)v = -Bp.$$

Multiplicando por p^T, obtemos $a(u^T p)^2 = p^T q$ e $b(v^T p)^2 = -p^T Bp$. Assim, conside-

rando $a = 1$ e $b = -1$, temos que

$$u = \frac{q}{u^T p} = \frac{q}{\sqrt{p^T q}} \quad \text{e} \quad v = \frac{Bp}{v^T p} = \frac{Bp}{\sqrt{p^T Bp}}.$$

Portanto,

$$B_+ = B + auu^T + bvv^T = B + \frac{qq^T}{p^T q} - \frac{Bpp^T B}{p^T Bp}. \tag{3.4.17}$$

Note a relação desta fórmula com a obtida por DFP. Uma segue da outra trocando os papéis de B e H, bem como de p e q.

O método BFGS consiste em escolher a nova H como a inversa de B_+. Isto pode ser feito com o auxílio da fórmula de Sherman-Morrison (A.1.2), a saber

$$(Q + uv^T)^{-1} = Q^{-1} - \frac{Q^{-1}uv^T Q^{-1}}{1 + v^T Q^{-1} u}.$$

Aplicando esta fórmula em (3.4.17) e voltando com os índices, obtemos

$$H_{k+1}^{\text{BFGS}} = H_k + \left(1 + \frac{(q^k)^T H_k q^k}{(p^k)^T q^k}\right) \frac{p^k (p^k)^T}{(p^k)^T q^k} - \frac{p^k (q^k)^T H_k + H_k q^k (p^k)^T}{(p^k)^T q^k}, \tag{3.4.18}$$

onde $H_k = B_k^{-1}$.

Apresentamos, a seguir, algumas propriedades do método BFGS, dentre as quais a positividade.

Lema 3.4.10 *Suponha que no Algoritmo 6 o tamanho do passo t_k é obtido por uma minimização local de $f(x^k + td^k)$ e que H_k é definida positiva. Então $(p^k)^T q^k > 0$ e H_{k+1}^{BFGS} é definida positiva.*

Demonstração: A prova de que $(p^k)^T q^k > 0$ é exatamente a mesma feita no Lema 3.4.9. Para verificar a positividade, note que $H_{k+1}^{\text{BFGS}} = B_{k+1}^{-1}$, onde de (3.4.17)

$$B_{k+1} = B_k + \frac{q^k (q^k)^T}{(p^k)^T q^k} - \frac{B_k p^k (p^k)^T B_k}{(p^k)^T B_k p^k}$$

e $B_k = H_k^{-1}$. Assim, trocando H por B e p por q na prova do Lema 3.4.9, pode-se concluir que H_{k+1}^{BFGS} é definida positiva, completando a demonstração. ■

3.4.5 MÉTODO DE GAUSS-NEWTON

Vamos descrever agora o método mais simples para minimização de uma função não-linear para problemas de Mínimos Quadrados da forma (3.3.1) que explora a estrutura do gradiente $\nabla f(x)$ (3.3.4) e da Hessiana (3.3.5). O Método de Gauss-Newton pode ser visto como um Método de Newton para Sistema de Equações modificado com Busca Linear. Ao invés de resolver as equações de Newton padrão $\nabla^2 f(x)d = -\nabla f(x)$, nós resolvemos o seguinte sistema para obter a direção de busca d_k^{GN}:

$$J_k^T J_k d_k^{GN} = -J_k^T r_k. \tag{3.4.19}$$

Esta simples modificação dá uma série de vantagens em relação ao Método de Newton. Primeiro, quando realizamos a aproximação

$$\nabla^2 f_k \approx J_k^T J_k \tag{3.4.20}$$

tiramos o trabalho de calcular as Hessianas individuais dos resíduos $\nabla^2 r_j$, $j = 1,2,\ldots,m$, que são necessárias no segundo termo de (3.3.5). De fato, se calcularmos a Jacobiana J_k no decurso da avaliação do gradiente dado por $\nabla f_k = J_k^T J_k$, a aproximação (3.4.20) não requer nenhum calculo diferencial adicional e a economia no tempo computacional pode ser bastante significativa em algumas aplicações. Em segundo, existem diversas situações interessantes nas quais o primeiro termo $J_k^T J_k$ em (3.3.5) domina o segundo termo (pelo menos perto da solução x^*), tais que $J_k^T J_k$ é uma boa aproximação para $\nabla^2 f_k$ e a taxa de convergência do Método de Gauss-Newton é similar à do Método de Newton. O primeiro termo de (3.3.5) será dominante quando a norma de cada termo de segunda ordem (isto é, $|r_j(x)|\|\nabla^2 r_j(x)\|$) for significativamente menor que os autovalores de $J^T J$.

A terceira vantagem do Método de Gauss-Newton é que sempre que J_k possui posto completo e o gradiente ∇f_k é não-nulo, a direção d_k^{GN} é uma direção de descida para f e, portanto, uma direção adequada para busca linear. De (3.3.4) e (3.4.19) temos

$$(d_k^{GN})^T \nabla f_k = (d_k^{GN})^T J_k^T r_k = -(d_k^{GN})^T J_k^T J_k d_k^{GN} = -\|J_k d_k^{GN}\|^2 \leq 0. \tag{3.4.21}$$

Esta última inequação é estrita a menos quando $J_k d_k^{GN} = 0$, que no caso teremos, por (3.4.19) e pelo posto completo de J_k, que $J_k^T r_k = \nabla f_k = 0$, isto é, x_k é um ponto estacionário. Finalmente, a quarta vantagem do Método de Gauss-Newton está na similaridade entre as equações (3.4.19) e (3.3.7) para o problema de mínimos-quadrados lineares. Essa co-

nexão nos diz que d_k^{GN} é uma solução do problema de mínimos-quadrados lineares

$$\min_p \frac{1}{2}\|J_k p + r_k\|^2. \tag{3.4.22}$$

Portanto podemos encontrar a direção de busca aplicando algoritmos de mínimos-quadrados lineares ao subproblema (3.4.22). De fato, se forem utilizados algoritmos baseados em fatoração QR ou decomposição SVD, não há necessidade de calcular explicitamente a Hessiana aproximada $J_k^T J_k$ em (3.4.19): podemos trabalhar diretamente com a Jacobiana J_k.

Se o número de resíduos m é grande enquanto o número de variáveis n é relativamente pequeno, pode ser imprudente armazenar a Jacobiana J explicitamente. Uma estratégia preferível pode ser calcular a matriz $J^T J$ e o vetor gradiente $J^T r$, avaliando r_j e ∇r_j sucessivamente para $j = 1, 2, \ldots, m$ e executando as acumulações

$$J^T J = \sum_{j=1}^{m} (\nabla r_j)(\nabla r_j)^T \quad \text{e} \quad J^T r = \sum_{j=1}^{m} r_j (\nabla r_j). \tag{3.4.23}$$

Os passos do Método de Gauss-Newton podem então ser calculados resolvendo o sistema (3.4.19) de equações normais diretamente.

O subproblema (3.4.22) sugere outra motivação para a direção de busca do Método de Gauss-Newton. Podemos ver essa equação como sendo obtida de um modelo linear para a função vetorial $r(x_k + d) \approx r_k + J_k d$, substituída na função $\frac{1}{2}\|\cdot\|^2$. Em outras palavras, usamos a aproximação

$$f(x_k + d) = \frac{1}{2}\|r(x_k + d)\|^2 \approx \frac{1}{2}\|J_k d + r_k\|^2,$$

e escolhendo d_k^{GN} para o ser minimizador dessa aproximação.

As implementações do Método de Gauss-Newton geralmente realizam uma busca linear na direção d_k^{GN}, exigindo que o tamanho do passo α_k satisfaça a condição de Armijo, por exemplo.

O método de Gauss-Newton possui convergência global sob o pressuposto que as Jacobianas $J(x)$ possuem seus valores singulares uniformemente limitadas por zero na região de interesse, isto é, existe uma constante $\gamma > 0$ tal que

$$\|J(x)z\| \geq \gamma\|z\| \tag{3.4.24}$$

para todo x em uma vizinhança $\mathscr{N}$ do conjunto de nível

$$\mathscr{L} = \{x | f(x) \leq f(x_0)\}, \tag{3.4.25}$$

onde x_0 é o ponto inical do algoritmo. Assumimos que $\mathscr{L}$ é um conjunto limitado.

Teorema 3.4.11 *Suponha que cada função residual $r_j(x)$ é uma função Lipschitz continuamente diferenciável em uma vizinhança $\mathscr{N}$ do conjunto de nível limitado* (3.4.25)*, e que as Jacobianas $J(x)$ satisfazem a condição uniforme de posto completo* (3.4.24) *em $\mathscr{N}$. Então se as iterações x_k são geradas pelo método de Gauss-Newton com os passos α_k satisfazendo as condições de Wolfe, temos que*

$$\lim_{k\to\infty} J_k^T r_k = 0.$$

Demonstração: Vide Nocedal e Wright (2006). ■

Se J_k possui posto incompleto para algum k (de modo que uma condição como (3.4.24) não é satisfeita), o coeficiente da matriz em (3.4.19) é singular. O sistema (3.4.19) ainda possui solução, devido à equivalência entre este sistema linear e o problema de minimização (3.4.22).

4 ESTUDO DE PROTEÍNAS

O estudo de proteínas é o foco central deste trabalho. Muitos conceitos que discutiremos neste Capítulo são específicos da Biologia e Química, bastante complexos para um leitor iniciante. Contudo, buscamos sintetizar as informações e ilustrar os conceitos da melhor maneira possível. As definições, nomenclaturas e demais conceitos podem ser encontrados em Lima (2012), Sit (2010) e, principalmente, Lesk (2008).

4.1 INTRODUÇÃO A PROTEÍNAS

Proteínas são compostos orgânicos constituídos de *aminoácidos* que estão conectados entre si por *ligações peptídicas* formando uma cadeia. A estrutura de uma proteína "começa" em um aminoácido (definido por convenção como aminoácido N-*terminal*) e "termina" no aminoácido que está mais longe do primeiro em termos de sua posição na cadeia (aminoácido C-*terminal*). Em geral, os aminoácidos são moléculas contendo os grupos funcionais *amina* ($NH2$) e *carboxila* ($COOH$). A Figura 9 ilustra sua estrutura. Os grupos amina e carboxila estão presos ao mesmo átomo de carbono, também conhecido como *carbono alfa* (C_α) e a letra G indica a presença de um *substituinte orgânico* (grupo de átomos preso ao carbono C_α). Cada aminoácido dentro da cadeia é chamado resíduo e é comumente representado por uma letra maiúscula (LESK, 2008). Por exemplo, A, L e V são, respectivamente, siglas para *Alanina*, *Leucina* e *Valina*.

O estudo da configuração tridimensional de uma proteína é fundamental para o conhecimento do papel que ela desempenha nos organismos e é tema de constante de pesquisas (GIBRAT et al., 1996; HASEGAWA; HOLM, 2009). Para fins didáticos, é comum organizarmos a proteína em quatro estruturas:

- **Primária**: são as sequências de todos os aminoácidos que formam uma proteína. A estrutura primária é representada por uma sequência de letras maiúsculas que correspondem aos aminoácidos. As letras são colocadas na mesma ordem em que os aminoácidos ocorrem na cadeia da proteína analisada.

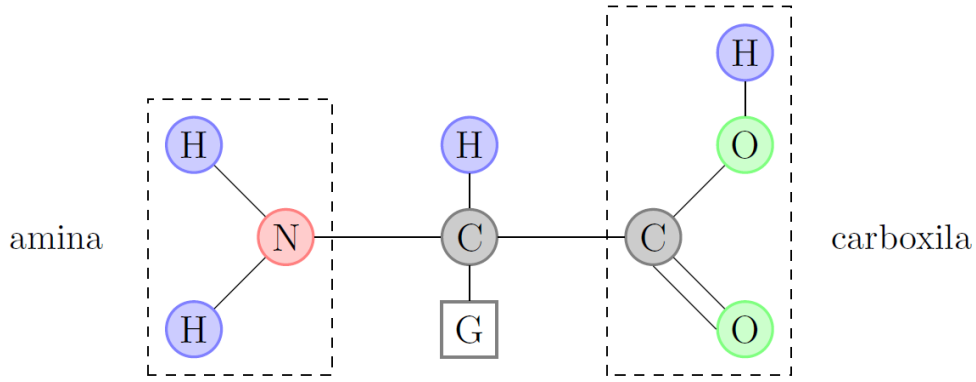


Figura 9: Estrutura geral de um aminoácido.

Fonte: (LIMA, 2012)

- **Secundária**: são as sequências de aminoácidos que se organizam no espaço tridimensional em forma de hélices ou fitas.
- **Terciária**: é a disposição tridimensional de cada estrutura secundária, ou seja, a posição e orientação espacial das hélices e fitas.
- **Quaternária**: é o modo como estão conectadas no espaço várias moléculas de proteínas.

Várias proteínas já foram descobertas e muitas outras surgem a cada dia. Como a quantidade de proteínas conhecidas é grande, faz-se necessária a criação de um banco de dados para guardar e administrar todas as informações relevantes sobre cada uma delas. O banco de dados mais conhecido e utilizado por pesquisadores da área da bioquímica é o *Protein Data Bank* (PDB) (`http://www.rcsb.org/pdb/`). Nele estão arquivados dados de 116772 proteínas (última atualização do banco de dados: 10/01/2017). O *site* permite aos usuários realizar buscas simples ou avançadas, baseadas em informações sobre sequências de aminoácidos, estrutura tridimensional e funções de uma determinada proteína. As moléculas podem ser visualizadas e vários tipos de arquivos podem ser baixados gratuitamente. Como exemplo, as Figuras 10 e 11 foram obtidas ao realizarmos uma busca no PDB pelo nome de *5K20*. A Figura 10 mostra a configuração tridimensional (estrutura terciária) da molécula. Observe a presença de hélices e fitas em sua estrutura. Como dissemos anteriormente, as sequências de aminoácidos que definem estas formas são as estruturas secundárias da proteína.

A Figura 11 mostra um trecho da sequência de aminoácidos (estrutura primária) de uma cadeia da molécula *1AQR*, onde cada aminoácido é representado por uma letra maiúscula. De acordo com esta figura, é possível saber quais aminoácidos formam hélices. Por exemplo, a sequência *EQC* pertence à primeira hélice (linha ondulada localizada na primeira fileira de

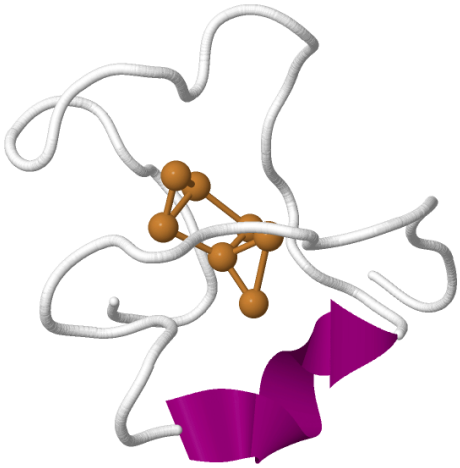


Figura 10: Configuração 3D da molécula *1AQR*.

cima para baixo). Essa proteína não possui fitas.

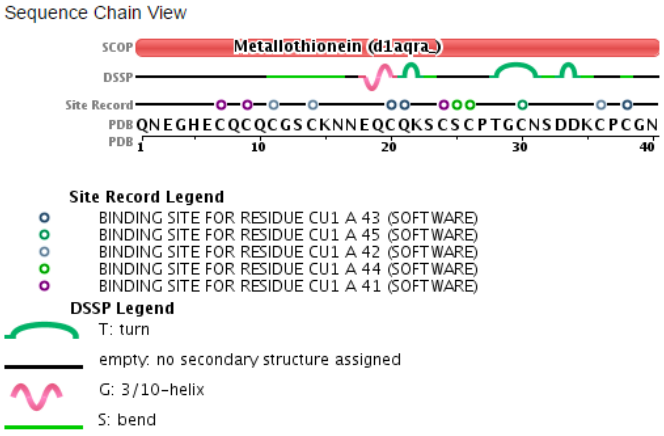


Figura 11: Trecho de uma sequência de aminoácidos da *1AQR*.

Cada proteína cadastrada no PDB possui um único código de identificação (sigla). Este código é composto por quatro caracteres: o primeiro caracter é sempre um número inteiro no intervalo $[1,9]$ enquanto os três últimos pode ser números inteiros em $[0,9]$ ou letras. Por exemplo, $1mbn$, $2hhd$ e $9ins$ são, respectivamente, siglas para mioglobina, hemoglobina humana e insulina. Essa regra permite a criação de 419904 códigos distintos, no entanto as proteínas cadastradas no PDB representam apenas 28% desse total. Como o PDB é constantemente atualizado com proteínas novas, será preciso num futuro próximo redefinir a forma de identificação das estruturas depositadas neste banco de dados.

Além do PDB, outras bases de dados utilizadas por cientistas são SCOP (*Structural Classification of Proteins*) e CATH (*Class, Architeture, Topology and Homologous superfamily*). Estas bases também possuem diversos recursos para que usuários possam estudar as

estruturas e funções das proteínas, além de ser possível comparar uma nova proteína com as proteínas da base de dados armazenada em cada servidor.

4.2 DETERMINAÇÃO DE ESTRUTURAS PROTEICAS

As proteínas formam uma importante classe das biomoléculas. Elas são codificadas nos genes e expressas em células via tradução genética. As proteínas são ingredientes que oferecem suporte à vida (ou as vezes, destruição) e são indispensáveis para a maioria dos processos biológicos. Com a conclusão do sequenciamento genômico do ser humano e de várias outras espécies, os estudos em proteínas tem se tornado mais importante do que nunca pela interpretação dos genes e suas implicações para a vida. Contudo, para compreender as proteínas e suas funções, é de fato essencial conhecer suas estruturas tridimensionais, o que, por diversas razões técnicas, são muito difíceis de se determinar.

Não existem meios físicos diretos para observar a estrutura de uma proteína em uma resolução desejável, por exemplo, em um nível residual. Diversas abordagens experimentais tem sido utilizadas para se obter dados estruturais indiretos sobre os quais as estruturas podem ser deduzidas. Por exemplo, os dados de difração para uma proteína cristalizada pode ser obtida por Cristalografia de Raios-X e utilizadas para encontrar a distribuição da densidade elétrica e consequentemente a estrutura da proteína; os espectros de ressonância magnética de *spins* nucleares em uma proteína pode ser detectada por experimentos NMR (do inglês *Nuclear Magnetic Resonance*) e usada para estimar as distâncias entre certos pares de átomos e, subsequentemente, as coordenadas dos átomos de uma proteína.

As abordagens experimentais possuem diversas limitações. Por exemplo, a Cristalografia de Raios-X requerem a cristalização da proteína, o que consome muito tempo e oferece falhas. Para obter sinais precisos o suficiente, os experimentos NMR só podem ser realizados para pequenas proteínas com menos de algumas centenas de resíduos. Assim, o número de estruturas que podem ser determinadas por essas abordagens experimentais está longe de ser adequado para as crescentes exigências de informação estrutural sobre as centenas de milhares de proteínas de importância médica e biológica.

Pesquisas sobre as estruturas de proteínas depositadas no PDB (BERMAN et al., 2000) mostram que 80% dessas estruturas são determinadas por Cristalografia de Raios-X, 15% por NMR e 5% por outras abordagens. Dessas estruturas, cerca de dezenas de milhares no total, contêm uma elevada porcentagem de replicações (várias estruturas para a mesma proteína determinadas com diferentes técnicas ou sob diferentes condições). Algumas estruturas também são

muito semelhantes, porque há apenas pequenas mutações entre elas. Sem contar as replicações e estruturas geneticamente similares, pode haver apenas cerca milhares de proteínas diferentes cujas estruturas foram determinadas. No entanto, existem pelo menos várias centenas de milhares de proteínas diferentes somente no corpo humano. A maioria de suas estruturas estão ainda desconhecidas.

Na Cristalografia de Raios-X, cientistas obtém um mapa de densidade elétrica para uma proteína cristalizada baseado nos dados de difração cristalizada (ver Figura 12). Eles então atribuem os átomos da proteína a certas posições no mapa de acordo com as formas e desidades das nuvens de elétrons em torno dessas posições. Após o refinamento adicional, a estrutura da proteína é determinada e documentada em um arquivo estrutural. Dentro do arquivo estrutural, os átomos da proteína são listados em uma certa ordem e suas coordenadas são registradas. Além disso, existe um valor chamado fator-B (ou fator de temperatura) determinado e atribuído para cada átomo (ver Figura 13). Seja $< r, r >$ a flutuação quadrática média de um átomo. Então o fator-B para esse átomo é definido como $8\pi^2 < r, r >$. Portanto, o fator-B é um importante indicador de como o átomo flutua em torno de sua posição de equilíbrio devido à sua condição estrutural ou física.

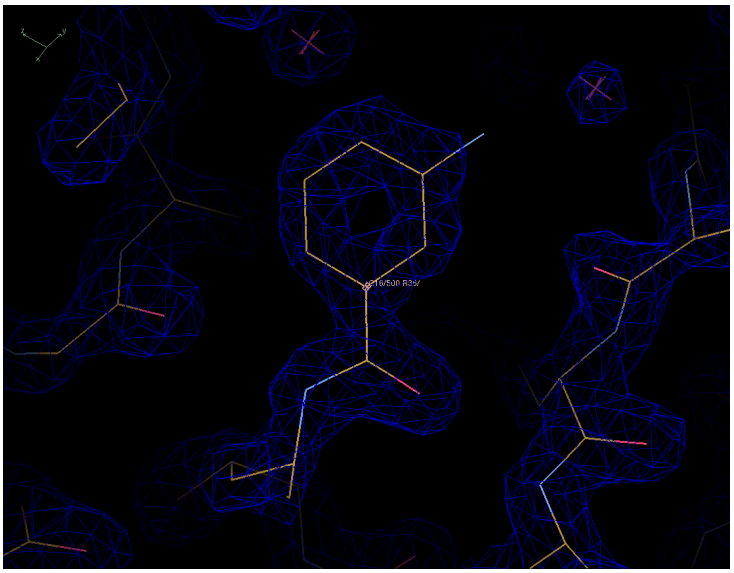


Figura 12: Mapa de densidade elétrica para uma proteína cristalizada.

A vantagem de usar o NMR é que a proteína não precisa estar cristalizada (o que ás vezes é impossível) e a estrutura pode ser determinada na solução. O NMR também pode ser usado para determinar propriedades dinâmicas das proteínas tais como a flexibilidade do suporte principal da proteína ou as correntes laterais na solução.

Nos experimentos com NMR são observados limitantes inferiores e superiores para as distâncias intra-átomos. Dado tal conjunto de limitantes de distâncias, um conjunto estrutural,

```
ATOM      1  N   LEU A  -1      24.643  20.085  43.684  1.00 56.30           N
ATOM      2  CA  LEU A  -1      23.491  19.590  42.867  1.00 55.33           C
ATOM      3  C   LEU A  -1      23.819  18.215  42.303  1.00 57.00           C
ATOM      4  O   LEU A  -1      24.705  18.089  41.445  1.00 55.78           O
ATOM      5  CB  LEU A  -1      23.191  20.554  41.708  1.00 53.09           C
ATOM      6  CG  LEU A  -1      21.892  20.587  40.882  1.00 51.34           C
ATOM      7  CD1 LEU A  -1      21.252  19.218  40.794  1.00 46.80           C
ATOM      8  CD2 LEU A  -1      20.919  21.623  41.456  1.00 50.09           C
ATOM      9  N   GLY A   0      23.120  17.194  42.826  1.00 60.32           N
ATOM     10  CA  GLY A   0      23.280  15.809  42.369  1.00 62.47           C
ATOM     11  C   GLY A   0      22.824  15.824  40.921  1.00 64.52           C
ATOM     12  O   GLY A   0      21.609  15.833  40.666  1.00 64.60           O
ATOM     13  N   SER A   1      23.816  15.981  40.021  1.00 66.30           N
```

Figura 13: Arquivo PDB para uma estrutura em Raios-X.

ao invés de somente uma estrutura, pode ser determinado pelo NMR. Usualmente, cerca de 20 a 100 estruturas são determinadas para representar todo um conjunto estrutural. Elas são alinhadas juntas para mostrar a estrutura global e a flutuação (ver Figura 14). Assim sendo, os arquivos estruturais para as estruturas determinadas por NMR normalmente contêm vários modelos como mostrado na Figura 15. Não existem fatores-B para os átomos, mas as flutuações atômicas pode ser estimadas com base na média das posições atômicas e seus desvios quadráticos médios das posições nos conjuntos estruturais.

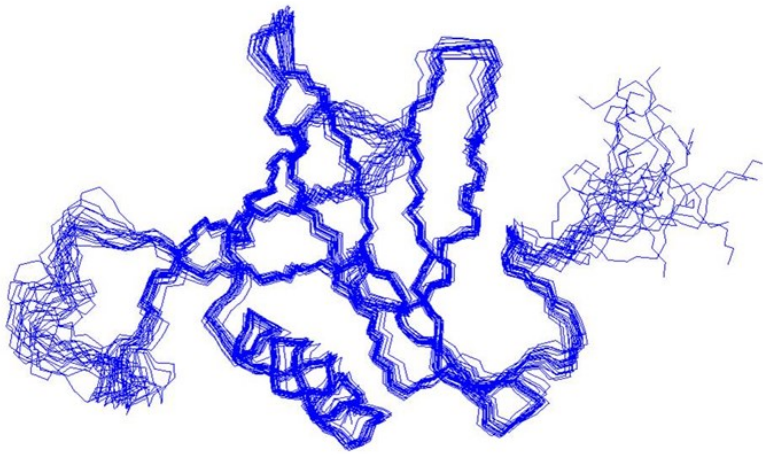


Figura 14: Conjunto estrutural gerado por NMR.

4.3 *PROTEIN DATA BANK*

Bancos de dados de estruturas arquivam, anotam e distribuem conjuntos de coordenadas atômicas. O principal banco de dados de estruturas de macromoléculas biológicas é o *Protein Data Bank* (PDB). Ele contém estruturas de proteínas, ácidos nucléicos e uns poucos carboidratos. Iniciado pelo pesquisador Walter Hamilton do *Brookhaven National Laboratories* em 1971, o PDB é gerenciado atualmente pelo *Research Collaboratory for Structural Bioinformatics* (RCSB). O endereço *web* do *Protein Data Bank* é `http://www.rcsb.org/pdb`.

```
MODEL        1
ATOM      1  N   VAL A 171       3.148   2.384   5.854  1.00  0.00           N
ATOM      2  CA  VAL A 171       3.495   2.127   4.420  1.00  0.00           C
ATOM      3  C   VAL A 171       2.427   1.356   3.559  1.00  0.00           C
ATOM      4  O   VAL A 171       2.123   1.866   2.480  1.00  0.00           O
ATOM      5  CB  VAL A 171       4.938   1.541   4.174  1.00  0.00           C
ATOM      6  CG1 VAL A 171       5.507   2.026   2.823  1.00  0.00           C
ATOM      7  CG2 VAL A 171       6.023   1.845   5.227  1.00  0.00           C
ATOM      8  H   VAL A 171       3.184   1.513   6.394  1.00  0.00           H
ATOM      9  HA  VAL A 171       3.465   3.113   3.981  1.00  0.00           H
ATOM     10  HB  VAL A 171       4.856   0.437   4.118  1.00  0.00           H
MODEL        2
ATOM      1  N   PRO A   1       8.041  10.551  -0.115  1.00  0.00           N
ATOM      2  CA  PRO A   1       8.700   9.672   0.903  1.00  0.00           C
ATOM      3  C   PRO A   1       8.290  10.125   2.312  1.00  0.00           C
ATOM      4  O   PRO A   1       7.772  11.207   2.504  1.00  0.00           O
ATOM      5  CB  PRO A   1      10.229   9.777   0.710  1.00  0.00           C
ATOM      6  CG  PRO A   1      10.469  10.743  -0.457  1.00  0.00           C
ATOM      7  CD  PRO A   1       9.089  11.146  -1.002  1.00  0.00           C
ATOM      8  H   PRO A   1       7.381   9.986  -0.687  1.00  0.00           H
ATOM      9  H3  PRO A   1       7.522  11.312   0.365  1.00  0.00           H
ATOM     10  HA  PRO A   1       8.383   8.651   0.754  1.00  0.00           H
MODEL        3
ATOM      1  N   PRO A   1       7.139  10.071   0.668  1.00  0.00           N
ATOM      2  CA  PRO A   1       7.802   9.158   1.651  1.00  0.00           C
ATOM      3  C   PRO A   1       7.701   9.772   3.060  1.00  0.00           C
ATOM      4  O   PRO A   1       7.106  10.810   3.249  1.00  0.00           O
ATOM      5  CB  PRO A   1       9.276   8.986   1.224  1.00  0.00           C
ATOM      6  CG  PRO A   1       9.518   9.948   0.054  1.00  0.00           C
ATOM      7  CD  PRO A   1       8.160  10.580  -0.298  1.00  0.00           C
ATOM      8  H   PRO A   1       6.400   9.549   0.153  1.00  0.00           H
ATOM      9  H3  PRO A   1       6.711  10.873   1.172  1.00  0.00           H
ATOM     10  HA  PRO A   1       7.305   8.196   1.640  1.00  0.00           H
```

Figura 15: Arquivo PDB para uma estrutura gerada por NMR.

A *home page* do PDB (Figura 16) possui conexões para os seus próprios arquivos de dados, para material expositivo e tutorial, incluindo artigos curtos e o PDB *Newsletter*, um boletim de notícias sobre o PDB, e recursos para depósito de novas entradas e *softwares* de pesquisa especializados para a recuperação e análise de dados.

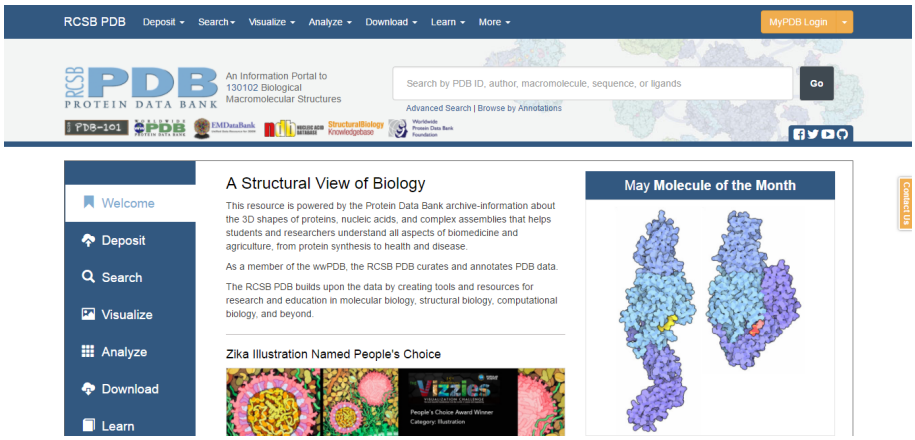


Figura 16: *Home page* do *Protein Data Bank*.

O PDB atribui um identificador, como já foi exposto, de quatro caracteres a cada uma das estruturas depositadas. Não existe nenhum significado mnemônico. Em muitos casos, diversas entradas correspondem a uma mesma proteína - determinada em diferentes estados de ligação, ou em diferentes formas cristalinas, ou resolvidas utilizando cristais de melhor qualidade ou técnicas mais apuradas de obtenção de dados. Por exemplo, existem quatro gerações de estruturas cristalográficas da mioglobina de baleia cachalote.

A tarefa de recuperar uma estrutura, se o usuÃ¡rio conhece o seu identificador, se torna

mais simples. Da *home page* do RCSB, informando como entrada um código PDB (PDB ID) e selecionando a opção *Site Search*, obté-se uma página sumária sobre a entrada. A Figura 17 mostra a página para a estrutura metalotioneína da *Saccharomyces cerevisiae* (levedura de cerveja), cujo identificador é *1AQR*. As informações presentes nessa página incluem:

- A publicação na qual essa entrada foi descrita, via banco de dados bibliográfico como o PubMed.
- Figuras tridimensionais da estrutura (algumas dessas podem exigir a instalação de um programa de visualização ou *plug-ins* no seu computador).
- O arquivo que contém as informações sobre a própria entrada (*Display Files* ou *Download Files*).
- Listas de estruturas relacionadas, de acordo com diferentes classificações de estruturas de proteínas.

Figura 17: Página de informações da proteína *1QR*.

Caso não possuir o identificador da estrutura, ainda assim é possível encontrar os dados da proteína desejada. A ferramenta simples, chamada *Site Search*, também possibilita uma pesquisa por palavras-chave. Entrando com "*metallothionein* and *Saccharomyces cerevisiae*", 6 resultados retornados, incluindo a *1AQR* e outras estruturas.

Para analisar as informações da estrutura da proteína pode-se utilizar os arquivos de estrutura (*Display Files* ou *Download Files*). A Figura 18 mostra a parte inicial do arquivo de informações da estrutura tridimensional da proteína *1AQR*. As informações incluem:

- Qual é a proteína e sua espécie.

- Quem determinou a estrutura e referências para publicações descrevendo a determinação dessa estrutura.
- Detalhes experimentais sobre a determinação da estrutura, incluindo informações relacionadas à qualidade do resultado.
- A sequência de aminoácidos.
- Quais moléculas adicionais aparecem na estrutura, incluindo co-fatores, inibidores e solvente, como moléculas de água.
- Coordenadas tridimensionais dos átomos (Figura 19).

```
HEADER    METALLOTHIONEIN                         31-JUL-97   1AQR
TITLE     CU-METALLOTHIONEIN FROM SACCHAROMYCES CEREVISIAE, NMR,
TITLE    2 MINIMIZED AVERAGE STRUCTURE
COMPND    MOL_ID: 1;
COMPND   2 MOLECULE: CU-METALLOTHIONEIN;
COMPND   3 CHAIN: A;
COMPND   4 SYNONYM: CU-MT
SOURCE    MOL_ID: 1;
SOURCE   2 ORGANISM_SCIENTIFIC: SACCHAROMYCES CEREVISIAE;
SOURCE   3 ORGANISM_COMMON: BAKER'S YEAST;
SOURCE   4 ORGANISM_TAXID: 4932;
SOURCE   5 STRAIN: 2180
KEYWDS    METALLOTHIONEIN, COPPER DETOXIFICATION, METAL-THIOLATE
KEYWDS   2 CLUSTER
EXPDTA    SOLUTION NMR
AUTHOR    C.W.PETERSON,S.S.NARULA,I.M.ARMITAGE
REVDAT   2   24-FEB-09 1AQR    1       VERSN
REVDAT   1   24-DEC-97 1AQR    0
JRNL        AUTH   C.W.PETERSON,S.S.NARULA,I.M.ARMITAGE
JRNL        TITL   3D SOLUTION STRUCTURE OF COPPER AND
JRNL        TITL 2 SILVER-SUBSTITUTED YEAST METALLOTHIONEINS.
JRNL        REF    FEBS LETT.                    V. 379    85 1996
JRNL        REFN                   ISSN 0014-5793
JRNL        PMID   8566237
JRNL        DOI    10.1016/0014-5793(95)01492-6
REMARK   1
REMARK   2
REMARK   2 RESOLUTION. NOT APPLICABLE.
REMARK   3
REMARK   3 REFINEMENT.
REMARK   3   PROGRAM     : X-PLOR 3.1
REMARK   3   AUTHORS     : BRUNGER
REMARK   3
REMARK   3  OTHER REFINEMENT REMARKS: REFINEMENT DETAILS CAN BE FOUND IN
REMARK   3  THE JRNL CITATION ABOVE.
REMARK   4
REMARK   4 1AQR COMPLIES WITH FORMAT V. 3.15, 01-DEC-08
REMARK 100
REMARK 100 THIS ENTRY HAS BEEN PROCESSED BY BNL.
REMARK 210
REMARK 210 EXPERIMENTAL DETAILS
REMARK 210  EXPERIMENT TYPE               : NMR
```

Figura 18: Parte inicial do arquivo de informações da proteína *1QR*.

```
ATOM      1  N   GLN A   1       8.259   1.294 -10.306  1.00  5.29           N
ATOM      2  CA  GLN A   1       8.543  -0.145 -10.039  1.00  4.64           C
ATOM      3  C   GLN A   1       9.623  -0.279  -8.962  1.00  3.69           C
ATOM      4  O   GLN A   1       9.551  -1.128  -8.097  1.00  4.00           O
ATOM      5  CB  GLN A   1       9.045  -0.704 -11.371  1.00  5.29           C
ATOM      6  CG  GLN A   1       8.008  -0.434 -12.463  1.00  5.64           C
ATOM      7  CD  GLN A   1       7.729  -1.725 -13.233  1.00  6.58           C
ATOM      8  OE1 GLN A   1       8.434  -2.702 -13.079  1.00  7.05           O
ATOM      9  NE2 GLN A   1       6.723  -1.771 -14.063  1.00  7.16           N
ATOM     10  H1  GLN A   1       8.132   1.795  -9.403  1.00  5.64           H
ATOM     11  H2  GLN A   1       9.054   1.713 -10.827  1.00  5.59           H
ATOM     12  H3  GLN A   1       7.393   1.376 -10.875  1.00  5.44           H
ATOM     13  HA  GLN A   1       7.644  -0.659  -9.738  1.00  4.86           H
ATOM     14  HB2 GLN A   1       9.978  -0.225 -11.633  1.00  5.77           H
ATOM     15  HB3 GLN A   1       9.199  -1.769 -11.279  1.00  5.41           H
ATOM     16  HG2 GLN A   1       7.093  -0.078 -12.012  1.00  5.63           H
ATOM     17  HG3 GLN A   1       8.388   0.314 -13.144  1.00  5.61           H
ATOM     18 HE21 GLN A   1       6.154  -0.983 -14.189  1.00  7.03           H
ATOM     19 HE22 GLN A   1       6.536  -2.594 -14.561  1.00  7.88           H
ATOM     20  N   ASN A   2      10.626   0.554  -9.013  1.00  2.99           N
ATOM     21  CA  ASN A   2      11.713   0.476  -7.995  1.00  2.43           C
ATOM     22  C   ASN A   2      11.594   1.635  -7.002  1.00  1.89           C
ATOM     23  O   ASN A   2      12.548   2.001  -6.344  1.00  2.70           O
ATOM     24  CB  ASN A   2      13.011   0.585  -8.796  1.00  3.11           C
ATOM     25  CG  ASN A   2      13.004   1.884  -9.604  1.00  3.79           C
ATOM     26  OD1 ASN A   2      12.826   1.863 -10.807  1.00  4.29           O
ATOM     27  ND2 ASN A   2      13.189   3.020  -8.991  1.00  4.37           N
ATOM     28  H   ASN A   2      10.665   1.231  -9.720  1.00  3.25           H
ATOM     29  HA  ASN A   2      11.678  -0.469  -7.477  1.00  2.77           H
ATOM     30  HB2 ASN A   2      13.853   0.587  -8.118  1.00  3.26           H
ATOM     31  HB3 ASN A   2      13.091  -0.255  -9.468  1.00  3.57           H
ATOM     32 HD21 ASN A   2      13.330   3.037  -8.022  1.00  4.38           H
ATOM     33 HD22 ASN A   2      13.187   3.858  -9.500  1.00  5.06           H
ATOM     34  N   GLU A   3      10.431   2.216  -6.888  1.00  1.28           N
ATOM     35  CA  GLU A   3      10.255   3.350  -5.936  1.00  1.73           C
ATOM     36  C   GLU A   3       8.842   3.334  -5.346  1.00  1.23           C
ATOM     37  O   GLU A   3       7.882   3.002  -6.013  1.00  1.64           O
ATOM     38  CB  GLU A   3      10.470   4.609  -6.778  1.00  2.79           C
ATOM     39  CG  GLU A   3      11.941   5.025  -6.711  1.00  3.76           C
ATOM     40  CD  GLU A   3      12.057   6.395  -6.038  1.00  4.70           C
ATOM     41  OE1 GLU A   3      11.488   7.340  -6.560  1.00  5.28           O
ATOM     42  OE2 GLU A   3      12.713   6.475  -5.012  1.00  5.16           O
```

Figura 19: Informações das coordenadas tridimensionais da proteína *1QR*.

5 O PROBLEMA DA GEOMETRIA DE DISTÂNCIAS

Neste Capítulo realizaremos um estudo aprofundado do Problema da Geometria de Distâncias (PGD), especificamente da aplicação em Geometria Molecular. Neste estudo, discutiremos as definições, resultados e abordagens para a resolução do PGD. As notações, definições e resultados estudados neste Capítulo foram baseados em Lima (2012), Lavor et al. (2008) e Liberti et al. (2014).

5.1 INTRODUÇÃO À GEOMETRIA DE DISTÂNCIAS

A Geometria de Distâncias (GD) é uma área de pesquisa consolidada, que possui a Matemática e a Computação como áreas fundamentais em seu alicerce. O conceito de distância é essencial para a experiência humana e a GD o coloca como objeto principal para o estudo de uma estrutura geométrica.

Atualmente, o principal problema da GD, denominado Problema da Geometria de Distâncias (PGD), consiste em determinar um conjunto de pontos em um dado espaço topológico, cujas distâncias, entre alguns deles, são conhecidas. Por mais que essa seja a definição geral do PGD, neste trabalho sempre consideraremos que o espaço topológico é um espaço Euclidiano, isto é, $\mathbb{R}^n$.

Considera-se que a GD surgiu em 1928, quando Menger (1928) caracterizou vários conceitos geométricos utilizando a ideia de distâncias. Contudo, apenas com os resultados de Blumenthal (1953) que o tema se tornou, de fato, uma nova área do conhecimento.

Em seus primórdios, a principal questão da GD era encontrar condições necessárias e suficientes para decidir se, dado um conjunto $\Delta = \{ \widehat{d_{ij}} \in \mathbb{R}_+ \mid 1 \leq i < j \leq n \}$ com $|\Delta| = n(n-1)/2$ números reais não negativos, é possível encontrar uma configuração de pontos em um determinado espaço, de modo que as distâncias Euclidianas entre pares desses pontos sejam dadas pelos elementos de Δ. Usualmente os elementos de Δ são organizados em uma matriz. Vale ressaltar que, nesse caso, todas as distâncias são conhecidas exatamente. Esse resultado

foi alcançado por Schoenberg (1935) e o exploraremos na Seção 5.2.

A primeira menção explícita ao problema fundamental da GD (PGD, mencionado mais acima), onde não são conhecidas todas as distâncias, foi dada por Yemini (1978). Essa mudança faz toda diferença, tanto do ponto de vista teórico quanto prático, pois além do aumento da dificuldade teórica para a criação um método de resolução, em muitas aplicações não se conhecem todas as distâncias relacionadas.

Visto que o conceito de distâncias está presente em várias áreas do conhecimento, consequentemente a GD possui diversas aplicações, como por exemplo: alocamento de roteadores de sinal *wi-fi*, robótica, biologia molecular, entre outras. No primeiro caso, o conjunto de pontos gerado pela resolução pode representar as posições em que os roteadores devem ser postos de modo que as distâncias entre eles respeitem os valores pré-estabelecidos, isto é, que faça com que o sinal *wi-fi* alcance todo o espaço e com potência suficiente. No segundo caso, o conjunto de pontos representa as posições em que os braços robóticos devem estar para exercer certas tarefas. E por fim, um problema muito conhecido na Biologia Molecular consiste em determinar a configuração tridimensional de uma proteína, sendo conhecidas as distâncias entre seus átomos (LIBERTI et al., 2014). Esta última aplicação constitui o foco principal do nosso trabalho e será explorado nas seções a seguir.

O PGD pode ser formalizado, matematicamente, da seguinte forma:

Encontre os pontos $x_1, x_2, \ldots, x_n$ tais que

$$\|x_i - x_j\| = d_{ij}, \quad \text{para} \quad (i,j) \in S, \tag{5.1.1}$$

onde S é um subconjunto de todos os pares de pontos cujas distâncias d_{ij} são conhecidas.

Em experimentos práticos ou até mesmo em estimativas teóricas, as distâncias obtidas podem conter erros. Assim, uma forma geral mais prática de definir o problema seria: encontre $x_1, x_2, \ldots, x_n$, dado um conjunto de limitantes inferiores e superiores, respectivamente l_{ij} e u_{ij}, para as distâncias d_{ij}, tais que

$$l_{ij} \leq \|x_i - x_j\| \leq u_{ij}, \quad \text{para} \quad (i,j) \in S, \tag{5.1.2}$$

onde S é um subconjunto de todos os pares de pontos cujos limitantes para as distâncias são conhecidos.

O PGD é resolvido em tempo polinomial se as distâncias para todos os pontos são conhecidas, porém é provado ser NP-difícil em geral (SIT, 2010). A definição da classe de problemas NP-difíceis pode ser vista em Bovet et al. (1994). Mesmo se os erros associados às

distâncias forem pequenos, o problema ainda é difícil. É importante salientar que o conjunto S em (5.1.1) e (5.1.2) pode não necessariamente conter todos os possíveis pares de pontos (i, j).

Definição 5.1.1 *Dizemos que o PGD possui um conjunto esparso de distâncias se S, em* (5.1.1) *ou* (5.1.2)*, possuir somente um subconjunto próprio de todos os pares de pontos* (i, j)*; caso contrário, dizemos que o PGD possui um conjunto completo de distâncias.*

Definição 5.1.2 *Dizemos que o PGD possui um conjunto de distâncias inexatas (ou distâncias limitadas) se as distâncias forem dadas em um intervalo estimado, que é o caso em* (5.1.2)*; caso contrário, que é o caso em* (5.1.1)*, dizemos que o PGD possui um conjunto de distâncias exatas.*

Na literatura existem basicamente duas maneiras distintas de tratar o PGD numericamente. Na primeira, pesquisadores formulam um problema de Otimização Não-Linear e emprega métodos contínuos para obterem uma solução aproximada. Na segunda, o problema é resolvido utilizando técnicas de otimização discreta e teoria de grafos (LIMA, 2012).

A abordagem discreta do PGD possui diversas vantagens e muitos pesquisadores renomados tratam o PGD desta forma, apresentando ótimos resultados (LAVOR et al., 2008). Contudo, neste trabalho será considerada a abordagem contínua.

Formulações de mínimos-quadrados são geralmente utilizadas para minimizar as discrepâncias entre os dados observados e preditos de uma certa variável. Nesta linha, podemos considerar as distâncias d_{ij} dadas na formulação do PGD como sendo o conjunto de dados observados e as distâncias $\|x_i - x_j\|$ como sendo o conjunto de dados preditos. Com isso, se torna natural e intuitivo formularmos o PGD na forma de um problema de mínimos quadrados.

Portanto, dado um conjunto de distâncias d_{ij}, as coordenadas dos pontos $x_1, x_2, \ldots, x_n$ em $\mathbb{R}^m$, satisfazendo (5.1.1), podem ser obtidas resolvendo o seguinte problema:

$$\begin{aligned} &\min \sum_{(i,j)\in S} \left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2 \\ &\text{sujeito a } x_i \in \mathbb{R}^m, i = 1, 2, \ldots, n. \end{aligned} \tag{5.1.3}$$

Se um conjunto de limitantes para as distâncias, como em (5.1.2), é dado em vez disso, uma formulação similar também pode ser feita, de modo que a soma dos quadrados dos erros é minimizada quando a estrutura encontrada se encaixa nos limitantes das distâncias:

$$\begin{aligned} &\min \sum_{(i,j)\in S} \left(\|x_i - x_j\|^2 - u_{ij}^2\right)^2 + \left(l_{ij}^2 - \|x_i - x_j\|^2\right)^2 \\ &\text{sujeito a } x_i \in \mathbb{R}^m, i = 1, 2, \ldots, n. \end{aligned} \tag{5.1.4}$$

Outras formulações similares a (5.1.3) e (5.1.4) também podem ser utilizadas, que simplesmente remove os quadrados das distâncias. Portanto, para distâncias exatas, o mesmo problema se torna:

$$\begin{aligned} \min \sum_{(i,j)\in S} \big(\|x_i - x_j\| - d_{ij}\big)^2 \\ \text{sujeito a } x_i \in \mathbb{R}^m, i = 1,2,\ldots,n, \end{aligned} \tag{5.1.5}$$

e, para distâncias limitadas,

$$\begin{aligned} \min \sum_{(i,j)\in S} \big(\|x_i - x_j\| - u_{ij}\big)^2 + \big(l_{ij} - \|x_i - x_j\|\big)^2 \\ \text{sujeito a } x_i \in \mathbb{R}^m, i = 1,2,\ldots,n. \end{aligned} \tag{5.1.6}$$

As funções objetivo (5.1.5) e (5.1.6) calculam os erros das distâncias diretamente em vez dos erros dos quadrados das distâncias e, desta forma, podem ser numericamente mais estáveis. Observe que as variáveis em (5.1.5) e (5.1.6) são as coordenadas dos pontos $x_i \in \mathbb{R}^m$ e as funções objetivo não são diferenciáveis em uma configuração de pontos onde $x_i = x_j$ para algum $(i,j) \in S$. Contudo, se estamos interessados em realizar distâncias d_{ij} sempre maiores que zero, não corremos o risco de obter configurações com pontos coincidentes e enfrentarmos problemas numéricos com a aplicação de um algoritmo de minimização que requer derivadas de primeira ordem analíticas. Essa última afirmação foi demonstrada por Leeuw (1984).

5.2 MATRIZES DE DISTÂNCIAS EUCLIDIANAS

Definição 5.2.1 *Uma matriz $D = \big[d_{ij}\big] \in \mathbb{R}^{n\times n}$ é chamada Matriz de Distâncias Euclidianas se existem n pontos $x^1, x^2, \ldots, x^n \in \mathbb{R}^m$, com $m \leq n-1$, tais que*

$$\|x^i - x^j\| = d_{ij}, \text{ para todo } 1 \leq i < j \leq n.$$

Deste modo, o conjunto $\{ x^1, x^2, \ldots, x^n \}$ é dito ser uma realização em $\mathbb{R}^m$ da matriz D.

Fica claro pela Definição 5.2.1 que se D é uma matriz de distâncias euclidianas, então D é simétrica com diagonal nula, isto é, $d_{ii} = 0,\ i = 1,\ldots,n$. O conjunto de matrizes de distâncias euclidianas de ordem n é comumente denotado por $EDM(n)$ (sigla para *Euclidean Distance Matrices of Order n*). Assim, dada uma matriz simétrica $\widehat{D}$ de ordem n com entradas não-negativas e diagonal nula, estamos interessados em saber que condições devem ser cumpridas para que esta matriz pertença ao conjunto $EDM(n)$. O teorema abaixo, devido a Schoenberg (1935), estabelece um critério de verificação:

Teorema 5.2.2 *Seja* $\widehat{D} = \left[\widehat{d_{ij}}\right] \in \mathbb{R}^{n\times n}$ *uma matriz simétrica com entradas não-negativas e diagonal nula. Sejam também* $M \in \mathbb{R}^{n\times n}$ *e* $V \in \mathbb{R}^{n\times(n-1)}$ *tais que*

$$M = \left[(\widehat{d_{ij}})^2\right] \text{ e } V = \begin{bmatrix} -\mathbb{1}^T \\ I_{n-1} \end{bmatrix}.$$

Então, $\widehat{D}$ *é uma matriz de distâncias euclidianas se, e somente se,*

$$V^T M V \leq 0. \tag{5.2.1}$$

Demonstração: Vamos mostrar primeiramente que a condição é necessária. Suponha $\widehat{D} \in EDM(n)$. Então, existem n pontos $x^1, x^2, \ldots, x^n$ em $\mathbb{R}^m$, $m \leq n-1$, tais que

$$\|x^i - x^j\| = \widehat{d_{ij}}, \quad \forall\ i, j = 1, 2, \ldots, n. \tag{5.2.2}$$

Elevando ambos os lados de (5.2.2) ao quadrado, segue que

$$\|x^i - x^j\|^2 = \|x^i\|^2 + \|x^j\|^2 - 2\langle x^i, x^j\rangle = (\widehat{d_{ij}})^2. \tag{5.2.3}$$

Sejam $X \in \mathbb{R}^{m\times n}$ tal que a j-ésima coluna contém as coordenadas de x^j. Então, considerando a relação (5.2.3), podemos reescrever M da seguinte forma:

$$M = diag(X^T X)\mathbb{1}^T + \mathbb{1}diag(X^T X) - 2X^T X, \tag{5.2.4}$$

onde $\mathbb{1}$ e $diag(X^T X)$ são vetores em $\mathbb{R}^n$. Logo, usando (5.2.4) e o fato de que

$$V^T \mathbb{1} = 0,$$

obtemos

$$V^T M V = -2V^T X^T X V = -2(XV)^T (XV) \leq 0. \tag{5.2.5}$$

Provemos agora que a condição é suficiente. Suponha que $V^T M V \leq 0$. Então, segue que $-V^T M V \geq 0$ e, como esta matriz admite decomposição espectral, podemos escrever

$$-V^T M V = Q\Lambda Q^T \Longrightarrow -\frac{1}{2}V^T M V = \left(\frac{1}{\sqrt{2}}Q\Lambda^{1/2}\right)\left(\frac{1}{\sqrt{2}}\Lambda^{1/2}Q^T\right) = XX^T, \tag{5.2.6}$$

onde $X \in \mathbb{R}^{(n-1)\times(n-1)}$ e $posto(X) = m \leq n-1$. Denotando as linhas da matriz X por $x^2, x^3, \ldots, x^n$

$\in \mathbb{R}^{n-1}$, da igualdade de matrizes em (5.2.6) obtemos

$$\begin{aligned} (\widehat{d}_{1i})^2 &= \|x^i\|^2, \quad i = 2,\ldots,n, \\ (\widehat{d}_{1i})^2 + (\widehat{d}_{1j})^2 - (\widehat{d}_{ij})^2 &= \|x^i\|^2 + \|x^j\|^2 - \|x^i - x^j\|^2, \quad i \neq j. \end{aligned} \tag{5.2.7}$$

Logo, de (5.2.7), concluímos que:

$$\|x^i\| = \widehat{d}_{1i} \quad \text{para} \quad i = j \quad \text{e} \quad \|x^i - x^j\| = \widehat{d}_{ij} \quad i \neq j, \tag{5.2.8}$$

onde $2 \leq i < j \leq n$. Assim, de acordo com (5.2.8), o ponto $x^1 = 0$ (origem do sistema de coordenadas) juntamente com os pontos $x^2, x^3, \ldots, x^n$ formam uma realização para a matriz $\widehat{D}$. Portanto, $\widehat{D} \in EDM(n)$. ■

A condição suficiente dada pelo Teorema 5.2.2 garante a realização de n pontos em um espaço euclidiano de dimensão no máximo igual a $(n-1)$. Além disso, as coordenadas dos pontos encontrados não são únicas. De fato, qualquer configuração $\{\tilde{x}^1, \tilde{x}^2, \ldots, \tilde{x}^n\}$ obtida de $\{x^1, x^2, \ldots, x^n\}$ por um movimento rígido (combinação de translação e rotação/reflexão) também será uma realização para a matriz $\widehat{D}$. A seguir, ilustraremos o Teorema através de dois exemplos.

Exemplo 5.2.3 *Considere a tarefa de alocar quatro pontos em $\mathbb{R}^3$ cujas distâncias dois a dois são dadas pelas entradas da matriz*

$$\hat{D} = \begin{bmatrix} 0 & 1 & \sqrt{3} & \sqrt{5} \\ 1 & 0 & \sqrt{2} & \sqrt{6} \\ \sqrt{3} & \sqrt{2} & 0 & \sqrt{2} \\ \sqrt{5} & \sqrt{6} & \sqrt{2} & 0 \end{bmatrix}. \tag{5.2.9}$$

Note que para essas dimensões temos

$$V = \begin{bmatrix} -1 & -1 & -1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$M = \begin{bmatrix} 0 & 1 & 3 & 5 \\ 1 & 0 & 2 & 6 \\ 3 & 2 & 0 & 2 \\ 5 & 6 & 2 & 0 \end{bmatrix}.$$

Desta forma, a matriz

$$-\frac{1}{2}V^T MV = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 3 & 3 \\ 0 & 3 & 5 \end{bmatrix} \geq 0,$$

possui todos os seus autovalores positivos e, portanto, é semi-definida positiva. Decompondo esta matriz da mesma forma que em (5.2.6)*, temos que*

$$X = \begin{bmatrix} 0.1943 & -0.1785 & 0.1089 \\ -0.9470 & -0.6632 & 0.6030 \\ 0.2557 & 1.5900 & 2.1505 \end{bmatrix}. \tag{5.2.10}$$

Assim, o vetor nulo juntamente com os vetores dados pelas colunas da matriz (5.2.10) *formam uma configuração de pontos em* $\mathbb{R}^3$ *que satisfazem a matrizes de distâncias D inicialmente dada em* (5.2.9)*. Dessa forma, temos*

$$\begin{aligned} x_1 &= [0,0,0]^T, \\ x_2 &= [0.1943, -0.9470, 0.2557]^T, \\ x_3 &= [-0.1785, -0.6632, 1.5900]^T, \\ x_4 &= [0.1089, 0.6030, 2.1505]^T. \end{aligned}$$

Notemos que a configuração encontrada não é única. De fato, $\bar{x}_1 = [0,0,0]^T$, $\bar{x}_2 = [0,0,1]^T$, $\bar{x}_3 = [1,1,1]^T$ e $\bar{x}_4 = [2,1,0]^T$ formam outra realização para $\hat{D}$ (veja Figura 20).

Exemplo 5.2.4 *Considere agora o problema de colocar três pontos em* $\mathbb{R}^2$*, de forma que as distâncias entre pares desses pontos sejam dadas pela matriz*

$$\hat{D} = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 15 \\ 2 & 15 & 0 \end{bmatrix}. \tag{5.2.11}$$

A matriz

$$-\frac{1}{2}V^T MV = \begin{bmatrix} 1 & -110 \\ -110 & 4 \end{bmatrix}$$

possui os autovalores $\lambda_1 = -107.5102$ *e* $\lambda_2 = 112.5102$ *e, portanto, não é semi-definida po-*

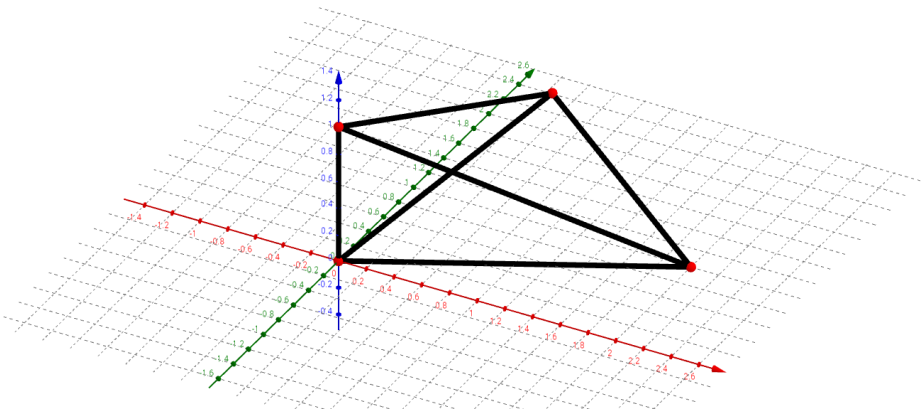


Figura 20: Uma configuração de pontos que realiza as distâncias de $\hat{D}$.
Fonte: Autoria própria.

sitiva, pois um de seus autovalores é negativo. Portanto, pelo Teorema 5.2.2, não é possível determinar uma configuração de três pontos no plano tal que as distâncias de (5.2.11) *sejam realizadas.*

Na prática, verificar se $\hat{D} \in EDM(n)$ não é tarefa fácil, sobretudo se a quantidade de pontos considerada é grande, pois o número de operações necessárias para verificar se uma matriz pertence a $EDM(n)$ é da ordem de n^3 (HUANG et al., 2003). J. Datorro elaborou uma rotina para verificar se uma matriz $\hat{D}$ simétrica, com diagonal nula e entradas não-negativas, é uma matriz de distâncias Euclidianas. A rotina, denominada *isedm*, pode ser obtida em `http://www.convexoptimization.com/wikimization`.

5.3 PROBLEMA DA GEOMETRIA DE DISTÂNCIAS MOLECULARES

A principal característica da Geometria de Distâncias está relacionada à sua riqueza de aplicações, associada à beleza da matemática envolvida. Atualmente, a aplicação de maior destaque da GD reside na conformação molecular, de forma mais específica, ao problema de determinar estruturas tridimensionais das moléculas de proteínas utilizando dados de Espectroscopia NMR. Vale mencionar que o Prêmio Nobel, de 2002, foi outorgado ao químico Kurt Wüthrich pelo desenvolvimento de aplicações de NMR à determinação de estruturas de proteínas (LAVOR; LIBERTI, 2014).

O destaque da aplicação em determinação de estruturas proteicas auxiliou em muito na consolidação da GD como uma área do conhecimento. Um marco importante dessa consolidação é o lançamento do livro de Crippen e Havel, em 1988, considerados os pioneiros da GD no

cálculo de proteínas (CRIPPEN; HAVEL, 1988). Essa consolidação foi gradual, de modo que já em 1984 e 1985 (HAVEL; WÜTHRICH, 1984, 1985), Havel e Wüthrich escreveram dois artigos mostrando como a GD pode ser utilizada no cálculo de estruturas proteicas utilizando dados de NMR.

Os dados de NMR estão relacionados a átomos próximos, cerca de 5 angstroms (Å), e a informação sobre a geometria estrutural das proteínas nos diz que, ao tentarmos determinar a estrutura tridimensional de uma dada proteína, as distâncias entre átomos ligados consecutivamente são conhecidas, *a priori*. Note que a proteína não é uma estrutura rígida, mas podemos considerar esses valores fixos (LAVOR; LIBERTI, 2014).

Isso sugere uma ordem natural entre os átomos da cadeia principal da proteína, formada por uma sequência de 3 átomos que sempre se repetem: N, C_α, C (veja Figura 21). A cadeia principal é o "esqueleto" da proteína, o que nos dá uma boa ideia de sua estrutura trimensional.

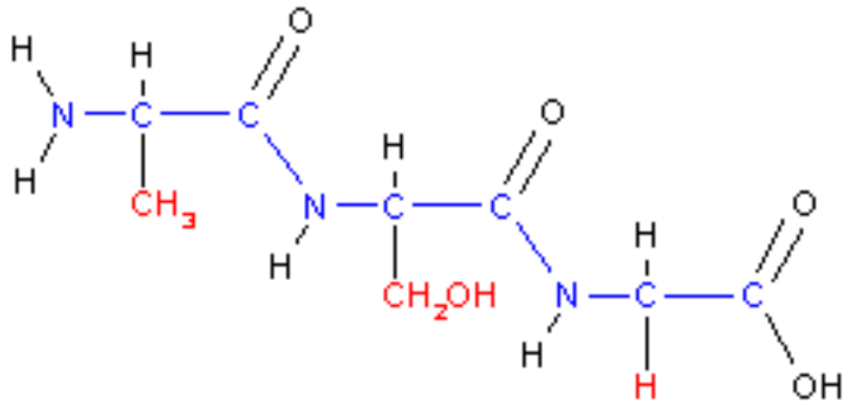


Figura 21: Cadeia principal e cadeias laterais de uma proteína.
Fonte: (LAVOR; LIBERTI, 2014)

De forma geral, as estruturas moleculares são representadas pelas coordenadas tridimensionais de seus átomos, porém, em particular, as estruturas proteicas podem ser representadas de uma forma mais simples, pelas coordenadas tridimensionais dos átomos de C_α de cada aminoácido. Essa representação capta as principais características das disposições tridimensionais dos aminoácidos das estruturas proteicas (ANDREANI et al., 2008). Dessa forma, no decorrer do presente trabalho, quando nos referirmos a átomos de uma proteína estaremos considerando os átomos de C_α de cada aminoácido dessa proteína.

Apesar de que, em geral, o PGD pode ser formulado em qualquer espaço topológico, grande parte das aplicações utilizam-se dos espaços Euclidianos (isto é, $\mathbb{R}^n$). Na aplicação com

reconstrução de proteínas não é diferente, visto que o problema de modelagem de proteínas é todo discutido no espaço Euclidiano tridimensional ($\mathbb{R}^3$). Dessa forma, podemos definir o Problema da Geometria de Distâncias Moleculares (PGDM), que busca determinar a estrutura tridimensional de uma dada proteína sabendo somente as distâncias intra-átomos, que pode ser formalizado, matematicamente, da seguinte forma.

Encontre as coordenadas $x_1, x_2, \ldots, x_n$ dos átomos tais que

$$\|x_i - x_j\| = d_{ij}, \quad \text{para} \quad (i,j) \in S, \tag{5.3.1}$$

onde S é um subconjunto de todos os pares de átomos cujas distâncias são conhecidas.

Quando as distâncias intra-átomos não são conhecidas exatamente, isto é, somente são conhecidos os limitantes para essas distâncias o PGDM será definido da seguinte forma: Encontre $x_1, x_2, \ldots, x_n$, dado um conjunto de limitantes inferiores e superiores, respectivamente l_{ij} e u_{ij}, para as distâncias d_{ij}, tais que:

$$l_{ij} \leq \|x_i - x_j\| \leq u_{ij}, \quad \text{para} \quad (i,j) \in S, \tag{5.3.2}$$

onde S é um subconjunto de todos os pares de átomos cujos limitantes para as distâncias são conhecidos.

Portanto, a fim de resolvermos o PGDM (em (5.3.1) ou (5.3.2)), formularemos o PGDM da seguinte forma:

- Dado um conjunto de distâncias exatas d_{ij}, as coordenadas $x_1, x_2, \ldots, x_n$ dos átomos em $\mathbb{R}^3$, satisfazendo (5.3.1), podem ser obtidas resolvendo os seguintes problemas:

$$\begin{aligned} &\min \sum_{(i,j)\in S} \left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2 \\ &x_i \in \mathbb{R}^3, i = 1,2,\ldots,n \end{aligned} \tag{5.3.3}$$

  e

$$\begin{aligned} &\min \sum_{(i,j)\in S} \left(\|x_i - x_j\| - d_{ij}\right)^2 \\ &x_i \in \mathbb{R}^3, i = 1,2,\ldots,n. \end{aligned} \tag{5.3.4}$$

- Dado um conjunto de limitantes para as distâncias, como em (5.3.2), as coordenadas dos pontos $x_1, x_2, \ldots, x_n$ em $\mathbb{R}^m$, quando a estrutura encontrada se encaixa nos limitantes das

distâncias, podem ser obtidas resolvendo os seguintes problemas:

$$\begin{aligned} \min \sum_{(i,j)\in S} \left(\|x_i - x_j\|^2 - u_{ij}^2\right)^2 + \left(l_{ij}^2 - \|x_i - x_j\|^2\right)^2 \\ x_i \in \mathbb{R}^3, i = 1,2,\ldots,n \end{aligned} \tag{5.3.5}$$

e

$$\begin{aligned} \min \sum_{(i,j)\in S} \left(\|x_i - x_j\| - u_{ij}\right)^2 + \left(l_{ij} - \|x_i - x_j\|\right)^2 \\ x_i \in \mathbb{R}^3, i = 1,2,\ldots,n. \end{aligned} \tag{5.3.6}$$

Como parte dos métodos clássicos de Otimização utilizam informações das derivadas de primeira ordem, convém analisarmos a diferença do vetor gradiente dos problemas (5.3.3) e (5.3.4). Para tal análise, consederaremos o caso particular com $n = 3$ átomos, sendo conhecidas as distâncias d_{12}, d_{13} e d_{23}. Vale notar que as variáveis desse problema são as coordenadas tridimensionais dos átomos x_1, x_2 e x_3, a saber: $x_1 = [x^1, x^2, x^3]^T$, $x_2 = [x^4, x^5, x^6]^T$ e $x_3 = [x^7, x^8, x^9]^T$.

Para o modelo (5.3.3), teremos

$$\begin{aligned} f_1 &= \sum_{i,j=1}^{3} \left(\|x_i - x_j\|^2 - d_{ij}^2\right)^2 \\ &= \left(\|x_1 - x_2\|^2 - d_{12}^2\right)^2 + \left(\|x_1 - x_3\|^2 - d_{13}^2\right)^2 + \left(\|x_2 - x_3\|^2 - d_{23}^2\right)^2 \\ &= \left(\left(x^1 - x^4\right)^2 + \left(x^2 - x^5\right)^2 + \left(x^3 - x^6\right)^2 - d_{12}^2\right)^2 + \\ &\quad + \left(\left(x^1 - x^7\right)^2 + \left(x^2 - x^8\right)^2 + \left(x^3 - x^9\right)^2 - d_{13}^2\right)^2 + \\ &\quad + \left(\left(x^4 - x^7\right)^2 + \left(x^5 - x^8\right)^2 + \left(x^6 - x^9\right)^2 - d_{23}^2\right)^2. \end{aligned}$$

Por outro lado, para o modelo (5.3.4), teremos

$$\begin{aligned} f_1 &= \sum_{i,j=1}^{3} \left(\|x_i - x_j\| - d_{ij}\right)^2 \\ &= \left(\|x_1 - x_2\| - d_{12}\right)^2 + \left(\|x_1 - x_3\| - d_{13}\right)^2 + \left(\|x_2 - x_3\| - d_{23}\right)^2 \\ &= \left(\sqrt{\left(x^1 - x^4\right)^2 + \left(x^2 - x^5\right)^2 + \left(x^3 - x^6\right)^2} - d_{12}\right)^2 + \\ &\quad + \left(\sqrt{\left(x^1 - x^7\right)^2 + \left(x^2 - x^8\right)^2 + \left(x^3 - x^9\right)^2} - d_{13}\right)^2 + \\ &\quad + \left(\sqrt{\left(x^4 - x^7\right)^2 + \left(x^5 - x^8\right)^2 + \left(x^6 - x^9\right)^2} - d_{23}\right)^2. \end{aligned}$$

Neste exemplo não calcularemos todas as entradas do vetor gradiente, pois somente a

primeira entrada, $\frac{\partial f}{\partial x^1}$ (derivada parcial da função f em relação à variável x^1), já é suficiente para analisarmos a principal diferença entre as derivadas de primeira ordem de ambos os modelos. De fato, utilizando as operações básicas de derivação, temos

$$\begin{aligned}
\frac{\partial f1}{\partial x^1} &= \frac{\partial\left[\left(\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2-d_{12}^2\right)^2\right]}{\partial x^1}+ \\
&+\frac{\partial\left[\left(\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2-d_{13}^2\right)^2\right]}{\partial x^1}+ \\
&+\frac{\partial\left[\left(\left(x^4-x^7\right)^2+\left(x^5-x^8\right)^2+\left(x^6-x^9\right)^2-d_{23}^2\right)^2\right]}{\partial x^1} \\
&= 2\left(\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2-d_{12}^2\right)2\left(x^1-x^4\right)(1)+ \\
&+2\left(\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2-d_{13}^2\right)2\left(x^1-x^7\right)(1)+ \\
&+0 \\
&= 4\left(x^1-x^4\right)\left(\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2-d_{12}^2\right)+ \\
&+4\left(x^1-x^7\right)\left(\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2-d_{13}^2\right).
\end{aligned}$$

E também,

$$\begin{aligned}
\frac{\partial f2}{\partial x^1} &= \frac{\partial\left[\left(\sqrt{\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2}-d_{12}\right)^2\right]}{\partial x^1}+ \\
&+\frac{\partial\left[\left(\sqrt{\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2}-d_{13}\right)^2\right]}{\partial x^1}+ \\
&+\frac{\partial\left[\left(\sqrt{\left(x^4-x^7\right)^2+\left(x^5-x^8\right)^2+\left(x^6-x^9\right)^2}-d_{23}\right)^2\right]}{\partial x^1} \\
&= \frac{2\left(\sqrt{\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2}-d_{12}\right)2\left(x^1-x^4\right)(1)}{2\sqrt{\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2}}+ \\
&+\frac{2\left(\sqrt{\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2}-d_{13}\right)2\left(x^1-x^7\right)(1)}{2\sqrt{\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2}}+ \\
&+0
\end{aligned}$$

$$\frac{\partial f2}{\partial x^1} = \frac{2\left(x^1-x^4\right)\left(\sqrt{\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2}-d_{12}\right)}{\sqrt{\left(x^1-x^4\right)^2+\left(x^2-x^5\right)^2+\left(x^3-x^6\right)^2}}+$$
$$+\frac{2\left(x^1-x^7\right)\left(\sqrt{\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2}-d_{13}\right)}{\sqrt{\left(x^1-x^7\right)^2+\left(x^2-x^8\right)^2+\left(x^3-x^9\right)^2}}.$$

É fácil verificar que a principal diferença entre as duas derivadas está no denominador. Se formos resolver o problema associado com $d_{12} = 0$, por exemplo, teremos que, perto da solução ($x_1 = x_2$), o denominador da derivada de primeira ordem do modelo (5.3.4) se aproxima de zero, gerando problemas numéricos.

Visto que o objetivo principal do nosso trabalho consiste na reconstrução de estruturas proteicas sendo conhecidas somente as distâncias intra-átomos, isto é, resolver o PGDM (em (5.3.3), (5.3.4), (5.3.5) ou (5.3.6)), e também não possuimos ferramentas práticas para adquirir dados de Espectroscopia NMR, faremos uma abordagem teórica que diversos pesquisadores (LIMA et al., 2012) também utilizam: resgatamos as estruturas tridimensionais de proteínas do *Protein Data Bank* (PDB) e calculamos todas as distâncias d_{ij} entre todas as coordenadas dos átomos das proteínas. Em um primeiro momento, utilizaremos apenas o conjunto completo de distâncias exatas.

Além disso, também temos a tarefa de alinhamento de proteínas, na qual utilizaremos a estrutura encontrada na solução do PGDM para comparar com a estrutura verdadeira da proteína analisada. A comparação é feita da seguinte forma: dadas a configuração original das coordenadas dos átomos e uma solução numérica do PGDM associado, determinamos um movimento rígido que sobrepõe as duas estruturas de maneira ótima, ou seja, mantendo fixa uma das estruturas, o movimento é aplicado à outra na tentativa de fazer com que elas coincidam. O problema de encontrar este movimento rígido é conhecido na literatura como *Procrustes* ou RMSD (do inglês *Root Mean Square Deviation*) e será descrito na Seção 5.4.

5.4 O PROBLEMA *PROCRUSTES* PARA ALINHAMENTO MOLECULAR

Uma propriedade importante de uma molécula é a sua conformação, ou seja, as coordenadas tridimensionais de seus átomos. Além disso, a conformação não é uma propriedade estática (rígida), então uma molécula pode ter duas ou mais conformações diferentes. É muito útil estudar as diferenças entre conformações de uma molécula, sobretudo quando o objetivo é

a reconstrução de uma molécula.

Uma boa ferramenta para mensurar o quanto duas conformações moleculares são similares é o cálculo do *RMSD* (Desvio Quadrático Médio, do inglês *Root Mean Square Deviation*), em outras palavras a raiz quadrada da soma dos quadrados das distâncias entre os átomos correspondentes. O problema é que a conformação de uma molécula é usualmente alinhada arbitrariamente no espaço, o que significa que calcular o *RMSD* de duas conformações diretamente não produz resultados relevantes. Para resolver esse problema é necessário realizar o alinhamento de ambas estruturas, por um movimento rígido (combinação de translação e rotação/reflexão), com o objetivo de minimizar o valor do *RMSD* (ver Figura 22).

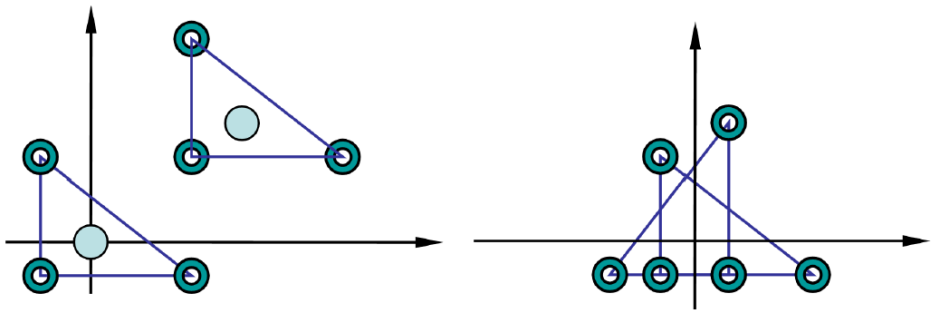


Figura 22: O RMSD pode ser calculado depois que as estruturas estiverem sido alinhadas (translação e rotação/reflexão).

Fonte: (SIT, 2010)

Existem diversas formas de abordar o problema de encontrar esse alinhamento ótimo e, na literatura, esse problema é conhecido como *Procrustes*. Acontece que encontrar a translação ótima é relativamente fácil, pois é provado que a translação ótima é aquela que sobrepõe os centróides das moléculas (HORN, 1987). Por outro lado, encontrar a rotação/reflexão ótima é muito mais complicado (no decorrer do texto, ao invés de falarmos sobre essa rotação ou reflexão, simplesmente falaremos movimento). Existem algumas abordagens iterativas para resolver esse problema, tais como o algoritmo apresentado por McLachlan (1972). Contudo, existe uma solução analítica para esse problema, primeiramente desenvolvido por Kabsch (1976), baseado em decomposição SVD. Mais tarde, um algoritmo baseado em quatérnios unitários foi desenvolvido por Horn (1987). A abordagem baseada em quatérnios para resolver o *Procrustes* se difere da abordagem baseada em SVD, pelo fato de que a primeira encontra um movimento rígido sempre de rotação, enquanto que, utilizando a decomposição SVD, garante-se o alinhamento ótimo mesmo quando uma estrutura se diferenciar da outra por uma reflexão.

5.4.1 DESVIO QUADRÁTICO MÉDIO - RMSD

O Desvio Quadrático Médio (*RMSD*) ou Erro Quadrático Médio (*RMSE*) é um frequente critério utilizado para mensurar as diferenças entre dois dados, que no caso do nosso trabalho se trata de proteínas. Para nossa aplicação, o *RMSD* mensura a média das distâncias entre os átomos correspondentes de duas proteínas sobrepostas. No estudo principal desta seção, usaremos o *RMSD* para mensurar a similariedade de duas estruturas tridimensionais pelas distâncias de seus átomos de C_α (Carbono alfa) correspondentes após a sobreposição por movimento rígido das proteínas (LIMA, 2012).

O Desvio Quadrático Médio - *RMSD* é dado por:

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\delta_i^2},$$

onde δ_i representa as distâncias entre os átomos i correspondentes das duas estruturas.

Normalmente a sobreposição rígida que minimiza o valor do *RMSD* é executada e o valor mínimo é recebido. Dessa forma, dados dois conjuntos, de n pontos, V e W, o *RMSD* entre os conjuntos V e W é calculado por:

$$\begin{aligned} RMSD(V,W) &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}\|V_i - W_i\|^2} \\ &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}((V_{ix}-W_{ix})^2 + (V_{iy}-W_{iy})^2 + (V_{iz}-W_{iz})^2)}. \end{aligned}$$

O valor do *RMSD* é expresso em unidades de comprimento. O mais comumente utilizado na biologia molecular é o Ångstrom que é equivalente a $10^{-10}m$.

Para o *Procrustes*, além do do valor do *RMSD*, devemos nos preocupar com outro dado: a sobreposição das estruturas. Dificilmente iremos nos deparar com as estruturas já sobrepostas (alinhadas) de maneira ótima para calcularmos o *RMSD* e tirarmos conclusões interessantes. Neste caso, vamos utilizar métodos que sobrepõem duas proteínas de maneira ótima através do movimento rígido, ou seja, mantendo uma estrutura fixa o objetivo é determinar uma matriz Q que minimize a função:

$$g(Q) = \sum_{i=1}^{n}\|QV_i - W_i\|^2.$$

Claramente Q^* é o valor mínimo da função $g(Q)$ quando $g(Q^*) = 0$, isto é, quando a estrutura V se sobrepõe a estrutura W.

5.4.2 ALGORITMO DE KABSCH E ALINHAMENTO ÓTIMO

O Algoritmo de Kabsch, criado por Wolfgang Kabsch (1976), é um método que calcula o movimento rígido ótimo que minimiza o valor do *RMSD* entre dois conjuntos de pontos analisados. O algoritmo somente calcula a matriz de movimento, mas também é preciso o cálculo de um vetor de translação.

O algoritmo inicia com dois conjuntos de pontos pareados, P e Q. Cada conjunto pode ser representado por uma matriz de dimensão $3 \times N$. A primeira coluna contém as coordenadas tridimensionais do primeiro ponto, a segunda coluna contém as coordenadas tridimensionais do segundo ponto, a N-ésima coluna contém as coordenadas tridimensionais do N-ésimo ponto. Por exemplo, um conjunto A será representado da seguinte forma:

$$A = \begin{bmatrix} x_1 & x_2 & \dots & x_N \\ y_1 & y_2 & \dots & y_N \\ z_1 & z_2 & \dots & z_N \end{bmatrix} \in \mathbb{R}^{3\times N}.$$

Vale ressaltar que o método de Kabsch, quando este o propôs, usa multiplicadores de Lagrange para resolver o problema de minimização de encontrar o movimento ótimo. Essa formulação pode ser encontrada em (KABSCH, 1976, 1978). Vejamos, então, uma versão mais intuitiva utilizando propriedades de Álgebra Linear, que chega ao mesmo resultado que o original (SEHNAL, 2008).

A ideia principal por trás dessa abordagem é encontrar uma matriz ortonormal $U \in \mathbb{R}^{3\times 3}$ de modo que a aplicação de U para as posições de átomos de uma das matrizes de dados, P, se alinhe da melhor forma possível com a outra matriz de dados, Q, no sentido que o valor a ser minimizado é a distância $D(P,Q)$, onde P e Q já possuem seus centróides coincindindo com a origem do sistema de coordenadas. Matematicamente, o problema pode ser determinado como a minimização do seguinte valor:

$$E = \frac{1}{N}\sum_{i=1}^{N} \|UP_i - Q_i\|^2.$$

Quando E é minimizado, o *RMSD* entre as matrizes P e Q se torna mínimo também. Sendo uma matriz ortonormal, U precisa satisfazer a condição de ortonomalidade $UU^T = I$, onde I é a matriz identidade. A condição de ortonomalidade garante que as linhas e colunas são mutualmente ortogonais e suas normas vetoriais são iguais a 1. Denotando UP como P' e

movendo o fator constante N para a esquerda da equação, a fórmula se torna:

$$NE = \sum_{i=1}^{N} \|P'_i - Q_i\|^2.$$

Como os pontos estão sendo representados por matrizes $\mathbb{R}^{3\times N}$, podemos escrever:

$$NE = \sum_{i=1}^{N} \|P'_i - Q_i\|^2 = Tr((P' - Q)^T (P' - Q)),$$

onde $P' = UP$ e $Tr(A)$ significa o traço da matriz A. É fácil ver que o traço da matriz à direita corresponde precisamente à soma à esquerda. O lado direito da equação pode ser expandido para:

$$Tr((P' - Q)^T (P' - Q)) = Tr(P'^T P') + Tr(Q^T Q) - 2Tr(Q^T P'),$$

que segue das propriedades do operador traço, que são: $Tr(A+B) = Tr(A) + Tr(B)$, $Tr(AB) = Tr(BA)$, $Tr(A^T) = Tr(A)$, $Tr(kA) = kTr(A)$. Além disso, os primeiros dois termos da expansão acima representam a soma dos quadrados de P_i e Q_i, então pode ser reescrito como:

$$NE = \sum_{i=1}^{N} (\|P_i\|^2 + \|Q_i\|^2) - 2Tr(Q^T P').$$

Note que os componentes de P não precisam ser separados (isto é, utilizar P') sabendo que a matriz de movimento U ao redor da origem não muda a norma dos vetores P_i. Note que o somatório acima não depende de U, então minimizar E é equivalente a maximizar $Tr(Q^T P')$. Por essa razão, o resto da discussão é focada em encontrar a matriz de rotação ótima U que maximiza $Tr(Q^T P')$. Relembrando que $P' = UP$, o valor que buscamos maximizar é, então, $Tr((Q^T U)P)$. Das propriedades do operador traço, isso é equivalente a $Tr((PQ^T)U)$. Sabendo que $PQ^T \in \mathbb{R}^{3\times 3}$, então ela possui uma decomposição em valores singulares (SVD), sendo $PQ^T = VSW^T$, onde V e W^T são as matrizes dos autovalores da esquerda (P) e direita (Q^T) (que são matrizes ortonormais), respectivamente, e $S \in \mathbb{R}^{3\times 3}$ é uma matriz cuja diagonal possui os valores singulares s_1, s_2 e s_3 em ordem decrescente. De novo, das propriedades do operador traço, obtemos que:

$$Tr(Q^T P') = Tr(VSW^T U) = Tr(SW^T UV) = \sum_{i=1}^{3} s_i w_i^T U v_i.$$

Se introduzirmos a matriz $T \in \mathbb{R}^{3\times 3}$ como o produto $T = W^T UV$, podemos reescrever a expressão acima como:

$$Tr(Q^T P') = \sum_{i=1}^{3} s_i T_{ii} \leq \sum_{i=1}^{3} s_i.$$

Sabendo que a matriz T é resultante de matrizes ortonormais, então ela própria é ortonormal e $det(T) = \pm 1$. Isso significa que o valor absoluto (módulo) de cada elemento dessa matriz não é maior que 1, de onde segue a última equação. É óbvio que o valor máximo do lado esquerdo da equação é atingido quando os elementos da diagonal de T são iguais a 1, e sabendo que ela é uma matriz ortonormal, todos os outros elementos devem ser nulos. Isso resulta em $T = I$. Além disso, sabendo que $T = W^T UV$, podemos reescrever $W^T UV = I$, e, como W e V são matrizes ortonormais, temos que $WW^T = I$ e $VV^T = I$. Multiplicando $W^T UV$ por W à esquerda e por V^T à direita, temos a solução para U:

$$U = WV^T,$$

onde V e W^T são as matrizes dos autovetores da esquerda e da direita, respectivamente, da matriz de covariância $C = PQ^T$. Essa fórmula assegura que U é uma matriz ortonormal. Todos esses argumentos que foram apresentados como uma prova podem ser postos sucintamente em um algoritmo para o cálculo da matriz de movimento ótimo para alinhar os dois conjuntos de dados P e Q.

Este algoritmo pode, então, ser resumido em três passos: translação, o cálculo da matriz de covariância e o cálculo da matriz de movimento ótimo.

- Passo 1: Translação.

Ambos os conjuntos de coordenadas devem ser primeiramente transladados, de modo que seus centróides coincidam com a origem do sistema de coordenadas. Isso é feito pela subtração das coordenadas dos centróides das coordenadas dos pontos em questão. Dessa forma

$$C_P = P\left(\frac{\mathbb{1}}{N}\right)^T \quad \text{e} \quad C_Q = Q\left(\frac{\mathbb{1}}{N}\right)^T,$$

onde $\mathbb{1} = [\ 1\ 1\ \ldots\ 1\ ]^T \in \mathbb{R}^N$, são os centróides do conjunto P e Q, respectivamente. Agora só falta transladar os conjuntos P e Q o centro da origem do sistema de coordenadas:

$$P_t = P - C_P\mathbb{1} \quad \text{e} \quad Q_t = Q - C_Q\mathbb{1}.$$

- Passo 2: Cálculo da Matriz de Covariância.

O segundo passo consiste em calcular a Matriz de Covariância. Em resumo, ela é dada por:

$$C = P_t diag(I_N m) Q_t^T,$$

onde $m = \dfrac{\mathbb{1}}{N}$ e I_N é a matriz identidade em $\mathbb{R}^{N\times N}$.

- Passo 3: Cálculo da Matriz de Movimento Ótimo.

A Matriz de Movimento Ótimo $U \in \mathbb{R}^{3\times 3}$ é calculada usando este simples algoritmo. Primeiramente, calculamos a decomposição SVD da matriz de Covariância C:

$$C = VSW^T.$$

A seguir, calculamos nossa matriz de movimento ótimo U fazendo:

$$U = WV^T.$$

Dessa forma, U minimiza a função

$$g(Q) = \sum_{i=1}^{N} \|UP_i - Q_i\|^2,$$

ou seja, UP sobrepõe Q.

5.4.3 QUATÉRNIOS E ROTAÇÃO ÓTIMA

Sejam $A = (a_i)_{i=1}^N$ e $B = (b_i)_{i=1}^N$ sequências de pontos no espaço tridimensional (ou seja, as duas estruturas tridimensionais que desejamos comparar). O objetivo é encontrar a transformação $T : \mathbb{R}^3 \to \mathbb{R}^3$ de modo que o valor de

$$RMSD(T(A), B) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|T(a_i) - b_i\|^2},$$

seja minimizado.

Seja $RMSD'(x,y) = \sum_{i=1}^{N} \|x_i - y_i\|^2$. O valor de $RMSD(x,y)$ é minimizado quando $RMSD'(x,y)$ é minimizado. Portanto, a partir de agora iremos nos referir ao $RMSD$ nessa notação simplificada.

O problema, então, se torna em encontrar a transformação T que possa ser decomposta em uma translação ótima e um movimento ótimo. Portanto, T pode ser escrita como $T(x) = R(x) - t$, onde R corresponde ao movimento e t à translação.

Como ocorre no Algoritmo de Kabsch, a translação que oferece o alinhamento ótimo é aquele que sobrepõe os centróides de ambas estruturas. O centróide de um conjunto de pontos

pode ser facilmente calculado da seguinte forma:

$$C_A = \frac{1}{N}\sum_{i=1}^{N} a_i \quad \text{e} \quad C_B = \frac{1}{N}\sum_{i=1}^{N} b_i.$$

Expressaremos ambas as estruturas com respeito a seus centróides (isto é, o centróide de ambas as estruturas sendo a origem do sistema de coordenadas). Para chegarmos a isso, simplesmente subtraimos o centróide de cada estrutura por cada ponto das estruturas:

$$A' = (a_i - C_A)_{i=1}^{N} \quad \text{e} \quad B' = (b_i - C_B)_{i=1}^{N}.$$

Note que

$$\sum_{i=1}^{N} a'_i = 0 \quad \text{e} \quad \sum_{i=1}^{N} b'_i = 0.$$

Usando a identidade $\|a\|^2 = a \cdot a$ e a propriedade distributiva do produto escalar, temos a seguinte expressão:

$$\begin{aligned} RMSD'(T(A),B) &= \sum_{i=1}^{N} \|T(a'_i) - b'_i\|^2 \\ &= \sum_{i=1}^{N} \|R(a'_i) - b'_i - t\|^2 \\ &= \sum_{i=1}^{N} \|R(a'_i) - b'_i\|^2 - 2t\sum_{i=1}^{N}\left(R(a'_i) - b'_i\right) + \sum_{i=1}^{N}\|t\|^2. \end{aligned}$$

O segundo termo $2t\sum_{i=1}^{N}\left(R(a'_i) - b'_i\right)$ é igual a 0, pois as estruturas A' e B' estão em coordenadas centróides. A expressão restante é:

$$\sum_{i=1}^{N} \|R(a'_i) - b'_i\|^2 + \sum_{i=1}^{N}\|t\|^2.$$

Essa expressão é minimizada quando $t = 0$ e, portanto, devemos encontrar a matriz de movimento R que minimize o valor de:

$$\sum_{i=1}^{N} \|R(a'_i) - b'_i\|^2.$$

Para trabalhar com o movimento rígido nos quatérnios, precisamos reformular nosso problema. Vamos utilizar propriedades de norma de vetores e produto escalar para expandir a

última expressão da seguinte forma:

$$\sum_{i=1}^{N} \|R(a_i') - b_i'\|^2 = \sum_{i=1}^{N} \|R(a_i')\|^2 + \sum_{i=1}^{N} \|b_i'\|^2 - 2\sum_{i=1}^{N} R(a_i') \cdot b_i'. \tag{5.4.1}$$

Como movimentos rígidos preservam a norma de vetores, os primeiros dois termos são constantes. Portanto, a fim de minimizar o valor da expressão (5.4.1), precisamos encontrar a matriz de movimento R que maximize o valor de

$$\sum_{i=1}^{N} R(a_i') \cdot b_i'.$$

Reescrevemos a matriz de movimento como sendo $R(x) = qxq^*$, onde q é um quatérnio unitário e o vetor x é expresso como um quatérnio puramente imaginário, descrevendo, assim, uma rotação do vetor x pelo quatérnio unitário. Com isso, teremos:

$$\sum_{i=1}^{N} R(a_i') \cdot b_i' = \sum_{i=1}^{N} \left(qa_i'q^*\right) \cdot b_i'.$$

Agora usando a propriedade (2.1.5) dos quatérnios, tiramos a seguinte expressão:

$$\sum_{i=1}^{N} \left(qa_i'\right) \cdot \left(b_i'q\right).$$

O próximo passo é expressar os quatérnios puramente imaginários a_i' e b_i' como matrizes:

$$\hat{A}_i = \begin{bmatrix} 0 & -a_{i,x} & -a_{i,y} & -a_{i,z} \\ a_{i,x} & 0 & a_{i,z} & -a_{i,y} \\ a_{i,y} & -a_{i,z} & 0 & a_{i,x} \\ a_{i,z} & a_{i,y} & -a_{i,x} & 0 \end{bmatrix},$$

$$B_i = \begin{bmatrix} 0 & -b_{i,x} & -b_{i,y} & -b_{i,z} \\ b_{i,x} & 0 & -b_{i,z} & b_{i,y} \\ b_{i,y} & b_{i,z} & 0 & -b_{i,x} \\ b_{i,z} & -b_{i,y} & b_{i,x} & 0 \end{bmatrix},$$

para obtermos a expressão

$$\sum_{i=1}^{N} \left(\hat{A}_i q\right) \cdot \left(B_i q\right),$$

que, ainda, pode ser simplificada para

$$\sum_{i=1}^{N} q^T \hat{A}_i^T B_i q.$$

Fatorando fora do somatório os termos q^T e q, obtemos expressão

$$q^T \left(\sum_{i=1}^{N} \hat{A}_i^T B_i \right) q.$$

Finalmente, considerando $N_i = \hat{A}_i^T B_i$ e $N = \sum_{i=1}^{N} N_i$, obtemos a seguinte expressão

$$q^T N q,$$

onde N é uma matriz simétrica. De fato, cada N_i tem a forma de

$$N_i = \begin{bmatrix} d_1 & a_z b_y - a_y b_z & a_x b_z - a_z b_x & a_y b_x - a_x b_y \\ a_z b_y - a_y b_z & d_2 & a_y b_x + a_x b_y & a_z b_x + a_x b_z \\ a_x b_z - a_z b_x & a_y b_x + a_x b_y & d_3 & a_z b_y + a_y b_z \\ a_y b_x - a_x b_y & a_z b_x + a_x b_z & a_z b_y + a_y b_z & d_4 \end{bmatrix},$$

a qual é uma matriz simétrica (os atuais valores d_i são irrelevantes) e, portanto, $N \in \mathbb{R}^{4\times4}$ é uma matriz simétrica (COUTSIAS et al., 2004).

Veja que os autovetores $(v)_i$ de N formam uma base ortonormal do $\mathbb{R}^4$. Expressando q como uma combinação linear dos vetores da base $(v)_i$ temos:

$$q = \sum_{i=1}^{4} \alpha_i v_i,$$

onde $\sum_{i=1}^{4} \alpha_i^2 = 1$, que segue do fato de q ser um quatérnio unitário. O produto Nq torna-se, então,

$$Nq = \sum_{i=1}^{4} \alpha_i \lambda_i v_i,$$

onde λ_i é o autovalor correspondente ao autovetor v_i. Até aqui, temos que

$$q^T N q = \sum_{i=1}^{4} \alpha_i \lambda_i q^T v_i = \sum_{i=1}^{4} \alpha_i \lambda_i \left(\sum_{j=1}^{4} \alpha_j v_j^T \right) v_i.$$

Lembrando que $(v)_i$ forma uma base ortonormal, tiramos que $v_i^T v_j = \delta_{ij}$, onde δ_{ij} é o

Delta de Kronecker. Então toda esta última expressão se simplifica para

$$\sum_{i=1}^{4} \alpha_i \lambda_i \alpha_i v_i^T v_i = \sum_{i=1}^{4} \alpha_i^2 \lambda_i,$$

que é uma média ponderada, sabendo que $\sum_{i=1}^{4} \alpha_i^2 = 1$ e $\alpha_i^2 \geq 0$. A média ponderada possui um valor máximo quando o maior autovalor λ_k é ponderado com $\alpha_k = 1$ e todos os outros α iguais a zero.

Isso corresponde à escolha da rotação q como

$$q = v_k,$$

onde v_k é o autovetor correspondente ao máximo autovalor positivo λ_k da matriz simétrica N.

Resolver o polinômio característico de N envolve encontrar as raízes de uma equação do 4° grau. A seguir, encontrar o correspondente autovetor para o maior autovalor λ_{max} pode ser resolvido utilizando eliminação de Gauss.

Vejamos, a seguir, o pseudo-código do método dos quatérnios, a fim encontrar a rotação ótima. Neste pseudo-código, assumimos que as estruturas A e B são representadas por uma lista de quatérnios puramente imaginários.

- Passo 1: Redefina A e B para as "coordenadas centróides".
- Passo 2: Converter os quatérnios em suas representações matriciais e fazer

$$N = \sum_{i=1}^{N} \hat{A}_i^T B_i.$$

- Passo 3: Encontrar o maior autovalor λ_{max} positivo de N.
- Passo 4: Encontrar o autovetor q correspondente ao autovalor λ_{max}.
- Passo 5: Normalizar o vetor q para eliminar erros de arredondamento.
- Passo 6: Rotacionar cada vetor em A

$$\bar{A} = Alinhar(A, q).$$

- Passo 7: Retornar o valor do *RMSD*

$$RMSD(\bar{A}, B).$$

6 RESULTADOS COMPUTACIONAIS

Para os experimentos numéricos, implementamos os métodos BFGS e Gauss-Newton, descritos no Capítulo 3, bem como os algoritmos para alinhamento molecular (de Kabsch e Horn) e as demais funções auxiliares, no *software* MATLAB[1], cujas linhas de código se encontram nos apêndices. Além disso, usaremos oito proteínas do *Protein Data Bank* e tomamos somente as coordenadas dos átomos de carbono alfa (C_α), presentes em cada estrutura. Propusemos duas simulações numéricas cujos detalhes são discutidos a seguir. A sigla de cada proteína e o número de átomos C_α estão presentes na Tabela 1. Os experimentos foram rodados em um computador com as seguintes características: sistema operacional Windows 10 Pro 64 bits, processador Intel Core $i7-2600$ com 3.7 Ghz e 32GB de memória RAM.

Tabela 1: Proteínas utilizadas nos experimentos.

Proteína	n_{C_α}
1A9W	572
1AQR	40
1BQX	77
1BSA	321
1F39	202
1FS3	124
1JK2	90
1MBN	153

Fonte: Autoria própria.

Na proposta, faríamos também uma formulação com um conjunto esparso de distâncias, isto é, não consideraríamos todas as distâncias e faríamos cortes no conjunto de dados, porém os testes númericos realizados mostraram que os métodos BFGS e Gauss-Newton não tiveram uma performance satisfatória nessa perspectiva.

[1] `https://www.mathworks.com/products/matlab.html`

6.1 SIMULAÇÕES NUMÉRICAS - PARTE A

Neste primeiro experimento, selecionamos todos os átomos de C_α das oito proteínas da Tabela 1 e calculamos os conjuntos completos de distâncias exatas, isto é, admitimos conhecidas todas as distâncias d_{ij} entre os pares de átomos C_α. Repetimos cada algoritmo 20 vezes e tomamos o melhor resultado em relação ao valor de função, sempre partindo de um ponto inicial gerado aleatoriamente e os resultados obtidos são apresentados na Tabela 2. As colunas da Tabela 2 obedecem a seguinte legenda: Proteína é a sigla da proteína retirada do PDB, *nvar* é o número de variáveis do problema, *iter* é o total de iterações realizadas pelo algoritmo, $f(x^*)$ é o valor da função na solução encontrada e $t(s)$ é o tempo de execução medido em segundos. Em todos os experimentos o número máximo de iterações foi 400 e a precisão de 10^{-4}.

Tabela 2: Resultado dos Testes Numéricos

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	53	$5.9390e-03$	39.7073	26	$1.4164e-20$	205.5401
1AQR	120	45	$2.2945e-05$	0.1629	30	$1.3944e-24$	0.0829
1BQX	231	51	$1.2723e-05$	0.5621	24	$1.1852e-23$	0.3156
1BSA	963	91	$2.0218e-03$	20.1517	26	$6.0874e-21$	26.2482
1F39	606	81	$2.3484e-03$	6.6483	20	$9.7204e-22$	4.2060
1FS3	372	57	$8.8267e-04$	1.6664	15	$8.7508e-23$	0.6524
1JK2	270	77	$9.0695e-04$	1.1818	23	$1.1712e-22$	0.3994
1MBN	459	49	$6.0768e-04$	2.2029	31	$1.7444e-22$	2.4517

Podemos verificar, na Tabela 2, que, apesar do número muito menor de função na solução e de iterações realizado pelo método de Gauss-Newton em relação ao método BFGS, o método de Gauss-Newton, com um grande número de variáveis, apresenta um tempo de execução muito maior. Isso se deve a dimensão da matriz Jacobiana: $(3n) \times ((n^2-n)/2)$, que é utilizada na resolução de um sistema linear no cáclulo da direção de busca.

Escolhemos a proteína *1BQX* para ilustrar a resolução em que, tanto com o método BFGS quanto com o método de Gauss-Newton, obteve uma configuração de pontos que difere da estrutura original por um movimento rígido de reflexão. A Figura 23 mostra a configuração original de *1BQX* com os seus 77 átomos de carbono alfa. Cada átomo está representado por um ponto em $\mathbb{R}^3$ e pontos consecutivos estão ligados por segmentos de reta. Já a Figura 24 compara as distâncias originais (eixo d^*_{ij}) e as distâncias obtidas numericamente pelo método de Gauss-Newton (eixo d_{ij}). Note que os ajustes são exatos, pois todos os pares ordenados $\left(d^*_{ij}, d_{ij}\right)$ estão sobre a reta $y = x$. As análises realizadas com o problema *Procrustes* estão indicadas nas Figuras 25 e 26.

Tabela 3: Resultado do problema *Procrustes* nos Testes Numéricos

Proteína	RMSD			
	BFGS		Gauss-Newton	
	SVD	Quatérnios	SVD	Quatérnios
1A9W	$4.0418e-06$	$2.4359e+01$	$2.7605e-14$	$2.5548e-14$
1AQR	$9.0517e-06$	$9.0517e-06$	$3.9455e-15$	$1.4217e-14$
1BQX	$2.4795e-06$	$1.1674e+01$	$5.6301e-15$	$1.1674e+01$
1BSA	$7.1585e-06$	$7.1585e-06$	$2.6593e-14$	$1.4778e+01$
1F39	$1.6895e-05$	$1.6895e-05$	$2.4409e-14$	$2.3217e-14$
1FS3	$8.8990e-06$	$8.8990e-06$	$2.1824e-14$	$1.1733e+01$
1JK2	$3.1789e-05$	$3.1789e-05$	$1.6826e-14$	$8.4986e+00$
1MBN	$1.0382e-05$	$1.1214e+01$	$2.0804e-14$	$1.1214e+01$

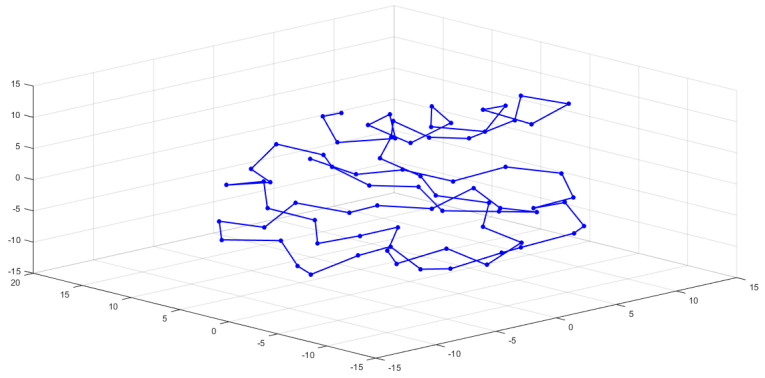


Figura 23: Visualização $3D$ da estrutura original proteína *1BQX*.

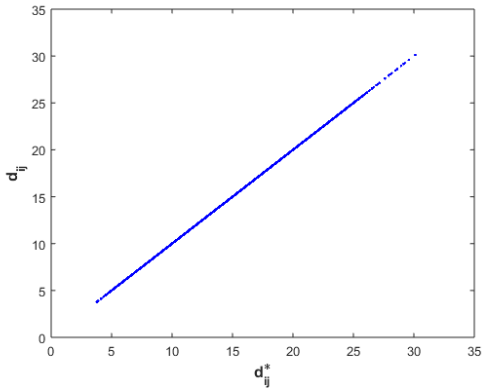


Figura 24: Distâncias (originais e geradas pela resolução do PGDM) da proteína *1BQX*.

As Figuras 25 e 26 mostram a sobreposição da configuração obtida pelo método de Gauss-Newton (pontos vermelhos) com a configuração original (pontos azuis que não aparecem na Figura 25, pois as estruturas estão perfeitamente alinhadas), onde tal sobreposição foi feita utilizando o algoritmo baseado em SVD e o algoritmo baseado quatérnios unitários, res-

pectivamente. A solução obtida na resolução do problema *Procrustes* e o valor do RMSD estão indicados na Tabela 4.

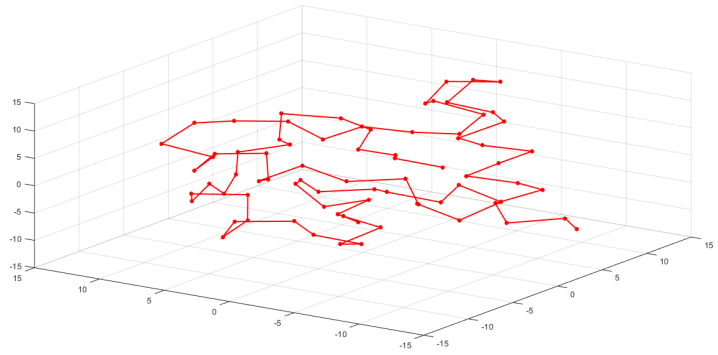


Figura 25: Sobreposição das estruturas original e gerada numericamente da proteína *1BQX* por SVD.

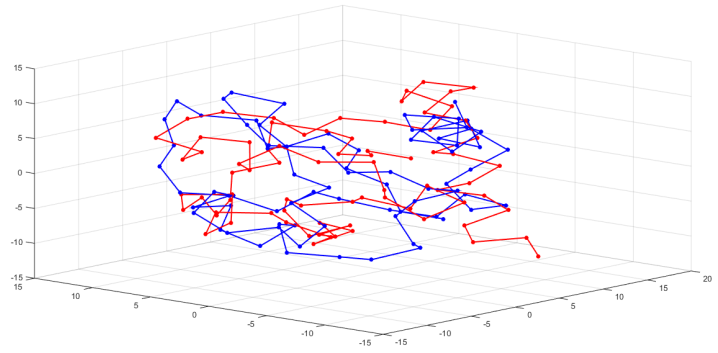


Figura 26: Sobreposição das estruturas original e gerada numericamente da proteína *1BQX* por Quatérnios Unitários.

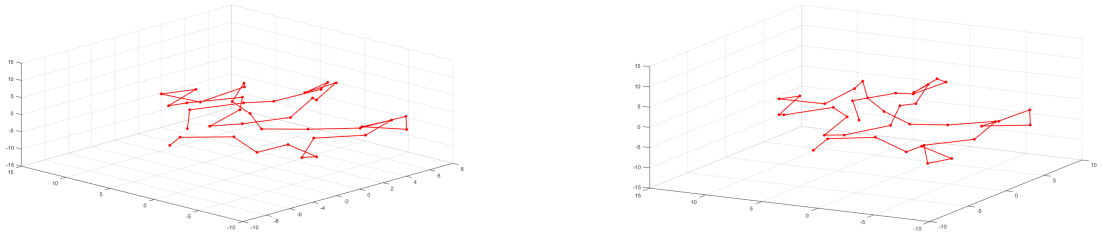
Tabela 4: Resultados Computacionais do problema *Procrustes* envolvendo a Proteína *1BQX* com a solução do Método de Gauss-Newton

Procrustes via Decomposição em Valores Singulares
$Q = \begin{bmatrix} 0.8561 & 0.0093 & 0.5168 \\ 0.3373 & -0.7676 & -0.5449 \\ -0.3916 & -0.6408 & 0.6603 \end{bmatrix}, \quad RMSD = 5.630143e-15.$
Procrustes via Quatérnios Unitários
$q = \begin{bmatrix} 0.3592 \\ 0.7596 \\ -0.0162 \end{bmatrix}, \quad RMSD = 1.167498e+01.$

Fonte: Autoria própria.

Após analisar o resultado obtido pela resolução do PGDM associado à proteína *1BQX*, fica claro que, em ambos os métodos (BFGS e Gauss-Newton), a estrutura tridimensional da solução numérica se difere da original por um movimento rígido de reflexão. Como o algoritmo de alinhamento baseado em decomposição SVD consegue detectar essa reflexão, enquanto o algoritmo baseado em quatérnios unitários somente consegue gerar movimento de rotação, as estruturas somente são alinhadas perfeitamente quando foi utilizado o primeiro algoritmo de alinhamento.

Contudo, na solução do PGDM associada à proteína *1AQR*, em ambos os métodos de otimização e alinhamento molecular, o valor do RMSD foi pequeno, o que pode-se verificar na Figura 27, onde as estruturas estão totalmente alinhadas. Como o algoritmo de alinhamento baseado em quatérnios unitários somente descreve rotações, podemos concluir que a estrutura gerada numericamente pelo PGDM associado à proteína *1AQR* se difere da estrutura original por um movimento de rotação. Neste caso, ambos os algoritmos de alinhamento conseguiram sobrepor as duas estruturas perfeitamente.



(a) Alinhamento por SVD

(b) Alinhamento por Quatérnios Unitários

Figura 27: Sobreposição das estruturas original e gerada numericamente da proteína *1AQR*.

6.2 SIMULAÇÕES NUMÉRICAS - PARTE B

Neste caso, consideramos todas as oito proteínas indicadas na Tabela 1 e todas as distâncias d_{ij} entre seus pares de átomos C_α. No entanto, para simular erros no conjunto de distâncias, perturbamos cada distância d_{ij} com um número gerado aleatoriamente no intervalo $[-\rho,\rho]$, onde $|\rho| \leq 1$. Para cada proteína e cada valor de ρ fixos, resolvemos o problema (5.3.3) 20 vezes, sempre partindo de um ponto inicial diferente. Utilizamos os seguintes valores para ρ: $10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}$ e 10^{-4}, cujos resultados estão sintetizados nas Tabelas 5, 6, 7, 8, 9 e 10. Para valores maiores que $\rho = 10^{-4}$, os algoritmos não convergiram para a solução do problema.

Podemos notar nas Tabelas 5, 6, 7, 8, 9 e 10 que à medida que aumentamos o valor de ρ os valores finais de função do resultados obtidos pelo método de Gauss-Newton também

aumentam, por um fator de 10^2. Um ponto importante a se ressltar é que os números totais de iterações e tempo de execução, indicados nas Tabelas 5, 6, 7, 8, 9 e 10, foram muito semelhantes aos valores obtidos nos experimentos da primeira bateria.

Em relação aos resultados obtidos pelo método BFGS, podemos verificar nas tabelas apresentadas que, de forma geral, os valores de função mantiveram-se próximas das apresentadas na Tabela 2 da primeira bateria, a não ser no caso $\rho = 10^{-4}$, no qual a solução teve uma piora significativa.

Tabela 5: Resultado dos Testes Numéricos com erro $\rho = 10^{-9}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	46	$1.5876e-02$	33.9240	27	$2.4361-10$	204.4772
1AQR	120	51	$4.0249e-04$	0.1898	21	$1.3593e-13$	0.0363
1BQX	231	45	$9.9262e-04$	0.5167	36	$8.4813e-13$	0.4344
1BSA	963	75	$1.5402e-02$	15.9080	24	$8.4272e-11$	22.7349
1F39	606	63	$1.9458e-02$	4.9892	21	$1.8842e-11$	4.4367
1FS3	372	54	$1.4055e-03$	1.5698	24	$3.8329e-12$	1.1035
1JK2	270	83	$1.4826e-03$	1.3464	21	$3.1394e-12$	0.4147
1MBN	459	49	$1.5140e-03$	2.2593	25	$6.7785e-12$	2.3588

Tabela 6: Resultado dos Testes Numéricos com erro $\rho = 10^{-8}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	56	$8.9339e-03$	41.9406	28	$2.4409e-08$	211.8461
1AQR	120	47	$2.8271e-04$	0.1763	26	$1.3675e-11$	0.0480
1BQX	231	41	$1.0726e-04$	0.4738	33	$8.5016e-11$	0.4044
1BSA	963	84	$5.1672e-02$	17.7921	29	$8.3691e-09$	28.9241
1F39	606	74	$2.2522e-02$	5.8885	24	$1.8775e-09$	5.1577
1FS3	372	54	$2.3108e-03$	1.5679	26	$3.8539e-10$	1.1802
1JK2	270	78	$3.9381e-03$	1.2310	20	$3.0142e-10$	0.3624
1MBN	459	54	$2.9087e-03$	2.3934	29	$6.8095e-10$	2.4565

Para analisar se o problema de alinhamento de estruturas com as estruturas encontradas no problema com erro $\rho = 10^{-4}$ foi resolvido com sucesso, escolhemos visualizar as estruturas sobrepostos $1AQR$ e $1BQX$, encontradas pelo método de Gauss-Newton. As Figuras 28 e 29 mostram a sobreposição da configuração obtida pelo método de Gauss-Newton (pontos vermelhos) com a configuração original (pontos azuis que não aparecem na figura, pois as estruturas estão totalmente alinhadas), onde tal sobreposição foi feita utilizando o algoritmo baseado em SVD e o algoritmo baseado quatérnios unitários, respectivamente. A solução obtida na resolução do problema *Procrustes* e o valor do RMSD estão indicados nas Tabelas 11 e 12.

Tabela 7: Resultado dos Testes Numéricos com erro $\rho = 10^{-7}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	46	$9.4533e-04$	34.2576	33	$2.4350e-06$	250.7572
1AQR	120	39	$7.2555e-05$	0.1478	25	$1.3287e-09$	0.0454
1BQX	231	38	$2.3556e-05$	0.4371	26	$8.4952e-09$	0.3074
1BSA	963	73	$6.3572e-02$	15.4956	30	$8.3673e-07$	28.5926
1F39	606	76	$8.7556e-03$	6.0399	21	$1.8719e-07$	4.3682
1FS3	372	141	$2.0912e-03$	4.1381	27	$3.8429e-08$	1.2078
1JK2	270	71	$7.6261e-03$	1.1582	23	$3.0223e-08$	0.4480
1MBN	459	52	$2.3834e-03$	2.4102	22	$6.8886e-08$	2.1134

Tabela 8: Resultado dos Testes Numéricos com erro $\rho = 10^{-6}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	56	$4.8038e-03$	41.1540	22	$2.4475e-04$	163.2213
1AQR	120	41	$6.2749e-04$	0.1548	23	$1.3841e-07$	0.0395
1BQX	231	43	$9.3918e-04$	0.4950	23	$8.5521e-07$	0.2706
1BSA	963	99	$1.7098e-02$	21.0565	36	$8.4082e-05$	33.7448
1F39	606	69	$8.1507e-03$	5.4797	32	$1.8912e-05$	6.6345
1FS3	372	53	$2.0879e-03$	1.5443	33	$3.9023e-06$	1.5455
1JK2	270	82	$8.2786e-04$	1.3179	28	$3.0852e-06$	0.5568
1MBN	459	53	$8.2385e-04$	2.4678	24	$6.6851e-06$	2.4140

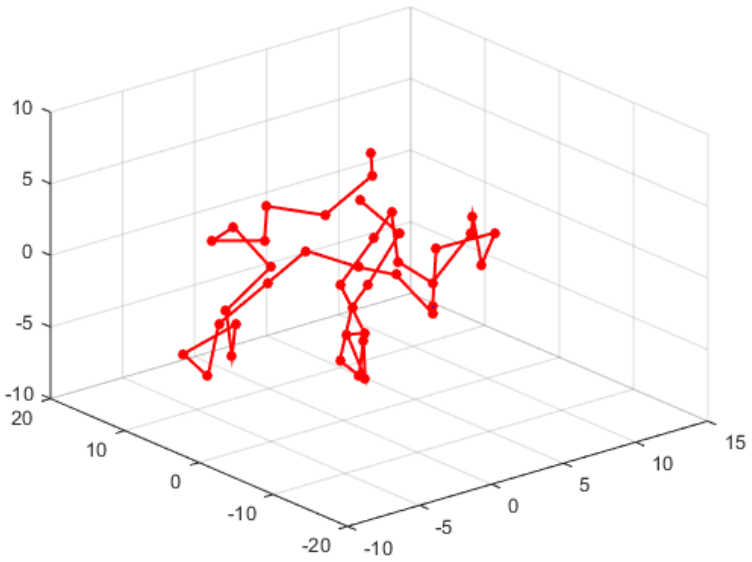


Figura 28: Sobreposição das estruturas original e gerada numericamente da proteína *1BQX* pelo Método Gauss-Newton, com erro $\rho = 10^{-4}$, realizada pelo Método baseado em Decomposição SVD.

Tabela 9: Resultado dos Testes Numéricos com erro $\rho = 10^{-5}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	50	$4.4889e-02$	37.3918	34	$2.4485e-02$	257.1028
1AQR	120	51	$7.5461e-05$	0.1758	32	$1.4174e-05$	0.0598
1BQX	231	78	$7.2708e-04$	0.8493	20	$8.6138e-05$	0.1899
1BSA	963	96	$1.5363e-02$	21.0314	29	$8.4201e-03$	27.5020
1F39	606	62	$2.1683e-02$	5.0729	21	$1.9174e-03$	4.2437
1FS3	372	49	$1.6311e-03$	1.4271	19	$3.8408e-04$	0.7477
1JK2	270	73	$7.4210e-04$	1.1021	28	$3.1770e-04$	0.4546
1MBN	459	107	$4.1756e-03$	4.7583	18	$6.7892e-04$	1.2932

Tabela 10: Resultado dos Testes Numéricos com erro $\rho = 10^{-4}$

Problema		BFGS			Gauss-Newton		
Proteína	*nvar*	*iter*	$f(x^*)$	$t(s)$	*iter*	$f(x^*)$	$t(s)$
1A9W	1716	50	$2.4533e-00$	37.2183	27	$2.4374e-00$	204.5073
1AQR	120	40	$1.5281e-03$	0.1369	32	$1.3205e-03$	0.0588
1BQX	231	36	$8.5581e-03$	0.3932	32	$8.2794e-03$	0.3108
1BSA	963	92	$8.4750e-01$	20.2114	32	$8.4707e-01$	30.2021
1F39	606	74	$1.9261e-01$	6.0414	23	$1.8900e-01$	4.6651
1FS3	372	48	$4.0969e-02$	1.4091	33	$3.8506e-02$	1.3415
1JK2	270	79	$3.3534e-02$	1.2000	20	$3.0474e-02$	0.3447
1MBN	459	73	$7.1870e-02$	3.2548	28	$6.8973e-02$	2.1296

Tabela 11: Resultados Computacionais do problema *Procrustes* envolvendo a Proteína *1BQX* com a solução do Método de Gauss-Newton, com erro $\rho = 10^{-4}$.

Procrustes via Decomposição em Valores Singulares

$$Q = \begin{bmatrix} 0.1976 & -0.8332 & 0.5164 \\ -0.9799 & -0.1833 & 0.0792 \\ 0.0287 & -0.5216 & -0.8527 \end{bmatrix}, \quad RMSD = 3.751825e-05.$$

Procrustes via Quatérnios Unitários

$$q = \begin{bmatrix} 0.2010 \\ -0.7473 \\ 0.6066 \end{bmatrix}, \quad RMSD = 3.751825e-05.$$

Fonte: Autoria própria.

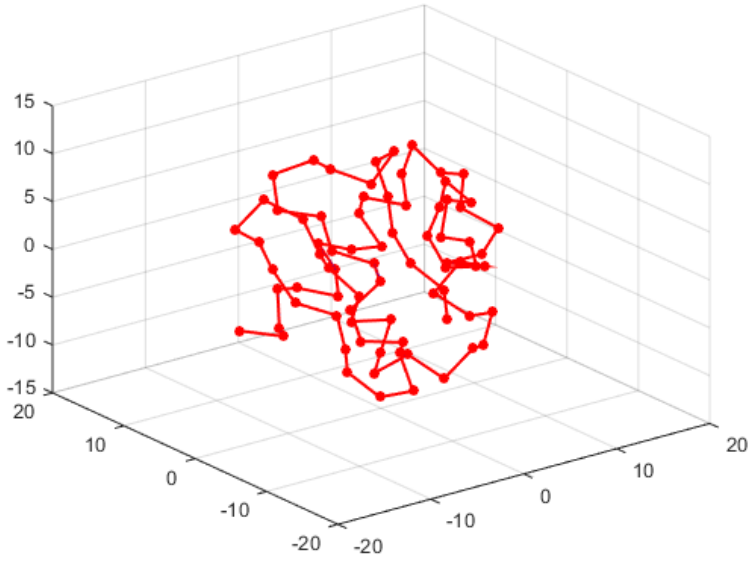


Figura 29: Sobreposição das estruturas original e gerada numericamente da proteína *1BQX* pelo Método Gauss-Newton, com erro $\rho = 10^{-4}$, realizada pelo Método baseado em Quatérnios Unitários.

Tabela 12: Resultados Computacionais do problema *Procrustes* envolvendo a Proteína *1BQX* com a solução do Método de Gauss-Newton, com erro $\rho = 10^{-4}$.

Procrustes via Decomposição em Valores Singulares
$Q = \begin{bmatrix} -0.5374 & -0.8379 & -0.0952 \\ 0.4405 & -0.1827 & -0.8790 \\ 0.7191 & -0.5143 & 0.4673 \end{bmatrix}, \quad RMSD = 2.416791e-05.$
Procrustes via Quatérnios Unitários
$q = \begin{bmatrix} 0.4322 \\ 0.2109 \\ -0.4710 \end{bmatrix}, \quad RMSD = 2.416791e-05.$

Fonte: Autoria própria.

7 CONCLUSÃO

Neste trabalho foi apresentado o Problema da Geometria de Distâncias aplicado em modelagem molecular de Proteínas, bem como os pré-requisitos necessários para a compreensão desse assunto. Além disso, ao ser feita a aplicação em Biologia Molecular, foi preciso um envolvimento com os conceitos da área da Biologia, gerando, desta forma, um momento muito oportuno de interdisciplinaridade curricular.

E, ainda, apresentamos uma formulação do Problema da Geometria de Distâncias, cuja resolução separamos em dois conjuntos de testes computacionais: a primeira é uma formulação simples do problema, cujas distâncias são exatas e são admitidas todas conhecidas; a segunda, para assimilar erros práticos, perturbamos todo o conjunto de distâncias com um número aleatório e consideramos conhecidas todas as distâncias.

Os resultados obtidos na primeira bateria de testes evidenciam o melhor desempenho, em partes, do método de Gauss-Newton, sobretudo no baixo número de iterações e a qualidade da solução, com um valor próximo de zero. Por outro lado, o método BFGS também resolveu todos os problemas e com menor tempo de execução, principalmente nos problemas com um grande número de variáveis.

Em relação aos resultados apresentados na segunda bateria de testes, quando adicionamos erros no conjunto de distâncias, podemos destacar que o método de Gauss-Newton não teve um comportamento satisfatório, mesmo adicionando erros bem pequenos (quando $\rho = 10^{-9}$), enquanto que o método BFGS apresentou resultados satisfatórios, similares as simulações numéricas realizadas na Seção 6.1. Vale destacar que o número de iterações e tempo de execução, em ambos os métodos, foram similares aos obtidos nas primeiras simulações.

Esse trabalho tem como perspectivas de continuidade o estudo da abordagem discreta do Problema da Geometria de Distâncias, com o intuito de analisar a possibilidade de mesclar ambas as abordagens para a resolução do problema. Além disso, pretende-se criar uma interface gráfica para a resolução do PGD no MATLAB, a fim de disponibilizar para a comunidade acadêmica.

REFERÊNCIAS

ALFAKIH, A. Y.; KHANDANI, A.; WOLKOWICZ, H. Solving euclidean distance matrix completion problems via semidefinite programming. **Computational optimization and applications**, Springer, v. 12, n. 1-3, p. 13–30, 1999.

ANDREANI, R. et al. Continuous optimization methods for structure alignments. **Mathematical Programming**, Springer, v. 112, n. 1, p. 93–124, 2008.

BERMAN, H. M. et al. The protein data bank. **Nucleic acids research**, Oxford Univ Press, v. 28, n. 1, p. 235–242, 2000.

BLUMENTHAL, L. M. **Theory and applications of distance geometry**. Oxford: Clarendon Press Oxford, 1953.

BOVET, D. P.; CRESCENZI, P.; BOVET, D. **Introduction to the Theory of Complexity**. [S.l.]: Prentice Hall London, 1994.

COELHO, F. U.; LOURENÇO, M. L. **Um Curso de Álgebra Linear**. São Paulo: Edusp, 2001.

COUTSIAS, E. A.; SEOK, C.; DILL, K. A. Using quaternions to calculate rmsd. **Journal of Computational Chemistry**, Wiley Online Library, v. 25, n. 15, p. 1849–1857, 2004.

CRIPPEN, G. M.; HAVEL, T. F. **Distance geometry and molecular conformation**. New York: Research Studies Press, 1988.

GIBRAT, J.-F.; MADEJ, T.; BRYANT, S. H. Surprising similarities in structure comparison. **Current opinion in structural biology**, Elsevier, v. 6, n. 3, p. 377–385, 1996.

GLUNT, W.; HAYDEN, T. L.; RAYDAN, M. Molecular conformations from distance matrices. **Journal of Computational Chemistry**, Wiley Online Library, v. 14, n. 1, p. 114–120, 1993.

GOLUB, G. H.; LOAN, C. F. V. **Matrix computations**. Baltimore: JHU Press, 2012.

HAMILTON, W. R. On quaternions; or on a new system of imaginaries in algebra. **Philosophical Magazine Series 3**, v. 25, n. 169, p. 489–495, 1844.

HASEGAWA, H.; HOLM, L. Advances and pitfalls of protein structural alignment. **Current opinion in structural biology**, Elsevier, v. 19, n. 3, p. 341–348, 2009.

HAVEL, T.; WÜTHRICH, K. A distance geometry program for determining the structures of small proteins and other macromolecules from nuclear magnetic resonance measurements of intramolecular 1h-1h proximities in solution. **Bulletin of mathematical biology**, Elsevier, v. 46, n. 4, p. 673–698, 1984.

HAVEL, T. F.; WÜTHRICH, K. An evaluation of the combined use of nuclear magnetic resonance and distance geometry for the determination of protein conformations in solution. **Journal of molecular biology**, Elsevier, v. 182, n. 2, p. 281–294, 1985.

HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. **JOSA A**, Optical Society of America, v. 4, n. 4, p. 629–642, 1987.

HUANG, H.-X.; LIANG, Z.-A.; PARDALOS, P. M. Some properties for the euclidean distance matrix and positive semidefinite matrix completion problems. **Journal of Global Optimization**, Springer, v. 25, n. 1, p. 3–21, 2003.

KABSCH, W. A solution for the best rotation to relate two sets of vectors. **Acta Crystallographica Section A**, International Union of Crystallography, v. 32, n. 5, p. 922–923, 1976.

KABSCH, W. A discussion of the solution for the best rotation to relate two sets of vectors. **Acta Crystallographica Section A**, International Union of Crystallography, v. 34, n. 5, p. 827–828, 1978.

LAVOR, C.; LIBERTI, L. **Um Convite à Geometria de Distâncias**. São Carlos: SBMAC, 2014.

LAVOR, C.; LIBERTI, L.; MACULAN, N. An overview of distinct approaches for the molecular distance geometry problem. **Encyclopedia of Optimization, 2nd edn, Springer (to appear)**, 2008.

LAVOR, C. et al. On the computation of protein backbones by using artificial backbones of hydrogens. **Journal of Global Optimization**, Springer, v. 50, n. 2, p. 329–344, 2011.

LEEUW, J. D. Differentiability of kruskal's stress at a local minimum. **Psychometrika**, Springer, v. 49, n. 1, p. 111–113, 1984.

LESK, A. M. **Introdução à bioinformática**. Porto Alegre: Artmed, 2008.

LIBERTI, L. et al. Euclidean distance geometry and applications. **Siam Review**, SIAM, v. 56, n. 1, p. 3–69, 2014.

LIBERTI, L. et al. On the number of solutions of the discretizable molecular distance geometry problem. In: SPRINGER. **COCOA**. [S.l.], 2011. p. 322–342.

LIMA, E. L. **Analise Real, volume 2: Funções de n Variáveis**. Rio de Janeiro: Impa, 2013.

LIMA, R.; LAVOR, C.; MARTÍNEZ, J. M. Solving molecular distance geometry problems with uncertain data. **Mecánica Computacional**, Asociación Argentina de Mecánica Computacional, XXXI, p. 2757–2763, 2012.

LIMA, R. S. **Reconstrução e Classificação de Estruturas Espaciais via Otimização Contínua: Ênfase em Proteínas**. Tese (Doutorado) — Unicamp, 2012.

LUENBERGER, D. G.; YE, Y. **Linear and nonlinear programming**. New York: Springer, 2015.

MCLACHLAN, A. D. A mathematical procedure for superimposing atomic coordinates of proteins. **Acta Crystallographica Section A**, International Union of Crystallography, v. 28, n. 6, p. 656–657, 1972.

MENGER, K. Untersuchungen über allgemeine metrik. **Mathematische Annalen**, Springer, v. 100, n. 1, p. 75–163, 1928.

MEYER, C. D. **Matrix Analysis and Applied Linear Algebra**. Philadelphia: Siam, 2000.

MILIES, C. P. Breve história da álgebra abstrata. **II Bienal da Sociedade Brasileira de Matemática**, 2004.

NOBRE, S. G. G. **A Decomposição em Valores Singulares e suas Aplicações**. Dissertação (Mestrado) — Universidade do Algarve, 2007.

NOCEDAL, J.; WRIGHT, S. **Numerical Optimization**. New York: Springer Science & Business Media, 2006.

POOLE, D. **Álgebra Linear**. São Paulo: Cengage Learning, 2014.

RIBEIRO, A. A.; KARAS, E. W. Otimização contínua: aspectos teóricos e computacionais. **Cengage Learning**, 2013.

SCHOENBERG, I. J. Remarks to maurice frechet's article "sur la definition axiomatique d'une classe d'espace distances vectoriellement applicable sur l'espace de hilbert". **Annals of Mathematics**, JSTOR, p. 724–732, 1935.

SEHNAL, D. Algorithms for comparing molecule conformations. **Bachelor thesis, Masaryk University Brno**, 2008.

SIT, A. **Solving distance geometry problems for protein structure determination**. Tese (Doutorado) — Iowa State University, 2010.

TROSSET, M. W. Distance matrix completion by numerical optimization. **Computational Optimization and Applications**, Springer, v. 17, n. 1, p. 11–22, 2000.

YEMINI, Y. The positioning problem-a draft of an intermediate summary. In: **Proceedings of the Conference on Distributed Sensor Networks**. Pittsburgh: Carnegie-Mellon University, 1978. p. 137–145.

APÊNDICE A – CONCEITOS PRELIMINARES

Neste Apêndice faremos uma breve revisão dos conceitos matemáticos mais relevantes que serão tratados neste Trabalho de Conclusão de Curso. Nessa revisão da literatura, assumiremos que o leitor está familiarizado com a linguagem matemática (lógica, conjuntos, etc). Os conceitos e as discussões são essenciais para que o leitor compreenda sem dificuldades os resultados mais avançados que serão tratados no decorrer do trabalho

Para esta revisão foram selecionados os tópicos de Álgebra Linear, normas e fatoração de matrizes. As notações, definições e os resultados, trabalhados nesse Capítulo, podem ser encontrados, em sua grande parte, em (GOLUB; LOAN, 2012), (MEYER, 2000), (COELHO; LOURENÇO, 2001) e (POOLE, 2014).

A.1 CONCEITOS BÁSICOS DE ÁLGEBRA LINEAR

A abordagem utilizada neste trabalho considera o isomorfismo entre o espaço das transformações lineares e o espaço das matrizes reais de ordem $m \times n$. Por esse motivo, todos os resultados são construídos por meio da aplicação de matrizes.

Definição A.1.1*Seja $\mathbb{R}$ o conjunto dos números reais. O Espaço Vetorial das matrizes reais de dimensão $m \times n$ é $\mathbb{R}^{m\times n}$, de modo que:*

$$A \in \mathbb{R}^{m\times n} \Leftrightarrow A = (a_{ij}) = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}, a_{ij} \in \mathbb{R}.$$

Observação A.1.2*Se uma letra maiúscula for utilizada para se referir a uma matriz (ou seja, A, B, Δ), então a letra minúscula correspondente com o subíndice ij se refere a entrada (i, j) (ou seja, a_{ij}, b_{ij}, δ_{ij}). Algumas vezes designaremos aos elementos de uma matriz com a notação $[A]_{ij}$ ou $A(i, j)$.*

Para prosseguirmos com as propriedades e proposições que as matrizes nos oferecem, precisamos primeiramente introduzir as operações básicas de matrizes.

Definição A.1.3*A operação transposição é uma função* $T : \mathbb{R}^{m\times n} \to \mathbb{R}^{n\times m}$ *tal que:*

$$C = A^T \Rightarrow c_{ij} = a_{ji}.$$

A matriz A^T *é dita a matriz transposta de A.*

Definição A.1.4*A operação adição é uma função* $A : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ *de modo que:*

$$C = A + B \Rightarrow c_{ij} = a_{ij} + b_{ij}.$$

Definição A.1.5*A operação multiplicação escalar-matriz é uma função* $E : \mathbb{R} \times \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$*:*

$$C = \alpha A \Rightarrow c_{ij} = \alpha a_{ij}.$$

Além das matrizes, os vetores sÃ£o importantes para os diversos resultados de Álgebra Linear.

Definição A.1.6*O Espaço Vetorial dos vetores* $n-$*dimensionais* $\mathbb{R}^n$ *é tal que:*

$$x \in \mathbb{R}^n \Leftrightarrow x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix},\ x_i \in \mathbb{R}.$$

Observação A.1.7*Nos referimos a* x_i *como o i-ésimo componente de x. Dependendo do contexto também podemos utilizar a notação* $[x]_i$ *ou* $x(i)$.

Dessa forma, bem como no caso das matrizes, precisamos definir as operações elementares dos vetores:

Definição A.1.8*Sejam* $a \in \mathbb{R}$, $x \in \mathbb{R}^n$ *e* $y \in \mathbb{R}^n$*. As principais operações elementares dos vetores são:*

•*Multiplicação escalar-vetor:*

$$z = ax \Rightarrow z_i = ax_i.$$

•*Soma vetorial:*

$$z = x + y \Rightarrow z_i = x_i + y_i.$$

•*Produto escalar (ou produto interno)*

$$z = x^T y \Rightarrow z = \sum_{i=1}^{n} x_i y_i,$$

onde $x^T \in \mathbb{R}^{1\times n}$ é o vetor transposto de x. Nesse caso, $x \in \mathbb{R}^n$ (ou ainda $\mathbb{R}^{n\times 1}$) indica que x é um vetor coluna, enquanto que $x^T \in \mathbb{R}^{1\times n}$ é um vetor linha.

Observação A.1.9*Uma maneira útil de especificar uma linha ou coluna de uma matriz é utilizando a notação "colon". Se $A \in \mathbb{R}^{m\times n}$, então $A(k,:)$ representa a k-ésima linha, isto é,*

$$A(k,:) = [a_{k1}, \ldots, a_{kn}].$$

A k-ésima coluna é especificada por

$$A(:,k) = \begin{bmatrix} a_{1k} \\ \vdots \\ a_{mk} \end{bmatrix}.$$

Definição A.1.10*Uma partição em linha de uma matriz $A \in \mathbb{R}^{m\times n}$ é dada por*

$$A = \begin{bmatrix} l_1^T \\ \vdots \\ l_m^T \end{bmatrix},\ l_k \in \mathbb{R}^n.$$

Definição A.1.11*Uma partição em colunas de uma matriz $A \in \mathbb{R}^{m\times n}$ é dada por*

$$A = [\, c_1 \mid \ldots \mid c_n \,],\ c_k \in \mathbb{R}^m.$$

Definição A.1.12*O produto externo de dois vetores $x \in \mathbb{R}^m$ e $y \in \mathbb{R}^n$ é dado por:*

$$A = xy^T \Rightarrow a_{ij} = x_i y_j.$$

Observação A.1.13*O produto externo entre um vetor $x \in \mathbb{R}^m$ e outro $y \in \mathbb{R}^n$ resulta em uma matriz $A \in \mathbb{R}^{m\times n}$.*

Definição A.1.14*A multiplicação matriz-vetor entre uma matriz $A \in \mathbb{R}^{m\times n}$ e um vetor $x \in \mathbb{R}^n$ é dado por:*

$$y = Ax \Rightarrow y_i = \sum_{j=1}^{n} a_{ij} x_j,\ \ para\ \ i = 1, \ldots, m.$$

Observação A.1.15*A multiplicação entre uma matriz $A \in \mathbb{R}^{m\times n}$ e um vetor $x \in \mathbb{R}^n$ retorna um vetor $y \in \mathbb{R}^m$.*

Agora chegamos no ponto principal da nossa discussão acerca das operações elementares de matrizes e vetores: a multplicação entre matrizes.

Definição A.1.16*A multiplicação matriz-matriz entre duas matrizes $A \in \mathbb{R}^{m\times r}$ e $B \in \mathbb{R}^{r\times n}$ é dada por:*

$$C = AB \Rightarrow c_{ij} = \sum_{k=1}^{r} a_{ik} b_{kj}.$$

Observação A.1.17*A multiplicação entre duas matrizes $A \in \mathbb{R}^{m\times r}$ e $B \in \mathbb{R}^{r\times n}$ retorna uma matriz $C \in \mathbb{R}^{m\times n}$.*

Exemplo A.1.18*Note que a multiplicação matriz-matriz pode ser vista como uma coleção de multplicações matriz-vetor, como mostra o exemplo a seguir:*

$$\begin{aligned}
\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} &= \begin{bmatrix} 1.5+2.7 & 1.6+2.8 \\ 3.5+4.7 & 3.6+4.8 \end{bmatrix} \\
&= \left[\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \end{bmatrix}, \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 6 \\ 8 \end{bmatrix} \right] \\
&= \begin{bmatrix} 19 & 22 \\ 42 & 50 \end{bmatrix}
\end{aligned}$$

Essa definição da multiplicação matriz-matriz abre espaço para outras formulações. Por meio do produto escalar de vetores podemos reescrever a Definição A.1.16 da seguinte forma:

Sejam $A \in \mathbb{R}^{m\times r}$ e $B \in \mathbb{R}^{r\times n}$ descritas nas seguintes formas particionadas:

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}, \; a_k \in \mathbb{R}^r, \text{ e } B = [\, b_1 \mid \ldots \mid b_n \,], \; b_k \in \mathbb{R}^r.$$

Então, a matriz $C \in \mathbb{R}^{m\times n}$ resultante da multiplicação entre as matrizes A e B é dada por:

$$C = AB \Rightarrow c_{ij} = a_i^T b_j.$$

Outra possibilidade é utilizar o produto externo de vetores:

Sejam $A \in \mathbb{R}^{m\times r}$ e $B \in \mathbb{R}^{r\times n}$ tais que,

$$A = [\, a_1 \mid \ldots \mid a_r \,],\ a_k \in \mathbb{R}^m, \text{ e } A = \begin{bmatrix} b_1^T \\ \vdots \\ b_r^T \end{bmatrix},\ b_k \in \mathbb{R}^n.$$

Então, a matriz $C \in \mathbb{R}^{m\times n}$ resultante da multiplicação entre as matrizes A e B é dada por:

$$C = AB \Rightarrow C = \sum_{k=1}^{r} a_k b_k^T.$$

Agora que definimos as principais operações de matrizes e vetores, falta ainda mostrar alguns exemplos de matrizes e vetores, que serão utilizadas ao decorrer do texto.

Definição A.1.19*(Matriz Simétrica): Seja $A = (a_{ij}) \in \mathbb{R}^{n\times n}$. A é dita matriz simétrica quando $A = A^T$, isto é, $a_{ij} = a_{ji}$, para todo $i, j = 1, 2, \ldots, n$.*

Definição A.1.20*(Matriz Identidade): A matriz $I_n \in \mathbb{R}^{n\times n}$ unitária em sua diagonal e nula nas outras entradas*

$$I_n = \begin{bmatrix} 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix}$$

é chamada de matriz identidade de ordem n. Para cada matriz $A \in \mathbb{R}^{m\times n}$, temos

$$AI_n = A \ \ e \ \ I_m A = A.$$

Observação A.1.21*O subíndice em I_n será omitido quando a dimensão for óbvia visto o contexto.*

Definição A.1.22*O vetor com entradas unitárias, denotado por $\mathbb{1}$, é definido como*

$$\mathbb{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^n.$$

A dimensão do vetor $\mathbb{1}$ *será compreendida visto o contexto do cálculo que estiver relacionada.*

Definição A.1.23*(Traço de Matriz): O traço de uma matriz* $A \in \mathbb{R}^{n\times n}$*, denotado por* $Tr(A)$*, é a soma dos elementos da diagonal da matriz A, isto é,*

$$Tr(A) = \sum_{i=1}^{n} a_{ii}.$$

Sejam $A, B \in \mathbb{R}^{n\times n}$ e $k \in \mathbb{R}$. As principais propriedades do operador traço são:

•O traço da soma é a soma dos traços:

$$Tr(A+B) = Tr(A) + Tr(B).$$

•O traço da multiplicação é o traço da multiplicação reversa:

$$Tr(AB) = Tr(BA).$$

•O traço do transposto é o traço da própria matriz:

$$Tr(A^T) = Tr(A).$$

•O traço da multiplicação por escalar é a multiplicação do escalar pelo traço da própria matriz:

$$Tr(kA) = kTr(A).$$

Definição A.1.24*(Diagonal de Matriz): O vetor diagonal de uma matriz* $A \in \mathbb{R}^{n\times n}$*, denotado por* $diag(A)$*, é tal que seus elementos são as entradas da diagonal principal da matriz A, isto é,*

$$diag(A) = \begin{bmatrix} a_{11} \\ a_{22} \\ \vdots \\ a_{nn} \end{bmatrix} \in \mathbb{R}^n.$$

Definição A.1.25*(Matriz Triangular): Seja* $A \in \mathbb{R}^{n\times n}$*. A é dita matriz triangular inferior quando seus elementos acima da diagonal são nulos, isto é,* $a_{ij} = 0,\ \forall\ i < j$*. De forma similiar, A é chamada de matriz triangular superior quando seus elementos abaixo da diagonal são nulos, isto é,* $a_{ij} = 0,\ \forall\ i > j$*.*

Definição A.1.26*(Matriz Diagonal): Uma matriz $A \in \mathbb{R}^{n\times n}$ é dita diagonal quando é, ao mesmo tempo, triangular inferior e superior, isto é, todos seus elementos fora da diagonal são nulos:*

$$A = \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{nn} \end{bmatrix}.$$

Definição A.1.27*(Matriz Inversa): Sejam $A, X \in \mathbb{R}^{n\times n}$. Se essas duas matrizes satisfazem $AX = XA = I$, então X é a matriz inversa de A e a denotamos por A^{-1}.*

A seguir, destacamos algumas propriedades relacionadas a matriz inversa.

•A inversa do produto é o produto reverso das inversas:

$$(AB)^{-1} = B^{-1}A^{-1}. \tag{A.1.1}$$

•A transposta da inversa é a inversa da transposta:

$$(A^{-1})^T = (A^T)^{-1} \equiv A^{-T}.$$

A propriedade (A.1.1) não é tão difícil de entender, porém a inversa de uma soma de matrizes é mais complicada. Para começar, $(A+B)^{-1}$ pode não existir mesmo se A^{-1} e B^{-1} existirem. Existe uma fórmula geral útil para a soma $(A+B)^{-1}$, sobretudo quando esta pode ser descrita como $(A+cd^T)$, onde $c,d \in \mathbb{R}^n$ tais que $1+d^Tc \neq 0$. A inversa dessa soma será utilizada no Capítulo (**??**) para a construção de um método quase-Newton. A inversa dessa soma é chamada de Fórmula Sherman-Morrison:

Definição A.1.28*(Fórmula Sherman-Morrison): Se $A \in \mathbb{R}^{n\times n}$ é não-singular e $c,d \in \mathbb{R}^n$ tais que $1+d^Tc \neq 0$, então a soma $(A+cd^T)$ é não-singular e*

$$\left(A+cd^T\right)^{-1} = A^{-1} - \frac{A^{-1}cd^TA^{-1}}{1+d^TA^{-1}c}. \tag{A.1.2}$$

Para verificar que a igualdade (A.1.2) é verdadeira, vamos utilizar a definição de matriz inversa (isto é, a matriz Y é a inversa de X se, e só se, $XY = YX = I$):

$$
\begin{aligned}
\left(A+cd^T\right)^{-1}\left(A^{-1}-\frac{A^{-1}cd^TA^{-1}}{1+d^TA^{-1}c}\right) &= AA^{-1}+cd^TA^{-1}-\frac{AA^{-1}cd^TA^{-1}+cd^TA^{-1}cd^TA^{-1}}{1+d^TA^{-1}c} \\
&= I+cd^TA^{-1}-\frac{cd^TA^{-1}+cd^TA^{-1}cd^TA^{-1}}{1+d^TA^{-1}c} \\
&= I+cd^TA^{-1}-\frac{c\left(1+d^TA^{-1}c\right)d^TA^{-1}}{1+d^TA^{-1}c} \\
&= I+cd^TA^{-1}-cd^TA^{-1}, \text{ pois } 1+d^TA^{-1}c\in\mathbb{R} \\
&= I.
\end{aligned}
$$

$$
\begin{aligned}
\left(A^{-1}-\frac{A^{-1}cd^TA^{-1}}{1+d^TA^{-1}c}\right)\left(A+cd^T\right)^{-1} &= A^{-1}A+A^{-1}cd^T-\frac{A^{-1}cd^TA^{-1}A+A^{-1}cd^TA^{-1}cd^T}{1+d^TA^{-1}c} \\
&= I+A^{-1}cd^T-\frac{A^{-1}cd^T+A^{-1}cd^TA^{-1}cd^T}{1+d^TA^{-1}c} \\
&= I+A^{-1}cd^T-\frac{A^{-1}c\left(1+d^TA^{-1}c\right)d^T}{1+d^TA^{-1}c} \\
&= I+A^{-1}cd^T-A^{-1}cd^T, \text{ pois } 1+d^TA^{-1}c\in\mathbb{R} \\
&= I.
\end{aligned}
$$

Definição A.1.29*(Matrizes Definidas): Seja $A\in\mathbb{R}^{n\times n}$ uma matriz. Dizemos que A é*

- *Definida Positiva quando $x^TAx>0$, para todo $x\in\mathbb{R}^n\backslash\{0\}$.*
- *Semi-Definida Positiva quando $x^TAx\geq 0$, para todo $x\in\mathbb{R}^n\backslash\{0\}$.*
- *Definida Negativa quando $x^TAx<0$, para todo $x\in\mathbb{R}^n\backslash\{0\}$.*
- *Semi-Definida Negativa quando $x^TAx\leq 0$, para todo $x\in\mathbb{R}^n\backslash\{0\}$.*
- *Indefinida quando existirem $x,y\in\mathbb{R}^n$ tais que $x^TAx>0$ e $y^TAy<0$.*

Definição A.1.30*(Matriz Ortogonal): Uma matriz $Q\in\mathbb{R}^{n\times n}$ é dita ortogonal se $Q^TQ=I$.*

Observação A.1.31*As matrizes ortogonais estão relacionadas as transformações que preservam distâncias, em outras palavras, relacionadas a rotação e reflexão.*

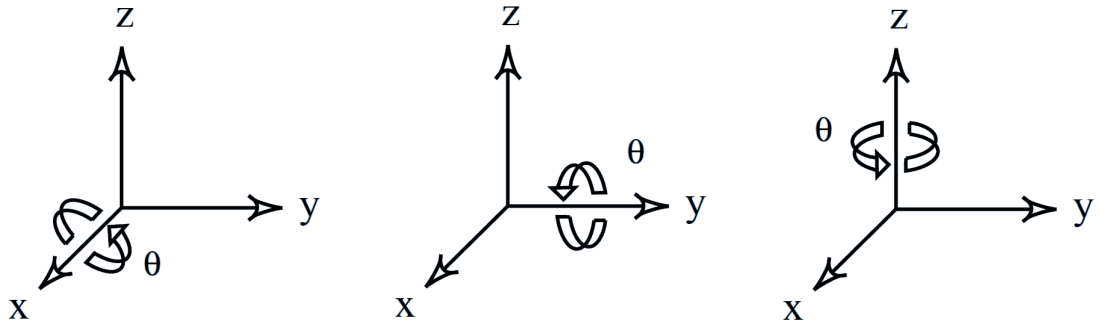
Definição A.1.32*(Matriz de Rotação): Um vetor $u\in\mathbb{R}^3$ pode ser rotacionado no sentido anti-horário através de um ângulo θ em torno de um eixo de coordenadas por meio de uma multiplicação*

$P_\star u$, onde $P_\star$ é uma matriz ortogonal como descrito a seguir:

$$P_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\theta) & -sen(\theta) \\ 0 & sin(\theta) & cos(\theta) \end{bmatrix}. \quad \text{(A.1.3)}$$

$$P_y = \begin{bmatrix} cos(\theta) & 0 & sen(\theta) \\ 0 & 1 & 0 \\ -sen(\theta) & 0 & cos(\theta) \end{bmatrix}. \quad \text{(A.1.4)}$$

$$P_z = \begin{bmatrix} cos(\theta) & -sen(\theta) & 0 \\ sen(\theta) & cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad \text{(A.1.5)}$$



(a) Rotação A.1.3 - Eixo-x (b) Rotação A.1.4 - Eixo-y (c) Rotação A.1.5 - Eixo-z

Figura 30: Rotações em torno do Eixo-x, Eixo-y e Eixo-z.

Fonte: (MEYER, 2000)

Estando definidas as operações elementares de vetores e matrizes, propriedades e seus principais casos, podemos, então, discutir algumas propriedades e resultados de Álgebra Linear que serão importantes para o desenvolvimento deste trabalho.

Definição A.1.33*(Conjunto LI): Um conjunto de vetores $\{a_1,\ldots,a_m\} \in \mathbb{R}^n$ é Linearmente Independente (L.I.) se*

$$\sum_{j=1}^{n} \alpha_j a_j = 0 \Rightarrow \alpha_j = 0,\ \forall j = 1,\ldots,m.$$

Definição A.1.34*(Posto): O posto de uma matriz $A \in \mathbb{R}^{m\times n}$, denotado por $posto(A)$, é o número de linhas ou colunas linearmente independentes de A, ou seja, $posto(A) \leq min\{m,n\}$. Uma matriz $A \in \mathbb{R}^{m\times n}$ é dita ser de posto completo se o seu posto for igual a $min\{m,n\}$.*

Definição A.1.35*(Combinação Linear): Um vetor $v \in \mathbb{R}^n$ é uma combinação linear dos vetores $v_1, \ldots, v_n \in \mathbb{R}^n$ se existiram escalares $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$ tais que*

$$v = \alpha_1 v_1 + \ldots + \alpha_n v_n = \sum_{i=1}^{n} \alpha_i v_i.$$

Definição A.1.36*(Conjunto gerador): Seja B um subconjunto de $\mathbb{R}^n$. Dizemos que B é um conjunto gerador de $\mathbb{R}^n$ (ou que B gera $\mathbb{R}^n$) se todo elemento de $\mathbb{R}^n$ for uma combinação linear de um número finito de elementos de B.*

Definição A.1.37*(Base): Dizemos que um subconjunto B de $\mathbb{R}^n$ é uma base de $\mathbb{R}^n$ se*

1.B for um conjunto gerador de $\mathbb{R}^n$; e

2.B for um conjunto linearmente independente.

Exemplo A.1.38*A base mais usual de $\mathbb{R}^n$ é a base canônica*

$$B = \{e_i, i = 1,2,\ldots,n\} = \{(1,0,\ldots,0),(0,1,0,\ldots,0),\ldots,(0,0,\ldots,1)\}.$$

Cada vetor $e_i \in \mathbb{R}^n$ é nulo a não ser pela i-ésima entrada que é unitária.

Definição A.1.39*O Delta de Kronecker, denotado por δ_{ij}, é definido por*

$$\delta_{ij} = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases}$$

Definição A.1.40*(Conjunto Ortogonal): Um conjunto de vetores $\{x_1, \ldots, x_p\}$ do $\mathbb{R}^n$ é ortogonal se $x_i^T x_j = 0$ para quaisquer $i \neq j$, e será ortonormal se $x_i^T x_j = \delta_{ij}$.*

Definição A.1.41*Se $A = (a) \in \mathbb{R}^{1\times 1}$, então seu determinante é dado por $det(A) = a$. O determinante de $A \in \mathbb{R}^{n\times n}$ é definido em termos de ordem $(n-1)$ determinantes:*

$$det(A) = \sum_{j=1}^{n} (-1)^{j+1} a_{1j} det(A_{1j}),$$

onde A_{1j} é uma matriz de dimensão $(n-1)\times(n-1)$ obtida deletando a primeira linha e a $j-$ésima coluna de A.

Sejam $A, B \in \mathbb{R}^{n\times n}$ e $c \in \mathbb{R}$. O determinante de uma matriz tem uma enorme importância em diversos resultados de Álgebra Linear e, por esta razão, apresentamos a seguir suas principais propriedades.

•O determinante da multiplicação é a multiplicação dos determinantes

$$det(AB) = det(A)det(B).$$

•O determinante de uma matriz transposta é o determinante da própria matriz

$$det(A^T) = det(A).$$

•O determinante da multiplicação de um escalar por uma matriz é a multiplicação do escalar elevado à dimensão da matriz pelo determinante da matriz

$$det(cA) = c^n det(A).$$

•O determinante de uma matriz ser não-nulo é uma condição necessária e suficiente para que a matriz seja inversível

$$det(A) \neq 0 \Leftrightarrow \exists\, A^{-1} \text{ tal que } AA^{-1} = I.$$

•O determinante de uma matriz $A \in \mathbb{R}^{n\times n}$ ortogonal é dado por $det(A) = \pm 1$. Em particular quando A for uma matriz de rotação o seu determinante será $det(A) = 1$ e será $det(A) = -1$ quando A for uma matriz de reflexão.

Definição A.1.42*Os autovalores de uma matriz $A \in \mathbb{C}^{n\times n}$ são as raízes do polinômio característico*

$$p(x) = det(A - xI).$$

Observação A.1.43*Todas as matrizes em $\mathbb{C}^{n\times n}$ possuem n autovalores. Denotamos o conjunto dos autovalores de A por*

$$\lambda(A) = \{x : det(A - xI) = 0\}.$$

Se os autovalores de A são reais, podemos indexá-los de forma descrescente:

$$\lambda_n(A) \leq \ldots \leq \lambda_2(A) \leq \lambda_1(A).$$

Nese caso, usaremos as notações $\lambda_{max}(A)$ e $\lambda_{min}(A)$ como sendo $\lambda_1(A)$ e $\lambda_n(A)$ respectivamente.

Definição A.1.44*Se $\lambda \in \lambda(A)$, então existe um vetor x associado a λ tal que $Ax = \lambda x$. Tal vetor é chamado de autovetor de A associado ao autovalor λ.*

Uma importante relação entre as matrizes definidas e seus respectivos autovalores está expressa no seguinte Teorema, cuja demonstração pode ser vista em (POOLE, 2014). Este resultado será muito útil quando precisarmos verificar a positiva-definição de uma matriz nos Capítulos a seguir.

Teorema A.1.45*Seja $A \in \mathbb{R}^{n\times n}$ uma matriz simétrica. A matriz A é:*

- *Definida Positiva se, e somente se, todos os seus autovalores são positivos.*
- *Semi-Definida Positiva se, e somente se, todos os seus autovalores são não-negativos.*
- *Definida Negativa se, e somente se, todos os seus autovalores são negativos.*
- *Semi-Definida Negativa se, e somente se, todos os seus autovalores são não-positivos.*
- *Indefinida se, e somente se, possuir autovalores positivos e negativos.*

Definição A.1.46*(Matrizes Semelhantes): Sejam $A, B \in \mathbb{R}^{n\times n}$. Dizemos que A é semelhante a B se existir uma matriz $P \in \mathbb{R}^{n\times n}$ invertível tal que $P^{-1}AP = B$. Se A é semelhante a B escrevemos $A \sim B$.*

Definição A.1.47*(Matriz Diagonalizável): Uma matriz $A \in \mathbb{R}^{n\times n}$ é diagonalizável se existe uma matriz diagonal $D \in \mathbb{R}^{n\times n}$ tal que A é semelhante a D, isto é, existe uma matriz $P \in \mathbb{R}^{n\times n}$ invertível tal que $P^{-1}AP = D$.*

Definição A.1.48*(Matriz Ortogonalmente Diagonalizável): Uma matriz $A \in \mathbb{R}^{n\times n}$ é ortogonalmente diagonalizável se existem uma matriz ortogonal Q e uma matriz diagonal D tais que $Q^T AQ = D$.*

Teorema A.1.49*(Teorema Espectral): Seja $A \in \mathbb{R}^{n\times n}$. Então A é simétrica se, e somente se, A for ortogonalmente diagonalizável.*

Observação A.1.50*Dizemos que uma matriz $A \in \mathbb{R}^{n\times n}$ admite Decomposição Espectral quando a hipótese necessária do Teorema Espectral A.1.49 é satisfeita, isto é, quando A é simétrica. Nesse sentido, sua Decomposição Espectral é $A = Q^T DQ$, onde D é uma matriz diagonal e Q é uma matriz ortogonal.*

A.2 NORMA DE VETORES

Definição A.2.1 *Uma norma de vetores do $\mathbb{R}^n$ é uma função $f : \mathbb{R}^n \to \mathbb{R}$ que satisfaz as seguintes propriedades:*

1. $f(x) \geq 0$, $x \in \mathbb{R}^n$, *(*$f(x) = 0$ *se, e só se,* $x = 0$*).*

2. $f(x+y) \leq f(x) + f(y)$, $x, y \in \mathbb{R}^n$.

3. $f(\alpha x) = |\alpha| f(x)$, $\alpha \in \mathbb{R}$, $x \in \mathbb{R}^n$.

Observação A.2.2 *Denotamos tal função por:* $f(x) = \|x\|$. *Os subíndices nas barras duplas são usadas para diferenciar as várias normas.*

Definição A.2.3 *Uma classe útil de normas de vetores são as normas-p, definidas por:*

$$\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{\frac{1}{p}},\ p \geq 1.$$

As principais normas são: A norma-1, norma-2 (também conhecida como Norma Euclidiana) e norma-∞ são:

- Norma-1:

$$\|x\|_1 = |x_1| + \ldots + |x_n|.$$

- Norma-2:

$$\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2 + \ldots + |x_n|^2}$$

- Norma-∞:

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

Observação A.2.4 *Um vetor unitário em relação a norma* $\|\cdot\|$ *é um vetor x que satisfaz* $\|x\| = 1$.

Observação A.2.5 *Frequentemente, omitiremos o subíndice quando estivermos desenvolvendo algum resultado e, nesse caso, estaremos nos referimos à Norma Euclidiana* $\|\cdot\|_2$.

Exemplo A.2.6 *Prove que se* $x \in \mathbb{R}^n$, *então*

$$\lim_{p \to \infty} \|x\|_p = \|x\|_\infty.$$

Demonstração: *Seja $x \in \mathbb{R}^n$. Assim, temos dois casos: se x é nulo ou se x não é nulo (isto é, possui ao menos um elemento diferente de zero). No primeiro caso, o resultado é obtido trivialmente. Agora para x não-nulo, temos:*

$$\|x\|_p = (|x_1|^p + \ldots + |x_n|^p)^{\frac{1}{p}}.$$

Sem perda de generalidade, tome $x_1 = max\{|x_i|,\ i = 1,2,\ldots,n\}$. Como $x_1 > x_i$, $\forall\, i = 2,3,\ldots,n$, então temos que:

$$\left|\frac{x_i}{x_1}\right|^p < 1$$

para todo $i = 2,3,\ldots,n$. Dessa forma, quando $p \to \infty$, temos

$$\begin{aligned}
\lim_{p\to\infty} \|x\|_p &= \lim_{p\to\infty} (|x_1|^p + \ldots + |x_n|^p)^{\frac{1}{p}} \\
&= \lim_{p\to\infty} \left(|x_1|^p \left(1 + \frac{|x_2|^p}{|x_1|^p} + \ldots + \frac{|x_n|^p}{|x_1|^p} \right) \right)^{\frac{1}{p}} \\
&= \lim_{p\to\infty} |x_1| \left(1 + \frac{|x_2|^p}{|x_1|^p} + \ldots + \frac{|x_n|^p}{|x_1|^p} \right)^{\frac{1}{p}} \\
&\underset{*}{=} \lim_{p\to\infty} |x_1| = |x_1| = max\{|x_i|\}.
\end{aligned}$$

(*) Note que $\lim_{p\to\infty} \frac{|x_k|^p}{|x_1|^p} = 0$. Logo $\lim_{p\to\infty} \left(1 + \frac{|x_2|^p}{|x_1|^p} + \ldots + \frac{|x_n|^p}{|x_1|^p} \right)^{\frac{1}{p}} = 1$.

Portanto, $\lim_{p\to\infty} \|x\|_p = \|x\|_\infty$. ■

A seguir, apresentamos algumas propriedades importantes das normas de vetores.

Definição A.2.7*A Desigualdade de Hölder para dois vetores $x \in \mathbb{R}^m$ e $y \in \mathbb{R}^n$ é dada por:*

$$|x^T y| \leq \|x\|_p \|y\|_q, \text{ onde } \frac{1}{p} + \frac{1}{q} = 1.$$

Um caso particular é a Desigualdade de Cauchy-Schwarz:

$$|x^T y| \leq \|x\|_2 \|y\|_2.$$

Vale ressaltar que as normas do $\mathbb{R}^n$ são equivalentes, isto é, se $\|\cdot\|_\alpha$ e $\|\cdot\|_\beta$ são normas do $\mathbb{R}^n$, então existem constantes positivas c_1 e c_2 tais que

$$c_1 \|x\|_\alpha \leq \|x\|_\beta \leq c_2 \|x\|_\alpha,$$

para todo $x \in \mathbb{R}^n$.

Exemplo A.2.8*Suponha que $x \in \mathbb{R}^n$. Então, algumas equivalências de normas do $\mathbb{R}^n$ são:*

$$\begin{aligned} \|x\|_2 &\leq \|x\|_1 \leq \sqrt{n}\|x\|_2. \\ \|x\|_\infty &\leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty. \\ \|x\|_\infty &\leq \|x\|_1 \leq n\|x\|_\infty. \end{aligned}$$

Exemplo A.2.9*A norma-2 é preservada numa transformação ortogonal. Em outras palavras, se $Q \in \mathbb{R}^{n\times n}$ é uma matriz ortogonal e $x \in \mathbb{R}^n$, então:*

$$\|Qx\|_2^2 = (Qx)^T(Qx) = (x^TQ^T)(Qx) = x^T(Q^TQ)x = x^Tx = \|x\|_2^2.$$

Lema A.2.10*Se $A \in \mathbb{R}^{n\times n}$ é uma matriz simétrica com λ_1 e λ_n sendo o menor e o maior autovalor, respectivamente, então*

$$\lambda_1\|x\|^2 \leq x^TAx \leq \lambda_n\|x\|^2,$$

para todo $x \in \mathbb{R}^n$.

Demonstração: Ver (RIBEIRO; KARAS, 2013). ■

A.3 NORMA DE MATRIZES

Definição A.3.1*Dizemos que $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ é uma norma de matrizes se as três seguintes propriedades forem satisfeitas:*

1. *$f(A) \geq 0$, $A \in \mathbb{R}^{m\times n}$, ($f(A) = 0$ se, e só se, $A = 0$).*
2. *$f(A+B) \leq f(A) + f(B)$, $A, B \in \mathbb{R}^{m\times n}$.*
3. *$f(\alpha A) = |\alpha| f(A)$, $\alpha \in \mathbb{R}$, $A \in \mathbb{R}^{m\times n}$.*

Observação A.3.2*Assim como a norma de vetores, usaremos barras duplas com subíndices para denotar normas de matrizes, isto é, $\|A\| = f(A)$.*

Definição A.3.3*A Norma Frobenius de uma matriz $A \in \mathbb{R}^n$ é dada por:*

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} = \sqrt{Tr(A^TA)}.$$

Definição A.3.4*As normas-p, para $p > 1$, de uma matriz $A \in \mathbb{R}^n$ são dadas por:*

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

Está claro que $\|A\|_p$ é a norma-p do maior vetor obtido aplicando A em um vetor unitário na norma-p para vetores, ou seja:

$$\|A\|_p = \sup_{x \neq 0} \left\| A\left(\frac{x}{\|x\|_p}\right) \right\|_p = \max_{\|x\|_p = 1} \|Ax\|_p.$$

As normas-p possuem uma importante propriedade, que para todo $A \in \mathbb{R}^{m \times n}$ e $x \in \mathbb{R}^n$ temos:

$$\|Ax\|_p \leq \|A\|_p \|x\|_p.$$

A principais propriedades das normas de matrizes, Frobenius e normas-p (especialmente $p = 1, 2, \infty$), são tais que, para $A \in \mathbb{R}^{m \times n}$,

$$\begin{aligned}
\|A\|_2 &\leq \|A\|_F \leq \sqrt{min\{m,n\}}\|A\|_2, \\
\max_{i,j} |a_{ij}| &\leq \|A\|_2 \leq \sqrt{mn} \max_{i,j} |a_{ij}|, \\
\|A\|_1 &= \max_{1 \leq j \leq n} \sum_{i=1}^{m} |a_{ij}|, \\
\|A\|_\infty &= \max_{1 \leq i \leq m} \sum_{j=1}^{n} |a_{ij}|, \\
\frac{1}{\sqrt{n}}\|A\|_\infty &\leq \|A\|_2 \leq \sqrt{m}\, \|A\|_\infty, \\
\frac{1}{\sqrt{m}}\|A\|_1 &\leq \|A\|_2 \leq \sqrt{n}\, \|A\|_1.
\end{aligned}$$

Além disso, a norma de qualquer sub-matriz é menor que a norma da própria matriz, isto é, se $A \in \mathbb{R}^{m \times n}$, $1 \leq i_1 \leq i_2 \leq m$ e $1 \leq j_1 \leq j_2 \leq n$, então

$$\|A(i_1 : i_2, j_1 : j_2)\|_p \leq \|A\|_p.$$

Uma característica da norma-1 e da norma-∞ para matrizes é que são facilmente calculadas e requerem somente $O(n^2)$ operações, pois são calculadas diretamente pelos elementos da matriz. Por outro lado, o cálculo da norma-2 é consideravelmente mais complicada, pois depende um vetor, alÃ©m disso.

Teorema A.3.5*Se $A \in \mathbb{R}^{m \times n}$, então existe um vetor unitário sobre a norma-2 com dimensão n tal que $A^T Az = \mu^2 z$, onde $\mu = \|A\|_2$.*

Demonstração: Ver (GOLUB; LOAN, 2012). ■

O Teorema A.3.5 implica que $\|A\|_2^2$ é uma raiz de

$$p(\lambda) = det(A^T A - \lambda I).$$

Em particular, $\|A\|_2 = \sqrt{\lambda_{max}(A^T A)}$. Uma estimativa para se obter o resultado da norma-2 é a seguinte:

Corolário A.3.6 *Se $A \in \mathbb{R}^{m\times n}$, então $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$.*

Demonstração: Se $z \neq 0$ é tal que $A^T Az = \mu^2 z$, onde $\mu = \|A\|_2$, então

$$\begin{aligned} \mu^2 \|z\|_1 = \|A^T Az\|_1 &\leq \|A^T\|_1 \|A\|_1 \|z\|_1 \\ &= \|A\|_\infty \|A\|_1 \|z\|_1. \end{aligned}$$

Logo,

$$\begin{aligned} \mu^2 \|z\|_1 &\leq \|A\|_\infty \|A\|_1 \|z\|_1 \\ \mu^2 &\leq \|A\|_\infty \|A\|_1 \\ \|A\|_2^2 &\leq \|A\|_\infty \|A\|_1 \\ \|A\|_2 &\leq \sqrt{\|A\|_\infty \|A\|_1}. \end{aligned}$$

■

A.4 FATORAÇÃO LU

Sistemas triangulares são resolvidos facilmente em $O(n^2)$ operações. A ideia por trás da Eliminação de Gauss é converter um dado sistema $Ax = b$ para um sistema triangular equivalente. A conversão é alcalçada selecionando apropriadas combinações lineares das equações. Por exemplo, no sistema

$$\begin{aligned} x_1 - 3x_2 &= -2, \\ 2x_1 + x_2 &= 3, \end{aligned}$$

se multiplicarmos a primeira equação por 2 e a subtrairmos da segunda equação, obtemos:

$$\begin{aligned} x_1 - 3x_2 &= -2, \\ 7x_2 &= 7. \end{aligned}$$

Para $n = 2$, essa é a Eliminação de Gauss. Nosso objetivo nessa seção é descrever os procedimentos na linguagem de fatoração de matrizes. Isso significa que iremos construir um algoritmo que calcula uma matriz triangular inferior unitária L e uma matriz triangular superior U tal que $A = LU$, como por exemplo:

$$\begin{bmatrix} 1 & -3 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ 0 & 7 \end{bmatrix}.$$

A solução do problema original $Ax = b$ é então encontrado por um processo em 2 passos de resolução de sistemas triangulares:

$$Ly = b,\ Ux = y \ \Rightarrow\ Ax = LUx = Ly = b.$$

Para obter uma descrição da fatoração para a eliminação de Gauss como é tradicionalmente apresentado, precisamos descrever o processo de zerar. No nível $n = 2$, se $v_1 \neq 0$ e $\tau = \frac{v_2}{v_1}$, então

$$\begin{bmatrix} 1 & 0 \\ -\tau & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ 0 \end{bmatrix}.$$

Genericamente, suponha $v \in \mathbb{R}^n$, onde $v_k \neq 0$. Se

$$\tau^T = [0, \ldots, 0, \tau_{k+1}, \ldots, \tau_n],\ \tau_i = \frac{v_i}{v_k},\ i = k+1, k+2, \ldots, n,$$

e definimos

$$M_k = I_n - \tau e_k^T,$$

então

$$M_k v = \begin{bmatrix} 1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & & & & \\ 0 & & 1 & 0 & & 0 \\ 0 & & \tau_{k+1} & 1 & & 0 \\ \vdots & & \vdots & \vdots & \ddots & \\ 0 & & \tau_n & 0 & & 1 \end{bmatrix} \begin{bmatrix} v_1 \\ \vdots \\ v_k \\ v_{k+1} \\ \vdots \\ v_n \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Uma matriz na forma $M_k = I_n - \tau e_k^T \in \mathbb{R}^{n \times n}$ é uma transformação de Gauss se os primeiros k componentes de $\tau \in \mathbb{R}^n$ são nulos. Tal matriz é triangular inferior unitária. Os componenetes $\tau(k+1:n)$ são chamados multiplicadores. O vetor τ é chamado de vetor de Gauss.

A multiplicacação por uma transformação de Gauss é particularmente simples. Se

$C \in \mathbb{R}^{n\times r}$ e $M_k = I_n - \tau e_k^T$ é uma transformação de Gauss, então

$$M_k C = (I_n - \tau e_k^T)C = C - \tau(e_k^T C) = C - \tau C(k,:)$$

é uma atualização do produto externo. Desde que $\tau(1:k) = 0$ somente $C(k+1:n,:)$ é afetada.

Exemplo A.4.1*Sejam a matriz $C \in \mathbb{R}^{3\times 3}$, o vetor de Gauss $\tau \in \mathbb{R}^3$ e o vetor $e_1 \in \mathbb{R}^3$, respectivamente,*

$$C = \begin{bmatrix} 3 & 2 & 4 \\ 6 & 1 & 2 \\ 12 & 13 & -6 \end{bmatrix}, \tau = \begin{bmatrix} 0 \\ 2 \\ 4 \end{bmatrix} \text{ e } e_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

Então, a multiplicação pela transformação de Gauss é

$$(I_3 - \tau e_1^T)C = \begin{bmatrix} 3 & 2 & 4 \\ 0 & -3 & -4 \\ 0 & 5 & -22 \end{bmatrix}.$$

Seja $A \in \mathbb{R}^{n\times n}$. As transformações de Gauss $M_1, \ldots, M_{n-1}$ usualmente podem ser encontradas de modo que $M_{n-1} \ldots M_2 M_1 A = U$ é uma matriz triangular superior. Para ver isso, primeiro vamos olhar o caso de $n = 3$.

Exemplo A.4.2*Suponha*

$$A = \begin{bmatrix} 3 & -4 & 1 \\ 1 & 2 & 2 \\ 4 & 0 & -3 \end{bmatrix}$$

e note que

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{3} & 1 & 0 \\ -\frac{4}{3} & 0 & 1 \end{bmatrix} \Rightarrow M_1 A = \begin{bmatrix} 3 & -4 & 1 \\ 0 & -\frac{10}{3} & \frac{5}{3} \\ 0 & \frac{16}{3} & \frac{13}{3} \end{bmatrix}.$$

Da mesma forma, no segundo passo temos que

$$M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{8}{5} & 1 \end{bmatrix} \Rightarrow M_2(M_1 A) = \begin{bmatrix} 3 & -4 & 1 \\ 0 & \frac{10}{3} & \frac{5}{3} \\ 0 & 0 & -7 \end{bmatrix}.$$

Extrapolando esse exemplo para o caso geral n, concluímos que:

- No início do k-ésimo passo temos uma matriz $A^{(k-1)} = M_{k-1} \ldots M_1 A$ que é uma matriz triangular superior nas colunas 1 até $(k-1)$.

•Os multiplicadores na k-ésima transformação de Gauss M_k é baseada em $A^{(k-1)}(k+1:n,k)$ e $a_{kk}^{(k-1)}$ deve ser não-nulo a fim de funcionar, pois Ã© utilizado numa divisÃ£o.

Note que o processo de triangulização superior é finalizado em $(n-1)$ passos. O Algoritmo 7 apresenta o processo de triangularização superior.

Algoritmo 7 TRIANGULIZAÇÃO SUPERIOR

1: **Dados de entrada**($A \in \mathbb{R}^{n\times n}$).
2: **Dados de saída**($A^{(n)} = M_{n-1}\dots M_2M_1A$).
3: Faça $A^{(1)} = A$.
4: **Para** $k = 1 : n-1$, faça
5: Determine os multiplicadores $\tau_i^{(k)} = a_{ik}^{(k)}/a_{kk}^{(k)}$.
6: Aplique $M_k = I - \tau^{(k)}e_k^T$ para obter $A^{(k+1)} = M_kA^{(k)}$.
7: **Fim Para**

Para que esse processo esteja bem definido, as entradas $a_{11}^{(1)}$, $a_{22}^{(2)}$, $a_{33}^{(3)}$, ..., $a_{n-1,n-1}^{(n-1)}$, chamadas de pivôs, devem ser não-nulas.

Se os pivôs não-nulos forem encontrados na linha 5 do Algoritmo 7, então as matrizes de transformação de Gauss $M_1,\dots,M_{n-1}$ são geradas de tal modo que $M_{n-1}\dots M_1A = U$ é uma matriz triangular superior. É fácil verificar que se $M_k = I_n - \tau^{(k)}e_k^T$, então sua inversa é $M_k^{-1} = I_n + \tau^{(k)}e_k^T$ e, então,

$$A = LU, \tag{A.4.1}$$

onde

$$L = M_1^{-1}\dots M_{n-1}^{-1}. \tag{A.4.2}$$

Sendo L uma matriz triangular inferior unitária, pois cada M_k^{-1} é uma matirz triangular inferior unitária. A fatorização (A.4.1) é chamada de Fatoração LU.

Exemplo A.4.3*A fatoração LU pode não existir. Por exemplo, é impossível encontrar l_{ij} e u_{ij} tal que*

$$\begin{bmatrix} 1 & 2 & 9 \\ 3 & 6 & 7 \\ 2 & 7 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}.$$

Para ver isso, iguale as entradas e observe que devemos ter que $u_{11} = 1$, $u_{12} = 2$, $l_{21} = 3$, $u_{22} = 0$ e $l_{31} = 2$. Mas então a entrada $(3,2)$ nos dá a contradição $7 = l_{31}u_{12} + l_{32}u_{22} = 4$. Nesse exemplo, o pivô $a_{22}^{(1)} = a_{22} - (a_{21}/a_{11})a_{12}$ é nulo.

Isso nos dá que o k-ésimo pivô em $(\star)$ é nulo se $A(1:k,1:k)$ é singular (não-inversível). A submatriz da forma $A(1:k,1:k)$ é chamada de submatriz líder principal.

Teorema A.4.4*(Fatoração LU) Se $A \in \mathbb{R}^{n\times n}$ e $det(A(1:k,1:k)) \neq 0$ para todo $k = 1:n-1$, então existe uma matriz triangular inferior $L \in \mathbb{R}^{n\times n}$ e uma matriz triangular superior $U \in \mathbb{R}^{n\times n}$ tal que $A = LU$. Se este é o caso e A é não-singular (inversível), então a fatoração é única e $det(A) = u_{11}\dots u_{nn}$.*

Demonstração: Suponha que $(k-1)$ passos do Algoritmo 7 tenham sido executados. No início do passo k a matriz A já foi sobrescrita por $M_{k-1}\dots M_1 A = A^{(k-1)}$. Visto que as tranformações de Gauss são matrizes triangulares inferiores, segue que olhando para a parte inicial da equação que

$$det(A(1:k,1:k)) = a_{11}^{(k-1)}\dots a_{kk}^{(k-1)}. \tag{A.4.3}$$

Portanto, se $A(1:k,1:k)$ é não-singular, então o k-ésimo pivô $a_{kk}^{(k-1)}$ é não-nulo.

Para a unicidade, se $A = L_1U_1$ e $A = L_2U_2$ são duas fatorações LU de uma matriz não-singular A, então $L_2^{-1}U_1 = U_2U_1^{-1}$. Visto que $L_2^{-1}L_1$ é uma matriz triangular inferior e $U_2U_1^{-1}$ é uma matriz triangular superior, segue que ambas as matrizes devem ser iguais a identidade. Consequentemente, temos que $L_1 = L_2$ e $U_1 = U_2$. Portanto, se $A = LU$, então

$$\begin{aligned} det(A) = det(LU) &= det(L)det(U) \\ &= det(U) \\ &= u_{11}u_{22}\dots u_{nn}. \end{aligned}$$

■

Verificação que a construção de L não é tão complicada como a equação (A.4.2) sugere. De fato,

$$\begin{aligned} L &= M_1^{-1}\dots M_{n-1}^{-1} \\ &= \left(I_n - \tau^{(1)}e_1^T\right)^{-1}\dots\left(I_n - \tau^{(n-1)}e_{n-1}^T\right)^{-1} \\ &= \left(I_n + \tau^{(1)}e_1^T\right)\dots\left(I_n + \tau^{(n-1)}e_{n-1}^T\right) \\ &= I_n + \sum_{k=1}^{n-1}\tau^{(k)}e_k^T. \end{aligned}$$

Mostrando que

$$L(k+1:n,k) = \tau^{(k)}(k+1:n),\ k = 1:n-1. \tag{A.4.4}$$

Em outras palavras, a k-ésima coluna de L é definida pelos multiplicadores que foram encontradas até o k-ésimo passo na linha 5 do Algoritmo 7. Considerando o exemplo A.4.2, temos,

$$\tau^{(1)} = \begin{bmatrix} 0 \\ \frac{1}{3} \\ \frac{4}{3} \end{bmatrix}, \tau^{(2)} = \begin{bmatrix} 0 \\ 0 \\ \frac{8}{5} \end{bmatrix} \Rightarrow \begin{bmatrix} 3 & -4 & 1 \\ 1 & 2 & 2 \\ 4 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{3} & 1 & 0 \\ \frac{4}{3} & \frac{8}{5} & 1 \end{bmatrix} \begin{bmatrix} 3 & -4 & 1 \\ 0 & \frac{10}{3} & \frac{5}{3} \\ 0 & 0 & -7 \end{bmatrix}.$$

Portanto, o Algoritmo 8 ilustra o procedimento para encontrar a matriz triangular inferior L e a matriz triangular superior U obtidas através da fatoração LU da matriz A.

Algoritmo 8 FATORAÇÃO LU

1: **Dados de entrada**($A \in \mathbb{R}^{n\times n}$).
2: **Dados de saída**($L \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times n}$).
3: Faça $A^{(1)} = A$.
4: **Para** $k = 1 : n-1$, faça
5: Determine os multiplicadores $\tau_i^{(k)} = a_{ik}^{(k)} / a_{kk}^{(k)}$.
6: Aplique $M_k = I - \tau^{(k)} e_k^T$ para obter $A^{(k+1)} = M_k A^{(k)}$.
7: **Fim Para**
8: Faça $U = M_{n-1} M_{n-2} \ldots M_1 A$ e $L = I_n + \sum_{k=1}^{n-1} \tau^{(k)} e_k^T$.

A.5 DECOMPOSIÇÃO SVD

Como vimos no Teorema Espectral A.1.49, toda matriz simétrica A pode ser fatorada como $A = PDP^T$, onde P é uma matriz ortogonal e D é uma matriz diagonal formada pelos autovalores de A. Contudo, nem toda matriz admite decomposição espectral. Nesta seção, vamos mostrar que toda matriz (simétrica ou não) possui uma fatoração da forma $A = PDQ^T$, onde P e Q são ortogonais e D é uma matriz diagonal. Esse resultado é a Decomposição em Valores Singulares (SVD, do inglês *Singular Values Decomposition*).

Para qualquer matriz $A \in \mathbb{R}^{m\times n}$, a matriz $A^TA \in \mathbb{R}^{n\times n}$ é simétrica e, portanto, pode ser diagonalizada ortogonalmente pelo Teorema Espectral. Não só os autovalores de A^TA são reais, como são todos não-negativos (POOLE, 2014). Portanto, faz sentido extrair a raiz quadrada

desses autovalores.

Definição A.5.1*(Valores Singulares) Se A é uma matriz em $\mathbb{R}^{m\times n}$, então os valores singulares de A são as raízes quadradas dos autovalores de A^TA denotados por $\sigma_1, \sigma_2, \ldots, \sigma_n$. É uma convenção ordenar os valores singulares de modo que $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$.*

Exemplo A.5.2*Encontre os valores singulares de*

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -2 & 2 \end{bmatrix}.$$

Solução: A matriz

$$A^TA = \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 5 & -4 \\ -4 & 5 \end{bmatrix}$$

tem autovalores $\lambda_1 = 9$ e $\lambda_2 = 1$. Consequentemente, os valores singulares de A são $\sigma_1 = \sqrt{\lambda_1} = 3$ e $\sigma_2 = \sqrt{\lambda_2} = 1$. ♦

Para entender o significado dos valores singulares de uma matriz $A \in \mathbb{R}^{m\times n}$, considere os autovalores de A^TA. Como A^TA é simétrica, sabemos que existe uma base ortonormal para $\mathbb{R}^n$ formada pelos autovetores de A^TA. Seja $\{v_1, v_2, \ldots, v_n\}$ a basse correspondente aos autovalores de A^TA, odenados de modo que $\lambda_1, \lambda_2, \ldots, \lambda_n$. Agora veja que

$$\|Av_i\|^2 = (Av_i)\cdot(Av_i) = (Av_i)^T(Av_i) = v_i^TA^TAv_i = v_i^T\lambda_i v_i = \lambda_i(v_i \cdot v_i) = \lambda_i.$$

Dessa forma, $\|Av_i\|^2 = \lambda_i$, para todo $i = 1, 2, \ldots, n$, ou seja,

$$\sigma_i = \sqrt{\lambda_i} = \|Av_i\|.$$

Em outras palavras, os valores singulares de A são os comprimentos dos vetores Av_1, Av_2, ..., Av_n.

Agora vamos descrever a decomposição em valores singulares de uma matriz. Queremos mostrar que uma matriz $A \in \mathbb{R}^{m\times n}$ pode ser fatorada como

$$A = U\Sigma V^T,$$

onde $U \in \mathbb{R}^{m\times m}$ é uma matriz ortogonal, $V \in \mathbb{R}^{n\times n}$ é uma matriz ortogonal e $\Sigma \in \mathbb{R}^{m\times n}$ é uma

matriz “diagonal”. Se os valores singulares não-nulos de A são

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0,$$

e $\sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_n = 0$, então Σ terá a forma em blocos

$$\Sigma = \begin{bmatrix} D & O \\ O & O \end{bmatrix}, \text{ onde } D = \begin{bmatrix} \sigma_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_r \end{bmatrix}$$

e cada matriz 0 é uma matriz nula de tamanho apropriado.

Para construir a matriz ortogonal V, vamos primeiro achar uma base ortonormal $\{v_1, v_2, \ldots, v_n\}$ de $\mathbb{R}^n$ formada por autovetores da matriz simétrica $A^TA \in \mathbb{R}^{n\times n}$. Então,

$$V = [v_1|\ldots|v_n] \in \mathbb{R}^{n\times n}$$

é uma matriz ortogonal.

Para a matriz ortogonal U, primeiro notamos que $\{Av_1, Av_2, \ldots, Av_r\}$ é um conjunto ortogonal de vetores de $\mathbb{R}^m$. Para ver isso, suponha que v_i seja o autovetor de A^TA correspondente ao autovalor λ_i. Então, para $i \neq j$, temos

$$\begin{aligned} (Av_i)\cdot(Av_j) &= (Av_i)^T(Av_j) \\ &= v_i^T A^T A v_j \\ &= v_i^T \lambda_j v_j \\ &= \lambda_j(v_i \cdot v_j) = 0, \end{aligned}$$

visto que os autovalores v_i são ortogonais. Agora, lembre-se de que o valor singular satisfaz $\sigma_i = \|Av_i\|$, e que os r primeiros desses valores são não-nulos. Portanto, podemos normalizar $Av_1, Av_2, \ldots, Av_r$, considerando

$$u_i = \frac{1}{\sigma_i} Av_i, \text{ para } i = 1, 2, \ldots, r.$$

Isso garante que $\{u_1, u_2, \ldots, u_r\}$ seja um conjunto ortonormal em $\mathbb{R}^m$, mas, se $r < m$, ele não será uma base para $\mathbb{R}^m$. Nesse caso estendemos o conjunto $\{u_1, u_2, \ldots, u_r\}$ para uma base ortonormal $\{u_1, u_2, \ldots, u_m\}$ de $\mathbb{R}^m$. Então, consideramos

$$U = [u_1|\ldots|u_m].$$

Só nos resta mostrar que essas escolhas funcionam, isto é, precisamos verificar que, com U, V e Σ como descritas anteriormente, temos $A = U\Sigma V^T$. Como $V^T = V^{-1}$, isso equivale a mostrar que

$$AV = U\Sigma.$$

Sabemos que

$$Av_i = \sigma_i u_i, \text{ para } i = 1, 2, \ldots, r$$

e $\|Av_i\| = \sigma_i = 0$ para $i = r+1, r+2, \ldots, n$. Portanto,

$$\begin{aligned}
AV &= A\,[\,v_1 \mid v_2 \mid \ldots \mid v_n\,] \\
&= [\,Av_1 \mid Av_2 \mid \ldots \mid Av_n\,] \\
&= [\,\sigma_1 u_1 \mid \ldots \mid \sigma_r u_r \mid 0 \mid \ldots \mid 0\,] \\
&= [\,u_1 \mid u_2 \mid \ldots \mid u_m\,] \begin{bmatrix} \sigma_1 & \ldots & 0 & \\ \vdots & \ddots & \vdots & O \\ 0 & \ldots & \sigma_r & \\ & O & & O \end{bmatrix} \\
&= U\Sigma.
\end{aligned}$$

Teorema A.5.3*(Decomposição por Valores Singulares) Seja $A \in \mathbb{R}^{m\times n}$ com valores singulares $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ e $\sigma_{r+1} = \sigma_{r+2} = \ldots = \sigma_n = 0$. Então existem uma matriz ortognal $U \in \mathbb{R}^{m\times m}$, uma matriz ortogonal $V \in \mathbb{R}^{n\times n}$ e uma matriz $\Sigma \in \mathbb{R}^{m\times n}$, de modo que*

$$A = U\Sigma V^T.$$

Demonstração: Vide (POOLE, 2014). ■

Exemplo A.5.4*Obtenha a decomposição em valores singulares de*

$$A = \begin{bmatrix} 1 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \\ 2 & 2 & 2 & 2 \end{bmatrix}.$$

Temos que a matriz

$$A^TA = \begin{bmatrix} 6 & 6 & 6 & 6 \\ 6 & 8 & 4 & 6 \\ 6 & 4 & 8 & 6 \\ 6 & 6 & 6 & 6 \end{bmatrix}$$

possui os autovalores 24, 4, 0 *e* 0. *Os autovetores correspondentes são:*

$$v_1 = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},\ v_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 \\ -1 \\ 1 \\ 0 \end{bmatrix},\ v_3 = \frac{1}{\sqrt{12}}\begin{bmatrix} -3 \\ 1 \\ 1 \\ 1 \end{bmatrix},\ e\ v_4 = \frac{1}{\sqrt{6}}\begin{bmatrix} 0 \\ -1 \\ -1 \\ 2 \end{bmatrix}.$$

Dessa forma teremos as matrizes:

$$V = \begin{bmatrix} 1/2 & 0 & -3/\sqrt{12} & 0 \\ 1/2 & -1/\sqrt{2} & 1/\sqrt{12} & -1/\sqrt{6} \\ 1/2 & 1/\sqrt{2} & 1/\sqrt{12} & -1/\sqrt{6} \\ 1/2 & 0 & 1/\sqrt{12} & 2/\sqrt{6} \end{bmatrix}\ e\ \Sigma = \begin{bmatrix} \sqrt{24} & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Para determinar a matriz U, calculamos

$$u_1 = \left(\frac{1}{\sqrt{24}}\right)Av_1\ e\ u_2 = \left(\frac{1}{2}\right)Av_2$$

para obter

$$u_1 = \frac{1}{\sqrt{6}}\begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}\ e\ u_2 = \frac{1}{\sqrt{2}}\begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}.$$

Como o contradomínio de A tem dimensão 3, *precisamos de mais um vetor para formar a base. Calculamos* u_3 *unitário e fazendo-o ortogonal a* u_1 *e* u_2. *Obtemos então* $u_3 = \left(1/\sqrt{3}\right)[\ 1\ 1\ -1\ ]^T$. *Com estes vetores montamos a matriz:*

$$U = \begin{bmatrix} 1/\sqrt{6} & -1/\sqrt{2} & 1/\sqrt{3} \\ 1/\sqrt{6} & 1/\sqrt{2} & 1/\sqrt{3} \\ 2/\sqrt{6} & 0 & -1/\sqrt{3} \end{bmatrix},$$

de modo que $A = U\Sigma V^T$.

A decomposição SVD é motivada pelo fato geométrico de que a imagem de uma esfera unitária $S \in \mathbb{R}^n$ sob uma matriz $A \in \mathbb{R}^{m\times n}$ resulta numa hiperelipse $AS \in \mathbb{R}^m$. Vejamos

primeiramente o caso bidimensional ($m = n = 2$).

Consideremos $S \in \mathbb{R}^2$ a circunferência unitária do $\mathbb{R}^2$ (ver Figura 31), à qual é aplicada uma transformação do tipo $A = U\Sigma V^T$. Seja $z = [\, x\, y\,]^T \in S$ um ponto da circunferência S. Então

$$Az = U\Sigma V^T z.$$

Considere $w = V^T z$. Como V é uma matriz ortogonal temos que, sem perda de generalidade, ela pode ser dada por

$$w = \begin{bmatrix} \cos(\theta) & -\operatorname{sen}(\theta) \\ \operatorname{sen}(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

que representa uma rotação (no sentido anti-horário) com uma amplitude θ. Assim, quando aplicamos V^T à circunferência S, esta rotaciona em um ângulo θ, no sentido anti-horário (ver Figura 31).

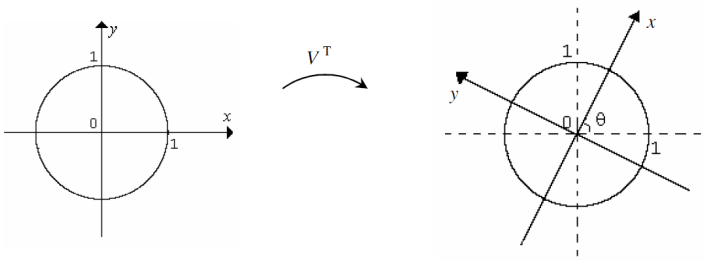


Figura 31: Rotação no sentido anti-horário.

Fonte: (NOBRE, 2007)

Continuamos agora, transformando S, depois de rotacionada, através de Σ. Como $v \equiv \Sigma w = [\, \sigma_1 w_1\ \sigma_1 w_1\,]^T$, vemos que a circunferência é transformada numa elipse (Figura 32) cuja excentricidade é dada por

$$e = \sqrt{1 - \frac{\sigma_2^2}{\sigma_1^2}}.$$

Finalmente a aplicação de U à elipse a fará rotacionar, no sentido horário, em um ângulo θ (Figura 33). A transformação dada pela matriz A é ilustrada nas Figuras 31, 32 e 33 para a circunferência unitária do $\mathbb{R}^2$, com os valores singulares $\sigma_1 = 2$ e $\sigma_2 = 1$.

Esta interpretação geométrica é estendida de forma similar no caso em que S é uma esfera unitária em $\mathbb{R}^n$. Neste caso, definimos uma hiperelipse (generalização de uma elipse) em $\mathbb{R}^m$, como sendo a superfície obtida pela deformação da esfera unitária ao longo das direções ortogonais $u_1, u_2, \ldots, u_m$ em $\mathbb{R}^m$, de acordo com os valores singulares $\sigma_1, \sigma_2, \ldots, \sigma_m$. Eventual-

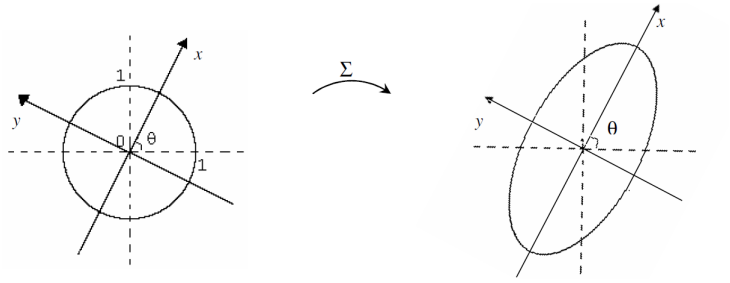


Figura 32: Transformação da circunferência numa elipse.

Fonte: (NOBRE, 2007)

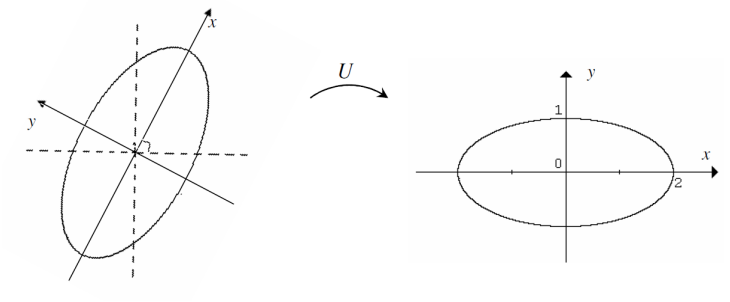


Figura 33: Rotação no sentido horário.

Fonte: (NOBRE, 2007)

mente, alguns destes valores singulares podem ser nulos e, assim, pode obter-se uma hiperelipse de dimensão n em um espaço de dimensão m, com $m > n$. Analogamente, os vetores $\sigma_i u_i$ são os semi-eixos principais da hiperelipse e os valores singulares $\sigma_1, \sigma_2, \ldots, \sigma_m$ correspondem aos respectivos comprimentos dos eixos.

A seguir, no Algoritmo 9, apresentamos o procedimento básico para obter a decomposição em valores singulares de uma matriz A qualquer.

Algoritmo 9 DECOMPOSIÇÃO SVD

1: **Dados de entrada**($A \in \mathbb{R}^{m \times n}$).
2: **Dados de saída**($U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$).
3: Encontre os valores singulares σ_i de A. Faça $\Sigma_{ii} = \sigma_i, i = 1, 2, \ldots, r$.
4: Determine um conjunto ortonormal $\{v_1, v_2, \ldots, v_n\}$ formado pelos autovetores de $A^T A$. Faça $V = [\, v_1 \mid v_2 \mid \ldots \mid v_n \,]$.
5: Calcule $u_i = \dfrac{Av_i}{\sigma_i}, i = 1, 2, \ldots, r$, complete o conjunto $\{u_i \mid i = 1, 2, \ldots, r\}$ para uma base ortonormal em $\mathbb{R}^m$ e faça $U = [\, u_1 \mid u_2 \mid \ldots \mid u_m \,]$.

APÊNDICE B – CÓDIGO DE ORGANIZAÇÃO DAS COORDENADAS

```
function [ X ] = org_PGDM( x )
%org_PGDM - Organizar a Configuração de Pontos x em uma matriz 3xn
% Os parâmetros esperados são:
% - x: Vetor x \in R^(3*n)
% A função retorna:
% - X: A configuração de pontos para criar matriz de distâncias
n = length(x); %dimensão do vetor
k = floor(n/3); %número de pontos (tridimensionais)
A = zeros(k,1); %vetor com as entradas x de todos os pontos
B = zeros(k,1); %vetor com as entradas y de todos os pontos
C = zeros(k,1); %vetor com as entradas z de todos os pontos
for i = 0:k-1
        A(i+1,1) = x(3*i + 1);
        B(i+1,1) = x(3*i + 2);
        C(i+1,1) = x(3*i + 3);
end
X = [ A B C ]';
end
```

APÊNDICE C – CÓDIGO DA MATRIZ DE DISTÂNCIAS

```
function [ A ] = matriz_PGDM( V, opt )
%matriz_PGDM - gerar uma Matriz de Distâncias Euclidianas a partir de uma matriz V com as coordenadas tridimensionais dos átomos de uma proteína
% Os parâmetros esperados são:
% V: matriz V com as coordenadas dos átomos da proteína
% opt: a opção em criar uma matriz triângular superior com diagonal nula (entrada: 0) ou matriz simétrica com diagonal nula (entrada: 1)
% A função retorna:
% A: uma Matriz de Distâncias Euclidianas das coordenadas tridimensionais dos átomos da proteína
n = length(V); %número de átomos
A = zeros(n,n); %matriz de distâncias inicia com entradas nulas
if (opt == 1) %vai gerar uma matriz simétrica com diagonal nula
        for j = 1:n
                for k = 1:n
                        A(j,k) = norm(V(:,j) - V(:,k)); %distância euclidiana
                end
        end
elseif (opt == 0) %vai gerar uma matriz triângular superior com diagonal nula
        for j = 1:n
                for k = j+1:n
                        A(j,k) = norm(V(:,j) - V(:,k)); %distância euclidiana
                end
        end
end
end
```

APÊNDICE D – CÓDIGO DA FUNÇÃO OBJETIVO

```
function [ f ] = funcao_PGDM( x, A )
%funcao_PGDM - Função Objetivo do Problema de Geometria de Distâncias:
% f = sum_i,j ( (—— x^i - x^j ——)^2 - (d_ij)^2)^2
% Os parâmetros esperados são:
% x: Vetor x \in R^(3*n)
% A: Matriz das Distâncias Euclidianas
% A função retorna:
% f: valor de função no ponto x
n = length(A); %dimensão do problema
k = 1; %marcar as entradas de vetor que representará cada parte da soma da função objetivo
ft = zeros(floor((n^2 - n)/2), 1); %número valores positivos da matriz triangular superior A, isto
é, número de fatores da soma da função objetivo
for j = 1:n
        for l = j+1:n
                ft(k) = ( (x(3*l - 2) - x(3*j - 2))^2 + (x(3*l - 1) - x(3*j - 1))^2 + (x(3*l) -
x(3*j))^2 - A(j, l)^2 )^2;
                %agora faz-se cada termo da somatório com o valor da função mesmo
                k = k + 1;
        end
end
f = sum(ft); %a cada loop acima, guarda no vetor ft as entradas de cada fator do somatório,
então no final é só somar as entradas de ft
end
```

APÊNDICE E – CÓDIGO DO VETOR GRADIENTE DA FUNÇÃO OBJETIVO

```
function [ gf ] = gradiente_PGDM( x, A )
%gradiente_PGDM - Gradiente da Função Objetivo do Problema de Geometria de Distâncias
% Os parâmetros esperados são:
% x: vetor x \in R^(3*n)
% A: matriz de Distâncias Euclidianas
% A função retorna:
% gf: Gradiente da Função Objetivo
n = length(A); %número de pontos do problema
gf = zeros(3*n, 1); %número de entradas do vetor x
for i = 1:n
        for j = i+1:n
                gf(3*i-2) = gf(3*i-2) + 4*(x(3*i-2) - x(3*j-2))*((x(3*i-2)-x(3*j-2))^2 + (x(3*i-
1)-x(3*j-1))^2 + (x(3*i)-x(3*j))^2 - A(i, j)^2);
                gf(3*j-2) = gf(3*j-2) - 4*(x(3*i-2) - x(3*j-2))*((x(3*i-2)-x(3*j-2))^2 + (x(3*i-
1)-x(3*j-1))^2 + (x(3*i)-x(3*j))^2 - A(i, j)^2);
                gf(3*i-1) = gf(3*i-1) + 4*(x(3*i-1) - x(3*j-1))*((x(3*i-1)-x(3*j-1))^2 + (x(3*i-
2)-x(3*j-2))^2 + (x(3*i)-x(3*j))^2 - A(i, j)^2);
                gf(3*j-1) = gf(3*j-1) - 4*(x(3*i-1) - x(3*j-1))*((x(3*i-1)-x(3*j-1))^2 + (x(3*i-
2)-x(3*j-2))^2 + (x(3*i)-x(3*j))^2 - A(i, j)^2);
                gf(3*i) = gf(3*i) + 4*(x(3*i) - x(3*j))*((x(3*i)-x(3*j))^2 + (x(3*i-2)-x(3*j-
2))^2 + (x(3*i-1)-x(3*j-1))^2 - A(i, j)^2);
                gf(3*j) = gf(3*j) - 4*(x(3*i) - x(3*j))*((x(3*i)-x(3*j))^2 + (x(3*i-2)-x(3*j-
2))^2 + (x(3*i-1)-x(3*j-1))^2 - A(i, j)^2);
        end
end
end
```

APÊNDICE F – CÓDIGO DA MATRIZ JACOBIANA

```
function [ J ] = jacobiana_PGDM( x, A )
%jacobiana_PGDM - Matriz Jacobiana do sistema não-linear do Problema da Geometria de Distâncias
% Os parâmetros esperados são:
% x: vetor x \in R^(3*n)
% A: matriz de Distâncias Euclidianas
% A função retorna:
% J: Matriz Jacobiana do sistema não-linear do Problema de Geometria de Distâncias
n = length(A); %dimensão do problema
m = floor((n^2 - n)/2); %número de entradas positivas da matriz A
J = zeros(m,3*n); %matriz Jacobiana inicia com zeros
k = 1; %linha 1 da Jacobiana
for i = 1:n
        for j = i+1:n
                J(k, 3*i - 2) = + 2*(x(3*i-2) - x(3*j-2));
                J(k, 3*i - 1) = + 2*(x(3*i-1) - x(3*j-1));
                J(k, 3*i) = + 2*(x(3*i) - x(3*j));
                J(k, 3*j - 2) = - 2*(x(3*i-2) - x(3*j-2));
                J(k, 3*j - 1) = - 2*(x(3*i-1) - x(3*j-1));
                J(k, 3*j) = - 2*(x(3*i) - x(3*j));
                k = k + 1; %próxima linha da Jacobiana
        end
end
end
```

APÊNDICE G – CÓDIGO DO MÉTODO BFGS

```
function [ x, R, f, k, t ] = BFGS_PGDM(f, tol, maxit, x, A)
%BFGS_PGDM - Minimiza uma função f(x) a partir de uma aproximação da Hessiana e de um ponto inicial x.
% Os parâmetros esperados são:
% f: função a ser minimizada
% tol: tolerância/precisão exigida
% maxit: número máximo de iterações
% x: aproximação/chute inicial
% A: matriz de distâncias euclidianas
% A função retorna:
% x: o mínimo aproximado da função
% R: 1 se o algoritmo convergiu e 0 caso contrário
% f: valor de função na solução
% k: número de iterações realizadas pelo algoritmo
% t: tempo utilizado pelo algoritmo
tic %inicia o contador de tempo
n = length(A); %número de variáveis
k = 0; %iteração inicial
B = eye(3*n); %aproximação inicial da Hessiana: matriz identidade
erro = inf; %erro inicial
while ( ( k < maxit ) && ( erro > tol ) )
        gf = gradfuncPEDM(x, A); %gradiente da função objetivo
        d = -B\(gf); %direção de descida: resolução do sistema por LU
        %retrocesso (backtracking) satisfazendo a condição de armijo
        alpha = 1;
        while f(x+alpha*d,A) > f(x,A) + alpha*0.0001*gf'*d;
                alpha = alpha*0.5; %diminui o tamanho do passo
        end
```

```
        xnovo = x + alpha*d; %novo ponto
        erro = max(abs(alpha*d)); %erro por norma infinita
        s = xnovo - x;
        gfnovo = gradfuncPEDM(xnovo, A);
        y = gfnovo - gf;
        B = B + (1/(y'*y))*(y*y') - (B*s)*(s'*B)*(1/(s'*B*s)); %atualização da matriz B
        x = xnovo; %o novo ponto é o atual da próxima iteração
        k = k + 1; %próxima iteração
end
if (k < maxit) %se não excedeu o limite de iterações
        R = 1; %o algoritmo convergiu
        f = f(x,A); %valor da função na solução
else
        R = 0; %o algoritmo não convergiu
end
t = toc; %finaliza o contador de tempo
end
```

APÊNDICE H – CÓDIGO DO MÉTODO DE GAUSS-NEWTON

```
function [ x, R, f, k, t ] = GaussNewton_PGDM( x, r, maxit, tol, A)
%GaussNewton_PGDM - Algoritmo de Gauss-Newton com Jacobiana exata para resolver o problema de mínimos quadrados
% min 1/2 —— r(x) ——_2^2
%onde r_i : R^(3*(n)) —> R.
% Os parâmetros esperados são:
% x: vetor x \in R^(3*n)
% r: sistema não-linear associado ao PGDM
% maxit: número máximo de iterações a ser utilizado pelo algoritmo
% tol: erro admitido pelo algoritmo
% A: matriz de Distâncias Euclidianas
% A função retorna:
% x: o mínimo aproximado
% R: 1 se o algoritmo convergiu (r(x) = 0) e 0 caso não (r(x)  =0)
% f: valor de função na solução
% k: número de iterações realizadas pelo algoritmo
% t: tempo utilizado pelo algoritmo
tic %inicia o contados de tempo
k = 0; %iteração inicial
erro = inf;
while ( ( k <= maxit ) && ( erro > tol ) )
        J = jacobianaPEDM(x, A); %matriz jacobiana do sistema
        T = J'*J;
        gf = J'*(r(x, A));
        d = (T)\(-gf); %direção de descida: resolução do sistema por LU
        %retrocesso (backtracking) satisfazendo a condição de armijo
        alpha = 1;
        while (.5*norm( r(x + alpha*d, A) )^2) > (.5*norm( r(x, A ) )^2 + alpha*0.0001*gf'*d)
```

```
                alpha = alpha*0.5; %diminui o tamanho do passo
        end
        x = x + alpha*d; %novo ponto
        erro = max(abs(alpha*d)); %erro por norma infinita
        k = k + 1; %próxima iteração
end
f = funcPEDM(x,A); %valor de função na solução
if ( k < maxit ) %se não excedeu o máximo de iterações
        if norm( r(x, A) ) < tol %se r(x) = 0
                R = 1; %o algoritmo convergiu
        else
                R = 0; %o algoritmo não convergiu
        end
else
        R = 0; %o algoritmo não convergiu
end
t = toc; %finaliza o contador de tempo
end
```

APÊNDICE I – CÓDIGO DE MULTIPLICAÇÃO DE QUATÉRNIOS A

```
function [ X ] = matriz_quaternio_direita( x )
%matriz_quaternio_direita Expressa o quaternio x na forma matricial para uma multiplicação yx = Xy (à direita)
% Os pârametros esperados são:
% - x: quaternio na forma vetorial
% A função retorna:
% - X: a forma matricial do quatérnio x
if (length(x) == 3) %se x é um quatérnio puramente imaginário
        x = [ 0; x ]; %x será um quatérnio com a parte real nula
end
%matriz para expressar a multiplicação yx = Xy
X = [ x(1), -x(2), -x(3), -x(4); x(2), x(1), x(4), -x(3); x(3), -x(4), x(1), x(2); x(4), x(3), -x(2), x(1)];
end
```

APÊNDICE J – CÓDIGO DE MULTIPLICAÇÃO DE QUATÉRNIOS B

```
function [ X ] = matriz_quaternio_esquerda( x )
%matriz_quaternio_esquerda Expressa o quaternio x na forma matricial para uma multiplicação
xy = Xy (à esquerda)
% Os pârametros esperados são:
% - x: quaternio na forma vetorial
% A função retorna:
% - X: a forma matricial do quatérnio x
if (length(x) == 3) %se x é um quatérnio puramente imaginário
        x = [ 0; x ]; %será um quartérnio com a parte real nula
end
%matriz para expressar a multiplicação xy = Xy
X = [ x(1), -x(2), -x(3), -x(4); x(2), x(1), -x(4), x(3); x(3), x(4), x(1), -x(2); x(4), -x(3), x(2),
x(1)];
end
```

APÊNDICE K – CÓDIGO DE ALINHAMENTO BASEADO EM QUATÉRNIOS

```
function [ q, lrms, Pt, Qt, newPt ] = procrustes_quaternios( P, Q )
%procrustes_quaternios Encontra o RMSD mínimo entre dois conjuntos de N pontos tridimensionais e o movimento rígido (translação e rotação) que sobrepõe os dois conjuntos de pontos de maneira ótima.
% Os parâmetros esperados são:
% - P: Uma matriz DxN onde P(a,i) é a a-ésima coordenada do i-ésimo ponto do primeiro conjunto de pontos
% - Q: Uma matriz DxN onde Q(a,i) é a a-ésima coordenada do i-ésimo ponto do segundo conjunto de pontos
% A função retorna:
% - q: o quatérnio que representa rotação ótima
% - lrms: o Last Root Mean Square (Raíz dos Mínimos Quadrados)
% - Pt: O primeiro conjunto de pontos transladado para a origem
% - Qt: O segundo conjunto de pontos transladado para a origem
% - NewPt: O primeiro conjunto de pontos alinhado ao conjunto de pontos Q
sz1 = size(P); %dimensões da matriz de entrada P
sz2 = size(Q); %dimensões da matriz de entrada Q
if ( (length(sz1)  = 2) —— (length(sz2)  = 2) )
        error 'P e Q devem ser Matrizes.';
end
if ( any( sz1  = sz2 ) )
        error 'P e Q devem ter as mesmas dimensões.';
end
N = sz1(2); %número de pontos
m = ones(1,N)/N; %pesos
p0 = P*m'; %centróide de P
q0 = Q*m'; %centróide de Q
v1 = ones(1,N); %vetor linha com N entradas unitárias
```

```
Pt = P - p0*v1; %conjunto de pontos P transladado para o centro da origem
Qt = Q - q0*v1; %conjunto de pontos Q transladado para o centro da origem
M = zeros(4,4); %matriz M inicia nula
for i = 1:N
        M = M + matriz_quaternio_direita(Pt(:,i))'*matriz_quaternio_esquerda(Qt(:,i));
end
[ G, H ] = eig(M); %matriz dos autovetores e autovalores da matriz M
lambda = max(diag(H)); %maior autovalor
[ , k] = find(H == lambda); %coluna que se encontra o autovetor (lambda)
q = G(:, k); %autovetor relacionado a lambda (rotação ótima)
qc = [ q(1); -q(2); - q(3); -q(4)]; %quaténio conjugado de q
E = zeros(4,N); %a matriz de alinhamento inicia nula
for j = 1:N
        %calcula a matriz de alinhamento
        E(:,j) = (matriz_quaternio_esquerda(matriz_quaternio_direita(Pt(:,j))*q)*qc);
end
newPt = E(2:4,:); %ignora a primeira linha, pois E tem 4 linhas
Diff = newPt - Qt;
lrms = sqrt( sum( sum( bsxfun( @times, m, Diff ).*Diff ) ) ); %RMSD(P,Q)
end
```

APÊNDICE L – CÓDIGO DE ALINHAMENTO BASEADO EM SVD

```
function [ U, r, lrms, Pt, Qt, newPt ] = procrustes_SVD( P, Q )
%procrustes_SVD Encontra o RMSD mínimo entre dois conjuntos de N pontos tridimensionais e o movimento rígido (translação e rotação/reflexão) que sobrepõe os dois conjuntos de pontos de maneira ótima.
% Os parâmetros esperados são:
% - P: Uma matriz DxN onde P(a,i) é a a-ésima coordenada do i-ésimo ponto do primeiro conjunto de pontos
% - Q: Uma matriz DxN onde Q(a,i) é a a-ésima coordenada do i-ésimo ponto do segundo conjunto de pontos
% A função retorna:
% - U: Uma adequada matriz DxD, representando a rotação/reflexão
% - r: Um vetor coluna D-dimensional, representando a translação
% - lrms: o Last Root Mean Square (Raíz dos Mínimos Quadrados)
% - Pt: O primeiro conjunto de pontos transladado para a origem
% - Qt: O segundo conjunto de pontos transladado para a origem
% - NewPt: O conjunto de pontos P alinhado ao conjunto de pontos Q
sz1 = size(P); %dimensões da matriz de entrada P
sz2 = size(Q); %dimensões da matriz de entrada Q
if ( (length(sz1)  = 2) —— (length(sz2)  = 2) )
        error 'P e Q devem ser Matrizes.';
end
if ( any( sz1  = sz2 ) )
        error 'P e Q devem ter as mesmas dimensões.';
end
N = sz1(2); %número de pontos
m = ones(1,N)/N; %pesos
p0 = P*m'; %centróide de P
q0 = Q*m'; %centróide de Q
```

```
v1 = ones(1,N); %vetor linha com N entradas unitárias
Pt = P - p0*v1; %conjunto de pontos P transladado para o centro da origem
Qt = Q - q0*v1; %conjunto de pontos Q transladado para o centro da origem
Pdm = bsxfun( @times, m, Pt ) ;
C = Pdm*Qt'; %matriz de covariância
[V, ,W] = svd(C); %decomposição SVD
U = W*V'; %matriz de rotação/reflexão ótima
r = q0 - U*p0; %vetor de translação
newPt = U*Pt; %conjuto de pontos P rotacionado
Diff = newPt - Qt; %Pt e Qt são os pontos já centrados
lrms = sqrt( sum( sum( bsxfun( @times, m, Diff ).*Diff ) ) ); %rmsd(P,Q)
end
```