



UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

JESUS MARCOS CAMARGO

Geometria de proteínas no espaço conforme

Campinas

2021

Jesus Marcos Camargo

Geometria de proteínas no espaço conforme

Tese apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Doutor em Matemática Aplicada.

Orientador: Carlile Campos Lavor

Este trabalho corresponde à versão final da Tese defendida pelo aluno Jesus Marcos Camargo e orientada pelo Prof. Dr. Carlile Campos Lavor.

Campinas

2021

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

C14g Camargo, Jesus Marcos, 1989-
Geometria de proteínas no espaço conforme / Jesus Marcos Camargo. –
Campinas, SP : [s.n.], 2021.

Orientador: Carlile Campos Lavor.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de
Matemática, Estatística e Computação Científica.

1. Proteínas. 2. Álgebra geométrica. 3. Espaço conforme. I. Lavor, Carlile
Campos, 1968-. II. Universidade Estadual de Campinas. Instituto de
Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Protein geometry on the conformal space

Palavras-chave em inglês:

Proteins

Geometric algebra

Conformal space

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Carlile Campos Lavor [Orientador]

José Mario Martínez Pérez

Nelson Maculan Filho

Rafael Santos de Oliveira Alves

Michael Ferreira de Souza

Data de defesa: 24-06-2021

Programa de Pós-Graduação: Matemática Aplicada

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0003-1061-1318>

- Currículo Lattes do autor: <http://lattes.cnpq.br/3047537764552800>

**Tese de Doutorado defendida em 24 de junho de 2021 e aprovada
pela banca examinadora composta pelos Profs. Drs.**

Prof(a). Dr(a). CARLILE CAMPOS LAVOR

Prof(a). Dr(a). JOSÉ MARIO MARTÍNEZ PÉREZ

Prof(a). Dr(a). NELSON MACULAN FILHO

Prof(a). Dr(a). RAFAEL SANTOS DE OLIVEIRA ALVES

Prof(a). Dr(a). MICHAEL FERREIRA DE SOUZA

A Ata da Defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria de Pós-Graduação do Instituto de Matemática, Estatística e Computação Científica.

*Ao meu Lucky,
tão querido e tão amado.*

Agradecimentos

Terminar um trabalho como este definitivamente não é uma tarefa fácil, e como foi muito bem colocado por John Donne “Nenhum homem é uma ilha isolada; cada homem é uma partícula do continente, uma parte da terra”. Este trabalho, assinado por mim, é também resultado da influência e do suporte de muitos, e a estes meus agradecimentos.

Primeiramente gostaria de agradecer à minha família por todo o suporte e apoio. À minha mãe, Neusa, por não permitir que eu desistisse do que estava destinado a mim, ainda que eu mesmo não tivesse certeza. Ao meu irmão, Anderson, por todas as orações e torcida e pelo amor implícito que existe entre nós. Ao meu pai, Jair, pelas longas e cansativas viagens, pela disposição em ajudar com todas as atribulações que uma mudança para Campinas envolve. Ainda, a todos os familiares que estiveram torcendo por mim e que certamente compartilham da minha alegria neste momento.

Agradeço a todos os meus amigos, pelo carinho e pela companhia que me proporcionaram durante essa jornada que costuma ser tão solitária, em especial à Carla, que, ainda que distante, esteve sempre ao meu lado durante esses quatro longos anos, obrigado por compartilhar cada momento de alegria e de sofrimento, é impossível para mim quantificar o valor da nossa amizade.

Gostaria de agradecer também a todos os servidores e professores que tornam a UNICAMP a excelente universidade que ela é, em especial aos funcionários e professores do Instituto de Matemática, Estatística e Computação Científica que direta ou indiretamente colaboraram na minha formação.

Agradeço aos Membros da banca de avaliação, Prof. José Mario Martínez, Prof. Nelson Maculan, Prof. Rafael Alves e Prof. Michael de Souza por suas generosas e valiosas contribuições, foi uma honra para mim ter o texto apreciado por vocês.

Agradeço ao Prof. Emerson Castelani, por todo incentivo para que eu continuasse após o mestrado e pela firme recomendação para que eu fosse à UNICAMP, certamente foi uma escolha maravilhosa.

Por fim, agradeço imensamente ao meu orientador, Professor Carlile Lavor, a quem tenho profunda admiração, pelo pesquisador e pela pessoa incrível que o senhor é. Muito obrigado por todos os conselhos, pela paciência e pela serenidade com que o senhor me conduziu por essa jornada. Foi um prazer e uma honra ter sido seu aluno.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

A representação da estrutura de uma proteína é um elemento fundamental para seu estudo. Entre as diversas formas de representação destacam-se o uso de coordenadas internas e coordenadas cartesianas. Modelos para construção e atualização de coordenadas cartesianas, a partir do conjunto de coordenadas internas, têm sido fundamentais na dinâmica molecular. Os modelos apresentados são baseados na técnica pioneira, desenvolvida por Thompson em 1967, e consiste no uso de rotações e translações para posicionar pontos no espaço tridimensional. Diferentes variações deste modelo são abordadas, como a ordem de propagação na cadeia, o encaixamento ou não de translações e o uso de matrizes ou rotores para estabelecer rotações. A partir destes esquemas de construção e atualização, já consolidados, um modelo de representação inovador é desenvolvido utilizando a álgebra geométrica conforme, as coordenadas conformes. Fornecendo uma descrição tão precisa quanto as coordenadas cartesianas, as coordenadas conformes também se mostram muito úteis no cálculo direto das distâncias interatômicas, partindo das coordenadas internas. Com isso, é possível estabelecer, de maneira mais concisa, as derivadas de primeira e segunda ordem destas funções de distância, essenciais no estudo das funções de energia.

Palavras-chave: proteínas. álgebra geométrica. espaço conforme.

Abstract

The representation of a protein structure is a fundamental element for its study. Among the various forms of representation, the use of internal coordinates and Cartesian coordinates stands out. Models for building and updating Cartesian coordinates, based on the set of internal coordinates, have been fundamental in molecular dynamics. The models presented are based on the pioneering technique, developed by Thompson in 1967, which consists of the use of rotations and translations to position points in three-dimensional space. Different variations of this model are addressed, such as the order of propagation in the chain, the fitting or not of translations, and the use of matrices or rotors to establish rotations. From these construction and updating schemes, already consolidated, an innovative representation model is developed using the conformal geometric algebra, the conformal coordinates. Providing a description as accurate as the Cartesian coordinates, the conformal coordinates are also very useful in the direct calculation of interatomic distances, starting from the internal coordinates. With this, it is possible to establish, in a more concise way, the first and second order derivatives of these distance functions, essential in the study of energy functions.

Keywords: protein structure. geometric algebra. conformal space.

Lista de ilustrações

Figura 1 – Esquema de ligações de uma proteína	16
Figura 2 – Estruturas secundária e terciária da molécula LCK.	17
Figura 3 – Coordenadas internas de uma molécula e proteína	18
Figura 4 – Elementos de representação de um <i>backbone</i>	20
Figura 5 – Operações realizadas em e_1 para posicionar um vetor de ligação.	21
Figura 6 – Pontos x_{i-1} , x_i e x_{i+1} e vetores b_i e b_{i+1} representando as ligações C-N-C $_{\alpha}$	21
Figura 7 – Ilustração dos modelos de propagação direta e propagação reversa.	22
Figura 8 – Backbone do problema BBN50 com 50 pontos	26
Figura 9 – Backbone do problema BBN200 com 200 pontos	27
Figura 10 – Backbone do problema BBN1000 com 1000 pontos	27
Figura 11 – Alteração em d_k	30
Figura 12 – Atualização de d_6	32
Figura 13 – Alteração de θ_k	33
Figura 14 – Atualização de θ_6	36
Figura 15 – Alteração em φ_k	37
Figura 16 – Atualização de φ_6	39
Figura 17 – Alteração do exemplo BBN50.	44
Figura 18 – Alterações do exemplo BBN200.	45
Figura 19 – Alterações do exemplo BBN1000.	45
Figura 20 – Representação do bivector $a \wedge b$	50

Lista de tabelas

Tabela 1 – Lista de aminoácias e suas respectivas abreviaturas	16
Tabela 2 – Tempo para conversão de coordenadas internas para cartesianas	28
Tabela 3 – Tempo (s) de execução do algoritmo de atualização com propagação reversa.	44
Tabela 4 – Custo para determinar um novo ponto.	59
Tabela 5 – Tempo para conversão de coordenadas internas para cartesianas com rotores	59
Tabela 6 – Custo computacional para atualização de coordenadas.	63
Tabela 7 – Tempo de execução do algoritmo de atualização com rotores com translações finais.	63
Tabela 8 – Custo para determinar um novo ponto.	79
Tabela 9 – Tempo para conversão de coordenadas internas para conformes	79
Tabela 10 – Tempo (s) para conversão de coordenadas.	80
Tabela 11 – Custo computacional para atualização de coordenadas.	82
Tabela 12 – Tempo de execução (em segundos) da atualização de coordenadas conformes com translações encaixadas.	82
Tabela 13 – Comparação dos algoritmos de atualização de coordenadas.	83
Tabela 14 – Maior diferença, em módulo, entre matrizes de distância.	83
Tabela 15 – Diferença absoluta máxima entre matrizes de distâncias - Alteração por construção.	84
Tabela 16 – Diferença absoluta máxima entre matrizes de distâncias - Alteração por atualização.	84
Tabela 17 – Custo computacional estimado para o cálculo de $r_{i,j}$ e suas derivadas de primeira e segunda ordem.	106
Tabela 18 – Maior diferença absoluta entre as distâncias geradas e a matriz de referência.	106
Tabela 19 – Diferença absoluta entre as derivadas $\frac{\partial r_{i,j}}{\partial \varphi_k}$ calculadas com matrizes e álgebra conforme.	107

Lista de algoritmos

Algoritmo 1 – Conversão de coordenadas com matrizes (espaço homogêneo). . . .	24
Algoritmo 2 – Conversão de coordenadas com matrizes com translações finais . . .	25
Algoritmo 3 – Atualização de coordenadas a partir de uma alteração em d_k	31
Algoritmo 4 – Atualização de coordenadas a partir de uma alteração em φ_k	36
Algoritmo 5 – Atualização de coordenadas a partir de uma alteração em φ_k	39
Algoritmo 6 – Atualização de coordenadas com translações encaixadas.	42
Algoritmo 7 – Atualização de coordenadas com translações finais.	43
Algoritmo 8 – Conversão de coordenadas com rotores.	59
Algoritmo 9 – Atualização com rotores e translações encaixadas.	61
Algoritmo 10 – Atualização com rotores e translações finais.	62
Algoritmo 11 – Construção de coordenadas conformes com translações finais. . .	78
Algoritmo 12 – Construção de coordenadas conformes com translações encaixadas.	78
Algoritmo 13 – Atualização de coordenadas conformes com translações encaixadas.	80
Algoritmo 14 – Atualização de coordenadas conformes com translações finais. . .	82

Sumário

	Introdução	14
1	GEOMETRIA E PROTEÍNAS	15
1.1	Sistemas de representação	16
1.2	Conversão de coordenadas	19
2	MANIPULANDO COORDENADAS A PARTIR DE MUDANÇAS CONFORMACIONAIS	29
2.1	Modificação em d_k	29
2.2	Modificação em θ_k	32
2.3	Modificação em φ_k	37
2.4	Modelos de atualização	40
3	QUATÉRNIOS, ROTORES E A ÁLGEBRA GEOMÉTRICA	46
3.1	Quatérnios	46
3.2	Rotores	48
3.3	O produto externo	49
3.4	O produto geométrico	52
3.5	Produto interno multivetorial	54
4	ESTUDO DE PROTEÍNAS COM ÁLGEBRA GEOMÉTRICA	56
4.1	Conversão de coordenadas com rotores	57
4.2	Atualização de coordenadas cartesianas por rotores	60
5	REPRESENTAÇÃO DE PROTEÍNAS NO ESPAÇO CONFORME	64
5.1	O espaço conforme	64
5.2	Álgebra geométrica conforme	66
5.3	Translações e Rotações no espaço conforme	68
5.4	Sistema de coordenadas conformes	76
5.5	Atualização de coordenadas conformes	80
5.6	Precisão e sensibilidade das coordenadas conformes	83
6	FUNÇÃO DE ENERGIA POTENCIAL	85
6.1	O cálculo de distâncias interatômicas	86
6.2	Derivadas de U	89
6.3	Derivadas de U no espaço conforme	96
6.4	Testes computacionais	104

7	CONSIDERAÇÕES FINAIS	108
	REFERÊNCIAS	110

Introdução

O estudo de proteínas se diversifica em muitos problemas, um deles a representação matemática de sua estrutura tridimensional. Para a representação de proteínas dois sistemas de coordenadas se destacam: as coordenadas internas, constituída por medidas de comprimento, ângulos das ligações químicas e ângulos de torção; e a coordenadas cartesianas, onde a cada ponto do espaço tridimensional corresponde um átomo da molécula.

Estes dois sistemas são codependentes por apresentarem diferentes finalidades. Com isso, o modo como se relacionam tem sido estudado ao longo dos anos, em especial a conversão de coordenadas internas para cartesianas e a atualização de coordenadas cartesianas a partir de modificações em coordenadas internas. A técnica pioneira para conversão, apresentado por [Thompson \(1967\)](#) e donde derivam os métodos descritos neste trabalho, se utiliza de rotações e translações para construção de vetores e pontos que representem adequadamente a estrutura da proteína. Tomando diferentes variações, a respeito de como translações são empregadas e quais operadores de rotação são utilizados, quatro modelos de conversão, são apresentados, juntamente às suas características e custo computacional. Estes mesmos modelos podem ser adaptados, permitindo que coordenadas cartesianas já conhecidas possam ser somente atualizadas, evitando assim que sejam recalculadas a cada mudança conformacional.

Partindo do fato de que rotações são necessárias, tanto na conversão quanto na atualização de coordenadas, diversos estudos passaram a sugerir o uso de rotores, elementos da álgebra dos quatérnios, para a realização de rotações tridimensionais. Isto deu abertura para o uso da chamada álgebra geométrica no estudo de geometria das proteínas e para a adaptação dos métodos já existentes.

Alguns dos métodos apresentados recorrem, ainda, ao uso do espaço homogêneo para permitir que translações tridimensionais sejam dadas por operadores lineares. Diante disso, fomos motivados a explorar a representação em outro espaço, com propriedades adicionais. No chamado espaço conforme, translações tridimensionais não apenas são lineares, mas também ortogonais. Este espaço, de dimensão cinco, permite uma representação do espaço tridimensional similar à do espaço homogêneo, o que nos permitiu desenvolver um novo modelo de representação, que não apenas descreve a estrutura de uma proteína, mas também encapsula o sistema de coordenadas cartesianas.

Além da questão da ortogonalidade dos operadores envolvidos na conversão e atualização de coordenadas, outras características do espaço conforme são aproveitadas na representação de proteínas, como o cálculo da função de distâncias interatômicas e suas derivadas.

1 Geometria e proteínas

Proteínas são macromoléculas compostas por uma cadeia linear de ligações peptídicas entre aminoácidos. Cada aminoácido é formado por um carbono central, chamado *carbono α* , ligado a um grupo amina (NH_3^+), um átomo de hidrogênio, um ácido carboxílico (COO^-) e uma *cadeia lateral*. Esta cadeia lateral é o que diferencia cada um dos vinte aminoácidos existentes. A formação de uma proteína ocorre quando o ácido carboxílico de um aminoácido liga-se a um grupo amina de outro, liberando uma molécula de água. Temos assim uma cadeia polipeptídica, onde cada unidade de aminoácido é chamada de resíduo.

Proteínas desempenham as mais diversas funções em um sistema biológico como o transporte e armazenamento de substâncias, a atuação como catalisadores em processos químicos e a transmissão de impulsos nervosos. Para compreender sua atuação em qualquer processo é fundamental identificá-la e conhecer sua estrutura, que é avaliada em quatro classes.

A primeira classe, chamada estrutura primária, é dada pela sequência de resíduos dos aminoácidos que compõe a molécula. A diversidade de combinações entre os vinte aminoácidos existentes é o que permite a grande variedade de proteínas, e o que as diferencia.

A estrutura secundária corresponde a padrões que se repetem dentro da estrutura molecular. Estes padrões são formados a partir das chamadas ligações não covalentes, interações entre átomos onde não há troca de elétrons, de modo que estas são mais fracas e mais distantes. Pontes de hidrogênio são exemplo destas ligações. Existem dois tipos de estruturas secundárias predominantes no enovelamento das proteínas, as α -hélices e as folhas- β . As α -hélices são estruturas espirais estabilizadas por pontes de hidrogênio ao longo do eixo da hélice. Já folhas- β são estruturas estabilizadas por pontes de hidrogênio entre cadeias polipeptídicas adjacentes. Estas estruturas estão ilustradas na Figura 2.

A estrutura terciária é formada por interações de longa distância entre os aminoácidos. Desse modo, na estrutura terciária temos a organização tridimensional da proteína como um todo, e não apenas uma visão local. São estas estruturas que definem diretamente as funções de uma proteína, de modo que, conhecer sua estrutura terciária é fundamental para compreender seu funcionamento. Aqui está nosso maior interesse ao desenvolver este trabalho. Usaremos ferramentas matemáticas para descrever e manipular a estrutura tridimensional da molécula.

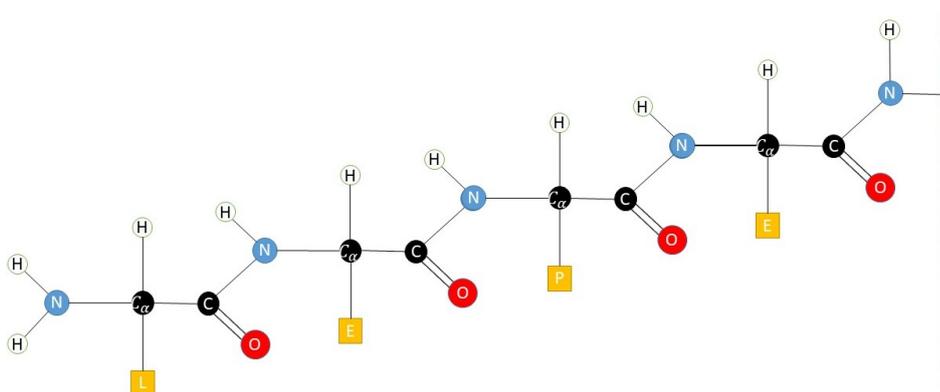
Por fim, temos a estrutura quaternária, referente a interação entre duas ou mais cadeias, cada uma chamada subunidade, e sua disposição espacial. É interessante ressaltar

que do ponto de vista estrutural talvez este seja o nível de menor importância, uma vez que a estrutura terciária descreve com mais propriedade as funções da proteína. Mais informações a respeito das estruturas e propriedades das proteínas podem ser encontradas em Berg, Tymoczko e Stryer (2008).

A fim de exemplificar as três estruturas de maior importância (primária, secundária e terciária), vamos considerar uma molécula de tirosina quinase LCK, cujos dados foram extraídos do RCSB PDB (PDB ID: 1LKL).

A molécula de LCK é composta por um total de 105 aminoácidos, sua estrutura primária é dada pela sequência LEPEPWFFKNLSRKAERQLLAPGNTGHSFLIRESESTAG-SFSLSVRDFDQNQGEVVKHYKIRNLDNGGFYISPRITFPGLHELVRHYTNASDGLC-TRLSRPCQT. O esquema de ligação dos primeiros 4 aminoácidos é ilustrado na Figura 1, mostrando o padrão de ligações de uma cadeia. A lista dos 20 aminoácidos e cada letra que o representa são apresentados na tabela 1. As estruturas secundárias, α -hélice e folha- β , são destacadas na estrutura terciária da molécula LCK na figura 2.

Figura 1 – Esquema de ligações de uma proteína



Fonte: Elaborada pelo autor.

Tabela 1 – Lista de aminoácias e suas respectivas abreviaturas

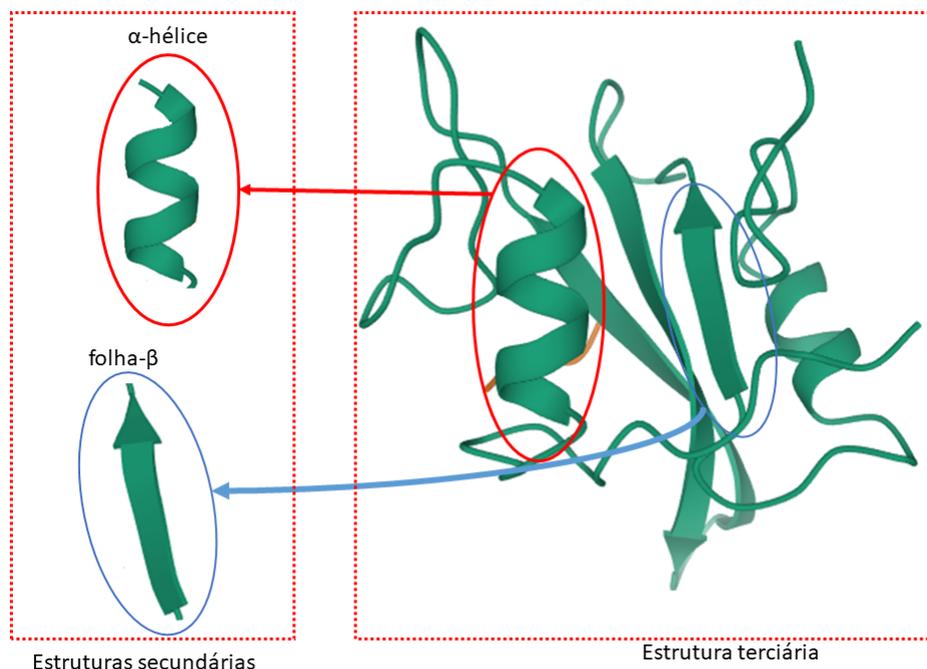
Aminoácido	Sigla	Aminoácido	Sigla	Aminoácido	Sigla	Aminoácido	Sigla
Alanina	A	Fenilalanina	F	Isoleucina	I	Serina	S
Arginina	R	Glicina	G	Leucina	L	Tirosina	Y
Aspartato	D	Ácido glutâmico	E	Lisina	K	Treonina	T
Asparagina	N	Glutamina	Q	Metionina	M	Triptofano	W
Cisteína	C	Histidina	H	Prolina	P	Valina	V

Fonte: Elaborada pelo autor.

1.1 Sistemas de representação

O enovelamento de uma molécula de proteína depende diretamente da sequência de aminoácidos que a compõe, mas, ainda que seja possível determinar tal sequência,

Figura 2 – Estruturas secundária e terciária da molécula LCK.



Fonte: Elaborada pelo autor.

estabelecer sua forma tridimensional partindo somente desta informação ainda é um enorme desafio. Neste trabalho, focaremos em modelos para descrever este enovelamento. Para isso, vamos utilizar estruturas geométricas que a representem adequadamente.

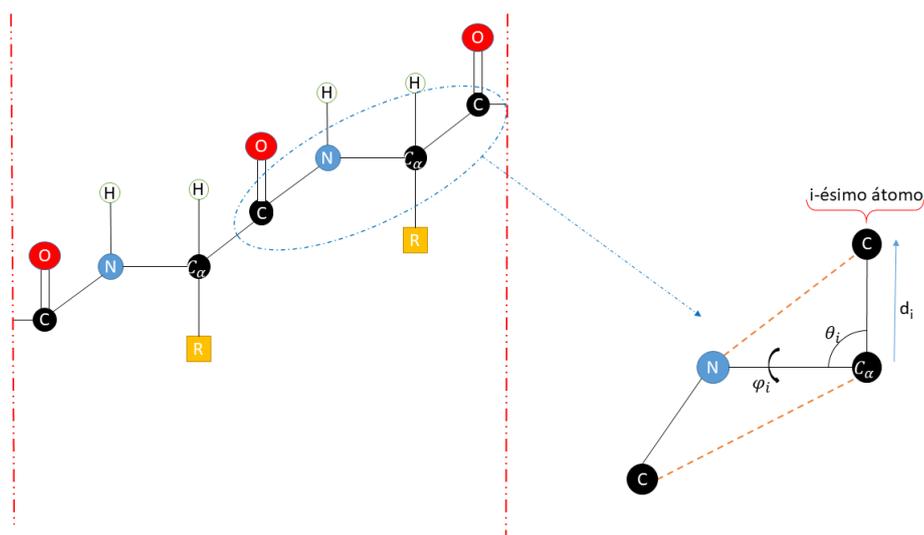
Tratando-se de estruturas espaciais, é intuitivo estabelecer uma primeira representação no espaço euclidiano tridimensional. Assim, para descrever a estrutura de uma proteína em \mathbb{R}^3 faremos cada átomo da molécula corresponder a um ponto do espaço. Neste modelo, a molécula é representada pelo conjunto de coordenadas cartesianas representantes dos átomos que a compõe. Indicaremos tais coordenadas ao longo do texto por C_C .

Uma alternativa talvez menos intuitiva, mas bastante conhecida, é a representação por coordenadas internas. Este modelo de representação usa propriedades geométricas das ligações químicas para descrever a estrutura da molécula. No chamado sistema de coordenadas internas, a posição de cada átomo é determinada a partir das distâncias entre átomos covalentes, ângulos formados entre duas ligações atômicas e ângulos de torção. Deste modo, cada molécula é representada por um conjunto de coordenadas triplas, aqui indicadas por C_I , com informações locais baseadas na estrutura da própria molécula. Uma apresentação mais aprofundada sobre coordenadas internas pode ser encontrada em [Todeschini e Consonni \(2009\)](#).

Para ilustrar o conjunto de coordenadas internas vamos tomar como exemplo um certo átomo de carbono em uma cadeia polipeptídica. Considerando este o i -ésimo átomo da cadeia, a primeira componente de sua coordenada interna, indicada por d_i , é

dada pelo comprimento da ligação covalente com o átomo C_α anterior, ou seja, d_i é o comprimento da ligação C_α -C. A segunda componente, θ_i , é dada pela medida do ângulo formado entre as duas ligações covalentes imediatamente anteriores ao átomo C, ou seja as ligações N- C_α e C_α -C. Por fim, a terceira componente, φ_i , é dada pela medida do ângulo diedral formado entre dois planos determinados pelas três ligações covalentes imediatamente anteriores ao átomo C. As ligações N- C_α -C determinam um plano, digamos π_2 , enquanto as ligações C-N- C_α (átomo C da unidade de aminoácido anterior na cadeia) formam um plano π_1 . Assim, a terceira componente, chamada ângulo de torção, é dada pelo ângulo formado entre os planos π_1 e π_2 . Neste caso a posição do i -ésimo átomo da cadeia tem coordenadas $(d_i, \theta_i, \varphi_i)$. As identificações das coordenadas internas de C são ilustradas na Figura 3.

Figura 3 – Coordenadas internas de uma molécula e proteína



Fonte: Elaborado pelo autor.

No modelo de representação por coordenadas cartesianas cada átomo é representado por coordenadas independentes. Isto significa que mudanças conformacionais (mudanças no formato da molécula) devem ser descritas atualizando todas coordenadas dos pontos cuja posição é afetada. Isto ocorre porque este modelo ignora, sob certos aspectos, as relações e interações entre átomos. Coordenadas internas, por outro lado, descrevem de maneira mais natural a estrutura da proteína. Como cada átomo é descrito por um conjunto de coordenadas locais, este sistema evidencia melhor as relações entre eles.

Conhecimentos químicos a respeito das ligações covalentes permitem saber, a ponto de serem consideradas constantes, as distâncias entre ligações covalentes e o ângulo entre duas ligações consecutivas. Deste modo, a liberdade de variação das coordenadas internas depende somente das componentes referentes aos ângulos de torção. Além disso, vale ressaltar que as coordenadas $d_1, \theta_1, \varphi_1, \theta_2, \varphi_2$ e φ_3 podem assumir qualquer valor, sem

interferir na conformação da proteína. Isto ocorre por que não existem pontos anteriores a posição 1 na cadeia, ou seja, estas medidas de distâncias e ângulos são estabelecidas a partir do que se conhece como “pontos virtuais”, cuja posição é irrelevante. Neste trabalho consideraremos tais coordenadas sempre iguais a zero, permitindo assim uma melhor estética aos métodos apresentados.

Ressaltadas as diferenças entre os dois sistemas de representação, é importante destacar que estes não são mutuamente excludentes. Por um lado, o uso das coordenadas internas permite reduzir o grau de liberdade nas mudanças conformacionais, e por essa razão tornam-se muito úteis em estudos da geometria molecular. Por outro lado, coordenadas cartesianas são extremamente úteis ao lidar com funções energia, centros de massa da molécula e funções de *score* de acoplamento (CHYS; CHACÓN, 2012). Assim mais que conhecer ambos os sistemas, é também importante estabelecer como transitar entre eles.

1.2 Conversão de coordenadas

Coordenadas internas podem ser facilmente obtidas a partir de coordenadas cartesianas utilizando conhecimentos básicos da geometria analítica. No entanto, obter coordenadas cartesianas a partir de um conjunto de coordenadas internas é uma tarefa um pouco mais complexa. Uma série de métodos foram criados e estudados a fim de obter algoritmos rápidos e eficientes para realização de tal conversão.

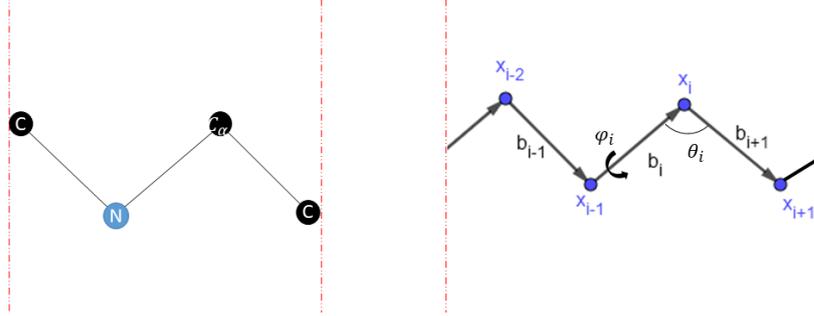
O método pioneiro, e ainda hoje um dos mais proeminentes, constrói tais coordenadas utilizando as tradicionais matrizes de rotação do espaço tridimensional euclidiano. Este modelo, apresentado por Thompson (1967), propõe a construção das coordenadas cartesianas a partir das ligações covalentes, representadas por vetores de ligação. Estes vetores são posicionados, por meio de rotações, em ângulos de ligação e torção apropriados. Por fim, as coordenadas de cada ponto são determinadas a partir do conjunto de vetores calculados.

Para compreender tanto a técnica elaborada por Thompson, quanto os demais modelos que aqui serão apresentados, vamos estabelecer alguns elementos utilizados na representação de uma proteína. Como mencionamos anteriormente, estamos interessados em descrever sua forma tridimensional e, dada sua composição, podemos descrevê-la analisando apenas a sequência de átomos centrais C, N e C_α. Se observarmos a estrutura completa, as ligações entre tais átomos formam a cadeia linear propriamente dita, enquanto os átomos remanescentes podem ser entendidos como ramificações. Embora todo o tratamento apresentado neste trabalho possa ser adaptado para uma descrição completa da molécula, atentaremos apenas a esta cadeia principal, conhecida como *backbone* da proteína. Esta abordagem é suficiente para resolver uma grande variedade de problemas.

Por padrão, adotaremos a seguinte notação: o ponto x_i representa o i -ésimo

átomo do *backbone*; o vetor b_i , chamado vetor de ligação, representa a ligação entre os átomos x_{i-1} e x_i (podemos identificar $b_i = x_i - x_{i-1}$); o comprimento da ligação b_i será denotado por $d_i = \|b_i\|$; θ_i , corresponde a medida do ângulo de ligação formado pelos átomos x_{i-1} , x_i e x_{i+1} e por fim, φ_i , corresponde o ângulo diedral formado entre o plano gerado pelos vetores $\{b_i, b_{i-1}\}$ e o plano gerado por $\{b_{i-1}, b_{i-2}\}$. A representação geométrica do *backbone* é ilustrada na Figura 4.

Figura 4 – Elementos de representação de um *backbone*.



Fonte: Elaborado pelo autor.

O modelo apresentado por Thompson propõe que a construção das coordenadas cartesianas seja realizada a partir da construção dos vetores de ligação. Partindo-se de um vetor que servirá como base para construção de cada b_i , por padrão escolhemos e_1 , vamos rotacioná-lo em torno do eixo orientado por e_3 a fim de posicionar o ângulo de ligação θ_i (medido entre o vetor em questão e o eixo orientado por e_1). Na sequência, aplicaremos uma nova rotação em torno do eixo orientado por e_1 para posicionar o ângulo de torção φ_i (medido a partir do plano gerado por $\{e_1, e_2\}$). Por fim, para ajustar o comprimento da ligação, aplicamos no vetor uma contração/extensão utilizando d_i . As etapas deste procedimento são ilustradas na Figura 5, onde B^θ indica a rotação em torno de e_3 e B^φ indica a rotação em torno de e_1 .

A matriz de rotação em torno de e_3 e a rotação em torno de e_1 são dadas, respectivamente, pelas matrizes

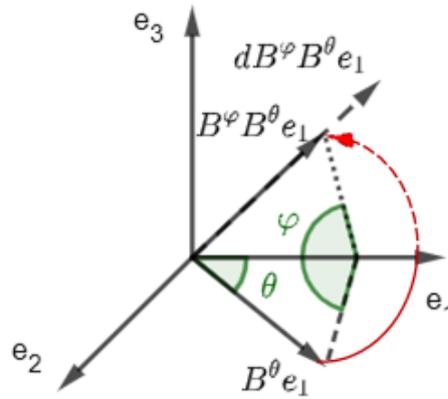
$$M_{e_3}(\theta) = \begin{bmatrix} \cos \theta & -\text{sen } \theta & 0 \\ \text{sen } \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

e

$$M_{e_1}(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \varphi & -\text{sen } \varphi \\ 0 & \text{sen } \varphi & \cos \varphi \end{bmatrix}.$$

Devemos, ainda, atentar para a necessidade de uma correção no ângulo de rotação na matriz M_{e_3} . Observe que o ângulo medido entre as ligações não corresponde ao

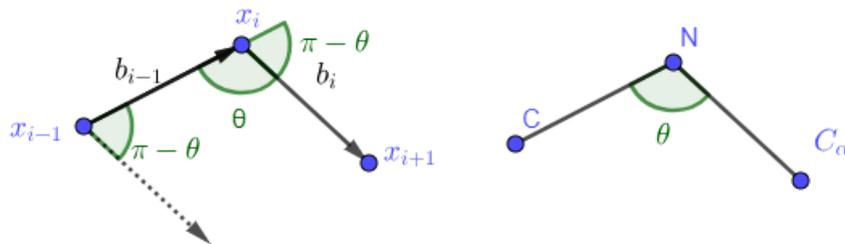
Figura 5 – Operações realizadas em e_1 para posicionar um vetor de ligação.



Fonte: Elaborado pelo autor.

ângulo formado entre os vetores de ligação. Conforme indicado em [Seok e Coutsiás \(2007\)](#), a relação entre tais ângulos é dada por uma diferença de π . Se o ângulo da ligação mede θ , o ângulo formado pelos vetores de ligação correspondentes será dado por $\pi - \theta$, como pode-se observar na Figura 6.

Figura 6 – Pontos x_{i-1} , x_i e x_{i+1} e vetores b_i e b_{i+1} representando as ligações C-N- C_α .



Fonte: Elaborada pelo autor.

Portanto, para obter o ângulo θ , devemos usar a matriz de rotação M_{e_3} com ângulo $\pi - \theta$. Logo,

$$B^\theta = \begin{bmatrix} -\cos \theta & -\sin \theta & 0 \\ \sin \theta & -\cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Uma vez que não se faça necessária nenhuma correção para a rotação φ , temos, simplesmente, $B^\varphi = M_{e_1}(\varphi)$.

Como as rotações são aplicadas em sequência no mesmo vetor, estas podem ser codificadas em uma única matriz, permitindo que ambas as rotações sejam realizadas simultaneamente. Desta maneira, a cada etapa, podemos determinar as duas rotações por

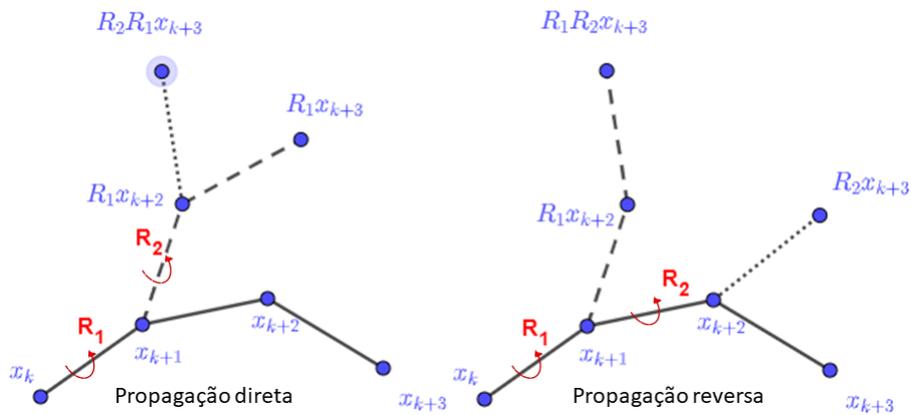
$B_i = B^{\varphi_i} B^{\theta_i}$. Por conseguinte,

$$B_i = \begin{bmatrix} -\cos \theta_i & -\sin \theta_i & 0 \\ \sin \theta_i \cos \varphi_i & -\cos \theta_i \cos \varphi_i & -\sin \varphi_i \\ \sin \theta_i \sin \varphi_i & -\cos \theta_i \sin \varphi_i & \cos \varphi_i \end{bmatrix}. \quad (1.1)$$

Dito isto, a construção das coordenadas cartesianas é dada pelo seguinte procedimento. O primeiro ponto da cadeia será fixado na origem do espaço tridimensional. Para a determinação dos demais pontos, cada vetor de ligação será dado por $B_i e_1$, o que implica que o ponto x_i será determinado pela soma de todos os vetores de ligação anteriores a i . É importante notar que o vetor $B_i e_1$ tem ângulos de torção e de ligação ajustados em relação aos eixos coordenados. Assim, é necessário que rotações anteriores a B_i sejam aplicadas a este para que os ângulos sejam posicionados corretamente em relação às ligações anteriores. Estas rotações anteriores devem ser aplicadas em ordem reversa para evitar o reposicionamento dos eixos de rotação. Vejamos, antes de seguir, a importância da escolha da ordem de propagação conforme descrito por [Chys e Chacón \(2012\)](#).

Na chamada propagação direta, cada rotação é realizada conforme a ordem de construção da cadeia. Isto significa que a ordem em que as rotações anteriores são aplicadas coincide com a ordem em que os ângulos aparecem na cadeia. Isto é, antes de realizar a translação pelo vetor de ligação determinado, aplica-se em ordem crescente todas as rotações que posicionaram corretamente os ângulos anteriores. O problema em aplicar as rotações desta forma é que os eixos de rotação devem ser reposicionados, pois são afetados pelas rotações anteriores. Para contornar tal situação, usamos a chamada propagação reversa. Na propagação reversa cada vetor é rotacionado partindo do último ângulo modificado para o primeiro. Ou seja, as rotações são realizadas em ordem reversa à ordem da cadeia. A figura 7 ilustra como funcionam a propagação direta e reversa.

Figura 7 – Ilustração dos modelos de propagação direta e propagação reversa.



Fonte: Elaborado pelo autor.

Voltando à construção das coordenadas cartesianas, o procedimento consiste em posicionar os ângulos de cada ligação e ajustar o comprimento do vetor, para então

somar os vetores de ligação anteriores e aplicar todas as rotações anteriores em ordem reversa. Em termos matemáticos, temos a seguinte sequência:

$$\begin{aligned}
 x_1 &= d_1 B_1 e_1 \\
 x_2 &= B_1(d_2 B_2 e_1) + d_1 B_1 e_1 \\
 x_3 &= B_1(B_2(d_3 B_3 e_1) + d_2 B_2 e_1) + d_1 B_1 e_1 \\
 &\vdots \\
 x_i &= B_1(B_2 \cdots B_{i-1}(d_i B_i e_1) + d_{i-1} B_{i-1} e_1) + \cdots + d_2 B_2 e_1) + d_1 B_1 e_1. \quad (1.2)
 \end{aligned}$$

Observe que, nas igualdades acima, temos $x_1 = d_1 B_1 e_1 = 0$ e $x_2 = B_1(d_2 B_2 e_1) + d_1 B_1 e_1 = d_2 B_2 e_1$. Isto decorre do fato de termos estabelecido as coordenadas $d_1 = \theta_1 = \varphi_1 = 0$.

Olhando a sequência que gera os pontos, podemos observar que cada vetor é rotacionado e transladado a cada etapa. Objetivando simplificar este processo, Thompson propõe, alternativamente, o uso do espaço homogêneo para permitir a linearização destas equações. O chamado espaço homogêneo, detalhado apropriadamente em [Kanatani \(2015\)](#), nos permite trabalhar com uma representação do espaço tridimensional dentro do espaço de dimensão quatro. Para isso, utilizamos um hiperplano paralelo ao subespaço gerado por $\{e_1, e_2, e_3\}$ e contendo o ponto e_4 . Deste modo, cada vetor $x \in \mathbb{R}^3$ é representado em \mathbb{R}^4 por $\begin{bmatrix} x \\ 1 \end{bmatrix}$.

A principal vantagem ao se trabalhar com o espaço homogêneo é que translações do espaço tridimensional são dadas por operadores lineares de \mathbb{R}^4 . Para ilustrar como isso acontece, vamos observar a Equação (1.3). Sendo $x, t \in \mathbb{R}^3$ e I_3 a matriz identidade de ordem 3, o produto matriz vetor, realizado em \mathbb{R}^4 resulta na translação do vetor x de \mathbb{R}^3 representada no espaço homogêneo:

$$\begin{bmatrix} I_3 & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ 1 \end{bmatrix} = \begin{bmatrix} x + t \\ 1 \end{bmatrix}. \quad (1.3)$$

Podemos, portanto, construir matrizes de ordem 4 que, além das rotações, codifiquem também as translações necessárias em cada etapa da construção. A linearidade das translações tridimensionais faz com que cada ponto seja diretamente calculado e todas as rotações e translações sejam combinadas em uma sequência de operadores lineares, evitando o cálculo por etapas.

O cálculo de coordenadas cartesianas pode, então, ser realizado da seguinte maneira. Se \hat{B}_i é a matriz de rotação de ordem 3, determinada pelos ângulos de ligação e de torção, e b_i o vetor de ligação, a matriz que codifica a rotação \hat{B}_i e a translação por

b_i é dada no espaço homogêneo por $B_i = \begin{bmatrix} \hat{B}_i & b_i \\ 0 & 1 \end{bmatrix}$. Uma vez que $b_i = d_i \hat{B}_i e_1$, então

$$B_i = \begin{bmatrix} \hat{B}_i & d_i \hat{B}_i e_1 \\ 0 & 1 \end{bmatrix}.$$

Colocando a matriz em termos das coordenadas internas, temos

$$B_i = \begin{bmatrix} -\cos \theta_i & -\sin \theta_i & 0 & -d_i \cos \theta_i \\ \sin \theta_i \cos \varphi_i & -\cos \theta_i \cos \varphi_i & -\sin \varphi_i & d_i \sin \theta_i \cos \varphi_i \\ \sin \theta_i \sin \varphi_i & -\cos \theta_i \sin \varphi_i & \cos \varphi_i & d_i \sin \theta_i \sin \varphi_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.4)$$

O processo de construção das coordenadas cartesianas pode ser então resumido pelo Algoritmo 1.

Algoritmo 1: Conversão de coordenadas com matrizes (espaço homogêneo).

Entrada: $C_I = \{(d_i, \theta_i, \varphi_i), i = 1, \dots, n\}$

Saída: $C_C = \{x_i, i = 1, \dots, n\}$

início

$x_1 \leftarrow (0, 0, 0, 1);$

$x_2 \leftarrow (d_2, 0, 0, 1);$

$B_1 \leftarrow I_3;$

para $k = 3, \dots, n$ **faça**

 Calcule $B_k;$

$B_{[k]} \leftarrow B_{[k-1]} B_k;$

$x_k \leftarrow B_{[k]} e_4;$

fim

fim

O custo computacional deste método é estimado pelo número de operações requeridas para adição de um átomo do *backbone*, excluindo-se o cálculo de multiplicações por 1 e somas por 0. Esta estimativa determina um custo total de 46 operações, sendo as seguintes: 12 multiplicações e 6 somas para o cálculo do produto matricial $B_{[i-1]} B_i^\varphi$; 12 multiplicações e 6 somas para o cálculo do produto matricial de $(B_{[i-1]} B_i^\varphi) B_i^\theta$; 3 multiplicações no ajuste do comprimento do vetor de ligação ($d_i b_i$); 3 somas na translação do vetor de ligação para posicionamento do i -ésimo átomo ($x_i = x_{i-1} + (d_i b_i)$). Este é o procedimento com número mínimo de operações usando matrizes conforme descrito em [Seok e Coutsiás \(2007\)](#). Não consideraremos na contagem operações trigonométricas por serem iguais em todos os métodos descritos ao longo do texto.

Olhando para a Equação (1.2), podemos aplicar a propriedade de distributividade, implicando que cada ponto da sequência é dado por

$$x_i = d_1 B_1 e_1 + d_2 B_1 B_2 e_1 + d_3 B_1 B_2 B_3 e_1 + \dots + d_i B_1 B_2 B_3 \dots B_i e_1. \quad (1.5)$$

Esta equação pode ser escrita em uma notação mais simplificada, substituindo o produto $B_1 B_2 B_3 \cdots B_i$ por $B_{[i]}$. Assim,

$$x_i = (d_1 B_{[1]} + d_2 B_{[2]} + d_3 B_{[3]} + \cdots + d_i B_{[i]}) e_1. \quad (1.6)$$

Isso nos permite determinar uma nova maneira de construir coordenadas, mais intuitiva e que oferecerá uma melhor percepção do processo de construção, permitindo outras abordagens que serão exploradas ao longo do texto.

Observe que, de acordo com a Equação (1.5), podemos determinar as coordenadas cartesianas do ponto x_i a partir de uma única translação do ponto anterior por meio do vetor de ligação. De fato, sabemos que $x_i = x_{i-1} + b_i$, também que b_i é determinado a partir de rotações B_i que posicionam os ângulos de ligação e de torção, e pelo produto por d_i que ajusta o comprimento do vetor. Mostramos ainda que precisamos aplicar todas as rotações anteriores por meio de propagação reversa para que os ângulos sejam corretamente posicionados. Como resultado temos $b_i = d_i B_1 B_2 B_3 \cdots B_i e_1$, confirmando a igualdade (1.5). A esta variação, chamaremos **construção por translações finais**, uma vez que determinamos o vetor de ligação para, então, aplicamos uma única translação ao final do processo. Para diferenciar o modelo elaborado por Thompson, iremos nos referir a ele como **construção por translações encaixadas**.

Vamos agora estabelecer o Algoritmo 2 que faz uso de translações finais para construção de coordenadas cartesianas.

Algoritmo 2: Conversão de coordenadas com matrizes com translações finais

Entrada: $C_I = \{(d_i, \theta_i, \varphi_i), i = 1, \dots, n\}$

Saída: $C_C = \{x_1, x_2, x_3 \cdots, x_n\}$

início

$x_1 \leftarrow (0, 0, 0);$

$x_2 \leftarrow (d_2, 0, 0);$

$B_1 \leftarrow I_3;$

para $k = 3, \dots, n$ **faça**

 Calcule $B_k;$

$B_{[k]} \leftarrow B_{[k-1]} B_k;$

$x_k \leftarrow x_{k-1} + d_k B_{[k]} e_1;$

fim

fim

É importante observar que as matrizes B_i , presentes no Algoritmo 2, são matrizes de ordem 3, e que realizam somente rotações. Enquanto que, no Algoritmo 1, B_i 's são matrizes de ordem 4 que realizam tanto translações quanto rotações. Outro fato que pode chamar atenção é que o modelo originalmente proposto por Thompson

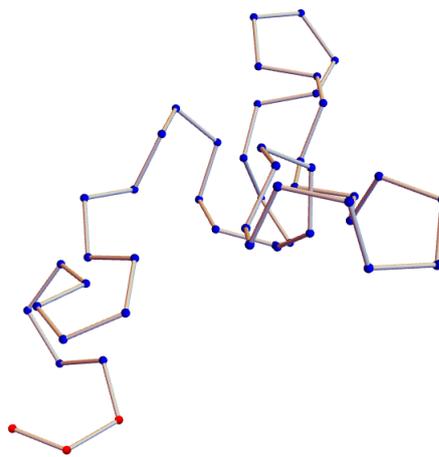
determina a construção da estrutura com base no vetor $-e_1$, alterando sua orientação. Contudo, partindo do mesmo conjunto de coordenadas internas e tomando como base para a construção o vetor e_1 , o resultado difere simplesmente por uma rotação do conjunto de coordenadas cartesianas. Sendo assim, visando uma padronização dos processos de construção, optaremos por manter e_1 e adequar as matrizes de ordem 4. Lembramos, ainda, que tal vetor é computado dentro da matriz.

Como o modelo homogêneo parte da formulação do modelo matricial, as operações realizadas em ambos são as mesmas, de modo que é possível estabelecê-los com o mesmo número de operações. Visto que ambos os modelos podem ser realizados com mesmo custo computacional, a vantagem para o modelo homogêneo está na linearização da equação de conversão.

O modelo homogêneo com translações encaixadas foi implementado no *software Wolfram Mathematica v 11.3*, onde calculamos exemplos gerados artificialmente simulando o *backbone* de proteínas com 50, 200 e 1000 átomos. As coordenadas internas foram estabelecidas com valores fixos de 1.5 para distâncias de ligação e 1.91 rad para ângulo de ligação. Para o ângulo de torção, valores aleatórios foram escolhidos no intervalo $[-\pi, \pi]$. Essas escolhas foram feitas em conformidade com o descrito em [Fernandes, Lavor e Oliveira \(2017\)](#), e geram instâncias adequadas para o problema real de proteínas. Usaremos estes três exemplos para avaliar todos os modelos discutidos neste texto.

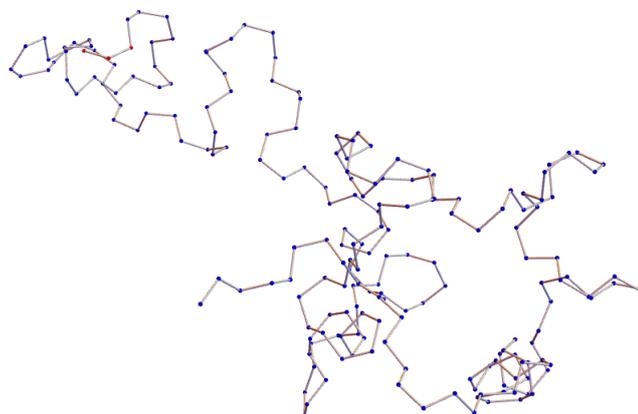
As Figuras 8, 9 e 10 ilustram graficamente os *backbones* gerados a partir dos problemas BBN50, BBN200 e BBN1000, respectivamente com 50, 200 e 1000 pontos.

Figura 8 – Backbone do problema BBN50 com 50 pontos .



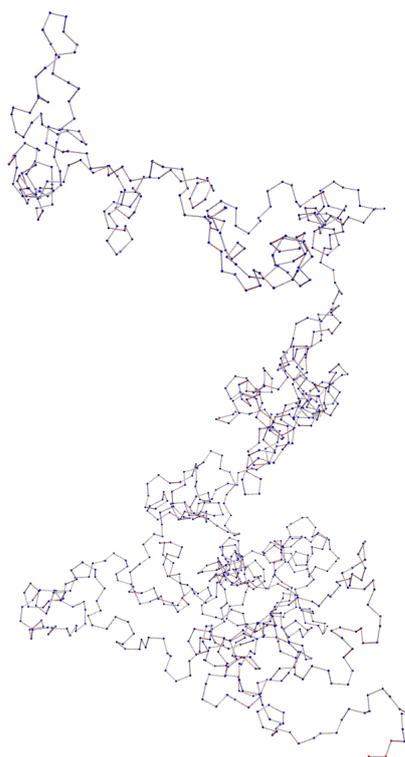
Fonte: Elaborado pelo autor.

Figura 9 – Backbone do problema BBN200 com 200 pontos .



Fonte: Elaborado pelo autor.

Figura 10 – Backbone do problema BBN1000 com 1000 pontos .



Fonte: Elaborado pelo autor.

A fim de melhor ilustrar nossa implementação, calculamos o tempo de execução do método para cada um dos conjuntos. Estes dados estão dispostos na Tabela 2.

Tabela 2 – Tempo para conversão de coordenadas internas para cartesianas

Conjunto	Tempo (s)
BBN50	0
BBN200	0.015
BBN1000	0.03

2 Manipulando coordenadas a partir de mudanças conformacionais

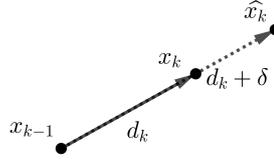
No capítulo anterior, falamos sobre os dois principais sistemas de representação de uma molécula e apresentamos técnicas para criação de coordenadas cartesianas a partir de um conjunto de coordenadas internas. No entanto, existe um outro elemento de grande importância nos estudos de proteínas, as matrizes de distâncias interatômicas. Tal matriz é composta por todas as distâncias entre os pares de átomos da proteína, para representar a distância entre os átomos x_i e x_j usaremos r_{ij} . Sabemos que proteínas atuam em diversos processos biológicos e, para tal, podem sofrer mudanças conformacionais. Isto significa que a estrutura de uma proteína tem regiões que se movem e regiões que permanecem imutáveis. Já mencionamos também que essas mudanças conformacionais são melhores descritas por coordenadas internas (CHOI, 2006). Deste modo, buscaremos, agora, entender como a representação cartesiana descreve tais mudanças e como estas afetam as distâncias interatômicas.

Em uma grande variedade de situações teóricas, os valores de distâncias e ângulos entre ligações covalentes podem ser tomados como fixos e conhecidos, de modo que a liberdade de movimento da molécula é dependente somente dos ângulos de torção e por esta razão avaliaremos de maneira detalhada tal situação. Além disso, optamos por também incluir uma breve análise da variação das distâncias e ângulos de ligação, pois entendemos que tal análise eventualmente poderá servir de aporte teórico para outras aplicações onde as coordenadas podem ser trabalhadas como variáveis livres.

2.1 Modificação em d_k

Iniciando pelo caso mais simples, vamos determinar como a alteração do comprimento de uma ligação covalente afeta a matriz de distância. Para tais estudos, vamos utilizar as equações com translações finais, que irão simplificar consideravelmente os cálculos.

Vamos considerar que a distância modificada se dá entre os átomos x_k e x_{k-1} , ou seja, a coordenada d_k sofre uma alteração do tipo $d_k + \delta$, conforme a Figura 11.

Figura 11 – Alteração em d_k .

Fonte: Elaborado pelo autor.

Da Equação (1.6), sabemos que as coordenadas cartesianas de um átomo são representadas em termos das coordenadas internas por

$$x_k = x_1 + d_2 B_{[2]} b_2 + d_3 B_{[3]} b_3 + \cdots + d_k B_{[k]} b_k, \quad (2.1)$$

onde d_i são os comprimentos das ligações covalentes e $B_{[i]} = B_2^{(\varphi)} B_2^{(\theta)} B_3^{(\varphi)} B_3^{(\theta)} \dots B_i^{(\varphi)} B_i^{(\theta)}$ é o conjunto de rotações determinadas pelas componentes angulares das coordenadas internas (ângulos de ligação e torção).

Vejamos o que ocorre com as distâncias $r_{ij} = \|x_i - x_{i-1}\|$ ao atualizarmos a coordenada interna d_k para $d_k + \delta$. Para isso, analisaremos três casos:

1. $i, j < k$

Neste caso, $r_{i,j}$ não se altera, pois o termo d_k não é computado no cálculo de x_i nem de x_j , conforme pode ser observado na Equação (1.6).

2. $i, j \geq k$

Dados \hat{x}_i e \hat{x}_j , atualizações de x_i e x_j , respectivamente, podemos escrever

$$\begin{aligned} \hat{x}_i &= x_1 + d_2 B_{[2]} b_2 + \cdots + (d_k + \delta) B_{[k]} b_k + \cdots + d_i B_{[i]} b_i \\ &= x_1 + d_2 B_{[2]} b_2 + \cdots + d_k B_{[k]} b_k + \cdots + d_i B_{[i]} b_i + \delta B_{[k]} b_k \\ &= x_i + \delta B_{[k]} b_k. \end{aligned}$$

Analogamente,

$$\hat{x}_j = x_j + \delta B_{[k]} b_k.$$

Assim,

$$\begin{aligned} \|\hat{x}_i - \hat{x}_j\| &= \|(x_i + \delta B_{[k]} b_k) - (x_j + \delta B_{[k]} b_k)\| \\ &= \|x_i - x_j\|. \end{aligned}$$

Logo, para o caso $i, j \geq k$ as distâncias não são alteradas.

3. $i < k$ e $j \geq k$

Neste caso, como $i < k$, temos que $\hat{x}_i = x_i$. Por outro lado, $j \geq k$, então $\hat{x}_j = x_j + \delta B_{[k]} b_k$. Deste modo, temos

$$\begin{aligned} \|\hat{x}_i - \hat{x}_j\| &= \|x_i - (x_j + \delta B_{[k]} b_k)\| \\ &= \|(x_i - x_j) - \delta B_{[k]} b_k\|, \end{aligned}$$

o que implica que as distâncias são alteradas e a alteração é dada em termos do vetor $\delta B_{[k]} b_k$.

Portanto, podemos concluir que, para alteração em uma entrada d_k , somente as distâncias $r_{i,j}$, com $i < k$ e $j \geq k$, precisam ser atualizadas.

Com base nessas informações podemos concluir uma mudança em d_k somente gera alteração nas coordenadas cartesianas de pontos posteriores a x_k . A atualização dessas coordenadas é realizada por meio de uma translação a partir do vetor de ligação b_k , que é incrementado pelo valor δ . O processo de atualização das coordenadas está descrito no Algoritmo 3.

Algoritmo 3: Atualização de coordenadas a partir de uma alteração em d_k .

Entrada: Coordenadas cartesianas, $\hat{d}_k = d_k + \delta$

Saída: Coordenadas cartesianas atualizadas

início

$b_k \leftarrow x_k - x_{k-1};$

para $i = k, \dots, n$ **faça**

$x_i \leftarrow x_i + \delta b_k;$

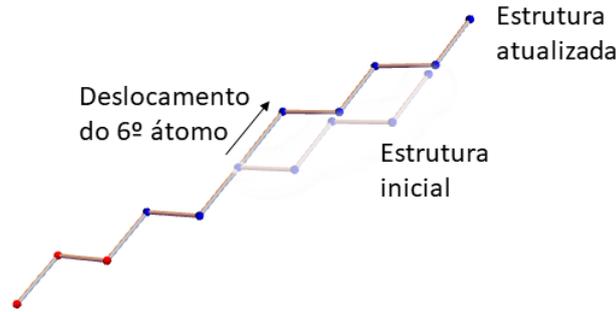
fim

fim

Para ilustrar os cálculos apresentados, vamos tomar como exemplo um conjunto de coordenadas internas bastante particular. Seja M um conjunto com dez pontos, cujas coordenadas internas são: $d_1 = 0$ e $d_k = 1.5$ para todo $k > 1$; $\theta_k = (-1)^k 1.91$ e $\varphi_k = 0$ para todo k . Embora tal conjunto se afaste da realidade das proteínas, ele permite uma ilustração mais clara das modificações realizadas a cada atualização. Utilizaremos este mesmo conjunto para ilustrar os demais casos.

Vamos determinar uma alteração $\delta = 1.8$ em d_6 . Na Figura 12, podemos observar a forma inicial do conjunto de pontos gerados por M e o resultado, após a alteração da distância.

Também podemos observar as alterações ocorridas na matriz de distâncias dos pontos. Para isso, vamos considerar D , a matriz de distâncias da estrutura de pontos inicial, e \hat{D} , a matriz de distâncias da estrutura atualizada. Deste modo, as entradas

Figura 12 – Atualização de d_6


Fonte: Elaborado pelo autor.

não nulas de $D - \hat{D}$ marcam as distâncias que sofreram alterações na atualização. Como matrizes de distâncias são simétricas, olharemos apenas para a forma triangular superior de tal matriz.

Considerando os valores mencionados, temos:

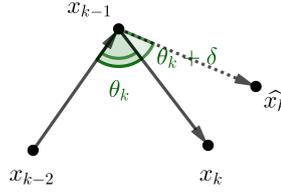
$$D - \hat{D} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -1.644 & -1.5305 & -1.605 & -1.5172 & -1.5799 \\ 0 & 0 & 0 & 0 & -1.5537 & -1.4103 & -1.5305 & -1.4272 & -1.5172 & \\ 0 & 0 & 0 & -1.7104 & -1.5537 & -1.644 & -1.5305 & -1.5305 & -1.605 & \\ 0 & 0 & -1.6049 & -1.3715 & -1.5537 & -1.4103 & -1.4103 & -1.5305 & -1.5305 & \\ 0 & -1.8 & -1.6049 & -1.7104 & -1.5537 & -1.4103 & -1.5537 & -1.644 & -1.644 & \\ & 0 & 0 & -8.8818^{-16} & -4.4409^{-16} & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & \\ & & 0 & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & -8.8818^{-16} & \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & \\ & & & & 0 & 0 & 0 & 0 & 0 & \\ & & & & & 0 & 0 & 0 & 0 & \\ & & & & & & 0 & 0 & 0 & \\ & & & & & & & 0 & 0 & \\ & & & & & & & & 0 & \\ & & & & & & & & & 0 \end{pmatrix}$$

De fato, podemos confirmar que, ao alterar d_6 , as distâncias que sofreram alteração foram $r_{i,j}$, com $i < 6$ e $j \geq 6$.

2.2 Modificação em θ_k

Neste segundo caso, avaliaremos as modificações na matriz de distâncias e nas coordenadas cartesianas, a partir da atualização de uma coordenada θ_k . Conforme ilustrado na Figura 13, esta atualização consiste em uma alteração na abertura do ângulo θ_k , que deve ser realizada pela rotação de um dos vetores que compõe o ângulo.

Como no caso anterior, também analisaremos as alterações das distâncias $r_{i,j} = \|x_i - x_j\|$ em três casos distintos. Vale observar o fato de que as distâncias $d_{k-1} = r_{k-2,k-1}$

Figura 13 – Alteração de θ_k .


Fonte: Elaborado pelo autor.

e $d_k = r_{k-1,k}$ não sofrem alteração, de modo que os dois últimos casos serão ligeiramente diferentes.

1. $i, j < k$

Novamente, para pontos associados a átomos anteriores ao k -ésimo, não ocorrem alterações nas distâncias, uma vez que as coordenadas cartesianas não são alteradas. Fato decorrente da própria construção dessas coordenadas.

2. $i, j \geq k - 1$

Para mostrar que as distâncias entre tais pontos não se alteram, vamos, inicialmente, considerar os pontos x_i e x_j com $i, j \geq k$. Na sequência, mostraremos que dado $j > k$ as distâncias $r_{k-1,j}$ também não se alteram.

Dados \hat{x}_i e \hat{x}_j atualizações de x_i e x_j , onde $i, j \geq k$, obtidas a partir da atualização de θ_k , podemos escrever

$$\hat{x}_i = x_1 + d_2 B_{[2]} b_2 + \cdots + d_k \hat{B}_{[k]} b_k + \cdots + d_i \hat{B}_{[i]} b_i \quad (2.2)$$

e também

$$\hat{x}_j = x_1 + d_2 B_{[2]} b_2 + \cdots + d_k \hat{B}_{[k]} b_k + \cdots + d_j \hat{B}_{[j]} b_j,$$

onde $\hat{B}_{[m]} = B_2^{(\varphi)} B_2^{(\theta)} \dots B_k^{(\varphi)} \hat{B}_k^{(\theta)} \dots B_m^{(\varphi)} B_m^{(\theta)}$, com $\hat{B}_k^{(\theta)}$ sendo a atualização da rotação $B_k^{(\theta)}$ de θ_k para $\theta_k + \delta$.

Assim,

$$\|\hat{x}_i - \hat{x}_j\| = \left\| (x_1 + d_2 B_{[2]} b_2 + \cdots + d_k \hat{B}_{[k]} b_k + \cdots + d_i \hat{B}_{[i]} b_i) - (x_1 + d_2 B_{[2]} b_2 + \cdots + d_k \hat{B}_{[k]} b_k + \cdots + d_j \hat{B}_{[j]} b_j) \right\|.$$

Sem perda de generalidade, vamos supor $i < j$, implicando em

$$\|\hat{x}_i - \hat{x}_j\| = \|d_{i+1} \hat{B}_{[i+1]} b_{i+1} + \cdots + d_j \hat{B}_{[j]} b_j\|.$$

Da linearidade de cada B_m , podemos evidenciar o produto de $B_m^{(\varphi)} B_m^{(\theta)}$, para $m = \{2, \dots, i+1\}$. Deste modo,

$$\|\hat{x}_i - \hat{x}_j\| = \left\| B_2^{(\varphi)} B_2^{(\theta)} \dots B_k^{(\varphi)} \hat{B}_k^{(\theta)} \dots B_{i+1}^{(\varphi)} B_{i+1}^{(\theta)} \left(d_{i+1} b_{i+1} + \sum_{p=i+2}^j d_p \left(\prod_{q=i+2}^p B_q^{(\varphi)} B_q^{(\theta)} \right) b_p \right) \right\|$$

Como o produto dos primeiros $i+1$ operadores representa uma sequência de rotações e sabemos que este não altera a norma do vetor, então

$$\|\hat{x}_i - \hat{x}_j\| = \left\| d_{i+1} b_{i+1} + \sum_{p=i+2}^j d_p \left(\prod_{q=i+2}^p B_q^{(\varphi)} B_q^{(\theta)} \right) b_p \right\|,$$

que não depende de $\theta_k + \delta$. Logo, podemos concluir que

$$\|\hat{x}_i - \hat{x}_j\| = \|x_i - x_j\| = r_{i,j}.$$

Agora, dado $j \geq k$, temos que $r_{k-1,j}$ é dado por

$$\begin{aligned} \|\hat{x}_{k-1} - \hat{x}_j\| &= \|d_k \hat{B}_{[k]} b_k + \dots + d_j \hat{B}_{[j]} b_j\| \\ &= \left\| B_2^{(\varphi)} B_2^{(\theta)} \dots B_k^{(\varphi)} \hat{B}_k^{(\theta)} \left(d_k b_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p B_q^{(\varphi)} B_q^{(\theta)} \right) b_p \right) \right\| \\ &= \left\| d_k b_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p B_q^{(\varphi)} B_q^{(\theta)} \right) b_p \right\| \\ &= r_{k-1,j}. \end{aligned}$$

Ou seja, estas distâncias também permanecem inalteradas.

3. $i \leq k-2$ e $j \geq k$.

Similarmente ao cálculo realizado no item anterior, a distância entre \hat{x}_i e \hat{x}_j é dada por

$$\|\hat{x}_i - \hat{x}_j\| = \|d_{i+1} B_{[i+1]} b_{i+1} + \dots + d_k \hat{B}_{[k]} b_k + \dots + d_j \hat{B}_{[j]} b_j\|,$$

naturalmente distinta de $r_{i,j}$, determinada por

$$\|x_i - x_j\| = \|d_{i+1} B_{[i+1]} b_{i+1} + \dots + d_k B_{[k]} b_k + \dots + d_j B_{[j]} b_j\|.$$

Podemos concluir, portanto, que uma atualização de θ_k gera somente alterações nas distâncias $r_{i,j}$ para $i \leq k-2$ e $j \geq k$.

Observe que a determinação das distâncias alteradas nos fornece a seleção de pontos cujas coordenadas cartesianas devem ser atualizadas. Então, para uma alteração em θ_k , precisamos atualizar somente as entradas da cadeia posteriores a x_k . Para isso, aplica-se uma rotação em todo o conjunto de pontos x_i , com $i = k, \dots, n$. O ângulo de tal rotação é dado pela variação δ , enquanto o eixo de rotação é determinado pelo vetor normal aos vetores de ligação b_{k-1} e b_k .

Alguns cuidados devem ser tomados para que se realize tal atualização. Note que as matrizes de rotação apresentadas anteriormente consideram os eixos coordenados como eixo de rotação. Contudo, a atualização de θ_k é dada em termos de um ângulo δ e um eixo arbitrário, orientado por um vetor unitário v . Deste modo, para realizar uma rotação sobre um eixo genérico, é necessário utilizar o seguinte procedimento: reposicionar o eixo de rotação sobre um dos eixos coordenados, realizar a rotação em questão e, então, retornar o eixo para a posição inicial. Tanto o reposicionamento quanto o retorno à posição inicial devem ser feitos por meio de rotações. Tradicionalmente, o reposicionamento ocorre de modo a sobrepor o eixo desejado no eixo z , implicando que uma rotação com relação ao eixo determinado por v é realizada da seguinte maneira,

$$R_{v,\theta} = R_{z,\alpha} R_{y,\beta} R_{z,\theta} R_{y,-\beta} R_{z,-\alpha}$$

onde $R_{z,\alpha}$ e $R_{y,\beta}$ são as rotações em torno de z e y , respectivamente, que reposicionam o eixo orientado por v .

Usando as coordenadas polares do vetor v e realizando o produto das matrizes, obtemos a seguinte matriz correspondente a rotação $R_{(v,\theta)}$.

$$\begin{pmatrix} v_1^2(1 - \cos \theta) + \cos \theta & v_1 v_2(1 - \cos \theta) - v_3 \sin \theta & v_3 v_1(1 - \cos \theta) + v_2 \sin \theta \\ v_1 v_2(1 - \cos \theta) + v_3 \sin \theta & v_2^2(1 - \cos \theta) + \cos \theta & v_2 v_3(1 - \cos \theta) - v_1 \sin \theta \\ v_3 v_1(1 - \cos \theta) - v_2 \sin \theta & v_2 v_3(1 - \cos \theta) + v_1 \sin \theta & v_3^2(1 - \cos \theta) + \cos \theta \end{pmatrix}. \quad (2.3)$$

Os detalhes de construção desta matriz são apresentados em [Flower \(1999\)](#) e explicitamente calculados em [Murray \(2013\)](#).

Dito isso, o método para atualização das coordenadas cartesianas pode ser

resumido pelo Algoritmo 4.

Algoritmo 4: Atualização de coordenadas a partir de uma alteração em φ_k .

Entrada: Coordenadas cartesianas, $\hat{\varphi}_k = \varphi_k + \delta$

Saída: Coordenadas cartesianas atualizadas

início

$b_k \leftarrow x_k - x_{k-1};$

$b_{k-1} \leftarrow x_{k-2} - x_{k-1};$

$v \leftarrow \frac{b_k \times b_{k-1}}{\|b_k \times b_{k-1}\|};$

Calcule a matriz de rotação $R_{(\delta,v)}$;

para $i = k, \dots, n$ **faça**

$\Delta x_i \leftarrow x_i - x_{k-1};$

$\Delta x_i \leftarrow R_{(\delta,n)} \Delta x_i;$

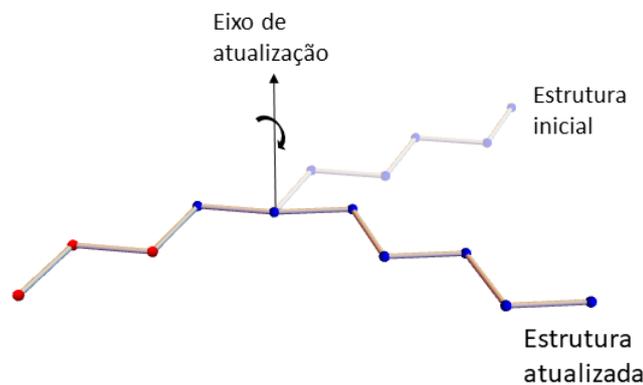
$x_i \leftarrow x_{k-1} + \Delta x_i;$

fim

fim

A Figura 14 ilustra as alterações promovidas por uma atualização de $\delta = \frac{\pi}{3}$ em θ_6 .

Figura 14 – Atualização de θ_6 .



Fonte: Elaborado pelo autor.

Podemos ainda analisar as alterações na matriz de distâncias. Conforme o exemplo anterior, temos a construção da seguinte matriz.

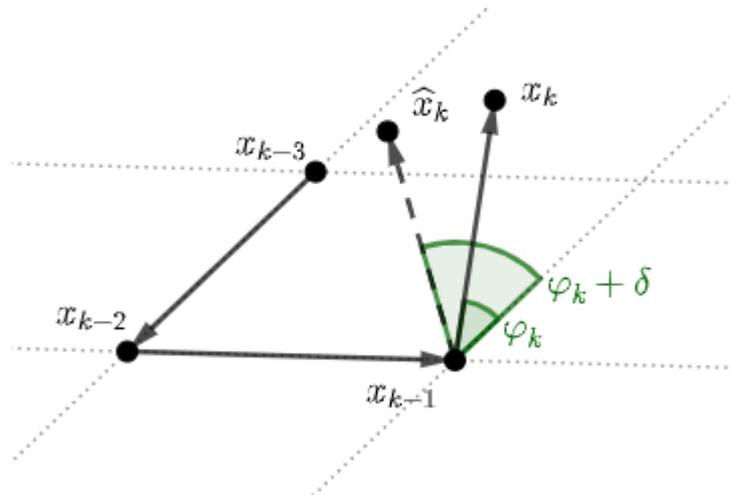
$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & -0.10844 & 0.867534 & 0.641521 & 1.31238 & 1.07628 \\ 0 & 0 & 0 & 0 & -0.354931 & 0.446373 & 0.117875 & 0.641521 & 0.641521 & 0.366132 \\ 0 & 0 & 0 & -0.088569 & 0.656192 & 0.446373 & 0.867534 & 0.867534 & 0.68405 & \\ 0 & 0 & -0.538317 & -0.088569 & -0.088569 & -0.354931 & -0.10844 & -0.10844 & -0.285674 & \\ 0 & -4.44089^{-16} & -4.44089^{-16} & -4.44089^{-16} & -4.44089^{-16} & 0 & 0 & 0 & 0 & \\ & 0 & 0 & 0 & 0 & 4.44089^{-16} & 0 & 0 & 0 & \\ & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \\ & & & 0 & 0 & 2.22044^{-16} & 0 & 0 & 0 & \\ & & & & 0 & 0 & 0 & 0 & 0 & \\ & & & & & 0 & 0 & 0 & 0 & \\ & & & & & & 0 & 0 & 0 & \\ & & & & & & & 0 & 0 & \\ & & & & & & & & 0 & \\ & & & & & & & & & 0 \end{pmatrix}$$

De fato, as alterações ocorreram nas entradas $r_{i,j}$ com $i \leq 4 = 6 - 2$ e $j \geq 6$.

2.3 Modificação em φ_k

Por fim, faremos a análise das alterações dadas a partir de uma atualização no ângulo de torção φ_k . Como no caso de θ_k , aqui a alteração também é expressa como uma modificação na abertura do ângulo, conforme ilustra a Figura 15.

Figura 15 – Alteração em φ_k



Fonte: Elaborado pelo autor.

O cálculo das entradas que são alteradas na matriz de distância para este caso é bastante similar ao caso anterior. Desde modo, também analisaremos três situações distintas.

1. $i, j < k$

Não há alterações nas coordenadas internas de x_i, x_j , uma vez que o ângulo de torção φ_k não é utilizado para seu cálculo.

2. $i, j \geq k - 2$

Similar ao item 2 do caso anterior, podemos mostrar que, para $i, j \geq k - 1$, as distâncias $r_{i,j}$ permanecem inalteradas após uma atualização do tipo $\varphi_k + \delta$.

Ademais, vamos mostrar que para $j \geq k - 1$, $r_{k-2,j}$ também não se altera.

Partindo dos pontos atualizados \hat{x}_i e \hat{x}_j , temos

$$\begin{aligned} \|\hat{x}_{k-2} - \hat{x}_j\| &= \|d_{k-1}B_{[k-1]}b_{k-1} + d_{k-1}\hat{B}_{[k]}b_k + \cdots + d_j\hat{B}_{[j]}b_j\| \\ &= \left\| B_{[k-1]} \left(d_{k-1}b_{k-1} + \sum_{p=k}^j d_p \left(\prod_{q=k}^p B_q^{(\varphi)} B_q^{(\theta)} \right) b_p \right) \right\| \\ &= \left\| d_{k-1}b_{k-1} + \hat{B}_k^{(\varphi)} B_k^{(\theta)} \left(d_k b_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p B_q^{(\varphi)} B_q^{(\theta)} \right) \right) b_p \right\|. \end{aligned}$$

Como sabemos, da construção da equação, a rotação $\hat{B}_k^{(\varphi)}$, bem como $B_k^{(\varphi)}$, tem o eixo de rotação orientado por b_{k-1} . Isto significa que tal vetor não é alterado por nenhum dos dois operadores, ou seja, $b_{k-1} = B_k^{(\varphi)} b_{k-1}$. Com isso, temos

$$\begin{aligned} \|\hat{x}_{k-2} - \hat{x}_j\| &= \left\| \hat{B}_k^{(\varphi)} \left(d_{k-1}b_{k-1} + B_k^{(\theta)} \left(d_k b_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p B_q^{(\varphi)} B_q^{(\theta)} \right) \right) b_p \right) \right\| \\ &= \left\| d_{k-1}b_{k-1} + B_k^{(\theta)} \left(d_k b_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p B_q^{(\varphi)} B_q^{(\theta)} \right) \right) b_p \right\| \\ &= r_{k-2,j}. \end{aligned}$$

Portanto, para $i, j \geq k - 2$, as distâncias $r_{i,j}$ não são alteradas.

3. $i \leq k - 3$ e $j \geq k$

Este caso é análogo ao caso 3 do item anterior, substituindo o operador de rotação de θ para φ .

Então, concluímos que em uma atualização de φ_k as únicas entradas de $r_{i,j}$ a sofrer alteração são $i \leq k - 3$ e $j \geq k$.

A atualização das coordenadas cartesianas a partir de uma mudança em φ_k são bastante similares ao caso em que θ_k é modificado. Deste modo, apenas as entradas

posteriores a x_k devem ser modificadas por meio de matrizes descritas em (2.3), onde o eixo de rotação é dado pelo vetor b_{k-1} normalizado. O Algoritmo 5 resume este processo.

Algoritmo 5: Atualização de coordenadas a partir de uma alteração em φ_k .

Entrada: Coordenadas cartesianas, $\hat{\varphi}_k = \varphi_k + \delta$

Saída: Coordenadas cartesianas atualizadas.

início

$b_{k-1} \leftarrow x_{k-1} - x_{k-2};$

$v = \frac{b_{k-1}}{d_{k-1}};$

Calcule a matriz de rotação $B_{(\delta,v)}$;

para $i = k, \dots, n$ **faça**

$\Delta x_i \leftarrow x_i - x_{k-1};$

$\Delta x_i \leftarrow B^{(\delta)} \Delta x_i;$

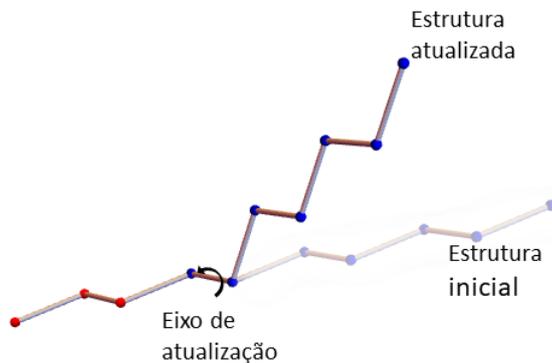
$x_i \leftarrow x_i + \Delta x_i;$

fim

fim

Para a exemplificação, vamos impor uma atualização $\delta = \frac{\pi}{3}$ em φ_6 . Na Figura 16 podemos observar as alterações realizadas.

Figura 16 – Atualização de φ_6 .



Fonte: Elaborado pelo autor.

As mudanças na matriz de distâncias podem ser observadas na matriz a seguir:

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0.332535 & 0.277593 & 0.477763 & 0.417416 & 0.557057 \\ & 0 & 0 & 0 & 0 & 0.208708 & 0.163971 & 0.277593 & 0.235476 & 0.311339 \\ & & 0 & 0 & 0 & 0.275101 & 0.208708 & 0.332535 & 0.277593 & 0.355731 \\ & & & 0 & 0 & 0 & 0 & 0 & 0 & -8.88178^{-16} \\ & & & & 0 & 0 & 0 & 0 & 0 & -8.88178^{-16} \\ & & & & & 0 & 0 & 0 & 8.88178^{-16} & -8.88178^{-16} \\ & & & & & & 0 & 0 & 4.44089^{-16} & -8.88178^{-16} \\ & & & & & & & 0 & 6.66133^{-16} & -8.88178^{-16} \\ & & & & & & & & 0 & -8.88178^{-16} \\ & & & & & & & & & 0 \end{pmatrix}$$

Conforme podemos observar, apenas sofreram alterações as entradas $r_{i,j}$ para $i \leq 3 = 6 - 3$ e $j \geq 6$.

2.4 Modelos de atualização

Mudanças conformacionais estão associadas às variações em ângulos de torção, de modo que um olhar mais atento para variações nessas coordenadas se faça necessário. Embora tenhamos estudado variações em todas as coordenadas internas, a partir de agora nos concentraremos somente nos casos em que há variação nos ângulos de torção.

Já estabelecemos como coordenadas cartesianas e distâncias interatômicas são afetadas por alterações individuais nas coordenadas internas. Porém, essa estratégia de atualização tem alguns problemas práticos. O número de operações necessárias para atualizar a cadeia cresce rapidamente à medida que mais coordenadas precisem ser alteradas. Além disso, cada atualização é tratada individualmente, implicando que pontos em posições posteriores as das alterações precisam ser recalculados. Ou seja, todas as posições intermediárias têm que ser calculadas explicitamente.

Essas situações são contornadas ao se utilizar o espaço homogêneo. Neste caso, como translações são linearizadas, pode-se codificar em um único operador a atualização de vários ângulos de torção. Desse modo, cada ponto a ser reposicionado é calculado uma única vez, diretamente para posição sua final. Conforme observado em [Zhang e Kavraki \(2002\)](#), para atualizar um ponto a partir de uma alteração em φ_k , deve-se utilizar o seguinte produto matricial:

$$\begin{pmatrix} I & x_k \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R_{(\delta,v)} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} I & -x_k \\ 0 & 1 \end{pmatrix}. \quad (2.4)$$

As matrizes à direita e à esquerda realizam translações que reposicionam o ponto x_k na origem do espaço tridimensional, enquanto a matriz central realiza a rotação. O elemento

$R_{(\delta,v)}$, presente na matriz intermediária, refere-se a matriz de rotação por um ângulo δ em relação ao eixo determinado pelo vetor unitário $v = \frac{b_{k-1}}{d_{k-1}}$, calculada conforme (2.3). Para simplificar a notação, vamos utilizar R_k para nomear a matriz de rotação que atualiza o ângulo φ_k .

Calculando o produto (2.4) temos como resultado a matriz

$$M_k = \begin{pmatrix} R_k & x_k - R_k x_k \\ 0 & 1 \end{pmatrix}, \quad (2.5)$$

que representa uma rotação seguida por uma translação, onde R_k realiza a rotação e $t_k = x_k - R_k x_k$ é o vetor de translação. Assim, para atualizar qualquer ponto, basta multiplicá-lo por M_k .

Uma vez determinada a matriz M_k , é necessário também estabelecer a maneira mais eficiente de compor tais movimentos. Sabemos que a composição dos movimentos é realizada mediante o produto das matrizes correspondentes. Logo, dadas M_k e M_j , a composição de ambas é dada por

$$M_k M_j = \begin{pmatrix} R_k R_j & R_k t_j + t_k \\ 0 & 1 \end{pmatrix}.$$

A composição das matrizes resulta em uma matriz de mesma natureza. Além disso, como observado em Choi (2006), para a construção da parte relativa à translação, pode-se usar o fato de $t_k = x_k - R_k x_k$, evitando calcular explicitamente t_k e, portanto, economizando o cálculo de uma rotação:

$$\begin{aligned} R_k t_j + t_k &= R_k t_j + x_k - R_k x_k \\ &= R_k (t_j - x_k) + x_k. \end{aligned}$$

Agora que translações e rotações foram fundidas em um único operador linear, o algoritmo de atualização das coordenadas pode ser reformulado, permitindo a atualização a partir da alteração de múltiplos ângulos de torção.

O Algoritmo 6 utiliza o espaço homogêneo para atualizar o conjunto de coordenadas cartesianas a partir da alteração de um conjunto de ângulos de torção. Este é o primeiro algoritmo que levaremos em consideração para a realização de testes e comparações entre os demais aqui estabelecidos. Indicaremos por C_C o conjunto de coordenadas cartesianas e por \mathcal{I} o conjunto de atualizações dos ângulos de torção. Quando necessário, usaremos uma barra superior para indicar o conjunto com coordenadas atualizadas.

O custo computacional dessas atualizações deve ser calculado baseado em dois fatores: o número de ângulos de torção alterados, indicado por n_r , e o número de pontos da cadeia a serem efetivamente atualizados, indicado por n_a . Para cada novo ângulo de

Algoritmo 6: Atualização de coordenadas com translações encaixadas.**Entrada:** $C_C = \{x_1, x_2, \dots, x_n\}, \mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$ **Saída:** $\overline{C_C} = \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}$ **início** $B_0 \leftarrow I_3;$ $t_0 \leftarrow 0;$ **para** $i = 1, \dots, p$ **faça** $b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{\|x_{k_i-2} - x_{k_i-1}\|};$ Calcule a matriz de rotação R_i , sobre o eixo b_{k_i-1} ; $t_i \leftarrow R_i(t_{i-1} - x_{k_i-2}) + x_{k_i-2};$ $B_i \leftarrow R_i B_{i-1};$ $M_i = \begin{pmatrix} B_i & t_i \\ 0 & 1 \end{pmatrix};$ **para** $k_i \leq j < k_{i+1}$ **faça** $\overline{x_j} \leftarrow M_{k_i} x_j;$ **fim****fim****fim**

torção a ser alterado, temos: 3 somas e 3 divisões para determinar o eixo; 15 multiplicações e 10 somas para determinar a matriz de rotação; 9 produtos e 12 somas para determinar o vetor de translação; 27 multiplicações e 18 somas para calcular o produto das matrizes de rotação, totalizando 54 multiplicações e 43 somas. O custo para a atualização das coordenadas de cada ponto é de 9 multiplicações e 9 somas. Assim, a estimativa do custo computacional deste modelo é de $97n_r + 18n_a$ operações.

Um olhar mais atento para o Algoritmo 6 revela duas escolhas para sua construção. A primeira é que a ordem em que as rotações são aplicadas é reversa. Assim como na construção de coordenadas cartesianas, o uso da propagação reversa na atualização também evita que recalculamos os eixos de rotação a cada nova iteração (veja a Figura 7). A segunda escolha é que translações são aplicadas de maneira intercalada, como no processo de construção desenvolvido por Thompson. O uso de translações encaixadas faz sentido ao se trabalhar com o espaço homogêneo, uma vez que o interesse é unificar translações e rotações em um único operador. Contudo, esta aplicação pode gerar acúmulo de erros, pois a cada atualização, a nova posição do ponto depende imediatamente da sua posição anterior. Além disso, são realizadas translações para reposicionar o vértice do ângulo de rotação, fazendo com que pontos com posições distantes na cadeia também possam afetar a precisão do método (ZHANG; KAVRAKI, 2002).

Um modelo mais moderno e com custo mais baixo pode ser obtido ao substituir o uso de translações encaixadas por translações finais. A principal diferença entre uma escolha e outra, é que no segundo avalia-se a atuação das rotações somente no vetor de

ligação. Isto significa que vetores e pontos intermediários não fazem parte do processo de rotação. Analisando este modelo, é possível observar que cada rotação pode ser calculada separadamente, a partir do conjunto inicial de coordenadas. Aplicam-se então as rotações em ordem reversa apenas nos vetores de ligação, cuja função é transladar o último ponto estabelecido a fim de obter o próximo ponto a ser determinado. Com isso, não há necessidade de atualizar as translações intermediárias a cada passo. O Algoritmo 7 resume tais procedimentos.

Algoritmo 7: Atualização de coordenadas com translações finais.

Entrada: $C_C = \{x_1, x_2, \dots, x_n\}$, $\mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$

Saída: $\overline{C_C} = \{\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n\}$

início

$B_{[0]} \leftarrow I_3$;

$t_0 \leftarrow 0$;

para $i = 1, \dots, p$ **faça**

$b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{\|x_{k_i-2} - x_{k_i-1}\|}$;

 Calcule B_i ;

$B_{[i]} \leftarrow B_{[i-1]}R_i$;

para $k_i \leq j < k_{i+1}$ **faça**

$b_j = x_j - x_{j-1}$;

$\overline{x}_j \leftarrow \overline{x}_{j-1} + B_{[i]}b_j$;

fim

fim

fim

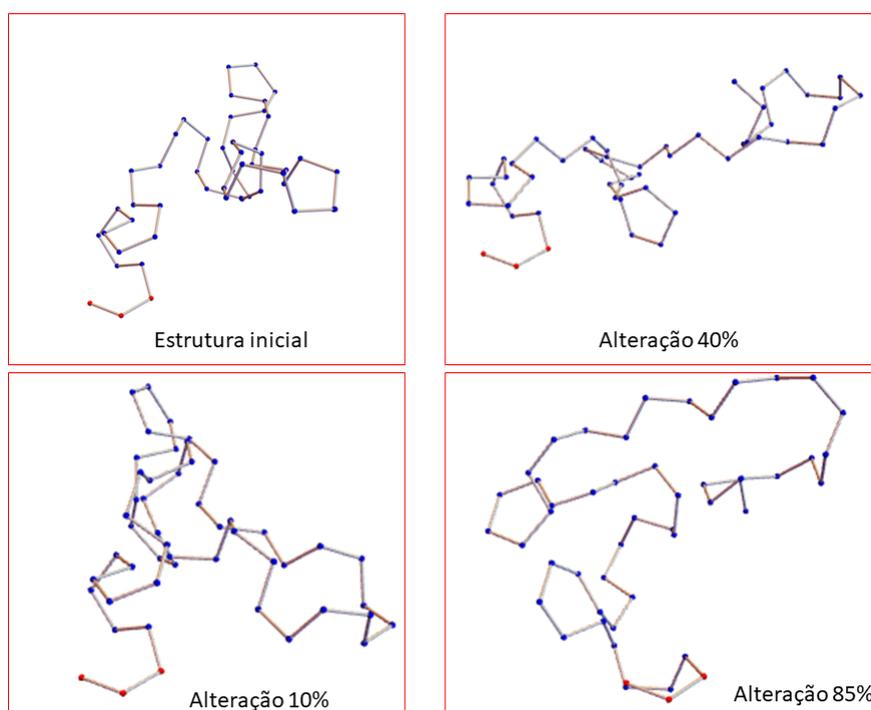
Este algoritmo apresenta um custo computacional menor que o anterior em relação ao número de rotações, sendo necessárias 3 subtrações e 3 divisões para determinar o eixo de rotação, 2 operações trigonométricas, 15 multiplicações e 10 somas para determinar a matriz de rotação, 27 multiplicações e 18 somas para calcular a matriz de rotações combinadas, totalizando 45 multiplicações/divisões e 22 somas/subtrações. Para a atualização de cada ponto, temos um custo de 3 somas para determinar b_j e 9 produtos e 9 somas para calcular as coordenadas. Assim, a estimativa do custo para a atualização da cadeia é dada por $67n_r + 21n_a$ operações.

O método de translações finais foi implementado no *software Wolfram Mathematica v11.3*. O código foi testado calculando uma atualização aleatória de cerca de 10%, 40% e 85% dos ângulos de torção de cada um dos exemplos BBN50, BBN200 e BBN1000, apresentados na seção anterior. Os ângulos de torção para os 3 primeiros elementos da cadeia são irrelevantes para a determinação do envelhecimento. Deste modo, consideramos atualizações apenas dos ângulos φ_i com $i \geq 4$. Os ângulos alterados foram escolhidos aleatoriamente e o valor da alteração também foi determinado de maneira aleatória, dentro do intervalo $[-1, 1]$. A escolha deste intervalo foi feita para que não gerasse variações

muito grandes no valor dos ângulos de torção, por uma questão puramente de escolha dos autores. Outros intervalos poderiam ser usados sem nenhuma perda teórica.

Para o exemplo BBN50, foi calculada a atualização de 5, 19 e 39 ângulos de torção, para o exemplo BBN200, 20, 78 e 166 ângulos, e para o exemplo BBN1000 foram alterados 100, 398 e 846 ângulos. As Figuras 17, 18 e 19 ilustram como as atualizações impostas modificam os backbones. Estes conjuntos de coordenadas e de atualização serão utilizados para testar todos os métodos aqui descritos.

Figura 17 – Alteração do exemplo BBN50.



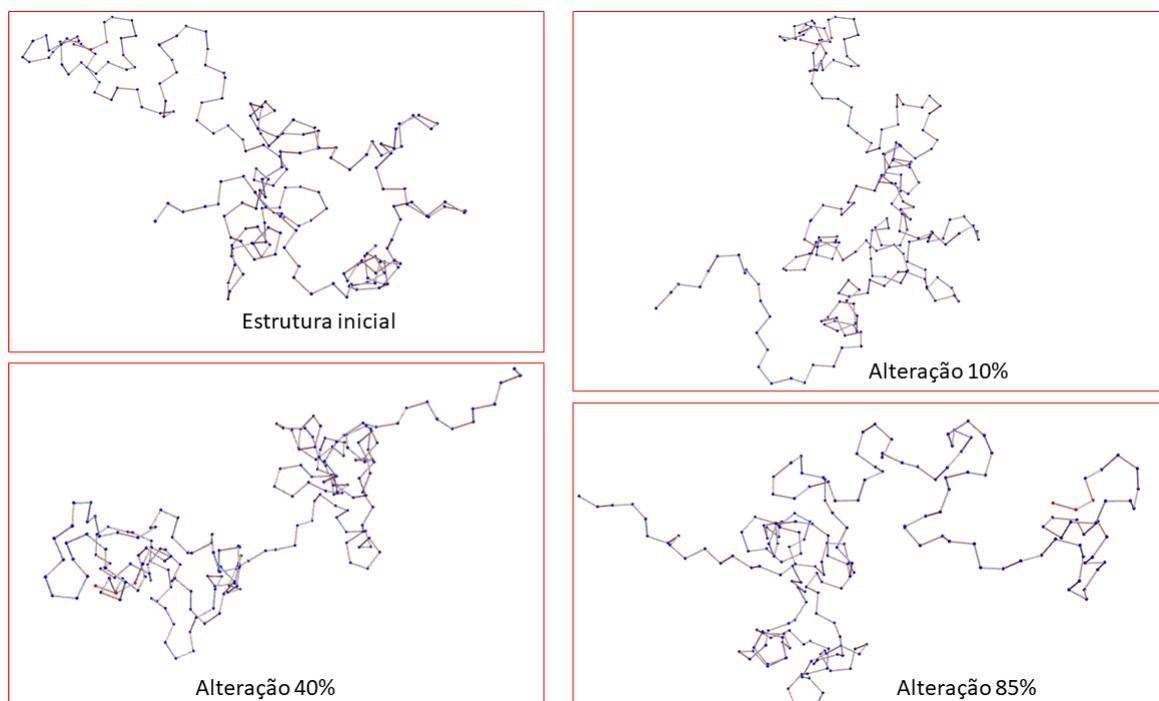
Fonte: Elaborado pelo autor.

A fim de ilustrar nossa implementação, foram medidos os tempos de execução para cada uma das atualizações. A Tabela 3 traz os tempos, em segundos, encontrados em tais medições.

Tabela 3 – Tempo (s) de execução do algoritmo de atualização com propagação reversa.

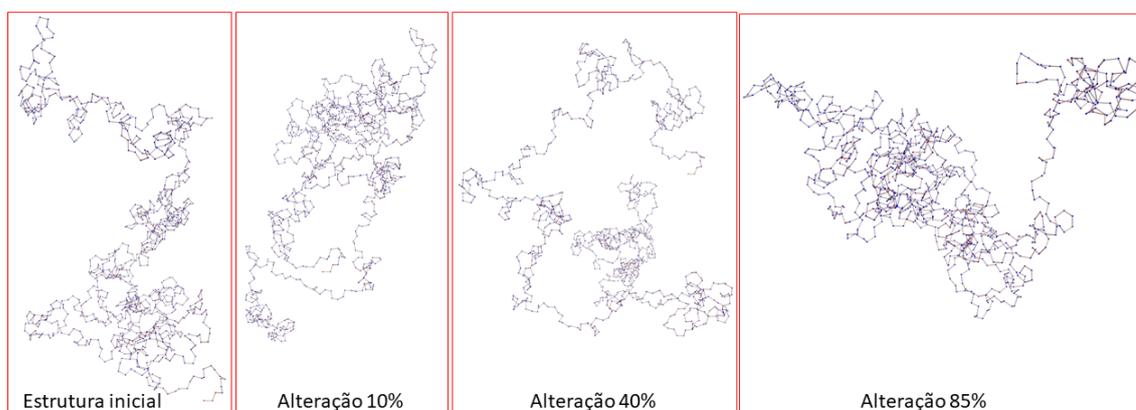
Conjunto	10%	40%	85%
BBN50	0.	0.015	0.015
BBN200	0.	0.015	0.03
BBN1000	0.015	0.062	0.11

Figura 18 – Alterações do exemplo BBN200.



Fonte: Elaborado pelo autor.

Figura 19 – Alterações do exemplo BBN1000.



Fonte: Elaborado pelo autor.

3 Quatérnios, Rotores e a Álgebra Geométrica

Temos explorado a manipulação de coordenadas de proteínas usando matrizes de rotação. Contudo, formas mais sofisticadas para representar rotações espaciais têm sido desenvolvidas ao longo da história. A seguir, apresentaremos de maneira sucinta as teorias relacionadas às chamadas álgebras de Clifford, que permitirão a descrição de novos modelos para construção e atualização de coordenadas cartesianas e o desenvolvimento de um novo modelo de representação para proteínas.

3.1 Quatérnios

Desenvolvida por William Rowan Hamilton, entre 1843 e 1866, por meio de um longo processo investigativo (DION; PACCA; MACHADO, 1995), a chamada álgebra dos quatérnios permite combinar vetores do espaço tridimensional e escalares reais em uma única entidade matemática, os quatérnios. Com uma estrutura algébrica adequada, o conjunto destes elementos possui interessantes características, como a possibilidade de calcular-se produtos internos e vetoriais simultaneamente, instituir a divisão de vetores e, a de maior interesse para este trabalho, calcular eficientemente rotações espaciais por meio de produtos entre quatérnios e vetores.

A forma padrão de um quatérnio é representada por

$$q = q_1 + q_2i + q_3j + q_4k, \quad (3.1)$$

onde $q_1, q_2, q_3, q_4 \in \mathbb{R}$ e i, j, k são unidades simbólicas. O conjunto dos quatérnios pode também ser entendido como uma extensão natural do conjunto dos números complexos, o que permite estabelecer muitas analogias entre suas álgebras.

Sendo o conjunto dos quatérnios um espaço vetorial, gerado pela base $\{1, i, j, k\}$, e considerando somas e produtos análogos aos de \mathbb{C} , é necessário definir apenas o produto entre os elementos i, j, k , a fim de compreender sua estrutura algébrica. Assim, temos as seguintes operações base:

$$\begin{aligned} i^2 &= -1, & j^2 &= -1, & k^2 &= -1, \\ jk &= i, & ki &= j, & ij &= k, \\ kj &= -i, & ik &= -j, & ji &= -k. \end{aligned}$$

Usando estas identidades e as propriedades de associatividade, comutatividade da soma e distributividade do produto para soma, podemos estabelecer o produto entre quatérnios.

Como mencionado anteriormente, quatérnios combinam a representação de escalares e do espaço tridimensional em um único elemento algébrico. Por esta razão, dado $q = q_1 + q_2i + q_3j + q_4k$, chama-se q_1 de parte escalar e q_2, q_3, q_4 de parte vetorial. Neste texto trataremos os quatérnios de maneira superficial, explorando apenas o suficiente para atingir nossos objetivos. Detalhes adicionais podem ser encontrados em [Kanatani \(2015\)](#).

Estamos interessados em trabalhar com a representação de rotações com quatérnios. Para tal, vamos considerar inicialmente o vetor $a \in \mathbb{R}^3$ e o quatérnio q unitário, isto é, $\|q\| = 1$. O conjugado de q , definido a partir da troca de sinal de sua parte vetorial, é denotado por q^\dagger . Como o conjunto dos quatérnios mapeia o espaço tridimensional, o vetor a pode ser escrito como o quatérnio $0 + a_1i + a_2j + a_3k$ e, portanto, o produto

$$a' = qaq^\dagger \quad (3.2)$$

fica bem definido.

Como q é unitário, temos que $\|q\| = q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$. Então, existe θ tal que $q_1 = \cos \frac{\theta}{2}$ e $\sqrt{q_2^2 + q_3^2 + q_4^2} = \sin \frac{\theta}{2}$ e podemos reescrever

$$q = \cos \frac{\theta}{2} + b \sin \frac{\theta}{2}, \quad (3.3)$$

onde b é o vetor unitário $\frac{1}{\sqrt{q_2^2 + q_3^2 + q_4^2}}(q_2, q_3, q_4)$. Dessa maneira, a Equação (3.2) pode ser reescrita como

$$\begin{aligned} a' &= \left(\cos \frac{\theta}{2} + b \sin \frac{\theta}{2} \right) a \left(\cos \frac{\theta}{2} - b \sin \frac{\theta}{2} \right) \\ &= a \cos^2 \frac{\theta}{2} - ab \cos \frac{\theta}{2} \sin \frac{\theta}{2} + ba \sin \frac{\theta}{2} \cos \frac{\theta}{2} + bab \sin^2 \frac{\theta}{2} \\ &= a \cos^2 \frac{\theta}{2} + (ba - ab) \cos \frac{\theta}{2} \sin \frac{\theta}{2} - bab \sin^2 \frac{\theta}{2}. \end{aligned}$$

Embora a e b sejam vetores do espaço tridimensional, ambos são representados como elementos vetoriais do espaço dos quatérnios. Assim, $(ba - ab) = 2(b \times a)$ (produto vetorial) e $bab = a - 2\langle a, b \rangle b$. Substituindo,

$$\begin{aligned} a' &= a \cos^2 \frac{\theta}{2} + 2(b \times a) \cos \frac{\theta}{2} \sin \frac{\theta}{2} - (a - 2\langle a, b \rangle b) \sin^2 \frac{\theta}{2} \\ &= a \left(\cos^2 \frac{\theta}{2} - \sin^2 \frac{\theta}{2} \right) + 2(b \times a) \cos \frac{\theta}{2} \sin \frac{\theta}{2} + 2 \sin^2 \frac{\theta}{2} \langle a, b \rangle b \\ &= a \cos \theta + (b \times a) \sin \theta + \langle a, b \rangle b (1 - \cos \theta) \end{aligned} \quad (3.4)$$

A Equação (3.4) é conhecida como fórmula de Rodrigues ([KANATANI, 2015](#)) e descreve a rotação do vetor a por um ângulo θ com relação a um eixo b .

Portanto, a rotação de um vetor qualquer, $x \in \mathbb{R}^3$, pode ser codificada pelo produto qxq^\dagger no espaço dos quatérnios.

3.2 Rotores

Retornando momentaneamente à descrição de proteínas, é importante observar que matrizes de rotação são projetadas para realizar rotações em torno dos eixos coordenados. Contudo, ao atualizar uma coordenada interna angular ou posicionar um novo átomo na cadeia, é preciso rotacionar pontos e vetores por um ângulo δ com relação a um eixo arbitrário. Essa questão foi contornada com uso de matrizes que codificam o reposicionamento do eixo, mas, como pudemos observar, uma alternativa mais simples pode ser obtida com o uso dos quatérnios.

Ao utilizar quatérnios, podemos calcular qualquer rotação genérica a partir do ângulo e do eixo desejado. Para isso, basta determinar $q = \cos \frac{\theta}{2} + b \sin \frac{\theta}{2}$, onde b é o vetor unitário que determina o eixo de rotação e θ o ângulo de rotação. Como rotações são operações lineares em \mathbb{R}^3 , é natural estabelecer esta sequência de produtos qxq^\dagger como um operador linear, chamado rotor, definido por

$$\begin{aligned} R &: \mathbb{R}^3 \mapsto \mathbb{R}^3 \\ x &\mapsto qxq^\dagger \end{aligned}$$

Este operador pode ser escrito na forma matricial. Para isso, aplicamos a rotação desejada nos vetores da base canônica, gerando as colunas da matriz correspondente. Dessa maneira, dado um rotor gerado por $q = q_1 + q_2i + q_3j + q_4k$, sua forma matricial é

$$2 \begin{bmatrix} q_1^2 + q_2^2 - \frac{1}{2} & q_2q_3 - q_1q_4 & q_2q_4 + q_1q_3 \\ q_2q_3 + q_1q_4 & q_1^2 + q_3^2 - \frac{1}{2} & q_3q_4 - q_1q_2 \\ q_2q_4 - q_1q_3 & q_3q_4 + q_1q_2 & q_1^2 + q_4^2 - \frac{1}{2} \end{bmatrix}. \quad (3.5)$$

Para ilustrar como esta matriz é determinada, consideremos o cálculo da primeira coluna, sendo as demais obtidas de maneira análoga. Dado e_1 , o primeiro vetor da base canônica, este corresponde ao quatérnio i . A ação de q em e_1 é dada por:

$$\begin{aligned} qi^\dagger &= (q_1 + q_2i + q_3j + q_4k)i(q_1 - q_2i - q_3j - q_4k) \\ &= (q_1 + q_2i + q_3j + q_4k)(q_1i + q_2 - q_3ij - q_4ik) \\ &= (q_1 + q_2i + q_3j + q_4k)(q_1i + q_2 - q_3k + q_4j) \\ &= (q_1 + q_2i + q_3j + q_4k)(q_2 + q_1i - q_3k + q_4j) \\ &= q_1^2i - 2q_1q_3k + 2q_1q_4j + q_2^2i + 2q_2q_3j + 2q_2q_4k - q_3^2i - q_4^2i \\ &= (q_1^2 + q_2^2 - q_3^2 - q_4^2)i + 2(q_1q_4 + q_2q_3)j + 2(q_2q_4 - q_1q_3)k. \end{aligned}$$

Como q é unitário, temos $q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1$, implicando em

$$qi q^\dagger = 2(q_1^2 + q_2^2 - \frac{1}{2})i + 2(q_2q_3 + q_1q_4)j + 2(q_2q_4 - q_1q_3)k.$$

Voltando o quatérnio vetorial resultante para \mathbb{R}^3 , vale

$$Re_1 = 2 \begin{bmatrix} q_1^2 + q_2^2 - \frac{1}{2} \\ q_2q_3 + q_1q_4 \\ q_2q_4 - q_1q_3 \end{bmatrix},$$

formando assim a primeira coluna da matriz.

Uma vez estabelecida a forma matricial, vamos tomar a notação simplificada Bx para representar a rotação qxq^\dagger . Para validar os métodos trabalhados anteriormente, basta então definir a composição de rotações dadas por rotores.

A composição de uma sequência de rotações é dada pelo produto das matrizes correspondentes. O mesmo vale para quatérnios, uma vez que, dados os rotores q e q' associados às rotações B e B' , a sequência de rotações $B'B$ de um ponto x é dada por $q'(qxq^\dagger)q'^\dagger = (q'q)x(q^\dagger q'^\dagger) = (q'q)x(q'q)^\dagger$, ou seja, pelo produto dos quatérnios correspondentes.

Com essas questões resolvidas, somos capazes de enunciar um novo método tanto para a conversão de coordenadas quanto para sua atualização por meio de produtos de quatérnios. Antes disso, vamos explorar a utilização de rotores sob uma perspectiva mais completa, usando a chamada álgebra geométrica, para só então retornar às conformações de proteínas.

3.3 O produto externo

A álgebra geométrica é o resultado da pesquisa desenvolvida por William Kingdon Clifford, matemático inglês, e apresentada em 1878. Clifford unificou o produto interno e o produto externo desenvolvido por Grassmann, definindo um novo produto chamado produto geométrico. Essa nova estrutura algébrica permitiu expandir os conceitos desenvolvidos por Hamilton na álgebra dos quatérnios, generalizando-os para espaços de dimensões maiores. Apresentaremos aqui os principais conceitos da álgebra geométrica, que permitirão a elaboração de novos métodos para conversão e atualização de coordenadas cartesianas. Uma pesquisa mais aprofundada a respeito de tais teorias e os aspectos históricos de seu desenvolvimento podem ser encontrado com mais detalhes em [Lounesto \(2001\)](#).

Embora a álgebra geométrica possa ser determinada para espaços de qualquer dimensão $n \in \mathbb{N}$, vamos nos concentrar na álgebra desenvolvida para \mathbb{R}^3 , uma vez que

moléculas de proteínas são elementos do espaço tridimensional. Começaremos definindo o produto externo de Grassmann e os novos elementos gerados por ele.

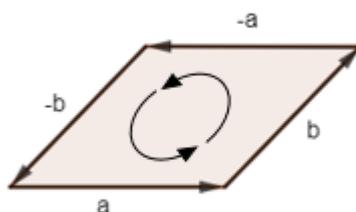
Hermann Grassmann apresentou seu produto externo em 1844 como a operação entre vetores que possui as seguintes propriedades:

1. **Antissimetria:** $a \wedge b = -b \wedge a$ (e consequentemente $a \wedge a = 0$);
2. **Distributividade:** $a \wedge (b + c) = a \wedge b + a \wedge c$;
3. **Associatividade:** $a \wedge (b \wedge c) = (a \wedge b) \wedge c$;
4. **Comutatividade de escalares:** $a \wedge \beta b = \beta a \wedge b = \beta(a \wedge b)$.

Com esta nova operação, Grassmann introduziu o conceito de bivector.

Dados dois vetores não colineares a e b , define-se como bivector o elemento bidimensional dotado de uma orientação e magnitude, nomeado $a \wedge b$. Este elemento é interpretado geometricamente como o paralelogramo cujos lados são determinados pelos vetores a e b . A orientação do bivector é definida como positiva para o sentido anti-horário, partindo de a em direção a b , e sua magnitude é dada pela área do paralelogramo. Veja que um bivector é um elemento do subespaço de dimensão 2. Além disso, o conceito de orientação dos bivectores permite verificar a natureza não comutativa do produto externo, ilustrada na Figura 20.

Figura 20 – Representação do bivector $a \wedge b$.



Fonte: Elaborado pelo autor.

A interpretação geométrica utilizada para revelar a natureza do elemento bivectorial também é útil para identificar que o produto externo de um vetor por um bivector (não coplanares) resulta em um elemento do espaço de dimensão 3 e, portanto, não deve ser de natureza escalar, vetorial ou bivectorial. De fato, o produto externo entre três vetores a , b e c (definido valendo-se da associatividade) corresponde geometricamente ao paralelepípedo cujos lados são determinados por estes. Tal elemento, chamado trivector,

tem orientação positiva se a sequência dos vetores a , b e c satisfazem a regra da mão direita, e magnitude dada pelo volume do paralelepípedo. Como estamos trabalhando com vetores de \mathbb{R}^3 , o produto de mais de três vetores não gera novos elementos (por não haver possibilidade de combinação LI). Além disso, escalares são interpretados como 0-vetores, elementos do subespaço de dimensão nula, com magnitude dada pelo módulo e direção determinada pelo sinal.

Generalizando estes conceitos para espaços de dimensões genéricas temos as seguintes definições.

Definição 3.1. *Chama-se k -blade o elemento resultante do produto externo entre k vetores linearmente independentes.*

Definição 3.2. *Chama-se k -vetor o elemento descrito como combinação linear não nula entre k -blades geradas a partir da base canônica.*

É preciso um certo cuidado com essas definições, pois toda k -blade é um k -vetor, no entanto a recíproca nem sempre é verdadeira. No espaço \mathbb{R}^3 estas duas definições coincidem, contudo para espaços de dimensões maiores isso não é verificado. Um exemplo simples para onde isso pode ser verificado é considerar o 2-vetor $e_1 \wedge e_2 + e_3 \wedge e_4$ gerado a partir da base de \mathbb{R}^4 , observe que este elemento não pode ser escrito como o produto externo entre dois vetores linearmente independentes e, portanto, não é uma 2-blade. Uma pesquisa mais aprofundada a respeito desse tema pode ser feita em [Macdonald \(2010\)](#).

Voltando ao espaço \mathbb{R}^3 , de nosso interesse, temos que o espaço definido por Grassmann, chamado de espaço exterior, contém todos estes elementos (escalares, vetores, bivectores, trivetores) e suas combinações lineares. Um elemento genérico do espaço exterior de Grassmann é chamado multivetor, e possui em si uma parte escalar, uma parte vetorial, uma parte bivetorial e uma parte trivetorial. Esses conceitos quando estendidos a espaços de dimensões maiores, culminam na criação de elementos associados a subespaços de dimensões maiores como quadrivectores, quintivetores e assim por diante.

A álgebra desenvolvida por Grassmann possui um forte apelo geométrico, permitindo explorar tal sentido para o produto externo. Uma análise geométrica detalhada pode ser encontrada em [Fernandes, Lavor e Oliveira \(2017\)](#) e [Kanatani \(2015\)](#). Aqui, focaremos em estabelecer axiomáticamente as características deste produto.

Qualquer elemento do espaço tridimensional pode ser representado em termos da base canônica. Usando as quatro propriedades do produto externo, temos também uma base para descrever os elementos do espaço exterior de Grassmann. Sendo $a, b \in \mathbb{R}^3$, onde $a = a_1e_1 + a_2e_2 + a_3e_3$ e $b = b_1e_1 + b_2e_2 + b_3e_3$, temos

$$\begin{aligned}
a \wedge b &= (a_1e_1 + a_2e_2 + a_3e_3) \wedge (b_1e_1 + b_2e_2 + b_3e_3) \\
&= (a_1b_1)e_1 \wedge e_1 + (a_1b_2)e_1 \wedge e_2 + (a_1b_3)e_1 \wedge e_3 \\
&\quad + (a_2b_1)e_2 \wedge e_1 + (a_2b_2)e_2 \wedge e_2 + (a_2b_3)e_2 \wedge e_3 \\
&\quad + (a_3b_1)e_3 \wedge e_1 + (a_3b_2)e_3 \wedge e_2 + (a_3b_3)e_3 \wedge e_3 \\
&= (a_1b_2 - a_2b_1)e_1 \wedge e_2 + (a_1b_3 - a_3b_1)e_1 \wedge e_3 + (a_2b_3 - a_3b_2)e_2 \wedge e_3,
\end{aligned}$$

implicando que todo bivector pode ser determinado como combinação linear entre os elementos do conjunto $\{e_1 \wedge e_2, e_1 \wedge e_3, e_2 \wedge e_3\}$. De maneira análoga, dados os vetores $a, b, c \in \mathbb{R}^3$, o trivector $a \wedge b \wedge c$ pode ser escrito em termos dos elementos da base canônica como

$$a \wedge b \wedge c = (a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_1b_3c_2 - a_2b_1c_3 - a_3b_2c_1)e_1 \wedge e_2 \wedge e_3.$$

Ou seja, todo trivector é múltiplo do trivector canônico.

A combinação de mais de três vetores, por meio do produto externo, não gera elementos novos além dos já conhecidos. Este fato pode ser verificado a partir da igualdade $e_i \wedge e_i = 0$, para $i = 1, 2, 3$. Sabemos que o produto externo gera elementos do espaço que podem ser representados em termos dos elementos da base (e seus produtos) e, conseqüentemente, o produto entre mais de três vetores zera automaticamente. Para exemplificar vamos multiplicar o bivector $e_1 \wedge e_3$ pelo bivector $e_1 \wedge e_2$:

$$(e_1 \wedge e_3) \wedge (e_1 \wedge e_2) = e_1 \wedge (e_3 \wedge e_1) \wedge e_2 = e_1 \wedge -(e_1 \wedge e_3) \wedge e_2 = -(e_1 \wedge e_1) \wedge (e_3 \wedge e_2) = 0.$$

Ou seja, o produto entre mais de três vetores quaisquer sempre resulta na combinação dos termos já apresentados, de modo que o elemento genérico do espaço exterior é dado por

$$a_1 + v_1e_1 + v_2e_2 + v_3e_3 + b_1e_1 \wedge e_2 + b_2e_1 \wedge e_3 + b_3e_2 \wedge e_3 + t_1e_1 \wedge e_2 \wedge e_3,$$

onde a_1 é a parte escalar, (v_1, v_2, v_3) a parte vetorial, (b_1, b_2, b_3) a parte bivectorial e t_3 a parte trivectorial do multivector.

3.4 O produto geométrico

Com esta noção geral da álgebra exterior de Grassmann e seu produto externo, vamos agora estudar a álgebra geométrica. A criação do chamado produto geométrico, desenvolvido por Clifford, permitiu unificar a álgebra dos quatérnios e a álgebra exterior de Grassmann. O produto geométrico, também conhecido como produto de Clifford, é associativo mas não comutativo, similar ao produto externo de Grassmann. A equação fundamental da álgebra geométrica estabelece uma relação direta entre estes dois produtos e definiremos aqui o produto geométrico a partir dela.

Definição 3.3. Dados dois vetores $a, b \in \mathbb{R}^3$, o produto geométrico de a por b , denotado simplesmente por ab , é definido por

$$ab = a \cdot b + a \wedge b. \quad (3.6)$$

A Equação (3.6), a partir da qual definimos o produto geométrico, é chamada Equação Fundamental da Álgebra Geométrica.

Ao apresentar o produto externo, usamos os elementos da base canônica para estabelecer seu funcionamento no espaço multivetorial. Baseando-se nesta mesma estratégia, podemos avaliar como o produto geométrico atua sobre os elementos da base de \mathbb{R}^3 .

Partindo da equação fundamental,

$$e_i e_i = e_i^2 = e_i \cdot e_i + e_i \wedge e_i = 1, \quad i \in \{1, 2, 3\},$$

e também

$$e_i e_j = e_i \cdot e_j + e_i \wedge e_j = e_i \wedge e_j, \quad i, j \in \{1, 2, 3\},$$

temos $e_i e_j = -e_j e_i$, para $i \neq j$, e $e_i e_i = e_i^2 = 1$. Estas igualdades, junto à propriedade de associatividade, nos dão condição de calcular uma grande diversidade de produtos, faltando ainda estabelecer o produto entre e_1, e_2 e e_3 .

Para determinar o produto geométrico entre os três elementos que compõem a base, vamos escrever

$$e_1 e_2 e_3 = e_1 (e_2 \wedge e_3).$$

Infelizmente, a relação é estabelecida entre os produtos interno, externo e geométrico, na equação fundamental, é válida somente para vetores. Entretanto, uma forma similar é dada para casos mais gerais. Segundo [Macdonald \(2010\)](#) e [Kanatani \(2015\)](#), dado um vetor a e um k -vetor B , vale

$$aB = a \cdot B + a \wedge B. \quad (3.7)$$

Com isso, podemos nos guiar pelas características geométricas dos elementos presentes nas equações, o que implica que

$$\begin{aligned} e_1 e_2 e_3 &= e_1 (e_2 \wedge e_3) \\ &= e_1 \cdot (e_2 \wedge e_3) + e_1 \wedge (e_2 \wedge e_3) \\ &= e_1 \wedge e_2 \wedge e_3. \end{aligned}$$

A nulidade do produto interno entre o vetor e_1 e o bivector $e_2 \wedge e_3$ é justificada pela ortogonalidade entre tais elementos, uma vez que e_1 é perpendicular ao plano em que o paralelogramo está contido. Com isso, o produto geométrico coincide com o trivetor $e_1 \wedge e_2 \wedge e_3$. Este elemento é comumente referido como pseudoescalar, denotado por ι , e tem grande importância dentro da álgebra geométrica.

O produto entre quatro ou mais vetores não representa novos elementos, uma vez que estamos trabalhando na dimensão três. Assim, o elemento genérico do espaço vetorial, com produto geométrico, é dado por

$$a + v_1e_1 + v_2e_2 + v_3e_3 + b_1e_1e_2 + b_2e_1e_3 + b_3e_2e_3 + te_1e_2e_3$$

Uma vez estabelecidas as regras operatórias do produto geométrico entre os elementos da base, pode-se calcular tal produto entre quaisquer elementos do espaço multivetorial. Vejamos o exemplo a seguir:

$$\begin{aligned} (3e_2 + 2e_1e_3)(4 - 2e_1e_2e_3) &= 12e_2 + 8e_1e_3 - 6e_2e_1e_2e_3 - 4e_1e_3e_1e_2e_3 \\ &= 12e_2 + 8e_1e_3 + 6e_1e_2e_2e_3 + 4e_3e_1e_1e_2e_3 \\ &= 12e_2 + 8e_1e_3 + 6e_1e_3 + 4e_3e_2e_3 \\ &= 12e_2 + 8e_1e_3 + 6e_1e_3 - 4e_2e_3e_3 \\ &= 12e_2 + 8e_1e_3 + 6e_1e_3 - 4e_2 \\ &= 8e_2 + 14e_1e_3. \end{aligned}$$

3.5 Produto interno multivetorial

Partir da equação fundamental é um ótimo caminho para a definição do produto geométrico, uma vez que os produtos interno e externo estão bem definidos. Contudo, conforme [Macdonald \(2010\)](#), o produto geométrico é o mais importante para as teorias da álgebra geométrica. Além disso, para o cálculo do produto entre três vetores, tivemos que recorrer às características geométricas de seus fatores, o que não seria muito prático em espaços de dimensões maiores. Deste modo, vamos definir os produtos interno e externo em termos do produto geométrico.

Utilizando a equação fundamental da álgebra geométrica, podemos reescrever o produto interno e o produto externo em termos do produto geométrico. Para isso, sendo a e b dois vetores, temos

$$a \cdot b = \frac{1}{2}(ab + ba)$$

e

$$a \wedge b = \frac{1}{2}(ab - ba).$$

Conforme a Equação (3.7), podemos fazer o mesmo para o produto entre um vetor e um k -vetor. Os produtos interno e externo podem, então, ser definidos como

$$a \cdot B = \frac{1}{2}(aB - (-1)^k Ba) \quad (3.8)$$

e

$$a \wedge B = \frac{1}{2}(aB + (-1)^k Ba).$$

A demonstração dessa igualdade pode ser encontrada em [Macdonald \(2010, p. 128\)](#), na versão com k -blades. A validade para k -vetores pode ser facilmente determinada a partir da distributividade do produto geométrico.

Ainda que a Equação (3.7) forneça uma generalização da equação fundamental, ela não é suficiente para estabelecer o produto interno entre multivetores, o que será necessário mais à frente. Para fazê-lo, vamos primeiramente determinar o produto interno entre k -vetores e, então, estendê-lo para o caso de multivetores. Aqui, vamos definir o produto interno conforme [Macdonald \(2010\)](#) e usaremos $\langle \rangle_k$ para indicar a parte k -vetorial de um multivetor.

Definição 3.4. *Dado A um k -vetor e B um j -vetor, define-se o produto interno entre A e B como a parte $(j - k)$ -vetorial do produto geométrico AB , ou seja,*

$$A \cdot B = \langle AB \rangle_{j-k}. \quad (3.9)$$

Se $j < k$ o produto é definido como 0.

Esta operação, por vezes, é referida como contração à esquerda e pode ser denotada pelo símbolo "J". Uma versão similar pode ser elaborada para escrever o produto externo em termos do produto geométrico (ver [Macdonald \(2010\)](#)).

A Equação (3.9) generaliza o produto interno entre vetores. De fato, o produto geométrico entre dois vetores a e b é dado pela soma de um escalar, $a \cdot b$, e um bivector $a \wedge b$. Desse modo, temos $a \cdot b = \langle ab \rangle_0$. Resta, apenas, definir o produto interno para multivetores quaisquer.

Sendo A um multivetor, de um espaço gerado a partir de \mathbb{R}^n , A pode ser escrito como a soma de seus k -vetores, isto é,

$$A = \langle A \rangle_0 + \langle A \rangle_1 + \langle A \rangle_2 + \cdots + \langle A \rangle_n.$$

Com isso, o produto interno pode ser definido a partir desta decomposição, ou seja, dados A e B dois multivetores,

$$A \cdot B = \sum_{j=0}^n \langle A \rangle_j \cdot \sum_{k=0}^n \langle B \rangle_k = \sum_{j,k=0}^n \langle A \rangle_j \cdot \langle B \rangle_k.$$

4 Estudo de proteínas com álgebra geométrica

Com o produto geométrico bem definido, analisaremos, agora, mais atentamente o resultado do produto entre dois vetores. Dados $a, b \in \mathbb{R}^3$, o produto ab resulta na soma de um escalar $a \cdot b$ e um bivector $a \wedge b$, de maneira tal que podemos escrever

$$ab = c_0 + c_1 e_2 e_3 + c_2 e_3 e_1 + c_3 e_1 e_2 = c_0 + \iota c, \quad (4.1)$$

onde $c = (c_1, c_2, c_3)$ e $\iota = e_1 e_2 e_3$, pseudoescalar. Agora, vamos estabelecer a seguinte identificação,

$$i = -e_2 e_3 = -\iota e_1 \quad j = -e_3 e_1 = -\iota e_2 \quad k = -e_1 e_2 = -\iota e_3. \quad (4.2)$$

Estas igualdades identificam dentro da álgebra de Clifford a álgebra dos quatérnios. Deste modo, podemos estabelecer uma correspondência entre o conjunto dos quatérnios e o produto geométrico entre dois vetores. Essa correspondência é de extrema importância, uma vez que nos permite trabalhar com rotores, apresentados anteriormente, como elementos da álgebra geométrica.

Para exemplificar este resultado, vamos considerar um quatérnio unitário q . Se v é um vetor qualquer do espaço, então o resultado da rotação de v por q é dado por qvq^\dagger . Com base na Equação (4.1), podemos escolher $c_0 = q_1$, $c_1 = q_2$, $c_2 = q_3$ e $c_3 = q_4$, fazendo com que $c_0 + \iota c$ corresponda a q^\dagger . Por outro lado, temos $c_0 - \iota c$ correspondendo a q . Assim, a rotação qvq^\dagger pode ser realizada no espaço geométrico pelo produto $(c_0 - \iota c)v(c_0 + \iota c)$. Em uma notação simplificada, podemos chamar $c_0 - \iota c$ de R e $c_0 + \iota c$ de R^\dagger , ou seja, a rotação é dada por RvR^\dagger .

Uma operação essencial para construção de rotações com quatérnios é a conjugação. Para definir adequadamente rotores a partir da álgebra geométrica é preciso, também, determiná-la dentro dessa álgebra. Neste caso, a conjugação é dada pelo produto dos conjugados, em ordem reversa, dos elementos da base que compõem a k -blade. Para qualquer vetor e_i na base, seu conjugado é $e_i^\dagger = -e_i$. Dessa maneira, valendo-se da linearidade da conjugação, para os demais k -vetores da base, usa-se o produto inverso dos conjugados, isto é,

$$(e_i e_j)^\dagger = e_j^\dagger e_i^\dagger = (-e_j)(-e_i) = e_j e_i = -e_i e_j.$$

Os elementos escalares são inalterados pela conjugação. Não entraremos em maiores detalhes a respeito de conjugação e dualidade, pois essa definição é suficiente para este texto. Para uma consulta mais aprofundada a respeito deste assunto, sugerimos a leitura de [Selig \(2004\)](#).

Sabemos que rotores são operadores ortogonais e podemos de fato confirmar tal afirmação com o uso da conjugação estabelecida para álgebra geométrica. Dado um rotor $R = \cos \frac{\theta}{2} - \text{sen} \frac{\theta}{2} \iota b$, este realiza a rotação com relação ao eixo determinado por b , por um ângulo θ . Naturalmente, a operação inversa é dada pela rotação em relação ao eixo de b , pelo mesmo ângulo com orientação oposta, identificado por $-\theta$. Temos então $R^{-1} = \cos \frac{-\theta}{2} - \text{sen} \frac{-\theta}{2} \iota b$. Pela paridade das funções seno e cosseno,

$$R^{-1} = \cos \frac{\theta}{2} + \text{sen} \frac{\theta}{2} \iota b = R^\dagger,$$

confirmando a ortogonalidade do operador. Por fim, vamos verificar que, de fato, R^{-1} é inverso de R .

$$\begin{aligned} RR^{-1} &= (\cos \frac{\theta}{2} - \text{sen} \frac{\theta}{2} \iota b)(\cos \frac{\theta}{2} + \text{sen} \frac{\theta}{2} \iota b) \\ &= \cos \frac{\theta}{2} \cos \frac{\theta}{2} - \text{sen} \frac{\theta}{2} \iota b \text{sen} \frac{\theta}{2} \iota b \\ &= \cos^2 \frac{\theta}{2} - \text{sen}^2 \frac{\theta}{2} \iota b \iota b \\ &= \cos^2 \frac{\theta}{2} - \text{sen}^2 \frac{\theta}{2} \iota^2 b^2 \\ &= \cos^2 \frac{\theta}{2} + \text{sen}^2 \frac{\theta}{2} \\ &= 1. \end{aligned}$$

Dentro da álgebra geométrica, temos, ainda, outra possibilidade de representação para um rotor. A forma exponencial, dada por $R = e^{-\frac{\theta}{2} \iota b}$, definida a partir da expansão de Taylor. A validade desta expressão é verificada pela propriedade do pseudoescalar ι que responde de maneira similar a unidade imaginária dos complexos em relação a potências, isto é, $\iota^2 = -1$.

Uma vez estabelecidos rotores como elementos do espaço multivetorial gerado a partir do produto geométrico, usaremos estes para determinar modelos de conversão e atualização das coordenadas cartesianas. Os métodos apresentados ao longo desta seção foram compilados a partir dos trabalhos de [Chys e Chacón \(2012\)](#), [Choi \(2006\)](#) e [Seok e Coutsias \(2007\)](#).

4.1 Conversão de coordenadas com rotores

Ao apresentar a álgebra desenvolvida por Hamilton, mostramos como quatérnios codificam de maneira eficiente rotações tridimensionais com relação a eixos arbitrários. Estabelecemos, também, uma relação entre os quatérnios e o produto geométrico de dois vetores, o que permite determinar uma nova forma de representação para rotores dentro da álgebra geométrica. Com isso, vamos determinar a construção das coordenadas cartesianas substituindo as matrizes de rotação por rotores.

Na álgebra geométrica rotores também podem ser convertidos para a forma matricial. Mais que isso, a matriz associada ao rotor tem a mesma forma, seja este determinado por um quatérnio, seja pela álgebra geométrica. Para relembrar esta construção, vamos considerar o rotor dado por $R = a_0 - \iota b$, onde $b = (b_1, b_2, b_3)$, e $R^{-1} = a_0 + \iota b$. Aplicando tal rotação em um vetor genérico $v = v_1 e_1 + v_2 e_2 + v_3 e_3$, temos

$$\begin{aligned} RvR^{-1} &= R(v_1 e_1 + v_2 e_2 + v_3 e_3)R^{-1} \\ &= Rv_1 e_1 R^{-1} + Rv_2 e_2 R^{-1} + Rv_3 e_3 R^{-1} \\ &= v_1 R e_1 R^{-1} + v_2 R e_2 R^{-1} + v_3 R e_3 R^{-1}. \end{aligned}$$

A rotação pode ser representada, portanto, por meio de um produto matriz-vetor, onde cada coluna da matriz é dada por $R e_i R^{-1}$, com $i = (1, 2, 3)$. Assim, a forma matricial de R é

$$2 \begin{bmatrix} a_0^2 + b_1^2 - \frac{1}{2} & b_1 b_2 - a_0 b_3 & b_1 b_3 + a_0 b_2 \\ b_1 b_2 + a_0 b_3 & a_0^2 + b_2^2 - \frac{1}{2} & b_2 b_3 - a_0 b_1 \\ b_1 b_3 - a_0 b_2 & b_2 b_3 + a_0 b_1 & a_0^2 + b_3^2 - \frac{1}{2} \end{bmatrix}. \quad (4.3)$$

Com essa versão matricial, basta alguns ajustes para que os algoritmos já estabelecidos funcionem. Neste modelo, as rotações são construídas com uso de rotores e aplicadas por meio de sua forma matricial. O uso da forma matricial de um rotor permite manter as Equações (1.2) e (1.6) usadas para determinar cada ponto. Para isso, basta substituir as matrizes de rotação tradicionais, que já foram utilizadas, pela forma matricial de rotores.

Para tal construção, aplicamos a rotação plana em torno do eixo z e na sequência a rotação de torção em torno do eixo x , os respectivos rotores são $R^{(\theta)} = e^{-\frac{\theta}{2}\iota e_3}$ e $R^{(\varphi)} = e^{-\frac{\varphi}{2}\iota e_1}$. A rotação será aplicada, por meio da forma matricial (4.3), no vetor e_1 para gerar os vetores de ligação. Deste modo, apenas a primeira coluna da matriz precisa de fato ser calculada. Este procedimento, baseado no modelo apresentado em [Chys e Chacón \(2012\)](#), que usa translações finais, é sintetizado no Algoritmo 8.

Estimando o custo computacional deste método pelo número de operações necessárias para adicionar um novo ponto na cadeia, podemos estabelecer uma comparação com o método matricial já apresentado.

O custo para estabelecer um novo ponto no sistema de coordenadas cartesianas é de: 4 operações trigonométricas, duas divisões por 2, 16 multiplicações e 8 somas para determinar o rotor $R_{[i]}$; 6 multiplicações e 4 somas para determinar a primeira coluna da matriz de rotação; 4 multiplicações para determinação de b_i e, por fim, 3 somas para translação de x_{i-1} . Totalizando 4 operações trigonométricas 28 multiplicações e 15 somas.

A Tabela 4 estabelece uma comparação entre os métodos, onde é possível verificar que ambos tem custos muito similares. Os símbolos \times , $+$ e \mathcal{T} indicam, respectivamente,

Algoritmo 8: Conversão de coordenadas com rotores.

Entrada: $I = \{(d_i, \theta_i, \varphi_i), i = 1, \dots, n\}$

Saída: $C = \{x_1, x_2, x_3, \dots, x_n\}$

início

$x_1 \leftarrow 0;$

$x_2 \leftarrow d_2 e_1;$

$R_{[2]} \leftarrow 1;$

para $i = 3, \dots, n$ **faça**

$R^{\theta_i} \leftarrow e^{-\frac{\theta_i}{2} \iota e_3};$

$R^{\varphi_i} \leftarrow e^{-\frac{\varphi_i}{2} \iota e_1};$

$R_i \leftarrow R^{\varphi_i} R^{\theta_i};$

$R_{[i]} \leftarrow R_{[i-1]} R_i;$

Determine a forma matricial $B_{[i]}$ do rotor $R_{[i]}$;

$b_i \leftarrow d_i B_{[i]} e_1;$

$x_i \leftarrow x_{i-1} + b_i;$

fim

fim

as operações de multiplicação/divisão, soma/subtração e operações trigonométricas.

Tabela 4 – Custo para determinar um novo ponto.

Método	\times	$+$	\mathcal{T}	Total
Matrizes	27	15	4	46
Rotores	28	15	4	47

A versão com rotores e translações encaixadas não é apresentada, embora possa ser realizada, pois seria necessária a construção total da versão matricial de um rotor, cujo custo é alto. Além disso, ao buscar minimizar o número de operações a implementação de tal método se tornaria redundante às demais. O que torna a versão com translações finais viável, além do baixo custo da composição de rotações, é o fato de que apenas a primeira coluna da forma matricial do rotor precisa ser calculada.

O algoritmo de conversão usando rotores foi implementado no *software Wolfram Mathematica v11.3*. Para a realização dos testes, utilizamos a conversão dos conjuntos de coordenadas internas para as cadeias BBN50, BBN200 e BBN1000 apresentados anteriormente. O tempo obtido na execução do algoritmo é descrito na Tabela 5.

Tabela 5 – Tempo para conversão de coordenadas internas para cartesianas com rotores

Conjunto	Tempo (s)
BBN50	0.015
BBN200	0.06
BBN1000	0.31

4.2 Atualização de coordenadas cartesianas por rotores

Na Seção 2, fizemos um estudo bastante detalhado sobre como mudanças nas coordenadas internas podem ser utilizadas para atualizar coordenadas cartesianas já calculadas. Estabelecemos também quais átomos do *backbone* devem ter coordenadas recalculadas e como determiná-las com o uso de matrizes de rotação. Agora que um novo caminho teórico para a conversão de coordenadas é possível, nada mais natural que explorar também a atualização de coordenadas a partir de tais teorias.

Vamos focar no cálculo da atualização dada a partir de uma modificação em φ_k , como fizemos para matrizes. Como a atualização por meio de rotores é bastante similar ao modelo apresentado no Algoritmo 6, vamos apenas ajustá-lo, substituindo as matrizes de rotação pelos rotores correspondentes.

O procedimento em questão consiste, basicamente, em transladar o centro de rotação para a origem do sistema coordenado, realizar a rotação adequada e então transladá-lo novamente para a posição inicial. Assim como no caso matricial, as translações podem ser intercaladas, sendo combinadas a cada novo ângulo, ou podem ser calculadas apenas ao final do processo, utilizando o último ponto atualizado para tal.

Considerando a atualização por translações encaixadas, se R corresponde à rotação necessária para atualizar os pontos da cadeia e t o vetor que translada o centro de rotação, a atualização de um certo x é dada algebricamente por $t + R(x - t)R^{-1}$. Usando a distributividade do produto, temos $t + RxR^{-1} - RtR^{-1}$. Isso implica que podemos separar a equação em duas partes, uma relativa a rotação RxR^{-1} e a outra a translação $t - RtR^{-1}$. Como rotores têm uma forma matricial, podemos determinar um método híbrido, combinando-a com o uso do espaço homogêneo. Para isso, basta substituir a matriz (2.3)

pela matriz (4.3) na Equação (2.5). O Algoritmo 9 estrutura estes procedimentos.

Algoritmo 9: Atualização com rotores e translações encaixadas.

Entrada: $C_C = \{x_1, x_2, \dots, x_n\}, \mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$

Saída: $\overline{C_C} = \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}$

início

$R_0 \leftarrow 1;$

$t_0 \leftarrow 0;$

para $i = 1, \dots, p$ **faça**

$b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{\|x_{k_i-2} - x_{k_i-1}\|};$

$R_i \leftarrow e^{-\frac{\varphi_{k_i}}{2} t b_{k_i-1}};$

Determine a forma matricial B_i para o rotor R_i ;

$t_i \leftarrow B_i(t_{i-1} - x_{k_{i-2}}) + x_{k_{i-2}};$

$R_{[i]} \leftarrow R_{[i-1]} R_i;$

Determine a forma matricial $B_{[i]}$ de $R_{[i]}$;

$M_i = \begin{pmatrix} B_{[i]} & t_i \\ 0 & 1 \end{pmatrix};$

para $j \geq k_1$ **faça**

se $j < k_p$ **então**

$l = \max_{k_r \leq j} \{r\};$

$\overline{x_j} \leftarrow M_{k_l} x_j;$

senão

$\overline{x_j} \leftarrow M_{k_n} x_j;$

fim

fim

fim

fim

A estimativa do custo computacional, para incluir a alteração de um ângulo de torção, é dada a partir da seguinte contagem de operações: 3 somas e 3 divisões para determinar o eixo de rotação; 3 multiplicações e 1 divisão por 2 para construção do rotor; 20 multiplicações e 20 somas para determinar as formas matriciais dos rotores; 9 multiplicações e 12 somas para determinar o vetor de translação; 16 multiplicações e 12 somas para o produto de rotores. Totalizando 52 multiplicações/divisões e 35 somas. Além disso, cada ponto para ser atualizado requer 9 multiplicações e 9 somas. Desta maneira, a estimativa de custo para a atualização da cadeia é dada por $87n_r + 18n_a$ operações.

No caso em que se utiliza translações finais, os vetores de translação não são combinados e atualizados a cada etapa. Ao invés disso, translada-se o último ponto com coordenadas atualizadas pelo vetor de ligação atualizado, fazendo com que as translações

sejam atualizadas implicitamente a partir do ponto anterior. Assim, como no caso de translações encaixadas, para estabelecer um novo algoritmo com rotores e translações finais, basta adaptar o Algoritmo 7. O Algoritmo 10 descreve o processo de atualização de coordenadas com uso de rotores e translações finais.

Algoritmo 10: Atualização com rotores e translações finais.

Entrada: $C_C = \{x_1, x_2, \dots, x_n\}, \mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$

Saída: $\overline{C_C} = \{\overline{x_1}, \overline{x_2}, \dots, \overline{x_n}\}$

início

$B_{[0]} \leftarrow I_3;$

para $i = 1, \dots, p$ **faça**

$b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{\|x_{k_i-2} - x_{k_i-1}\|};$

Calcule $R_i;$

$R_{[i]} \leftarrow R_{[i-1]}R_i;$

Determine a forma matricial $B_{[i]}$ de $R_{[i]}$;

para $k_i \leq j < k_{i+1}$ **faça**

$b_j = x_j - x_{j-1};$

$\overline{x_j} \leftarrow \overline{x_{j-1}} + B_{[i]}b_j;$

fim

fim

fim

Para este método de atualização, a contagem de operações é dada por: 3 somas e 3 divisões para determinar o eixo de rotação; 3 multiplicações e 1 divisão para determinar o rotor R_i ; 16 multiplicações e 12 somas para calcular o produto dos rotores; 10 multiplicações e 10 somas para determinar a forma matricial do rotor. Totalizando, assim, 33 multiplicações/divisões e 25 somas para a obtenção da rotação. Além disso, ainda são necessárias 3 somas para o cálculo do vetor de ligação, 9 produtos e 9 somas para atualizar cada ponto. Portanto, a estimativa para atualização da cadeia é de $58n_r + 21n_a$ operações.

Na Tabela 6, temos a comparação entre o número de operações necessárias para atualização de um ângulo em cada um dos métodos apresentados. Nesta tabela, TF indica o uso de translações finais e TE o uso de translações encaixadas. Com base nesses dados, podemos concluir que existe uma ligeira vantagem no uso de translações finais e, por esta razão, implementaremos apenas algoritmos que utilizem este modelo. Analisando o uso de matrizes *versus* o uso de rotores, podemos verificar uma diminuição de cerca de 10% no número de operações necessárias com o uso de translações encaixadas, e uma diferença ainda maior, em torno de 16%, quando utilizamos translações finais. Isso mostra a eficiência do uso de rotores para estabelecer rotações tridimensionais. Vale ainda ressaltar que, na conversão de coordenadas, as expectativas são de que ambos os métodos exijam praticamente o mesmo esforço computacional, com uma vantagem irrisória para matrizes.

Tabela 6 – Custo computacional para atualização de coordenadas.

Método	×	+	\mathcal{T}	Total
Matrizes - TE	54	43	2	99
Matrizes - TF	45	22	2	69
Rotores - TE	52	35	2	89
Rotores - TF	33	25	2	60

Fonte: Elaborada pelo autor com base em [Choi \(2006\)](#) e [Chys e Chacón \(2012\)](#).

Para verificar o tempo de execução, o algoritmo com rotores utilizando translações finais foi implementado no *software Wolfram Mathematica v11.3* e submetido aos mesmos testes de atualização já apresentados. O tempo (em segundos) de cada atualização está descrito na Tabela 7.

Tabela 7 – Tempo de execução do algoritmo de atualização com rotores com translações finais.

Conjunto	10%	40%	85%
BBN50	0.	0.	0.015
BBN200	0.	0.015	0.05
BBN1000	0.06	0.11	0.20

É possível utilizar rotores somente sob a perspectiva dos quatérnios. O uso da álgebra geométrica, contudo, permitirá avanços teóricos que a visão “limitada” dos quatérnios não pode explorar. É o caso, por exemplo, de explorar representações em dimensões maiores, objetivo que apresentaremos na seção seguinte.

5 Representação de proteínas no espaço conforme

O espaço homogêneo estabelecido na Subseção 1.2 utiliza um hiperplano do espaço de dimensão 4 para representar o espaço tridimensional. Na conversão de coordenadas, o uso deste espaço permite aplicar translações tridimensionais por meio de operadores lineares. Este fato torna possível a combinação de translações e rotações para atualização direta das coordenadas cartesianas, isto é, sem que seja preciso atualizar um ângulo de torção por vez. Nossa intenção agora é investigar um novo sistema de representação no chamado espaço conforme.

Para isso, é necessário fazer alguns apontamentos acerca do espaço conforme e como a álgebra geométrica se desenvolve neste caso. Uma consulta mais aprofundada pode ser realizada em [Kanatani \(2015\)](#), [Lounesto \(2001\)](#) e [Macdonald \(2010\)](#).

5.1 O espaço conforme

O espaço conforme é um espaço de dimensão cinco gerado a partir do completamento da base canônica de \mathbb{R}^3 com os vetores e_0 e e_∞ e da definição de um novo produto interno. Estes vetores são escolhidos de modo que sejam ortogonais à base $\{e_1, e_2, e_3\}$. O novo produto interno é estabelecido sem exigir a condição de positividade, ou seja, neste novo espaço o produto interno pode resultar valores negativos. Quando necessária a diferenciação entre este e o produto interno usual, iremos nos referir a tal como *produto interno conforme*.

O produto interno conforme é diferenciado do produto interno usual a partir de sua aplicação nos vetores e_0 e e_∞ , isto é, quando restrito ao espaço tridimensional, o produto interno conforme coincide com o produto interno usual. Assim, as igualdades a seguir são suficientes para defini-lo:

$$e_0 \cdot e_0 = 0, \quad e_\infty \cdot e_\infty = 0, \quad e_0 \cdot e_\infty = -1.$$

A última igualdade identifica a perda da positividade, enquanto as duas primeiras caracterizam os dois novos vetores como ortogonais a si mesmos. Além disso, temos $e_0 \cdot e_i = 0$ e $e_\infty \cdot e_i = 0$ para todo $i \in \{1, 2, 3\}$, uma vez que estes vetores foram escolhidos para a formação de uma nova base ortogonal para um espaço de dimensão 5. A única propriedade, característica de um produto interno, perdida no produto interno conforme é a positividade. Sendo assim, as demais propriedades de simetria e linearidade permanecem válidas.

Portanto, é suficiente definir o produto interno conforme apenas entre os elementos da base $\{e_0, e_1, e_2, e_3, e_\infty\}$.

Agora que o espaço conforme foi definido, é preciso determinar a forma de representação do espaço tridimensional neste. Para isto, considere o ponto $x = x_1e_1 + x_2e_2 + x_3e_3$ de \mathbb{R}^3 . Usaremos letras maiúsculas para denotar o correspondente de um ponto do espaço tridimensional no espaço conforme. De tal modo, a representação de x no espaço conforme é dada por

$$X = e_0 + x_1e_1 + x_2e_2 + x_3e_3 + \frac{1}{2}\|x\|^2e_\infty.$$

A fim de simplificar essa notação, vamos escrever simplesmente x^2 para denotar $\|x\|^2$, o que é justificado pela atuação do produto geométrico no espaço tridimensional. O produto geométrico foi definido a partir do produto externo e do produto interno usual, que coincide com o produto interno conforme quando restrito ao subespaço tridimensional. Com isso, temos $x^2 = x \cdot x + x \wedge x$, onde $x \cdot x = \|x\|^2$ e $x \wedge x = 0$.

A escolha dos símbolos e_0 e e_∞ para representar os vetores que completam a base do espaço tridimensional está associada ao papel desempenhado pelos mesmos no espaço conforme. De fato, a e_0 corresponde a origem do espaço tridimensional, e e_∞ está relacionado ao “infinito” deste. Esta segunda relação não é tão imediata quanto a primeira, mas pode ser verificada de maneira simples. Considerando X a representação do ponto x no espaço conforme, os pontos X e αX representam o mesmo ponto para todo α não nulo, então

$$p = \frac{X}{\frac{1}{2}x^2} = \frac{e_0}{\frac{1}{2}x^2} + \frac{x}{\frac{1}{2}x^2} + e_\infty$$

também é uma representação de x . Tomando $x^2 \rightarrow \infty$, resulta em $p \rightarrow e_\infty$, dessa maneira e_∞ representa, de fato, um ponto “infinito” do espaço tridimensional.

Esta forma de representação também permite codificar distâncias entre pontos a partir do produto interno conforme. Esta é uma das vantagens do uso deste espaço, uma vez que estamos interessados em analisar a matriz de distâncias da proteína. Para entender como esta relação se dá, vamos considerar os pontos $x, y \in \mathbb{R}^3$ e suas respectivas representações no espaço conforme,

$$X = e_0 + x + \frac{x^2}{2}e_\infty \quad \text{e} \quad Y = e_0 + y + \frac{y^2}{2}.$$

Temos, então,

$$\begin{aligned}
X \cdot Y &= (e_0 + x + \frac{x^2}{2} e_\infty) \cdot (e_0 + y + \frac{y^2}{2} e_\infty) \\
&= e_0 \cdot \frac{y^2}{2} e_\infty + x \cdot y + \frac{x^2}{2} e_\infty \cdot e_0 \\
&= x \cdot y + \frac{y^2}{2} e_0 \cdot e_\infty + \frac{x^2}{2} e_0 \cdot e_\infty \\
&= x \cdot y - \frac{y^2}{2} - \frac{x^2}{2} \\
&= x \cdot y - \frac{1}{2} (\|x\|^2 + \|y\|^2) \\
&= -\frac{1}{2} (\|x\|^2 - 2(x \cdot y) + \|y\|^2) \\
&= -\frac{1}{2} \|x - y\|.
\end{aligned}$$

Outra característica que vale ser mencionada, e é destacada em [Fernandes, Lavor e Oliveira \(2017\)](#), é que vetores que representam pontos de \mathbb{R}^3 são nulos, isto é, X é representante de algum ponto do espaço 3D somente se $X \cdot X = 0$. De fato, a codificação das distâncias apresentada resulta que

$$X \cdot X = -\frac{1}{2} \|x - x\| = 0.$$

5.2 Álgebra geométrica conforme

Uma vez que um forma de representação do espaço tridimensional foi estabelecido no espaço conforme, e desejamos trabalhar com proteínas usando este modelo, é preciso analisar como a álgebra geométrica se desenvolve sobre este espaço. Note que, pela equação fundamental (Equação (3.6)), o produto geométrico pode ser calculado a partir dos produtos externo e interno. Uma vez que um novo produto interno foi definido, faz-se necessária uma análise de quais consequências este terá sobre o produto geométrico.

Ao estabelecer o produto geométrico, é suficiente defini-lo para os vetores da base, fazendo com que o produto entre os demais elementos do espaço seja dado pelas propriedades de associatividade e distributividade. Para os vetores que compõem a base do espaço tridimensional $\{e_1, e_2, e_3\}$, as propriedades já foram determinadas, de modo que nos resta identificar como o produto geométrico atua sobre os novos vetores e_0 e e_∞ .

O caráter geométrico do produto externo explicita as primeiras relações estabelecidas entre os novos vetores adicionados, implicando em

$$e_0 \wedge e_0 = 0, \quad e_\infty \wedge e_\infty = 0, \quad e_0 \wedge e_\infty = -e_\infty \wedge e_0. \quad (5.1)$$

Da mesma maneira, os produtos entre os demais elementos da base são dados por

$$e_0 \wedge e_i = -e_i \wedge e_0, \quad e_\infty \wedge e_i = -e_i \wedge e_\infty, \quad \forall i \in \{1, 2, 3\}. \quad (5.2)$$

O produto externo está, portanto, definido para todos os elementos da base do espaço conforme e, conseqüentemente, utilizando suas propriedades de associatividade e distributividade, para todos os demais elementos do espaço.

Como o produto interno conforme já foi estabelecido para todos os elementos da base e, portanto, para todos os elementos do espaço conforme, o produto geométrico já pode ser definido.

O produto geométrico é definido nos elementos da base a partir das seguintes igualdades:

1. $e_0e_i + e_ie_0 = 0$, para $i \in \{1, 2, 3\}$;
2. $e_\infty e_i + e_ie_\infty = 0$, para $i \in \{1, 2, 3\}$;
3. $e_0^2 = 0$;
4. $e_\infty^2 = 0$;
5. $e_0e_\infty + e_\infty e_0 = -2$.

No caso do espaço tridimensional, o produto geométrico determina novos elementos, sendo estes associados a seus subespaços, gerando assim bivectores e trivectores. Como o espaço conforme tem dimensão cinco, além destes, ainda são definidos quadrivectores e quintivectores. Desta maneira, determinam-se 32 elementos para a base do espaço multivetorial construído sobre o espaço conforme, resultado das combinações do produto geométrico entre os elementos da base. Os elementos desta base são: um escalar, 5 vetores, 10 bivectores, 10 trivectores, 5 quadrivectores e um quintivector. Para uma simplificação da notação, vamos indicar os elementos por subíndices, por exemplo, o elemento denotado por $e_{03\infty}$ corresponde ao trivector $e_0e_3e_\infty$.

Para ilustrar o produto geométrico no espaço conforme, vamos considerar os multivetores $p_1 = 2e_{3\infty} - e_\infty$ e $p_2 = 3e_{03} + e_{12\infty}$. Com isso,

$$\begin{aligned}
 p_1p_2 &= 2e_3e_\infty 3e_0e_3 + 2e_3e_\infty e_1e_2e_\infty - e_\infty 3e_0e_3 - e_\infty e_1e_2e_\infty \\
 &= 6e_3(-2 - e_0e_\infty)e_3 - 2e_3e_1e_\infty e_2e_\infty - 3(-2 - e_0e_\infty)e_3 + e_1e_\infty e_2e_\infty \\
 &= 6e_3(-2e_3 - e_0e_\infty e_3) - 2e_1e_3e_2e_\infty e_\infty - 3(-2e_3 - e_0e_\infty e_3) - e_1e_2e_\infty e_\infty \\
 &= -12e_3e_3 - 6e_3e_0e_\infty e_3 - 0 + 6e_3 + 3e_0e_\infty e_3 - 0 \\
 &= -12 - 6e_0e_3e_3e_\infty + 6e_3 - 3e_0e_3e_\infty \\
 &= -12 + 6e_3 - 6e_0e_\infty - 3e_0e_3e_\infty.
 \end{aligned}$$

Agora que a álgebra geométrica conforme está bem definida, e foi estabelecido um meio de caracterizar pontos do espaço tridimensional dentro do espaço conforme, é

possível estabelecer um novo sistema de coordenadas para representação de proteínas. Neste novo sistema, o qual chamaremos coordenadas conformes, cada átomo será representado pelo ponto do espaço conforme correspondente ao ponto do espaço tridimensional que o descreve no sistema cartesiano. Para indicar o conjunto de coordenadas conformes usaremos C_O .

A conversão de coordenadas cartesianas para coordenadas conformes é demasiadamente simples. Para tal, basta calcular a distância de cada ponto até a origem do espaço tridimensional e, assim, determinar o coeficiente de e_∞ . O caminho contrário é mais simples ainda, uma vez que o conjunto de coordenadas cartesianas está embutido no conjunto de coordenadas conformes.

Como mencionado ao longo do texto, a mudança de coordenadas internas para cartesianas, com destaque para o cálculo das matrizes de distâncias, é a situação de maior importância prática. Assim, por duas razões, faz sentido estabelecer um modelo de conversão de coordenadas internas diretamente para coordenadas conformes. A primeira razão é que, como visto, coordenadas conformes encapsulam as coordenadas cartesianas, de modo que não faria sentido a conversão para coordenadas cartesianas como um passo intermediário para o cálculo das coordenadas conformes. A segunda razão, e talvez a mais interessante, é que devido às características do espaço conforme, as distâncias interatômicas podem ser simplesmente calculadas a partir do produto interno conforme.

5.3 Translações e Rotações no espaço conforme

Ao representar o espaço tridimensional como subconjunto de um espaço de dimensão maior, precisamos tomar alguns cuidados ao realizar operações de qualquer natureza. Com base nos algoritmos já estabelecidos, sabemos que, para a construção de coordenadas, são necessárias translações e rotações. Embora estas operações devam ser realizadas no espaço de representação, isto é, no espaço conforme, os movimentos devem ser observados no espaço de origem, ou seja, no espaço tridimensional. Para exemplificar esta situação, tomemos como exemplo o espaço homogêneo. Dados $x, t \in \mathbb{R}^3$, a translação do ponto x pelo vetor t é calculada pela soma $x + t$. Entretanto, ao passarmos para o espaço homogêneo, temos as respectivas representações de x e t dadas por $x + e_4$ e $t + e_4$. Se simplesmente calcularmos a soma, teremos como resultado $(x + t) + 2e_4$. Este elemento não representa nenhum ponto do espaço tridimensional, pois saiu do subconjunto cuja quarta coordenada é sempre 1. Por esta razão, é necessário definir um operador linear no espaço homogêneo que realize translações no espaço tridimensional. Vamos agora apresentar operadores do espaço conforme que realizam rotações e translações no espaço tridimensional.

Como sabemos, a representação de um ponto x no espaço conforme é dada por

$e_0 + x + \frac{1}{2}x^2$. Assim, a translação de x por um vetor t deve ser representada no espaço conforme por $e_0 + x + t + \frac{1}{2}(x + t)^2 e_\infty$.

Conforme definido em Kanatani (2015), as translações do espaço conforme são realizadas por

$$T_t X T_t^{-1},$$

onde X é a representação de x e T_t , chamado translador, é o operador de translação definido por

$$T_t = 1 - \frac{1}{2}te_\infty.$$

Sendo T_t um operador de translação, podemos presumir que o operador inverso T_t^{-1} corresponde à translação inversa, dada por $-t$, isto é,

$$T_t^{-1} = T_{-t} = 1 - \frac{1}{2}(-t)e_\infty = 1 + \frac{1}{2}te_\infty.$$

De fato,

$$\begin{aligned} T_t T_{-t} &= \left(1 - \frac{1}{2}te_\infty\right)\left(1 + \frac{1}{2}te_\infty\right) \\ &= 1 + \frac{1}{2}te_\infty - \frac{1}{2}te_\infty - \frac{1}{4}te_\infty te_\infty \\ &= 1. \end{aligned}$$

Para entender a atuação do translador T_t sobre um ponto X , vamos estabelecer como este age sobre cada elemento da base do espaço conforme.

Começando por $T_t e_0 T_t^{-1}$, temos

$$\begin{aligned} T_t e_0 T_t^{-1} &= \left(1 - \frac{1}{2}te_\infty\right)e_0\left(1 + \frac{1}{2}te_\infty\right) \\ &= \left(e_0 - \frac{1}{2}te_\infty e_0\right)\left(1 + \frac{1}{2}te_\infty\right) \\ &= e_0 - \frac{1}{2}te_\infty e_0 + \frac{1}{2}e_0 te_\infty - \frac{1}{4}te_\infty e_0 te_\infty \\ &= e_0 + \frac{1}{2}(e_0 te_\infty - te_\infty e_0) + \frac{1}{4}te_\infty te_0 e_\infty \\ &= e_0 + \frac{1}{2}(-t)(e_0 e_\infty + e_\infty e_0) - \frac{1}{4}t^2 e_\infty e_0 e_\infty \\ &= e_0 + \frac{1}{2}(-t)(-2) - \frac{1}{4}t^2(-2 - e_0 e_\infty)e_\infty \\ &= e_0 + t + \frac{1}{2}t^2 e_\infty. \end{aligned}$$

Isto significa que o translador mapeia a origem do espaço tridimensional no ponto correspondente a t no espaço conforme, de fato realizando a translação desejada.

Para os vetores que compõem a base do espaço 3D, a operação pode ser avaliada de maneira simultânea, então, sendo $x \in \mathbb{R}^3$, temos

$$\begin{aligned}
 T_t x T_t^{-1} &= \left(1 - \frac{1}{2} t e_\infty\right) x \left(1 + \frac{1}{2} t e_\infty\right) \\
 &= \left(x - \frac{1}{2} t e_\infty x\right) \left(1 + \frac{1}{2} t e_\infty\right) \\
 &= x - \frac{1}{2} t e_\infty x + \frac{1}{2} x t e_\infty - \frac{1}{4} t e_\infty x t e_\infty \\
 &= x + \frac{1}{2} (x t e_\infty - t e_\infty x) \\
 &= x + \frac{1}{2} (x t + t x) e_\infty \\
 &= x + (x \cdot t) e_\infty.
 \end{aligned}$$

A última linha é resultado da igualdade $x \cdot t = \frac{1}{2} (x t + t x)$.

Por fim, temos a atuação sobre e_∞ , dada por

$$\begin{aligned}
 T_t e_\infty T_t^{-1} &= \left(1 - \frac{1}{2} t e_\infty\right) e_\infty \left(1 + \frac{1}{2} t e_\infty\right) \\
 &= \left(e_\infty - \frac{1}{2} t e_\infty e_\infty\right) \left(1 + \frac{1}{2} t e_\infty\right) \\
 &= e_\infty + \frac{1}{2} e_\infty t e_\infty \\
 &= e_\infty.
 \end{aligned}$$

Lembrando que o vetor e_∞ está associado ao “infinito” do espaço tridimensional, logo, faz sentido este não ser alterado por uma translação.

Combinando os resultados obtidos, a atuação do translador no ponto X do espaço conforme é dada por

$$\begin{aligned}
 T_t X T_t^{-1} &= T_t \left(e_0 + x + \frac{x^2}{2} e_\infty\right) T_t^{-1} \\
 &= T_t e_0 T_t^{-1} + T_t x T_t^{-1} + \frac{x^2}{2} T_t e_\infty T_t^{-1} \\
 &= e_0 + t + \frac{1}{2} \|t\|^2 e_\infty + x + (x \cdot t) e_\infty + \frac{1}{2} \|x\|^2 e_\infty \\
 &= e_0 + (x + t) + \frac{1}{2} (\|x\|^2 + 2(x \cdot t) + \|t\|^2) e_\infty \\
 &= e_0 + (x + t) + \frac{1}{2} \|x + t\|^2 e_\infty.
 \end{aligned}$$

Com isso, mostramos que tal operador, de fato, realiza translações tridimensionais no espaço conforme.

O uso do espaço homogêneo permitiu realizar translações por meio de operadores lineares, resultando em alguns avanços teóricos tanto na conversão de coordenadas, como na sua atualização. Em particular, na atualização de coordenadas o uso do espaço homogêneo tornou possível combinar alterações de múltiplos ângulos de torção, evitando que as

coordenadas de cada átomo precisassem ser recalculadas a cada ângulo alterado. É desejável que esta característica permaneça válida ao se trabalhar com o espaço conforme. Felizmente, além dessa característica, que é mantida, surge também uma nova e interessante propriedade. A linearidade do transladador pode ser facilmente verificada a partir das propriedades do produto geométrico, sob o qual o operador é construído. Além disso, é possível verificar também sua ortogonalidade a partir de seu conjugado. Temos,

$$\begin{aligned}
(T_t)^\dagger &= \left(1 - \frac{1}{2}(t_1 e_1 e_\infty + t_2 e_2 e_\infty + t_3 e_3 e_\infty)\right)^\dagger \\
&= 1 - \frac{1}{2}(t_1 e_1 e_\infty + t_2 e_2 e_\infty + t_3 e_3 e_\infty)^\dagger \\
&= 1 - \frac{1}{2}(t_1 (e_1 e_\infty)^\dagger + t_2 (e_2 e_\infty)^\dagger + t_3 (e_3 e_\infty)^\dagger) \\
&= 1 - \frac{1}{2}(t_1 (-e_1 e_\infty) + t_2 (-e_2 e_\infty) + t_3 (-e_3 e_\infty)) \\
&= 1 + \frac{1}{2}(t_1 e_1 e_\infty + t_2 e_2 e_\infty + t_3 e_3 e_\infty) \\
&= T_t^{-1}.
\end{aligned}$$

Esta é uma vantagem extremamente significativa, pois se antes foi possível linearizar equações, estas agora serão dadas por operadores ortogonais. Vale observar que nos processos de conversão e atualização, translações são combinadas com rotações que, por sua vez, também são operadores ortogonais. Com isso, as equações utilizadas serão dadas por operadores lineares ortogonais.

Tal qual foi feito com rotores na Subseção 3.2, vamos estabelecer aqui uma forma matricial para transladores. De maneira similar, vamos gerar as colunas da matriz aplicando os transladores em cada um dos vetores que compõem a base do espaço conforme $\{e_0, e_1, e_2, e_3, e_\infty\}$. Como já fizemos um estudo da atuação do translador nestes elementos, a primeira e última coluna, $T_t e_0 T_t^{-1}$ e $T_t e_\infty T_t^{-1}$, respectivamente, já são conhecidas. Para as demais colunas, basta substituir os vetores e_1, e_2, e_3 na equação $T_t x T_t^{-1}$, assim temos a seguinte matriz:

$$\begin{pmatrix}
1 & 0 & 0 & 0 & 0 \\
t_1 & 1 & 0 & 0 & 0 \\
t_2 & 0 & 1 & 0 & 0 \\
t_3 & 0 & 0 & 1 & 0 \\
\frac{t^2}{2} & t_1 & t_2 & t_3 & 1
\end{pmatrix},$$

que pode ser escrita em blocos como

$$\begin{pmatrix}
1 & 0 & 0 \\
t & I_3 & 0 \\
\frac{t^2}{2} & t^t & 1
\end{pmatrix}.$$

A construção de rotações é mais simples, uma vez que rotores estabelecidos no espaço tridimensional permanecem funcionais no espaço conforme. Para verificar isso, é suficiente analisar sua atuação nos novos vetores e_0 e e_∞ . Como vimos nas seções anteriores, dado um ângulo θ e um eixo orientado pelo vetor unitário b , o rotor $R = \cos \frac{\theta}{2} - \text{sen} \frac{\theta}{2} \iota b$, onde $\iota = e_{123}$, rotaciona o vetor $x \in \mathbb{R}^3$ por meio do produto RxR^{-1} . Temos, então,

$$Re_0R^{-1} = RR^{-1}e_0 = e_0$$

e

$$Re_\infty R^{-1} = RR^{-1}e_\infty = e_\infty.$$

Uma vez que e_0 e e_∞ estão associados, respectivamente, à origem e ao infinito do espaço tridimensional, faz sentido que estes permaneçam inalterados por rotações.

Portanto, rotações no espaço conforme são realizadas da seguinte maneira:

$$\begin{aligned} R(e_0 + x + \frac{x^2}{2}e_\infty)R^{-1} &= Re_0R^{-1} + RxR^{-1} + \frac{x^2}{2}Re_\infty R^{-1} \\ &= e_0 + RxR^{-1} + \frac{x^2}{2}e_\infty \\ &= e_0 + RxR^{-1} + \frac{(RxR^{-1})^2}{2}e_\infty. \end{aligned}$$

Esta última igualdade decorre do fato de que rotações são isometrias e, com isso, não alteram a norma do vetor. Logo, rotores construídos no espaço 3D realizam também rotações em suas representações no espaço conforme. Na verdade, mais que isso, rotores realizam rotações de quaisquer elementos do espaço conforme, incluindo outros operadores, informação que será particularmente útil ao combinarmos operadores.

A forma matricial de um rotor no espaço conforme pode ser facilmente obtida, uma vez que já estabelecemos a atuação dos rotores nos vetores da base. Dado um rotor $R = a_0 - \iota b$, sua versão matricial é dada por

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2a_0^2 + 2b_1^2 - 1 & 2b_1b_2 - 2a_0b_3 & 2b_1b_3 + 2a_0b_2 & 0 \\ 0 & 2b_1b_2 + 2a_0b_3 & 2a_0^2 + 2b_2^2 - 1 & 2b_2b_3 - 2a_0b_1 & 0 \\ 0 & 2b_1b_3 - 2a_0b_2 & 2b_2b_3 + 2a_0b_1 & 2a_0^2 + 2b_3^2 - 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

ou, ainda, em uma forma reduzida

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

onde B é a forma matricial do rotor R , estabelecida em (4.3), para o espaço tridimensional.

Voltando brevemente ao problema de atualização de coordenadas, conforme apontado na Subseção 4.2, rotações são sempre realizadas em torno de um eixo que contém a origem do espaço. Desse modo, uma vez construído o rotor, é necessário ainda transladar todo o conjunto de pontos a ser rotacionado para que o vértice do ângulo de rotação coincida com a origem do espaço. Então, ao final do processo, é necessário ainda que os pontos sejam transladados de volta à posição anterior. Esta sequência de movimentos é descrita, no espaço conforme, pela sequência de produtos

$$T_t R T_{-t} X T_{-t}^{-1} R^{-1} T_t^{-1},$$

onde t é o vetor que posiciona o eixo de rotação sobre a origem, R é o rotor a ser aplicado e X o ponto que deve ser rotacionado.

Da ortogonalidade dos rotores e transladores e das propriedades da conjugação temos

$$T_{-t}^{-1} R^{-1} T_t^{-1} = T_{-t}^\dagger R^\dagger T_t^\dagger = (T_t R T_{-t})^\dagger$$

Por outro lado, $T_{-t}^{-1} R^{-1} T_t^{-1} = (T_t R T_{-t})^{-1}$, pois

$$(T_t R T_{-t})(T_{-t}^{-1} R^{-1} T_t^{-1}) = 1.$$

Com isso, podemos concluir que o produto $T_t R T_{-t}$ corresponde a um operador linear ortogonal.

Embora estejamos acostumados com a atuação de operadores de rotação e translação apenas sobre pontos e vetores, já mencionamos que estes atuam de maneira genérica sobre todos os elementos do espaço conforme. Um olhar mais atento para o produto $T_t R T_{-t}$ revela a natureza deste elemento. Podemos reescrevê-lo como $T_t R T_t^{-1}$, ou seja, este corresponde geometricamente à translação do rotor R por um vetor t . Se a álgebra dos quatérnios de Hamilton nos permitiu determinar rotações a partir de um eixo genérico, independentemente da decomposição em rotações pelos eixos coordenados, a álgebra geométrica conforme nos permite dar um passo além: rotacionar quaisquer elementos do espaço em torno de um eixo qualquer, passando pela origem ou não.

Vamos denotar por $R_{t,b}^\theta$ o rotor genérico, que realiza uma rotação por um ângulo θ com vértice do ângulo de rotação em t e cujo eixo de rotação é determinado por b , ou seja, $R_b^\theta = R_{0,b}^\theta$. Para estabelecermos a equação geral do rotor $R_{t,b}^\theta$, vamos calcular o produto que o determina:

$$\begin{aligned} T R T^{-1} &= \left(1 - \frac{1}{2} t e_\infty\right) R \left(1 + \frac{1}{2} t e_\infty\right) \\ &= \left(R - \frac{1}{2} t e_\infty R\right) \left(1 + \frac{1}{2} t e_\infty\right) \\ &= R - \frac{1}{2} t e_\infty R + \frac{1}{2} R t e_\infty - \frac{1}{4} t e_\infty R t e_\infty \\ &= R - \frac{1}{2} (R t e_\infty - t e_\infty R). \end{aligned}$$

Observe que a rotação com eixo passando pela origem pode ser identificada como um dos termos da soma que compõe o operador. Dessa forma, a segunda parcela da soma está associada ao movimento de deslocamento da origem, que é

$$\begin{aligned}
 Rt e_\infty - t e_\infty R &= \left(\cos \frac{\theta}{2} - \operatorname{sen} \frac{\theta}{2} e_{123} b\right) t e_\infty - t e_\infty \left(\cos \frac{\theta}{2} - \operatorname{sen} \frac{\theta}{2} e_{123} b\right) \\
 &= -\operatorname{sen} \frac{\theta}{2} (e_{123} b t e_\infty - t e_\infty e_{123} b) \\
 &= -\operatorname{sen} \frac{\theta}{2} (b t - t b) e_{123} e_\infty \\
 &= -2 \operatorname{sen} \frac{\theta}{2} (b \wedge t) e_{123} e_\infty.
 \end{aligned}$$

Voltando à equação anterior, a forma geral de um rotor é dada por

$$R_{t,b}^\theta = \cos \frac{\theta}{2} - \operatorname{sen} \frac{\theta}{2} (b + (t \wedge b) e_\infty) e_{123}. \quad (5.3)$$

Sabemos que o produto entre dois transladores resulta em um novo translador, isso porque a composição de translações resulta em uma translação. O mesmo vale para rotações com eixo passando pela origem, ou seja, a composição de duas rotações é, de fato, uma rotação. Infelizmente, isto não se aplica para rotações com vértice arbitrário. Isto significa que o produto entre rotores genéricos não necessariamente resulta em um rotor. Para verificar esta relação, tome dois rotores $R_{t,b}^\theta = R_1 + \alpha e_\infty$, com $R_1 = R_b^\theta$ e $\alpha = \operatorname{sen} \frac{\theta}{2} (b \times t)$, e $R_{u,c}^\varphi = R_2 + \beta e_\infty$, com $R_2 = R_c^\varphi$ e $\beta = \operatorname{sen} \frac{\varphi}{2} (c \times u)$. Então, o produto entre eles é dado por

$$\begin{aligned}
 R_{t,b}^\theta R_{u,c}^\varphi &= (R_1 + \alpha e_\infty)(R_2 + \beta e_\infty) \\
 &= R_1 R_2 + R_1 \beta e_\infty + \alpha e_\infty R_2 + \alpha e_\infty \beta e_\infty \\
 &= R_1 R_2 + \alpha e_\infty R_2 + R_1 \beta e_\infty.
 \end{aligned}$$

Novamente, é possível identificar dentro do produto uma rotação com eixo na origem e um termo de deslocamento, que analisaremos a parte. Para simplificar a notação vamos escrever $R_1 = a - e_{123} b$ e $R_2 = d - e_{123} c$, com isso,

$$\begin{aligned}
 R_1 \beta e_\infty + \alpha e_\infty R_2 &= (a - e_{123} b) \beta e_\infty + \alpha e_\infty (d - e_{123} c) \\
 &= (a \beta + d \alpha) e_\infty - e_{123} b \beta e_\infty - \alpha e_\infty e_{123} c \\
 &= (a \beta + d \alpha) e_\infty - (b \beta + \alpha c) e_{123} e_\infty.
 \end{aligned}$$

Observe, portanto, que o termo $b \beta + \alpha c$ resulta na soma de um escalar com um bivector. Deste modo, o quadrivector $e_{123} e_\infty$ é multiplicado por um escalar não nulo, dado por $b \cdot \beta + \alpha \cdot c$. Consequentemente, o resultado não pode ser um rotor, cuja coordenada em questão é nula. Portanto, o produto entre rotores nem sempre resulta em um novo rotor.

Este novo operador é o que se conhece como movimento de corpo rígido. Aqui, vamos nos referir a tal operador como motor, indicado geralmente por M . Um motor é o

resultado da composição de uma rotação com uma translação e, portanto, também são operadores ortogonais no espaço conforme. Embora o produto de dois rotores quaisquer não resulte, necessariamente, em um novo rotor, o produto entre dois motores resulta em um novo motor. Os resultados obtidos e apresentados a respeito de motores foram inicialmente baseados nas teorias contidas em [Selig \(2013\)](#).

Um motor é definido como uma rotação seguida por uma translação. Algebricamente, $M = TR = (1 - \frac{1}{2}t e_\infty)(a + be_{123})$. Calculando o produto geométrico, a forma geral de um motor, colocado em termos dos elementos da base do espaço multivetorial, é

$$M = a_0 + a_1e_{23} + a_2e_{31} + a_3e_{12} + a_4e_{1\infty} + a_5e_{2\infty} + a_6e_{3\infty} + a_7e_{123\infty}. \quad (5.4)$$

Os coeficientes a_0, a_1, a_2 e a_3 correspondem à parte rotacional do motor, enquanto os coeficientes a_4, a_5, a_6 e a_7 correspondem à parte associada à translação. Esta equação pode ser escrita também, de forma simplificada, como

$$M = a + be_{123} + ce_\infty + de_{123\infty},$$

onde $a, d \in \mathbb{R}$ e $b, c \in \mathbb{R}^3$. Faremos isso em alguns momentos para dar mais clareza ao texto.

Como já mencionamos, motores são, por construção, operadores ortogonais, ou seja,

$$M^{-1} = M^\dagger = a - be_{123} - ce_\infty + de_{123\infty}.$$

Por outro lado,

$$\begin{aligned} MM^\dagger &= (a + be_{123} + ce_\infty + de_{123\infty})(a - be_{123} - ce_\infty + de_{123\infty}) \\ &= a^2 - be_{123}be_{123} - be_{123}ce_\infty + be_{123}de_{123\infty} - ce_\infty be_{123} - de_{123\infty}be_{123} + 2ade_{123\infty} \\ &= a^2 + b \cdot b + (bc + cb)e_{123\infty} + 2ade_{123\infty} \\ &= a^2 + b \cdot b + 2(b \cdot c + ad)e_{123\infty} \end{aligned}$$

A ortogonalidade de M indica que $MM^\dagger = 1$. Desta forma, devemos ter necessariamente que $a^2 + b \cdot b = 1$, uma vez que esta é a parte rotacional do motor e, conseqüentemente, $ad + b \cdot c = 0$. Este fato é corroborado por [Selig e Bayro-Corrochano \(2010\)](#) e utilizado para definir formalmente um motor. Com base nestas igualdades, podemos estabelecer M como a soma entre um rotor com eixo passando na origem, um bivector e um quadrivector, formulação que será útil para determinar o produto entre motores.

Sejam $M_1 = R_1 + c_1 e_\infty + d_1 e_{123\infty}$ e $M_2 = R_2 + c_2 e_\infty + d_2 e_{123\infty}$, com R_1, R_2 rotores; c_1, c_2 vetores e d_1, d_2 escalares. O produto entre motores é dado por

$$\begin{aligned} M_1 M_2 &= (R_1 + c_1 e_\infty + d_1 e_{123\infty})(R_2 + c_2 e_\infty + d_2 e_{123\infty}) \\ &= R_1 R_2 + R_1 c_2 e_\infty + c_1 e_\infty R_2 + d_2 R_1 e_{123\infty} + d_1 R_2 e_{123\infty} \\ &= R_1 R_2 + (R_1 c_2 + c_1 R_2) e_\infty + (d_2 R_1 + d_1 R_2) e_{123\infty}. \end{aligned}$$

Para uma melhor organização dos cálculos, vamos avaliar separadamente a segunda e a terceira parcela da soma. Para isso, vamos considerar $R_1 = a_1 + b_1 e_{1213}$ e $R_2 = a_2 + b_2 e_{123}$, com a_1, a_2 escalares e b_1, b_2 vetores. Desta maneira,

$$\begin{aligned} (R_1 c_2 + c_1 R_2) e_\infty &= ((a_1 + b_1 e_{1213}) c_2 + c_1 (a_2 + b_2 e_{123})) e_\infty \\ &= (a_1 c_2 + a_2 c_1 + c_1 \times b_2 - b_1 \times c_2) e_\infty. \end{aligned}$$

Por outro lado,

$$\begin{aligned} (d_2 R_1 + d_1 R_2) e_{123\infty} &= (d_2 (a_1 + b_1 e_{1213}) + d_1 (a_2 + b_2 e_{123})) e_{123\infty} \\ &= (d_1 b_2 + d_2 b_1) e_\infty + (d_1 a_2 + d_2 a_1) e_{123\infty}. \end{aligned}$$

Portanto, o produto é dado por $M_1 M_2 = R_3 + c_3 e_\infty + d_3 e_{123\infty}$, onde

$$\begin{aligned} R_3 &= R_1 R_2, \\ c_3 &= a_1 c_2 + a_2 c_1 + c_1 \times b_2 - b_1 \times c_2 + d_1 b_2 + d_2 b_1, \\ d_3 &= d_1 a_2 + d_2 a_1. \end{aligned}$$

Note que o produto entre motores generaliza o produto entre rotores arbitrários. Também é possível calcular, por meio deste, o produto entre um motor e um rotor.

Resta, ainda, determinar a forma matricial de um motor, elemento particularmente útil para os problemas aqui apresentados. Para isso, as colunas da matriz serão determinadas pela atuação de M sobre cada um dos vetores da base do espaço conforme. De tal modo, a forma matricial é dada por

$$[M e_0 M^\dagger \mid M e_1 M^\dagger M e_2 M^\dagger \mid M e_3 M^\dagger \mid M e_\infty M^\dagger]$$

A fim de compactar o texto, vamos usar índices para indicar os produtos entre os coeficientes de M , ou seja, vamos escrever a_{ij} para indicar o produto $a_i a_j$. Dito isto, dado o motor M , definido conforme a Equação (5.4), tem forma matricial

$$2 \begin{pmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 \\ -(a_{17} + a_{26} + a_{35} + a_{04}) & a_0^2 + a_1^2 - \frac{1}{2} & a_{03} - a_{12} & a_{02} + a_{13} & 0 \\ -a_{05} - a_{16} + a_{27} + a_{34} & -(a_{12} + a_{03}) & a_0^2 + a_2^2 - \frac{1}{2} & a_{01} - a_{23} & 0 \\ -a_{06} + a_{15} + a_{24} - a_{37} & a_{13} - a_{02} & -(a_{01} + a_{23}) & a_0^2 + a_3^2 - \frac{1}{2} & 0 \\ a_4^2 + a_5^2 + a_6^2 + a_7^2 & -a_{04} + a_{35} + a_{26} - a_{17} & -a_{34} - a_{05} + a_{16} + a_{27} & -a_{24} - a_{15} - a_{06} - a_{37} & \frac{1}{2} \end{pmatrix}. \quad (5.5)$$

5.4 Sistema de coordenadas conformes

Com rotações e translações definidas no espaço conforme, e uma representação do espaço tridimensional, podemos gerar o novo sistema de coordenadas para identificar

conformações de proteínas. Para a construção das coordenadas conformes, vamos seguir o protocolo de construção das coordenadas cartesianas. Partindo de um conjunto de coordenadas internas já conhecidas, usaremos rotações e translações combinadas para realizar sua construção e atualização.

Conforme o protocolo de construção das coordenadas cartesianas com translações finais, dado o conjunto de coordenadas internas, fixa-se o primeiro ponto da cadeia na origem do espaço tridimensional, dado no espaço conforme por e_0 . O segundo ponto é estabelecido com posição correspondente ao eixo gerado por e_1 , assim este será fixado no ponto $e_0 + d_2 e_1 + \frac{d_2^2}{2} e_\infty$. Para o cálculo da quinta coordenada, a distância é dada pelo próprio conjunto de coordenadas internas, pois o vetor e_1 , que tem norma unitária, é usado para gerar os vetores de ligação. A determinação dos demais pontos é realizada pela sequência de rotações em ordem reversa que posiciona o vetor de ligação. A adequação do comprimento do vetor é realizada por um produto por escalar nas coordenadas associadas a e_1, e_2, e_3 , a coordenada e_0 permanece inalterada, e a coordenada e_∞ tem como coeficiente $\frac{d_i^2}{2}$, pois rotações não alteram o comprimento do vetor e_1 . Por fim para a construção do i -ésimo ponto, basta realizar a translação $T_{b_i} x_{i-1} T_{b_i}^{-1}$.

Na prática, a realização das rotações e translações envolvidas no processo são obtidas pela forma matricial, dispensando a necessidade de envolver os demais elementos do espaço multivetorial. Isto permite simplificar os cálculos, uma vez que dois produtos geométricos são substituídos por um produto matriz vetor. Esta construção também permite validar, no espaço conforme, a forma das equações já estabelecidas anteriormente.

O Algoritmo 11 estrutura os procedimentos para o modelo de construção de coordenadas conformes usando translações finais.

O custo computacional do Algoritmo 11 é dado por: 2 operações trigonométricas e 2 divisões para determinar os rotores; 16 produtos e 8 somas para compor as rotações 6 produtos e 4 somas para calcular a primeira coluna da matriz $B_{[i]}$; 4 produtos para construção do vetor b_i ; 3 produtos e 7 somas para calcular o produto $T_i b_i$. A construção do translador e de sua forma matricial não requer nenhuma operação. Totalizando 2 operações trigonométricas, 31 multiplicações divisões e 19 somas.

Ao apresentar a construção de coordenadas cartesianas com rotores, consideramos apenas o caso em que se utiliza translações finais. Porém, um olhar mais atento para o Algoritmo 1 oportuniza o uso de motores. Desta maneira, é interessante estabelecer a construção de coordenadas conformes por translações encaixadas.

Para este modelo de construção, a cada novo ponto a ser adicionado na cadeia, transladamos a origem pelo vetor $d_i e_1$, gerando um vetor com comprimento adequado que será rotacionado para gerar o vetor de ligação. Então, aplicamos a rotação correspondente R_i , que posiciona o vetor de ligação com ângulos adequados em relação aos eixos coor-

Algoritmo 11: Construção de coordenadas conformes com translações finais.

Entrada: $CI = \{(d_i, \theta_i, \varphi_i)\}$ com $i = 1, \dots, n$

Saída: $CO = \{X_1, X_2, \dots, X_n\}$.

início

$X_1 \leftarrow e_0;$

$R_{[1]} \leftarrow 1;$

para $i = 2, \dots, n$ **faça**

$R^{\theta_i} \leftarrow e^{\frac{\theta_i}{2} \iota e_3};$

$R^{\varphi_i} \leftarrow e^{\frac{\varphi_i}{2} \iota e_1};$

$R_i \leftarrow R^{\varphi_i} R^{\theta_i};$

$R_{[i]} \leftarrow R_{[i-1]} R_i;$

Determine a forma matricial $B_{[i]}$ do rotor $R_{[i]}$;

$b_i \leftarrow d_i B_{[i]} e_1;$

$T_{x_{i-1}} \leftarrow 1 - \frac{1}{2} x_{i-1} e_\infty;$

Calcule a forma matricial T_i de $T_{x_{i-1}}$;

$X_i \leftarrow T_i b_i.$

fim

fim

denados. Estas três operações podem ser combinadas em um motor M_i . Para ajustar a medida dos ângulos conforme as ligações anteriores, são aplicadas em ordem reversa a sequência de motores usadas, que posiciona cada um dos vetores de ligação componentes da cadeia. Assim, temos o Algoritmo 12.

Algoritmo 12: Construção de coordenadas conformes com translações encaixadas.

Entrada: $CI = \{(d_i, \theta_i, \varphi_i)\}$ com $i = 1, \dots, n$.

Saída: $CO = \{X_1, X_2, \dots, X_n\}$.

início

$X_1 \leftarrow e_0;$

$M_{[1]} \leftarrow 1;$

para $i = 2, \dots, n$ **faça**

$T_i \leftarrow 1 - \frac{1}{2} d_i e_{1\infty};$

$R^{\theta_i} \leftarrow e^{\frac{\theta_i}{2} \iota e_3};$

$R^{\varphi_i} \leftarrow e^{\frac{\varphi_i}{2} \iota e_1};$

$M_i \leftarrow R^{\varphi_i} R^{\theta_i} T_i;$

$M_{[i]} \leftarrow M_{[i-1]} M_i;$

Determine a forma matricial $B_{[i]}$ de $M_{[i]}$;

$X_i \leftarrow B_{[i]} e_0.$

fim

fim

O custo computacional do Algoritmo 12 é calculado da seguinte maneira: 1

divisão por 2 para construção do translador; 2 divisões por 2, para construção dos rotores; 31 produtos e 17 somas para composição de $M_{[i]}$; 16 produtos e 12 somas para o cálculo da primeira coluna da matriz $B_{[i]}$. Totalizando, assim, 50 multiplicações/divisões e 29 somas. Embora o custo para determinar as coordenadas conformes utilizando translações encaixadas seja consideravelmente mais alto, quando comparado ao custo utilizando translações finais, essa formulação traz equações mais simples, que serão extremamente úteis ao se manipular coordenadas para o cálculo de funções energia, pois, nesta formulação

$$X_i = M_{[i]}e_0M_{[i]}^{-1}$$

A Tabela 8 resume o custo de cada um dos modelos de construção de coordenadas cartesianas e conformes.

Tabela 8 – Custo para determinar um novo ponto.

Método	×	+	\mathcal{T}	Total
Matrizes	27	15	4	46
Rotores	28	15	4	47
Álgebra conforme TF	31	19	4	54
Álgebra conforme TE	50	29	4	83

Os algoritmos de conversão para coordenadas conformes foram implementados no *software Wolfram Mathematica v11.3*. Para os testes, foram utilizados os conjuntos de coordenadas internas BBN50, BBN200 e BBN1000. A Tabela 9 traz os tempos de execução do algoritmo que utiliza translações finais, mais barato computacionalmente.

Tabela 9 – Tempo para conversão de coordenadas internas para conformes

Conjunto	Tempo (s)
BBN50	0.015
BBN200	0.06
BBN1000	0.31

O tempo de execução foi ligeiramente maior que a conversão para coordenadas cartesianas usando matrizes e foi o mesmo na conversão para coordenadas cartesianas usando rotores. A Tabela 10 faz uma comparação entre a conversão utilizando cada um dos três métodos apresentados neste trabalho. Como podemos observar, não há alterações significativas entre os tempos de execução de cada algoritmo.

Tabela 10 – Tempo (s) para conversão de coordenadas.

Modelo	BBN50	BBN200	BBN1000
Matrizes	0.	0.02	0.03
Rotores	0.02	0.06	0.03
Álgebra conforme	0.02	0.06	0.03

5.5 Atualização de coordenadas conformes

Uma vez que um novo sistema de representação foi estabelecido, e considerando as questões já apresentadas em relação a mudanças conformacionais, é necessário determinar um meio para atualizar estas coordenadas, a partir de alterações em ângulos de torção. Para isto, baseado nos algoritmos de atualização já escritos, podemos determinar dois novos modelos de atualização, utilizando translações encaixadas e translações finais.

Como rotores, transladores e motores são operadores ortogonais no espaço conforme o processo de atualização por translações encaixadas torna-se mais simples e direto. O uso de rotores genéricos da álgebra conforme permitirá evitar a sequência translação-rotação-translação necessária para realizar rotações cujo eixo não contém a origem. Isto acontece porque rotores genéricos realizam rotações em relação a qualquer eixo e em qualquer posição, contendo ou não a origem.

A representação por coordenadas conformes é um mapeamento do conjunto de coordenadas cartesianas no espaço conforme, logo, sua atualização segue os mesmos princípios. No espaço conforme, cada uma dessas rotações é determinada por um rotor geral, conforme estabelecido na sessão anterior. Assim, o Algoritmo 13 resume este processo.

Algoritmo 13: Atualização de coordenadas conformes com translações encaixadas.

Entrada: $C_O = \{X_1, X_2, \dots, X_n\}, \mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$

Saída: $\overline{C}_O = \{\overline{X}_1, \overline{X}_2, \dots, \overline{X}_n\}$

início

$R_{[1]} = 1;$

para $i = 1, \dots, n$ **faça**

$b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{d_{k_i-1}};$

$R_i \leftarrow \cos \frac{\varphi_{k_i}}{2} - \text{sen} \frac{\varphi_{k_i}}{2} (b_{k_i-1} + (x_{k_i-2} \wedge b_{k_i-1}) e_\infty) e_{123};$

$R_{[i]} \leftarrow R_{[i-1]} R_i;$

 Determine a forma matricial $B_{[i]}$ de $R_{[i]}$;

para $k_i \leq j < k_{i+1}$ **faça**

$\overline{X}_j \leftarrow B_{[i]} X_j;$

fim

fim

fim

O custo computacional para o cálculo das rotações é dado por: 3 somas e 3 divisões para o cálculo do eixo de rotação; 12 produtos e 3 somas para determinar o rotor R_i ; 44 produtos e 36 somas para determinar $R_{[i]}$; 27 produtos e 33 somas para determinar a forma matricial. Cada ponto a ser atualizado tem ainda um custo de 16 produtos e 16 somas. Deste modo, a estimativa do custo total para atualização da cadeia é de $161n_r + 32n_a$ operações.

É interessante observar que, em oposição aos métodos de translações encaixadas já apresentados, no espaço conforme não há uma preocupação em calcular o vetor de translação separadamente. Neste caso, as translações são componentes indissociáveis do operador de rotação. Apenas um motor é necessário para realizar todo o conjunto de rotações combinadas, e apenas o produto geométrico é suficiente para codificar todos os movimentos requeridos para posicionar corretamente cada ponto da cadeia.

Um modelo para atualização com translações finais pode ser estabelecido no espaço conforme, embora a natureza desse modelo não apresente mudanças significativas em relação ao espaço tridimensional. O procedimento para este tipo de atualização consiste em aplicar rotações diretamente nos vetores de ligação. Estas rotações ampliam ou reduzem o ângulo de torção conforme a modificação necessária. Desta forma, como o próprio nome sugere, apenas uma translação é aplicada ao final do processo.

Como rotações são operações isométricas, a quinta coordenada conforme, associada ao vetor e_∞ , não é modificada (fato mostrado anteriormente). Isto significa que, ao aplicá-las no vetor de ligação, não importa se este está ou não representado no espaço conforme. Além disso, para determinar o translador e sua forma matricial, não há cálculo adicional, pois $\|b_i\| = d_i$. O Algoritmo 14, que atualiza coordenadas conformes com translações finais, permite identificar esta sutil diferença.

A diferença de custo entre este algoritmo e o Algoritmo 10 está somente no cálculo de pontos que devem ser rotacionados, pois o custo para atualizar um novo ângulo é exatamente o mesmo. Para atualizar a coordenada conforme de um ponto, o custo é dado por: 3 somas/subtrações para calcular b_j ; 9 multiplicações e 6 somas para rotacionar b_j ; 7 somas e 3 produtos para transladar o ponto X_{i-1} . Totalizando $58n_r + 28n_a$ operações.

A Tabela 11 mostra o custo computacional, a cada novo ângulo de torção modificado, em cada um dos métodos apresentados.

A Tabela 12 mostra o tempo de execução do Algoritmo 13 para calcular as atualizações dos exemplos trabalhados no texto.

Algoritmo 14: Atualização de coordenadas conformes com translações finais.

Entrada: $C_O = \{X_1, X_2, \dots, X_n\}, \mathcal{I} = \{\varphi_{k_1}, \varphi_{k_2}, \dots, \varphi_{k_p}\}$

Saída: $\overline{C_O} = \{\overline{X_1}, \overline{X_2}, \dots, \overline{X_n}\}$

início

$R_{[0]} \leftarrow 1;$

para $i = 1, \dots, p$ **faça**

$b_{k_i-1} \leftarrow \frac{x_{k_i-2} - x_{k_i-1}}{\|x_{k_i-2} - x_{k_i-1}\|};$

Calcule $R_i;$

$R_{[i]} \leftarrow R_{[i-1]}R_i;$

Determine a forma matricial $B_{[i]}$ de $R_{[i]}$;

para $k_i \leq j < k_{i+1}$ **faça**

$b_j \leftarrow x_j - x_{j-1};$

$b_j \leftarrow B_{[i]}b_j;$

$T_{b_j} = 1 - \frac{1}{2}b_j e_\infty;$

Determine a forma matricial T_j de $T_{b_j};$

$\overline{X_j} \leftarrow T_j \overline{X_{j-1}};$

fim

fim

fim

Tabela 11 – Custo computacional para atualização de coordenadas.

Método	\times	$+$	\mathcal{T}	Total
Matrizes - TE	54	43	2	99
Matrizes - TF	45	22	2	69
Rotores - TE	52	35	2	89
Rotores - TF	33	25	2	60
A. Conforme - TE	86	75	2	163
A. Conforme - TF	33	25	2	60

Fonte: Elaborada pelo autor.

Tabela 12 – Tempo de execução (em segundos) da atualização de coordenadas conformes com translações encaixadas.

Conjunto	10%	40%	85%
BBN50	0.02	0.02	0.03
BBN200	0.03	0.08	0.16
BBN1000	0.11	0.37	0.72

Como podemos verificar na tabela, o tempo de execução está relacionado mais diretamente a quantidade de ângulos modificados, e é menos sensível ao tamanho da cadeia numa primeira análise.

A Tabela 13 resume os tempos de execução de cada método, em cada teste

realizado.

Tabela 13 – Comparação dos algoritmos de atualização de coordenadas.

Método	BBN50			BBN200			BBN1000		
	10%	20%	85%	10%	20%	85%	10%	20%	85%
Matrizes	0.	0.02	0.02	0.	0.02	0.03	0.02	0.06	0.11
Rotores	0.	0.	0.02	0.	0.02	0.05	0.06	0.11	0.20
AC	0.02	0.02	0.03	0.03	0.08	0.16	0.11	0.37	0.72

5.6 Precisão e sensibilidade das coordenadas conformes

Embora a manipulação de coordenadas conformes pareça um tanto quanto complexa, o processo, tanto de construção quanto de atualização, é tão eficiente quanto os demais modelos em relação à precisão das coordenadas e no cálculo de distâncias interatômicas.

Para avaliar a precisão do conjunto de coordenadas conformes, foram realizados os seguintes testes. Inicialmente, geramos cadeias com tamanhos variando de 10 a 1000 átomos. Estas cadeias foram geradas usando o mesmo protocolo de construção das cadeias apresentadas nos exemplos anteriores. A partir deste conjunto de coordenadas internas, foram gerados: um conjunto de coordenadas cartesianas utilizando matrizes; um conjunto de coordenadas cartesianas utilizando rotores; um conjunto de coordenadas conformes. Os três métodos com translações finais por apresentar um menor custo computacional. Em seguida, foram geradas as matrizes de distâncias de cada conjunto, sendo usada a norma euclidiana para coordenadas cartesianas e o produto interno para coordenadas conformes.

A Tabela 14 indica a entrada com maior diferença em módulo entre as matrizes de distância, onde n indica o tamanho da cadeia, M o uso de matrizes de rotação tradicionais, R o uso de rotores, e A o uso da álgebra geométrica conforme.

Tabela 14 – Maior diferença, em módulo, entre matrizes de distância.

n	M vs. R	M vs. A	A vs. R
10	$6.661 \cdot 10^{-15}$	$8.437 \cdot 10^{-15}$	$1.110 \cdot 10^{-14}$
20	$1.598 \cdot 10^{-14}$	$5.573 \cdot 10^{-14}$	$5.817 \cdot 10^{-14}$
50	$1.065 \cdot 10^{-13}$	$3.586 \cdot 10^{-13}$	$3.626 \cdot 10^{-13}$
100	$6.057 \cdot 10^{-13}$	$1.840 \cdot 10^{-12}$	$1.791 \cdot 10^{-12}$
300	$2.941 \cdot 10^{-12}$	$2.679 \cdot 10^{-11}$	$2.712 \cdot 10^{-11}$
500	$1.340 \cdot 10^{-11}$	$1.221 \cdot 10^{-10}$	$1.213 \cdot 10^{-10}$
1000	$5.722 \cdot 10^{-11}$	$8.598 \cdot 10^{-10}$	$8.596 \cdot 10^{-10}$

Fonte: Elaborada pelo autor.

Para avaliar a sensibilidade com relação a mudanças conformacionais, foram alterados aleatoriamente todos os ângulos de torção dos conjuntos gerados. A alteração

foi realizada de maneira aleatória com valores entre 1 e 5 graus. Para avaliar a eficiência de cada modelo, foi calculada a maior diferença em módulo entre a matriz de distâncias original e a matriz de distâncias alterada. A alteração foi avaliada de duas maneiras: a Tabela 15 indica as alterações calculadas construindo uma nova cadeia, a partir da atualização das coordenadas internas, enquanto a Tabela 16 avalia os métodos de alteração descritos ao longo do texto, onde as coordenadas são atualizadas diretamente.

Tabela 15 – Diferença absoluta máxima entre matrizes de distâncias - Alteração por construção.

n	M	R	A
10	0.423	0.423	0.4231
20	0.913	0.913	0.913
50	1.731	1.731	1.731
100	2.946	2.946	2.946
300	7.829	7.828	7.828
500	12.281	12.281	12.281
1000	35.448	35.448	35.448

Fonte: Elaborada pelo autor.

Tabela 16 – Diferença absoluta máxima entre matrizes de distâncias - Alteração por atualização.

n	Matriz	Rotor	Conforme
10	0.423441	0.423441	0.423441
20	0.913077	0.913077	0.913077
50	1.73193	1.73194	1.73195
100	2.94633	2.94633	2.94633
300	7.82881	7.82881	7.82881
500	12.2813	12.2813	12.2813
1000	35.448	35.448	35.448

Fonte: Elaborada pelo autor.

Como pode-se observar nas Tabelas 14, 15 e 16, as coordenadas conformes descrevem a conformação da molécula tão eficientemente quanto as coordenadas cartesianas. Os algoritmos, tanto de construção quanto de atualização, em coordenadas conformes são tão precisos quanto os modelos para coordenadas cartesianas, sejam estes com as tradicionais matrizes de rotação ou com rotores.

6 Função de energia potencial

Ao longo deste texto foi mencionado por diversas vezes a importância de se conhecer as distâncias interatômicas de uma molécula. Muito disso se deve aos cálculos da mecânica molecular e, mais especificamente em dinâmica molecular. A dinâmica molecular é a simulação computacional dos movimentos realizados por uma molécula em um sistema. Esse procedimento tem uma grande diversidade de aplicações que variam do desenvolvimento de novas drogas (NAMBA; SILVA; SILVA, 2008) à determinação da estrutura molecular por meio de otimização (PHILLIPS; ROSEN; WALKE, 1995).

Conforme mencionado no início do texto, a estrutura terciária de uma molécula é dada pelo posicionamento tridimensional de seus átomos. Estas posições, por sua vez, são resultado das energias presentes nas ligações químicas componentes da molécula e de interações entre átomos que não possuem ligações covalentes. Esses termos de energia compõem um campo de forças empírico, que é estabelecido como a função de energia potencial.

Existem diversas maneira de se caracterizar a função de energia potencial de uma molécula. Como nosso objetivo é justificar o cálculo das distâncias interatômicas, vamos partir da mesma formulação utilizada por Phillips, Rosen e Walke (1995) que é algebricamente definida pela sequência de somas de termos de energia descritos pela equação

$$U = \sum_i U_d + \sum_i U_\theta + \sum_i U_\varphi + \sum_{i,j} U_r,$$

onde:

- $U_d = k_d(d_i - d_0)^2$ é a energia de estiramento da ligação, sendo d_i a i -ésima distância de ligação, d_0 o comprimento padrão para a ligação e k_d a constante de força para restituição do equilíbrio no comprimento da ligação;
- $U_\theta = k_\theta(\theta_i - \theta_0)^2$ é a energia de deformação do ângulo de ligação, com θ_i o ângulo de ligação, θ_0 o valor padrão do ângulo de ligação e k_θ a constante de força para restituição do equilíbrio do ângulo em questão;
- $U_\varphi = \frac{V_n}{2}(1 + \cos(n\varphi_i + \delta))$ é a energia de torção resultante da torção de uma ligação, sendo φ_i o ângulo de torção, n é o número de máximos da energia que a ligação atinge em uma torção completa, com δ o valor de defasagem no ângulo de torção que gera um ponto máximo de energia;

- $U_r = 4\varepsilon_{i,j} \left(\left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right)$ é a energia de atração e repulsão descrita pelo potencial de Lennard-Jones, com $\varepsilon_{i,j}$ e $\sigma_{i,j}$ constantes definidas pela interação entre os pares de átomos e $r_{i,j}$ a distância entre os átomos x_i e x_j .

Este conjunto de descrições, extraídos de [Namba, Silva e Silva \(2008\)](#) e [Phillips, Rosen e Walke \(1995\)](#), define adequadamente a função energia potencial U , que pode ser reescrita como:

$$\sum_i k_d (d_i - d_0)^2 + \sum_i k_\theta (\theta_i + \theta_0)^2 + \sum_i \frac{V_n}{2} (1 + \cos(n\varphi_i + \delta)) + \sum_{i,j} 4\varepsilon_{i,j} \left(\left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right). \quad (6.1)$$

Como pode ser observado, embora a função de energia esteja diretamente relacionada às coordenadas internas, o cálculo das distâncias interatômicas faz-se necessário para determinar o termo potencial de Lennard-Jones. Desta maneira, estabeleceremos agora formas que permitam o cálculo das distâncias partindo diretamente das coordenadas internas. Outras formulações para a função de energia potencial podem incluir ainda um termo relacionado a energia eletrostática, mas cabe ressaltar que sua ausência, neste caso, não afeta a teoria a ser desenvolvida.

6.1 O cálculo de distâncias interatômicas

Conforme [Phillips, Rosen e Walke \(1995\)](#), distâncias intermoleculares podem ser determinadas com base na equação de construção com translações encaixadas, isto é, se

$$x_i = B_1 B_2 \cdots B_i e_4 \quad \text{e} \quad x_j = B_1 B_2 \cdots B_j e_4$$

com B_k definida por (1.1), então o quadrado da distância entre x_i e x_j é resultado do produto interno

$$r_{i,j}^2 = (x_j - x_i)^t (x_j - x_i).$$

Para todos os efeitos, e sem perda de generalidade, vamos considerar aqui que $j > i$. Assim,

$$\begin{aligned} (x_j - x_i) &= (B_1 B_2 \cdots B_j - B_1 B_2 \cdots B_i) e_4 \\ &= \left(\prod_{l=1}^i B_l \right) \left(\prod_{l=i+1}^j B_l - I \right) e_4, \end{aligned} \quad (6.2)$$

onde I é a matriz identidade de ordem 4.

Observando a Equação (6.2), podemos notar que todas as matrizes de construção B_l , com $l = 1 \cdots j$, foram calculadas. Deste modo, ainda que as coordenadas cartesianas não tenham sido explicitamente calculadas, a determinação das distâncias realiza, basicamente, o mesmo processo que a construção das coordenadas.

Conforme as equações estabelecidas ao longo da Seção 2, as coordenadas internas $(d_k, \theta_k, \varphi_k)$, com $k = 1, \dots, i$, não afetam $r_{i,j}$. De fato, é bastante intuitivo que a distância $r_{i,j}$ não sofra alterações resultantes da variação nas coordenadas internas anteriores a x_i . Contudo, não é possível validar este pensamento utilizando a expressão (6.2). Uma alternativa eficaz que encontramos para evitar o cálculo de parte das matrizes B_l , é utilizar a equação de translações finais (Equação (1.5)) para o cálculo das distâncias. De tal maneira, propomos que dado

$$x_i = d_1 B_{[1]} e_1 + d_2 B_{[2]} e_1 + d_3 B_{[3]} e_1 + \dots + d_i B_{[i]} e_1$$

e

$$x_j = d_1 B_{[1]} e_1 + d_2 B_{[2]} e_1 + d_3 B_{[3]} e_1 + \dots + d_j B_{[j]} e_1,$$

onde $B_{[k]} = B_1 B_2 B_3 \dots B_k$. Com isso, B_i refere-se às matrizes de rotação de ordem 3 e o vetor e_4 é substituído pelo vetor e_1 , pois não há uso do espaço homogêneo. Então,

$$\begin{aligned} r_{i,j} &= \|x_j - x_i\| \\ &= \|(d_1 I + d_2 B_{[2]} + \dots + d_j B_{[j]}) e_1 - (d_1 I + d_2 B_{[2]} + \dots + d_i B_{[i]}) e_1\| \\ &= \|(d_{i+1} B_{[i+1]} + \dots + d_j B_{[j]}) e_1\| \\ &= \left\| \left(\prod_{k=1}^{i+1} B_k \right) \left(d_{i+1} I + \sum_{s=i+2}^j d_s \prod_{m=i+2}^s B_m \right) e_1 \right\| \\ &= \left\| \left(d_{i+1} I + \sum_{s=i+2}^j d_s \prod_{m=i+2}^s B_m \right) e_1 \right\|. \end{aligned}$$

A passagem da penúltima para a última igualdade vem do fato de que rotações são isometrias e, portanto, não alteram a norma. Assim, temos que $r_{i,j}$ não depende dos valores de d_k, θ_k e φ_k , para $k = 1, \dots, i$.

Com este novo modelo, é evitado o cálculo das coordenadas cartesianas de x_1 a x_i , uma vez que as matrizes B_1, \dots, B_i sequer precisam ser determinadas. Porém, ainda é necessário determinar parte da estrutura da cadeia, do ponto x_i ao ponto x_j , sendo x_i transferido para a origem do espaço e o vetor de ligação b_{i+1} posicionado sobre o eixo orientado por e_1 .

O uso de translações encaixadas torna as equações para o cálculo das distâncias consideravelmente mais simples, mas não permite um cálculo direto. Por outro lado, o uso de translações finais permite a redução do número de pontos determinados no processo, embora uma equação mais complexa seja necessária. Seria desejável combinar a simplicidade das equações que utilizam translações encaixadas com a redução no cálculo de coordenadas presente no uso de translações finais, resultando em um método capaz de calcular distâncias mais diretamente e com equações simples. Felizmente, a solução para esta questão está no uso de coordenadas conformes.

A ortogonalidade dos operadores, utilizados na construção das coordenadas conformes, permite simplificar as equações excluindo termos que envolvam coordenadas internas anteriores ao ponto x_i , assim como foi feito para translações finais. Para isso, vamos tomar

$$X_i = M_{[i]}e_0M_{[i]}^{-1} \quad \text{e} \quad X_j = M_{[j]}e_0M_{[j]}^{-1},$$

onde $M_{[k]} = M_1M_2M_3 \cdots M_k$.

Sabemos que o produto interno conforme codifica distâncias entre pares de pontos, ou seja,

$$-\frac{1}{2}r_{i,j}^2 = X_i \cdot X_j = \langle X_i X_j \rangle_0. \quad (6.3)$$

Então, para o cálculo de $r_{i,j}$, basta investigar o produto geométrico $X_i X_j$. Vamos denotar $M_{[s,t]} = M_s M_{s+1} \cdots M_t$, dessa maneira

$$\begin{aligned} X_i X_j &= M_{[i]}e_0M_{[i]}^{-1}M_{[j]}e_0M_{[j]}^{-1} \\ &= M_{[i]}e_0M_{[i]}^{-1}M_{[i]}M_{[i,j]}e_0(M_{[i+1,j]})^{-1}M_{[i]}^{-1} \\ &= M_{[i]}e_0M_{[i+1,j]}e_0(M_{[i+1,j]})^{-1}M_{[i]}^{-1} \end{aligned}$$

Considerando o multivetor $V = e_0M_{[i+1,j]}e_0(M_{[i+1,j]})^{-1}$, sabemos que V pode ser escrito como a soma de seus k -vetores ($k = 1, \dots, 5$):

$$V = \langle V \rangle_0 + \langle V \rangle_1 + \langle V \rangle_2 + \langle V \rangle_3 + \langle V \rangle_4 + \langle V \rangle_5.$$

Da distributividade do produto geométrico, temos

$$\begin{aligned} M_{[i]}VM_{[i]}^{-1} &= M_{[i]}\langle V \rangle_0M_{[i]}^{-1} + M_{[i]}\langle V \rangle_1M_{[i]}^{-1} + M_{[i]}\langle V \rangle_2M_{[i]}^{-1} + M_{[i]}\langle V \rangle_3M_{[i]}^{-1} + \\ &+ M_{[i]}\langle V \rangle_4M_{[i]}^{-1} + M_{[i]}\langle V \rangle_5M_{[i]}^{-1}. \end{aligned}$$

Voltando à Equação (6.3), apenas a parte escalar do produto $M_{[i]}VM_{[i]}^{-1}$ deve ser considerada. Como motores são compostos por rotores e transladores, que por sua vez não alteram a natureza do elemento k -vetorial, segue que

$$\langle M_{[i]}VM_{[i]}^{-1} \rangle_0 = M_{[i]}\langle V \rangle_0M_{[i]}^{-1}.$$

Por outro lado, $\langle V \rangle_0$ é escalar, então

$$M_{[i]}\langle V \rangle_0M_{[i]}^{-1} = \langle V \rangle_0M_{[i]}M_{[i]}^{-1} = \langle V \rangle_0.$$

Portanto,

$$-\frac{1}{2}r_{i,j}^2 = \langle e_0M_{[i+1,j]}e_0(M_{[i+1,j]})^{-1} \rangle_0,$$

ou ainda

$$r_{i,j}^2 = -2\langle e_0M_{[i+1,j]}e_0(M_{[i+1,j]})^{-1} \rangle_0. \quad (6.4)$$

Como $M_{[i+1,j]}$ é um motor, pode ser escrito conforme a Equação (5.4). Assim, se

$$M_{[i+1,j]} = a_0 + a_1 e_{23} + a_2 e_{31} + a_3 e_{12} + a_4 e_{1\infty} + a_5 e_{2\infty} + a_6 e_{3\infty} + a_7 e_{123\infty},$$

o escalar $\langle e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} \rangle_0$ é dado por $-(a_4^2 + a_5^2 + a_6^2 + a_7^2)$ e, conseqüentemente,

$$r_{i,j}^2 = 2(a_4^2 + a_5^2 + a_6^2 + a_7^2). \quad (6.5)$$

O modelo conforme permite, portanto, calcular as distâncias entre dois pontos utilizando somente as coordenadas internas entre o par pontos. Além disso, vale ressaltar, não há necessidade de construção das coordenadas conformes (ou cartesianas) em nenhuma etapa. Elimina-se, inclusive, a necessidade de se calcular a forma matricial do motor. Estes resultados somente são possíveis devido à ortogonalidade dos motores, que permite o cancelamento de termos que envolvam M_1, M_2, \dots, M_i .

Como o cálculo de distâncias foi estabelecido diretamente a partir das coordenadas internas da molécula, é possível calcular a função energia potencial sem a necessidade de calcular outro conjunto de coordenadas, sejam cartesianas ou conformes.

6.2 Derivadas de U

Existem diversas técnicas para se determinar a estrutura tridimensional de uma molécula de proteína, entre elas, a determinação de valores mínimos globais da função de energia potencial. A minimização de funções pode incluir o cálculo de suas derivadas de primeira e segunda ordem, por esta razão, vamos explorar como as funções derivadas da função de energia podem ser geradas a partir das coordenadas internas.

Considerando a função U , descrita por (6.1), as derivadas de primeira ordem de U , com relação a cada coordenada interna, são dadas por

$$\frac{\partial U}{\partial d_k} = 2k_d(d_k - d_0) + \sum_{i,j} \frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial d_k}, \quad (6.6)$$

$$\frac{\partial U}{\partial \theta_k} = 2k_\theta(\theta_k + \theta_0) + \sum_{i,j} \frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \theta_k}, \quad (6.7)$$

$$\frac{\partial U}{\partial \varphi_k} = \frac{nV_n}{2} \text{sen}(n\varphi_k + \delta) + \sum_{i,j} \frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \varphi_k}. \quad (6.8)$$

Observando as três derivadas parciais, podemos notar que a primeira parcela da soma é obtida diretamente da variável de derivação. A segunda parcela, derivada do potencial de Lennard-Jones, é comum às três derivadas parciais, exceto pelo termo derivado de $r_{i,j}$. Já estabelecemos como $r_{i,j}$ pode ser calculado a partir das coordenadas

internas, agora, utilizaremos uma técnica similar para determinar suas derivadas parciais de primeira e segunda ordem.

Iniciaremos descrevendo o método desenvolvido por [Phillips, Rosen e Walke \(1995\)](#), que utilizam as equações de translações encaixadas para determinar as derivadas parciais de primeira ordem.

Sabemos que

$$r_{i,j}^2 = (B_{i,j}e_4)^t B_{i,j}e_4,$$

onde

$$B_{i,j} = \left(\prod_{k=1}^i B_k \right) \left(\prod_{k=i+1}^j B_k - I \right).$$

Além disso,

$$\frac{\partial r_{i,j}}{\partial \alpha_k} = \frac{1}{2r_{i,j}} \frac{\partial r_{i,j}^2}{\partial \alpha_k} = \frac{e_4^t B_{i,j}^t}{r_{i,j}} \frac{\partial B_{i,j}}{\partial \alpha_k} e_4, \quad (6.9)$$

em que α_k representa uma das coordenadas: d_k , θ_k ou φ_k . Desta maneira, resta definir $\frac{\partial B_{i,j}}{\partial \alpha_k}$.

Uma vez que a matriz B_l , dada por (1.4), depende exclusivamente de $(d_l, \theta_l, \varphi_l)$, temos

$$\frac{\partial B_{i,j}}{\partial \alpha_k} = \begin{cases} \bar{0}, & \text{se } j < k; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^j B_l \right), & \text{se } i < k \leq j; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^i B_l \right) \left(\prod_{l=i+1}^j B_l - I \right), & \text{se } 1 \leq k \leq i; \end{cases} \quad (6.10)$$

onde $\bar{0}$ indica a matriz nula. Substituindo α_k por cada um dos parâmetros, podemos determinar as seguintes derivadas parciais:

- para $\alpha_k = d_k$, temos

$$\frac{\partial B_k}{\partial d_k} = \begin{bmatrix} 0 & 0 & 0 & -\cos \theta_k \\ 0 & 0 & 0 & \sin \theta_k \cos \varphi_k \\ 0 & 0 & 0 & \sin \theta_k \sin \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

- para $\alpha_k = \theta_k$,

$$\frac{\partial B_k}{\partial \theta_k} = \begin{bmatrix} \sin \theta_k & -\cos \theta_k & 0 & d_k \sin \theta_k \\ \cos \theta_k \cos \varphi_k & \sin \theta_k \cos \varphi_k & 0 & d_k \cos \theta_k \cos \varphi_k \\ \cos \theta_k \sin \varphi_k & \sin \theta_k \sin \varphi_k & 0 & d_k \cos \theta_k \sin \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

- para $\alpha_k = \varphi_k$,

$$\frac{\partial B_k}{\partial \varphi_k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\text{sen } \theta_k \text{ sen } \varphi_k & \cos \theta_k \text{ sen } \varphi_k & -\cos \varphi_k & -d_k \text{ sen } \theta_k \text{ sen } \varphi_k \\ \text{sen } \theta_k \cos \varphi_k & -\cos \theta_k \cos \varphi_k & -\text{sen } \varphi_k & d_k \text{ sen } \theta_k \cos \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Dessa maneira, as derivadas parciais de primeira ordem de U estão completamente definidas.

Partindo dessas ideias, optamos por dar prosseguimento, estabelecendo também as derivadas de segunda ordem. Para isso, vamos nos concentrar no termo derivado do potencial de Lennard-Jones, que é comum às funções (6.6), (6.70 e (6.8). Faremos este cálculo de maneira genérica, uma vez que este depende somente da função de distância $r_{i,j}$.

Como anteriormente, tomando α_k como representante de uma das variáveis d_k, θ_k ou φ_k , temos,

$$\frac{\partial U_r}{\partial \alpha_k} = \sum_{i,j} \frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \alpha_k}.$$

Para calcular a segunda derivada, vamos utilizar β_p como representante de algum d_p, θ_p ou φ_p . Desta forma,

$$\begin{aligned} \frac{\partial^2 U_r}{\partial \beta_p \partial \alpha_k} &= \frac{\partial}{\partial \beta_p} \left[\sum_{i,j} \frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \alpha_k} \right] \\ &= \sum_{i,j} \frac{\partial}{\partial \beta_p} \left[\frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \alpha_k} \right] \\ &= \sum_{i,j} \left(\frac{\partial}{\partial \beta_p} \left[\frac{-4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \right] \frac{\partial r_{i,j}}{\partial \alpha_k} + \right. \\ &\quad \left. - \left(\frac{4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \right) \frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} \right) \\ &= \sum_{i,j} \left(\frac{24\varepsilon_{i,j}}{r_{i,j}^2} \left(26 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 7 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \beta_p} \frac{\partial r_{i,j}}{\partial \alpha_k} + \right. \\ &\quad \left. - \left(\frac{4\varepsilon_{i,j}}{r_{i,j}} \left(12 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 6 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \right) \frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} \right) \end{aligned}$$

Portanto, a derivada de segunda ordem $\frac{\partial^2 U_r}{\partial \beta_p \partial \alpha_k}$ é

$$\sum_{i,j} \frac{24\varepsilon_{i,j}}{r_{i,j}} \left[\left(26 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - 7 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial r_{i,j}}{\partial \beta_p} \frac{\partial r_{i,j}}{\partial \alpha_k} - \frac{1}{r_{i,j}} \left(2 \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^{12} - \left(\frac{\sigma_{i,j}}{r_{i,j}} \right)^6 \right) \frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} \right].$$

As funções $r_{i,j}$, $\frac{\partial r_{i,j}}{\partial \alpha_k}$ e $\frac{\partial r_{i,j}}{\partial \beta_p}$ já foram calculadas, então, para que $\frac{\partial^2 U_r}{\partial \beta_p \partial \alpha_k}$ esteja bem definida, precisamos calcular $\frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k}$.

Partindo de (6.13), temos,

$$\begin{aligned} \frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} &= \frac{\partial}{\partial \beta_p} \left(\frac{e_4^t B_{i,j}^t}{r_{i,j}} \frac{\partial B_{i,j}}{\partial \alpha_k} e_b \right) \\ &= -\frac{1}{r_{i,j}^2} \frac{\partial r_{i,j}}{\partial \beta_p} \left(e_4^t B_{i,j}^t \frac{\partial B_{i,j}}{\partial \alpha_k} e_4 \right) + \frac{1}{r_{i,j}} \left(e_4^t \frac{\partial B_{i,j}^t}{\partial \beta_p} \frac{\partial B_{i,j}}{\partial \alpha_k} e_4 + e_4^t B_{i,j}^t \frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k} e_4 \right). \end{aligned}$$

Logo, a derivada segunda de $r_{i,j}$ é dada por

$$\frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} = \frac{1}{r_{i,j}} \left[-\frac{\partial r_{i,j}}{\partial \beta_p} \frac{\partial r_{i,j}}{\partial \alpha_k} + e_4^t \left(\frac{\partial B_{i,j}^t}{\partial \beta_p} \frac{\partial B_{i,j}}{\partial \alpha_k} + B_{i,j}^t \frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k} \right) e_4 \right] \quad (6.11)$$

Novamente, chegamos a uma expressão cujos termos $r_{i,j}$, $\frac{\partial r_{i,j}}{\partial \beta_p}$, $\frac{\partial r_{i,j}}{\partial \alpha_k}$, $\frac{\partial B_{i,j}}{\partial \beta_p}$ e $B_{i,j}$ já foram calculados, restando, portanto, calcular $\frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k}$, que é dada a cada caso:

$$\left\{ \begin{array}{ll} \bar{0}, & \text{se } j < \max\{p, k\}; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^{p-1} B_l \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{l=p+1}^j B_l \right), & \text{se, } \max\{k, i\} < p \leq j; \\ \left(\prod_{l=1}^{p-1} B_l \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{l=p+1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^j B_l \right), & \text{se, } \max\{p, i\} < k \leq j; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial^2 B_k}{\partial \beta_k \partial \alpha_k} \left(\prod_{l=k+1}^j B_l \right), & \text{se, } i < k = p \leq j; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^{p-1} B_l \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{l=p+1}^i B_l \right) \left(\prod_{l=i+1}^j B_l - I \right), & \text{se, } 1 \leq k < p \leq i; \\ \left(\prod_{l=1}^{p-1} B_l \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{l=p+1}^{k-1} B_l \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{l=k+1}^i B_l \right) \left(\prod_{l=i+1}^j B_l - I \right), & \text{se, } 1 \leq p < k \leq i; \\ \left(\prod_{l=1}^{k-1} B_l \right) \frac{\partial^2 B_k}{\partial \beta_k \partial \alpha_k} \left(\prod_{l=k+1}^i B_l \right) \left(\prod_{l=i+1}^j B_l - I \right), & \text{se, } 1 \leq p = k \leq i; \end{array} \right.$$

Observando as partes que compõem a função $\frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k}$, podemos notar que falta ainda calcular as matrizes $\frac{\partial^2 B_k}{\partial \beta_k \partial \alpha_k}$, que são dadas da seguinte maneira:

- para $\alpha_k = \beta_k = d_k$, $\frac{\partial^2 B_k}{\partial d_k^2} = \bar{0}$;

- para $\alpha_k = d_k$ e $\beta_k = \theta_k$,

$$\frac{\partial^2 B_k}{\partial \theta_k \partial d_k} = \begin{bmatrix} 0 & 0 & 0 & \text{sen } \theta_k \\ 0 & 0 & 0 & \cos \theta_k \cos \varphi_k \\ 0 & 0 & 0 & \cos \theta_k \text{sen } \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix} = \frac{\partial^2 B_k}{\partial d_k \partial \theta_k};$$

- para $\alpha_k = d_k$ e $\beta_k = \varphi_k$,

$$\frac{\partial^2 B_k}{\partial \varphi_k \partial d_k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\text{sen } \theta_k \text{sen } \varphi_k \\ 0 & 0 & 0 & \text{sen } \theta_k \cos \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix} = \frac{\partial^2 B_k}{\partial d_k \partial \varphi_k};$$

- para $\alpha_k = \beta_k = \theta_k$,

$$\frac{\partial^2 B_k}{\partial \theta_k^2} = \begin{bmatrix} \cos \theta_k & \text{sen } \theta_k & 0 & d_k \cos \theta_k \\ -\text{sen } \theta_k \cos \varphi_k & \cos \theta_k \cos \varphi_k & 0 & -d_k \text{sen } \theta_k \cos \varphi_k \\ -\text{sen } \theta_k \text{sen } \varphi_k & \cos \theta_k \text{sen } \varphi_k & 0 & -d_k \text{sen } \theta_k \text{sen } \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix};$$

- para $\alpha_k = \theta_k$ e $\beta_k = \varphi_k$,

$$\frac{\partial^2 B_k}{\partial \varphi_k \partial \theta_k} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\cos \theta_k \text{sen } \varphi_k & -\text{sen } \theta_k \text{sen } \varphi_k & 0 & -d_k \cos \theta_k \text{sen } \varphi_k \\ \cos \theta_k \cos \varphi_k & \text{sen } \theta_k \cos \varphi_k & 0 & d_k \cos \theta_k \cos \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix} = \frac{\partial^2 B_k}{\partial \varphi_k \partial \theta_k};$$

- para $\alpha_k = \beta_k = \varphi_k$,

$$\frac{\partial^2 B_k}{\partial \varphi_k^2} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -\text{sen } \theta_k \cos \varphi_k & \cos \theta_k \cos \varphi_k & \text{sen } \varphi_k & -d_k \text{sen } \theta_k \cos \varphi_k \\ -\text{sen } \theta_k \text{sen } \varphi_k & \cos \theta_k \text{sen } \varphi_k & -\cos \varphi_k & -d_k \text{sen } \theta_k \text{sen } \varphi_k \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Embora tenha sido demonstrado na Seção 2 que a variação na k -ésima coordenada interna não interfere em distâncias $r_{i,j}$, com $k < i$, isto não é verificado na abordagem utilizada por Phillips, Rosen e Walke (1995), pois a derivada parcial nesses casos é diferente de 0. Este fato decorre da impossibilidade de eliminar as coordenadas anteriores a i no cálculo das distâncias $r_{i,j}$, situação contornada com uso de translações finais.

A construção de coordenadas por translações finais é bastante similar ao uso de translações encaixadas. Podemos escrever

$$r_{i,j}^2 = (B_{i,j}e_1)^t B_{i,j}e_1,$$

onde,

$$B_{i,j} = \left(d_{i+1}I + \sum_{s=i+2}^j d_s \prod_{m=i+1}^s B_m \right). \quad (6.12)$$

Logo,

$$\frac{\partial r_{i,j}}{\partial \alpha_k} = \frac{1}{2r_{i,j}} \frac{\partial r_{i,j}^2}{\partial \alpha_k} = \frac{e_1^t B_{i,j}^t}{r_{i,j}} \frac{\partial B_{i,j}}{\partial \alpha_k} e_1. \quad (6.13)$$

Assim, devemos calcular as derivadas parciais de $B_{i,j}$.

Nesta formulação, as matrizes B_m , que compõe $B_{i,j}$, dependem somente de φ_m e θ_m . Com isso, o cálculo da derivada com relação a d_k precisa ser feito separadamente. Temos então

$$\frac{\partial B_{i,j}}{\partial d_k} = \begin{cases} \bar{0}, & \text{se } k \leq i \text{ ou } j < k \\ I, & \text{se } k = i + 1 \\ \prod_{m=i+1}^k B_m, & \text{se } i + 2 \leq k \leq j \end{cases}.$$

Para as variáveis θ_k e φ_k , podemos calcular genericamente $\frac{\partial B_{i,j}}{\partial \alpha_k}$ e então determinar a derivada de cada matriz de rotação. Assim,

$$\frac{\partial B_{i,j}}{\partial \alpha_k} = \begin{cases} \bar{0}, & \text{se } k \leq i + 1 \text{ ou } j < k; \\ \sum_{s=k}^j d_s \left(\prod_{m=i+2}^{k-1} B_m \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{m=k+1}^s B_m \right), & \text{se } i + 1 < k \leq j; \end{cases}.$$

As derivadas das matrizes B_k , definidas em (1.1), são dadas por:

$$\frac{\partial B_k}{\partial \theta_k} = \begin{bmatrix} \text{sen } \theta_k & -\text{cos } \theta_k & 0 \\ \text{cos } \theta_k \text{ cos } \varphi_k & \text{sen } \theta_k \text{ cos } \varphi_k & 0 \\ \text{cos } \theta_k \text{ sen } \varphi_k & \text{sen } \theta_k \text{ sen } \varphi_k & 0 \end{bmatrix},$$

e

$$\frac{\partial B_k}{\partial \varphi_k} = \begin{bmatrix} 0 & 0 & 0 \\ -\text{sen } \theta_k \text{ sen } \varphi_k & \text{cos } \theta_k \text{ sen } \varphi_k & -\text{cos } \varphi_k \\ \text{sen } \theta_k \text{ cos } \varphi_k & -\text{cos } \theta_k \text{ cos } \varphi_k & -\text{sen } \varphi_k \end{bmatrix}.$$

O caminho para construção das derivadas de segunda ordem, usando translações finais, é muito similar ao já traçado anteriormente, com translações encaixadas. A derivada $\frac{\partial^2 r_{i,j}}{\partial \beta_p \alpha_k}$ tem mesma forma da expressão (6.11), onde e_4 é substituído por e_1 e $B_{i,j}$ é dada por (6.12). Falta então calcular a derivada segunda desta matriz. Note que as funções $\frac{\partial B_{i,j}}{\partial d_k}$, $\frac{\partial B_{i,j}}{\partial \theta_k}$ e $\frac{\partial B_{i,j}}{\partial \varphi_k}$ já foram calculadas, de modo que, partiremos destas para calcular as derivadas de segunda ordem.

Sabemos que derivações envolvendo a variável d_k devem ser calculadas a parte, de modo que as estabeleceremos primeiro. Considerando que β_p representa uma das coordenadas θ_p ou φ_p , temos,

$$\frac{\partial^2 B_{i,j}}{\partial \beta_p \partial d_k} = \frac{\partial^2 B_{i,j}}{\partial d_k \partial \beta_p} = \begin{cases} \left(\prod_{m=i+1}^{p-1} B_m \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{m=p+1}^k B_m \right), & \text{se } i+2 \leq p \leq k \leq j; \\ \bar{0}, & \text{c.c.} \end{cases},$$

além disso, $\frac{\partial^2 B_{i,j}}{\partial d_p \partial d_k} = \bar{0}$, $\forall p, k \in \{1, \dots, n\}$.

Para os demais casos, temos,

$$\frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k} = \begin{cases} \sum_{s=p}^j d_s \left(\prod_{m=i+2}^{k-1} B_m \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{m=k+1}^{p-1} B_m \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{m=p+1}^s B_m \right), & \text{se } i+1 < k < p \leq k; \\ \sum_{s=k}^j d_s \left(\prod_{m=i+2}^{p-1} B_m \right) \frac{\partial B_p}{\partial \beta_p} \left(\prod_{m=p+1}^{k-1} B_m \right) \frac{\partial B_k}{\partial \alpha_k} \left(\prod_{m=k+1}^s B_m \right), & \text{se } i+1 < p < k \leq k; \\ \sum_{s=k}^j d_s \left(\prod_{m=i+2}^{k-1} B_m \right) \frac{\partial^2 B_k}{\partial \beta_k \partial \alpha_k} \left(\prod_{m=k+1}^s B_m \right), & \text{se } i+1 < k = p \leq k; \\ \bar{0}, & \text{c.c.} \end{cases}.$$

Por fim, vamos calcular $\frac{\partial^2 B_k}{\partial \beta_p \partial \alpha_k}$:

- para $\beta_k = \alpha_k = \theta_k$, temos,

$$\frac{\partial^2 B_k}{\partial \theta_k^2} = \begin{bmatrix} \cos \theta_k & \sin \theta_k & 0 \\ -\sin \theta_k \cos \varphi_k & \cos \theta_k \cos \varphi_k & 0 \\ -\sin \theta_k \sin \varphi_k & \cos \theta_k \sin \varphi_k & 0 \end{bmatrix};$$

- para $\beta_k = \theta_k$ e $\alpha_k = \varphi_k$, temos,

$$\frac{\partial^2 B_k}{\partial \theta_k^2} = \begin{bmatrix} 0 & 0 & 0 \\ -\cos \theta_k \sin \varphi_k & -\sin \theta_k \sin \varphi_k & 0 \\ \cos \theta_k \cos \varphi_k & \sin \theta_k \cos \varphi_k & 0 \end{bmatrix} = \frac{\partial^2 B_k}{\partial \varphi_k \partial \theta_k};$$

- para $\beta_k = \alpha_k = \varphi_k$, temos,

$$\frac{\partial^2 B_k}{\partial \varphi_k^2} = \begin{bmatrix} 0 & 0 & 0 \\ -\sin \theta_k \cos \varphi_k & \cos \theta_k \cos \varphi_k & \sin \varphi_k \\ -\sin \theta_k \sin \varphi_k & \cos \theta_k \sin \varphi_k & -\cos \varphi_k \end{bmatrix};$$

A redução no número de casos que compõe as derivadas segundas de $B_{i,j}$ ficam ainda mais evidenciadas. Além disso, o caso em que $i + 1 < k < p \leq k$, na definição de $\frac{\partial^2 B_{i,j}}{\partial \beta_p \partial \alpha_k}$ permite uma nova redução no número de matrizes B_m que devem ser calculadas. No entanto, consoante ao caso das derivadas de primeira ordem, as derivações com relação a d_k também precisam ser determinadas separadamente. Esta situação também pode ser contornada com uso de coordenadas conformes.

6.3 Derivadas de U no espaço conforme

Vamos, agora, determinar as derivadas de primeira e segunda ordem utilizando as coordenadas conformes.

Partindo da igualdade

$$\frac{\partial r_{i,j}}{\partial \alpha_k} = \frac{1}{2r_{i,j}} \frac{\partial r_{i,j}^2}{\partial \alpha_k}, \quad (6.14)$$

devemos calcular

$$\begin{aligned} \frac{\partial r_{i,j}^2}{\partial \alpha_k} &= \frac{\partial}{\partial \alpha_k} \left[-2 \langle e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} \rangle_0 \right] \\ &= -2 \left\langle \frac{\partial}{\partial \alpha_k} \left[e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} \right] \right\rangle_0. \end{aligned}$$

A regra do produto para derivadas pode ser aplicada ao produto geométrico, respeitando sua não comutatividade (MACDONALD, 2012), assim,

$$\frac{\partial}{\partial \alpha_k} (e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1}) = e_0 \frac{\partial M_{[i+1,j]}}{\partial \alpha_k} e_0 (M_{[i+1,j]})^{-1} + e_0 M_{[i+1,j]} e_0 \frac{\partial (M_{[i+1,j]})^{-1}}{\partial \alpha_k}.$$

A derivada do inverso de um motor M pode ser escrita em razão da derivada do próprio motor. Esta relação, dada pela igualdade

$$\frac{\partial M^{-1}}{\partial \alpha} = -M^{-1} \frac{\partial M}{\partial \alpha} M^{-1},$$

pode ser facilmente verificada derivando ambos os lados da expressão $MM^{-1} = 1$.

Com isso, a derivada de $r_{i,j}^2$ pode ser expressa somente em termos da derivada de $M_{[i+1,j]}$, isto é,

$$-\frac{1}{2} \frac{\partial r_{i,j}^2}{\partial \alpha_k} = e_0 \frac{\partial M_{[i+1,j]}}{\partial \alpha_k} e_0 (M_{[i+1,j]})^{-1} - e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} \frac{\partial M_{[i+1,j]}}{\partial \alpha_k} (M_{[i+1,j]})^{-1}. \quad (6.15)$$

Para estabelecer as derivadas parciais do motor $M_{[i+1,j]}$, é necessário analisar a natureza deste elemento. Conforme notação estabelecida, o motor $M_{[i+1,j]}$ é o resultado do produto entre os motores de índice $i + 1$ até j , que, por sua vez, são compostos pela

sequência de uma translação seguida por duas rotações, isto é, $M_i = R_{\varphi_i} R_{\theta_i} T_{d_k}$. Deste modo, para determinar suas derivadas, é necessário avaliar, antes de mais nada, as derivadas de R_{φ_k} , R_{θ_k} e T_{d_k} . Para isso, vamos utilizar suas formas exponenciais.

Sendo $T_{d_k} = e^{-\frac{1}{2}d_k e_{1\infty}}$, temos

$$\begin{aligned} \frac{\partial T_{d_k}}{\partial d_k} &= \frac{\partial}{\partial d_k} \left(e^{-\frac{1}{2}d_k e_{1\infty}} \right) \\ &= \left(e^{-\frac{1}{2}d_k e_{1\infty}} \right) \left(-\frac{1}{2} e_{1\infty} \right) \\ &= \frac{-1}{2} T_{d_k} e_{1\infty}. \end{aligned}$$

Analogamente, para $R_{\theta_k} = e^{-\frac{1}{2}\theta_k e_{12}}$ e $R_{\varphi_k} = e^{-\frac{1}{2}\varphi_k e_{23}}$, temos,

$$\frac{\partial R_{\theta_k}}{\partial \theta_k} = -\frac{1}{2} e_{12} R_{\theta_k},$$

e

$$\frac{\partial R_{\varphi_k}}{\partial \varphi_k} = -\frac{1}{2} e_{23} R_{\varphi_k}.$$

Podemos notar que a derivação de cada um dos operadores é dada pelo produto entre o próprio operador e um bivector. A construção de T_{d_k} , R_{θ_k} e R_{φ_k} permite comutar o produto entre o operador e o bivector. Para facilitar o cálculo das derivadas, optamos por manter o bivector à esquerda, no caso do translador, e à direita para os rotores. Agora, uma vez estabelecidas as derivadas dos componentes de M_k , vamos calcular suas derivadas parciais.

Como $M_k = R_{\varphi_k} R_{\theta_k} T_{d_k}$, as derivadas com relação à φ_k e d_k são obtidas imediatamente, veja:

$$\begin{aligned} \frac{\partial M_k}{\partial \varphi_k} &= \frac{\partial}{\partial \varphi_k} (R_{\varphi_k} R_{\theta_k} T_{d_k}) \\ &= \frac{\partial R_{\varphi_k}}{\partial \varphi_k} R_{\theta_k} T_{d_k} \\ &= -\frac{1}{2} e_{23} R_{\varphi_k} R_{\theta_k} T_{d_k} \\ &= -\frac{1}{2} e_{23} M_k. \end{aligned}$$

Similarmente, temos

$$\frac{\partial M_k}{\partial d_k} = R_{\varphi_k} R_{\theta_k} T_{d_k} \frac{-1}{2} e_{1\infty} = -\frac{1}{2} M_k e_{1\infty}.$$

A derivada com relação a θ_k requer um pequeno ajuste. A fim de evidenciar M_k , pois

$$\frac{\partial M_k}{\partial \theta_k} = -\frac{1}{2} R_{\varphi_k} e_{12} R_{\theta_k} T_{d_k},$$

vamos multiplicar o termo $R_{\varphi_k}^{-1}R_{\varphi_k}$ entre e_{12} e R_{θ_k} , assim,

$$\begin{aligned} -\frac{1}{2}R_{\varphi_k}e_{12}R_{\theta_k}T_{d_k} &= -\frac{1}{2}R_{\varphi_k}e_{12}(R_{\varphi_k}^{-1}R_{\varphi_k})R_{\theta_k}T_{d_k} \\ &= -\frac{1}{2}(R_{\varphi_k}e_{12}R_{\varphi_k}^{-1})M_k. \end{aligned}$$

Vamos padronizar a posição deste bivector sempre à esquerda, para isso, reposicionaremos o bivector da derivada em relação à d_k . Isso é facilmente resolvido ao multiplicar à direita o termo $M_k^{-1}M_k$. Desta maneira,

$$\begin{aligned} \frac{\partial M_k}{\partial d_k} &= -\frac{1}{2}M_k e_{1\infty} \\ &= -\frac{1}{2}M_k e_{1\infty} (M_k^{-1}M_k) \\ &= -\frac{1}{2}(M_k e_{1\infty} M_k^{-1}) M_k. \end{aligned}$$

Com isso, a derivada de M_k é, também, determinada pelo produto entre M_k e um bivector, afinal, rotores e motores não alteram a natureza dos elementos sobre os quais atuam. Então, genericamente, podemos escrever

$$\frac{\partial M_k}{\partial \alpha_k} = -\frac{1}{2}\mathcal{B}_{\alpha_k}M_k,$$

onde \mathcal{B}_{α_k} é o bivector que gera a derivada em α_k , sendo:

- $\mathcal{B}_{d_k} = M_k e_{1\infty} M_k^{-1}$;
- $\mathcal{B}_{\theta_k} = R_{\varphi_k} e_{12} R_{\varphi_k}^{-1}$;
- $\mathcal{B}_{\varphi_k} = e_{23}$.

A derivada de $M_{[i+1,j]}$ será igual a zero para os casos em que $k \leq i$ ou $j < k$. Para os casos em que $i < k \leq j$, temos,

$$\begin{aligned} \frac{\partial M_{[i+1,j]}}{\partial \alpha_k} &= M_{i+1} \cdots M_{k-1} \frac{\partial M_k}{\partial \alpha_k} M_{k+1} \cdots M_j \\ &= -\frac{1}{2}M_{i+1} \cdots M_{k-1} \mathcal{B}_{\alpha_k} M_k M_{k+1} \cdots M_j, \end{aligned}$$

logo podemos estabelecer

$$\frac{\partial M_{[i+1,j]}}{\partial \alpha_k} = -\frac{1}{2}M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]}. \quad (6.16)$$

Com as derivadas do motor $M_{[i+1,j]}$ devidamente estabelecidas, vamos voltar a derivada de $r_{i,j}^2$. Para tal cálculo, vamos substituir (6.16) em (6.15), assim, temos

$$\begin{aligned}
 \frac{\partial r_{i,j}^2}{\partial \alpha_k} &= -2 \left\langle e_0 \left(-\frac{1}{2} M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]} \right) e_0 (M_{[i+1,j]})^{-1} + \right. \\
 &\quad \left. - e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} \left(-\frac{1}{2} M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]} \right) (M_{[i+1,j]})^{-1} \right\rangle_0 \\
 &= \langle e_0 M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[i+1,j]})^{-1} + \\
 &\quad - e_0 M_{[i+1,j]} e_0 (M_{[i+1,j]})^{-1} M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]} (M_{[i+1,j]})^{-1} \rangle_0 \\
 &= \langle e_0 M_{[i+1,k-1]} \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[i+1,j]})^{-1} - e_0 M_{[i+1,j]} e_0 (M_{[k,j]})^{-1} \mathcal{B}_{\alpha_k} (M_{[i+1,k-1]})^{-1} \rangle_0 \\
 &= \langle e_0 M_{[i+1,k-1]} (\mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} - M_{[k,j]} e_0 (M_{[k,j]})^{-1} \mathcal{B}_{\alpha_k}) (M_{[i+1,k-1]})^{-1} \rangle_0 \\
 &= -2 \langle e_0 M_{[i+1,k-1]} (M_{[k,j]} e_0 (M_{[k,j]})^{-1} \cdot \mathcal{B}_{\alpha_k}) (M_{[i+1,k-1]})^{-1} \rangle_0.
 \end{aligned}$$

A passagem da penúltima para última linha utiliza a Equação (3.8), que identifica o produto interno em termos do produto geométrico.

Portanto, temos que a derivada de $r_{i,j}$ é dada por

$$\frac{\partial r_{i,j}}{\partial \alpha_k} = \begin{cases} \frac{1}{r_{i,j}} \langle e_0 M_{[i+1,k-1]} \langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1 (M_{[i+1,k-1]})^{-1} \rangle_0, & \text{se } i < k \leq j \\ 0, & \text{c.c.} \end{cases} \quad (6.17)$$

Comparando as diferentes maneiras de calcular $\frac{\partial r_{i,j}}{\partial \alpha_k}$, podemos observar que o uso de coordenadas conformes mantém as características identificadas no cálculo de distâncias. Permanecem aqui, tanto a redução do número de operadores a serem calculados quanto a simplificação das equações utilizadas. Também, o uso de coordenadas conformes permite padronizar a equação com relação a qualquer variável, ao contrário do uso de translações finais. Além disso, ao utilizar coordenadas conformes, há uma nova e importante diferença com relação aos demais métodos, nenhum novo elemento precisa ser calculado. De fato, conforme a Equação (6.17), a determinação dos motores $M_{i+1}, M_{i+2}, \dots, M_j$ é suficiente tanto para o cálculo da função de distâncias, quanto o cálculo de suas derivadas. O mesmo não ocorre quando utilizado coordenadas cartesianas, onde é necessário calcular, ainda, as derivadas das matrizes B_k .

Embora a equação sugira o uso da forma matricial de $M_{[k,j]}$ e $M_{[i+1,k-1]}$, isto não é, de fato, necessário. No cálculo de $\langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1$, estamos interessados apenas na parte vetorial do produto entre o bivector \mathcal{B}_{α_k} e o vetor $M_{[k,j]} e_0 (M_{[k,j]})^{-1}$. As componentes deste vetor são obtidas a partir dos coeficientes de $M_{[k,j]}$, e não precisam ser todas determinadas. O produto com o bivector, aliada ao fato de estarmos interessados apenas na sua parte vetorial, elimina a necessidade de calcular parte das componentes.

Para compreender melhor, vamos analisar $\frac{\partial r_{i,j}}{\partial \varphi_k}$, uma vez que as coordenadas

φ_k são, de fato, as de maior importância. Dado

$$\frac{\partial r_{i,j}^2}{\partial \varphi_k} = 2 \langle e_0 M_{[i+1,k-1]} \langle \mathcal{B}_{\varphi_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1 (M_{[i+1,k-1]})^{-1} \rangle_0,$$

vamos identificar o vetor

$$v = M_{[k,j]} e_0 (M_{[k,j]})^{-1} = v_0 e_0 + v_1 e_1 + v_2 e_2 + v_3 e_3 + v_\infty e_\infty.$$

A parte vetorial no centro da equação é dado por

$$\bar{v} = \langle e_{23} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1 = v_3 e_2 - v_2 e_3.$$

Isto significa que, neste momento, apenas os coeficientes v_3 e v_2 são necessários. Este mesmo processo se repete ao aplicar o motor $M_{[i+1,k-1]}$ no vetor resultante, uma vez que, neste segundo momento, nosso interesse é determinar apenas a parte escalar do produto entre e_0 e $M_{[i+1,k-1]} \bar{v} (M_{[i+1,k-1]})^{-1}$. Escrevendo

$$u = M_{[i+1,k-1]} \bar{v} (M_{[i+1,k-1]})^{-1} = u_0 e_0 + u_1 e_1 + u_2 e_2 + u_3 e_3 + u_\infty e_\infty,$$

a parte escalar do produto entre e_0 e u é dada por

$$\bar{u} = \langle e_0 M_{[i+1,k]} \bar{v} (M_{[i+1,k]})^{-1} \rangle_0 = -u_\infty,$$

ou seja, precisamos apenas do coeficiente de e_∞ .

Aplicando o motor $M_{[i+1,k-1]}$ sobre o vetor \bar{v} , temos

$$\begin{aligned} M_{[i+1,k-1]} \bar{v} (M_{[i+1,k-1]})^{-1} &= M_{[i+1,k-1]} (v_3 e_2 - v_2 e_3) (M_{[i+1,k-1]})^{-1} \\ &= (v_3 M_{[i+1,k-1]} e_2 (M_{[i+1,k-1]})^{-1} + v_2 M_{[i+1,k-1]} e_3 (M_{[i+1,k-1]})^{-1}). \end{aligned}$$

Falta, ainda, calcular v_2 , v_3 e o coeficiente de e_∞ dos produtos $M_{[i+1]} e_2 (M_{[i+1]})^{-1}$ e $M_{[i+1]} e_3 (M_{[i+1]})^{-1}$, que indicaremos, respectivamente, por w_1 e w_2 . Todos estes valores podem ser encontrados na forma matricial dos motores, dada pela matriz (5.5), já que estes são aplicados nos vetores da base. Dessa maneira, se

$$M_{[i+1,k-1]} = a_0 + a_1 e_{23} + a_2 e_{13} + a_3 e_{12} + a_4 e_{1\infty} + a_5 e_{2\infty} + a_6 e_{3\infty} + a_7 e_{123\infty}$$

e

$$M_{[k,j]} = b_0 + b_1 e_{23} + b_2 e_{13} + b_3 e_{12} + b_4 e_{1\infty} + b_5 e_{2\infty} + b_6 e_{3\infty} + b_7 e_{123\infty},$$

então,

$$\begin{aligned} v_2 &= 2(-b_0 b_5 - b_1 b_6 + b_2 b_7 + b_3 b_4) \\ v_3 &= 2(-b_0 b_6 + b_1 b_5 + b_2 b_4 - b_3 b_7) \\ w_1 &= 2(-a_3 a_4 - a_0 a_5 + a_1 a_6 - a_2 a_7) \\ w_2 &= 2(a_2 a_4 + a_1 a_5 + a_0 a_6 + a_3 a_7). \end{aligned}$$

Resumindo, a derivada, neste caso, é dada por

$$\frac{\partial r_{i,j}}{\partial \theta_k} = \frac{4}{r_{i,j}} \left((-b_0 b_6 + b_1 b_5 + b_2 b_4 - b_3 b_7)(-a_3 a_4 - a_0 a_5 + a_1 a_6 - a_2 a_7) + \right. \\ \left. (-b_0 b_5 - b_1 b_6 + b_2 b_7 + b_3 b_4)(a_2 a_4 + a_1 a_5 + a_0 a_6 + a_3 a_7) \right).$$

É possível, ainda, fazer uma análise similar tanto para a derivada com relação θ_k quanto com relação a d_k , embora, em geral, estas tenham menor relevância.

Para o cálculo da derivada de segunda ordem, vamos partir da Equação (6.14), então

$$\begin{aligned} \frac{\partial^2 r_{i,j}}{\partial \beta_p \partial \alpha_k} &= -\frac{1}{2r_{i,j}^2} \frac{\partial r_{i,j}}{\partial \beta_p} \frac{\partial r_{i,j}^2}{\partial \alpha_k} + \frac{1}{2r_{i,j}} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_p} \\ &= -\frac{1}{r_{i,j}} \frac{\partial r_{i,j}}{\partial \beta_p} \frac{1}{2r_{i,j}} \frac{\partial r_{i,j}^2}{\partial \alpha_k} + \frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_p} \\ &= \frac{1}{r_{i,j}} \left(-\frac{\partial r_{i,j}}{\partial \beta_p} \frac{\partial r_{i,j}}{\partial \alpha_k} + \frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_p} \right). \end{aligned} \quad (6.18)$$

Como as derivadas de primeira ordem de $r_{i,j}$ já foram estabelecidas, devemos calcular ainda

$$\frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_p} = \frac{\partial}{\partial \beta_p} \left[-\langle e_0 M_{[i+1,k-1]} \langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1 (M_{[i+1,k-1]})^{-1} \rangle_0 \right].$$

Dessa maneira, temos que avaliar três casos: $p \in [i+1, k-1]$, $p \in [k+1, j]$ e $p = k$.

Se p está no intervalo $[i+1, k-1]$, então o vetor $\langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \rangle_1$ pode ser tomado como constante. Vamos chamar tal vetor de V . Então,

$$\begin{aligned} \frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_k} &= \frac{\partial}{\partial \beta_p} \left[-\langle e_0 M_{[i+1,k-1]} V (M_{[i+1,k-1]})^{-1} \rangle_0 \right] \\ &= -\left\langle \frac{\partial}{\partial \beta_p} [e_0 M_{[i+1,k-1]} V (M_{[i+1,k-1]})^{-1}] \right\rangle_0 \\ &= -\left\langle e_0 \frac{\partial M_{[i+1,k-1]}}{\partial \beta_p} V (M_{[i+1,k-1]})^{-1} + e_0 M_{[i+1,k-1]} V \frac{\partial (M_{[i+1,k-1]})^{-1}}{\partial \beta_p} \right\rangle_0 \\ &= -\left\langle e_0 \frac{\partial M_{[i+1,k-1]}}{\partial \beta_p} V (M_{[i+1,k-1]})^{-1} + \right. \\ &\quad \left. -e_0 M_{[i+1,k-1]} V (M_{[i+1,k-1]})^{-1} \frac{\partial M_{[i+1,k-1]}}{\partial \beta_p} (M_{[i+1,k-1]})^{-1} \right\rangle_0 \\ &= \frac{1}{2} \langle e_0 M_{[i+1,p-1]} \mathcal{B}_{\beta_p} M_{[p,k-1]} V (M_{[i+1,k-1]})^{-1} + \\ &\quad -e_0 M_{[i+1,k-1]} V (M_{[i+1,k-1]})^{-1} M_{[i+1,p-1]} \mathcal{B}_{\beta_p} M_{[p,k]} (M_{[i+1,k-1]})^{-1} \rangle_0 \\ &= \frac{1}{2} \langle e_0 M_{[i+1,p-1]} \mathcal{B}_{\beta_p} M_{[p,k-1]} V (M_{[i+1,k-1]})^{-1} + \\ &\quad -e_0 M_{[i+1,k-1]} V (M_{[p,k-1]})^{-1} \mathcal{B}_{\beta_p} (M_{[i+1,p-1]})^{-1} \rangle_0 \\ &= \frac{1}{2} \langle e_0 M_{[i+1,p-1]} (\mathcal{B}_{\beta_p} M_{[p,k]} V (M_{[p,k]})^{-1} - M_{[p,k]} V (M_{[p,k]})^{-1} \mathcal{B}_{\beta_p}) (M_{[i+1,p-1]})^{-1} \rangle_0 \\ &= -\langle e_0 M_{[i+1,p-1]} (M_{[p,k]} V (M_{[p,k]})^{-1} \cdot \mathcal{B}_{\beta_p}) (M_{[i+1,p-1]})^{-1} \rangle_0, \end{aligned}$$

ou seja,

$$\frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_k} = \left\langle e_0 M_{[i+1,p-1]} \left\langle \mathcal{B}_{\beta_p} M_{[p,k-1]} \left\langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \right\rangle_1 (M_{[p,k-1]})^{-1} \right\rangle_1 (M_{[i+1,p-1]})^{-1} \right\rangle_0.$$

Para o caso em que p está no intervalo $[k+1, j]$, precisamos derivar apenas o vetor V , pois os demais elementos da equação são constantes. A derivação deste vetor é muito similar ao processo de derivação de $r_{i,j}^2$, uma vez que \mathcal{B}_{α_k} é constante, desse modo,

$$\begin{aligned} \frac{\partial V}{\partial \beta_p} &= \frac{\partial}{\partial \beta_p} \left\langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \right\rangle_1 \\ &= \left\langle \mathcal{B}_{\alpha_k} M_{[k,p-1]} \left\langle \mathcal{B}_{\beta_p} M_{[p,j]} e_0 (M_{[p,j]})^{-1} \right\rangle_1 (M_{[k,p-1]})^{-1} \right\rangle_1. \end{aligned}$$

Portanto, neste caso,

$$\frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_k} = \left\langle e_0 M_{[i+1,k-1]} \left\langle \mathcal{B}_{\alpha_k} M_{[k,p-1]} \left\langle \mathcal{B}_{\beta_p} M_{[p,j]} e_0 (M_{[p,j]})^{-1} \right\rangle_1 (M_{[k,p-1]})^{-1} \right\rangle_1 M_{[i+1,k-1]}^{-1} \right\rangle_0$$

Para o caso em que $p = k$ deve-se, também, avaliar a derivada de V . Entretanto, não podemos considerar \mathcal{B}_{α_k} constante. Partimos, então, da derivada de V , desse modo,

$$\begin{aligned} \frac{\partial V}{\partial \beta_k} &= \frac{\partial}{\partial \beta_k} \left\langle \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \right\rangle_1 \\ &= \left\langle \frac{\partial \mathcal{B}_{\alpha_k}}{\partial \beta_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} + \mathcal{B}_{\alpha_k} \frac{\partial M_{[k,j]}}{\partial \beta_k} e_0 (M_{[k,j]})^{-1} + \mathcal{B}_{\alpha_k} M_{[k,j]} e_0 \frac{\partial M_{[k,j]}^{-1}}{\partial \beta_k} \right\rangle_1 \\ &= \left\langle \frac{\partial \mathcal{B}_{\alpha_k}}{\partial \beta_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} + \mathcal{B}_{\alpha_k} \left\langle \mathcal{B}_{\beta_k} M_{[k,j]} e_0 (M_{[k,j]})^{-1} \right\rangle_1 \right\rangle_1. \end{aligned}$$

Logo, devemos avaliar as derivadas de \mathcal{B}_{α_k} . Para cada caso, temos:

- Se $\alpha_k = \varphi_k$, a derivada é igual a zero, pois, neste caso, o bivector, $\mathcal{B}_{\alpha_k} = e_{23}$, é constante;
- Se $\alpha_k = \theta_k$, o bivector, $\mathcal{B}_{\theta_k} = R_{\varphi_k} e_{12} R_{\varphi_k}^{-1}$, depende somente de φ_k , assim

$$\begin{aligned} \frac{\partial \mathcal{B}_{\theta_k}}{\partial \varphi_k} &= \frac{\partial}{\partial \varphi_k} (R_{\varphi_k} e_{12} R_{\varphi_k}^{-1}) \\ &= \frac{\partial R_{\varphi_k}}{\partial \varphi_k} e_{12} R_{\varphi_k}^{-1} + R_{\varphi_k} e_{12} \frac{\partial R_{\varphi_k}^{-1}}{\partial \varphi_k} \\ &= -\frac{1}{2} (e_{23} R_{\varphi_k} e_{12} R_{\varphi_k}^{-1} - R_{\varphi_k} e_{12} R_{\varphi_k}^{-1} e_{23}) \\ &= -\frac{1}{2} (\mathcal{B}_{\varphi_k} \mathcal{B}_{\theta_k} - \mathcal{B}_{\theta_k} \mathcal{B}_{\varphi_k}); \end{aligned} \tag{6.19}$$

- Se $\alpha_k = d_k$, então o bivector $\mathcal{B}_{d_k} = M_k e_{1\infty} M_k^{-1}$ pode ser derivado para β_k substituindo d_k, θ_k ou φ_k :

$$\begin{aligned} \frac{\partial \mathcal{B}_{d_k}}{\partial \beta_k} &= \frac{\partial}{\partial \beta_k} (M_k e_{1\infty} M_k^{-1}) \\ &= \frac{\partial M_k}{\partial \beta_k} e_{1\infty} M_k^{-1} + M_k e_{1\infty} \frac{\partial M_k^{-1}}{\partial \beta_k} \\ &= -\frac{1}{2} (\mathcal{B}_{\beta_k} M_k e_{1\infty} M_k^{-1} - M_k e_{1\infty} M_k^{-1} \mathcal{B}_{\beta_k}) \\ &= -\frac{1}{2} (\mathcal{B}_{\beta_k} \mathcal{B}_{d_k} - \mathcal{B}_{d_k} \mathcal{B}_{\beta_k}). \end{aligned}$$

Para o caso em que $\beta_k = d_k$ a derivada do bivector é zero.

Com isso, podemos definir a derivada de segunda ordem para os casos em que $\alpha_k = \theta_k$ e $\beta_k = \varphi_k$ e $\alpha_k = d_k$ e $\beta_k = \varphi_k$ ou $\beta_k = \theta_k$, a partir da seguinte expressão:

$$\begin{aligned} \frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \beta_k \partial \alpha_k} &= - \left\langle e_0 M_{[i+1,k-1]} \left\langle \mathcal{B}_{\alpha_k} \left\langle \mathcal{B}_{\beta_k} M_{[k,j]} e_0 M_{[k,j]}^{-1} \right\rangle_1 \right\rangle_1 + \right. \\ &\quad \left. + \frac{1}{2} (\mathcal{B}_{\alpha_k} \mathcal{B}_{\beta_k} - \mathcal{B}_{\beta_k} \mathcal{B}_{\alpha_k}) M_{[k,j]} e_0 M_{[k,j]}^{-1} \right\rangle_1 M_{[i+1,k-1]}^{-1} \right\rangle_0 \end{aligned}$$

Mencionamos por diversas vezes, ao longo do texto, que o caso de maior atenção é termos relacionados às variáveis φ_k , pois tanto d_k quanto θ_k podem ser considerados fixos. Com isso, a expressão para derivada de segunda ordem, em relação somente às φ_k , tornam-se ainda mais simples, pois o caso em que $p = k$ se enquadra nas demais expressões. Com isso, $\frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \varphi_p \partial \varphi_k}$ é dada conforme os seguintes casos:

- para $i < p < k \leq j$,

$$\left\langle e_0 M_{[i+1,p-1]} \left\langle e_{23} M_{[p,k-1]} \left\langle e_{23} M_{[k,j]} e_0 (M_{[k,j]}^{-1})^{-1} \right\rangle_1 (M_{[p,k-1]}^{-1})^{-1} \right\rangle_1 (M_{[i+1,p-1]}^{-1})^{-1} \right\rangle_0;$$

- para $i < k \leq p \leq j$,

$$\left\langle e_0 M_{[i+1,k-1]} \left\langle e_{23} M_{[k,p-1]} \left\langle e_{23} M_{[p,j]} e_0 (M_{[p,j]}^{-1})^{-1} \right\rangle_1 (M_{[k,p-1]}^{-1})^{-1} \right\rangle_1 (M_{[i+1,k-1]}^{-1})^{-1} \right\rangle_0;$$

- para todos os outros casos $\frac{1}{2} \frac{\partial^2 r_{i,j}^2}{\partial \varphi_p \partial \varphi_k} = 0$

É possível notar que as vantagens obtidas para o cálculo da derivada de primeira ordem permanecem válidas quando calculamos as de segunda ordem. Os motores que precisam ser calculados são os mesmos que utilizamos para calcular as distâncias e as derivadas de primeira ordem. Além disso, nenhum novo elemento precisa ser determinado. As derivadas de segunda ordem são estabelecidas exclusivamente em função dos coeficientes

dos motores, e podem ser estabelecidos de maneira similar ao que foi feito para o caso da derivada primeira. É possível notar ainda uma redução muito significativa no número de casos que definem $\frac{\partial^2 r_{i,j}^2}{\partial \beta_p \partial \alpha_k}$ com relação ao uso de coordenadas cartesianas. Isto é possível porque as derivadas de motores não envolvem o cálculo de nenhum novo operador, além dos já calculados. Por fim, ao contrário do que ocorre com o uso de translações finais, o uso de coordenadas conformes permite codificar todas as derivadas em uma única expressão, onde α_k e β_p podem ser substituídos por qualquer uma das variáveis θ , φ e d .

O uso de coordenadas conformes se mostra, portanto, muito prático tanto no cálculo das funções energia quanto suas derivadas.

6.4 Testes computacionais

Embora tenhamos partido das funções de energia, olhando para os cálculos realizados podemos observar que a parte crucial de sua resolução recai exatamente sobre as funções de distância. Dessa maneira, vamos nos focar na implementação de tais funções e suas derivadas.

Com intuito de validar as expressões encontradas, implementamos, no software *Wolfram Mathematica*, as funções $r_{i,j}$ e $\frac{\partial r_{i,j}}{\partial \varphi_k}$ utilizando a formulação matricial para coordenadas cartesianas desenvolvida por [Phillips, Rosen e Walke \(1995\)](#) a fim de comparar com nossa versão usando coordenadas conformes. Embora as derivadas com relação a θ_k e d_k tenham sido apresentadas, estas variáveis, em geral, podem ser tomadas como fixas, conforme já mencionamos, e, por esta razão, optamos por realizar testes apenas considerando as variáveis φ_k . O cálculo das derivadas de segunda ordem seguem, basicamente, os mesmos procedimentos do cálculo das derivadas de primeira ordem, tanto para coordenadas cartesianas quanto conformes, com isso entendemos que validar a derivada de primeira ordem é suficiente. Ainda, antes de prosseguir para os testes, vamos estimar o custo computacional dos métodos implementados.

Primeiramente, vamos avaliar o número de operações necessárias para calcular a distância $r_{i,j}$ e suas derivadas, a partir das coordenadas cartesianas. Começaremos pelo cálculo de $r_{i,j}$, dada pela Equação (6.2). Temos, 46 operações para cada inclusão de B_i no produto $B_{[i-1]}$. Este valor foi estabelecido na conversão de coordenadas, na Seção 1.2. Assim, temos $46i$ operações para o cálculo do primeiro produtório que define $r_{i,j}$, mais $46(j-i-1)$ para o cálculo do segundo produtório. Além disso, são necessárias 3 subtrações para diferença da identidade, 36 produtos 27 somas para o cálculo de $B_{i,j}$ e 3 produtos, 2 somas e uma raiz para calcular a norma da quarta coluna da matriz $B_{i,j}$. Totalizando, assim, $46j + 26$ operações.

Para o cálculo da primeira derivada, seguindo a Equação (6.9), vamos considerar

apenas o caso em que $i < k \leq j$, pois na versão conforme somente este é diferente de 0. Ao calcular a derivada de primeira ordem, o cálculo de $r_{i,j}$ se faz necessário, de modo que aqui são requeridas as $46j + 26$ operações. Contudo, as matrizes B_i , com i de 1 a j , já foram calculadas, deste modo, resta determinar o número de operações necessárias para calcular $\frac{\partial B_{i,j}}{\partial \varphi_k}$. São necessários 6 produtos para determinar $\frac{\partial B_k}{\partial \varphi_k}$. Ainda, são necessários 60 produtos e 39 somas para multiplicar substituir B_k por $\frac{\partial B_k}{\partial \varphi_k}$ na construção de $\frac{\partial B_{i,j}}{\partial \varphi_k}$. Feito isso, são precisos ainda 7 produtos e 4 somas para efetivamente calcular $\frac{\partial r_{i,j}}{\partial \varphi_k}$. Totalizando, assim, $46(j + 3) + 4$ operações.

Para o cálculo da segunda derivada, dada pela equação 6.11, vamos analisar apenas o caso em que $1 < p < k < j$, tendo em vista que o modelo conforme tem um número de casos reduzido. Esta análise deve ser suficiente para estimar a diferença entre as duas formulações. Observe que, conforme a equação 6.11, as derivadas de $r_{i,j}$ em relação à φ_p e φ_k também devem ser calculadas. Com isso, temos, $46(j + 3) + 4$ operações para determinar $\frac{\partial r_{i,j}}{\partial \varphi_p}$. Uma vez que este tenha sido calculado, a determinação de $\frac{\partial r_{i,j}}{\partial \varphi_k}$ requer somente 66 produtos e 39 somas adicionais para construção de $\frac{\partial B_{i,j}}{\partial \varphi_k}$. Tendo sido, os demais elementos da equação, calculados, vamos calcular $\frac{\partial^2 B_{i,j}}{\partial \varphi_p \partial \varphi_k}$, o que inclui 72 produtos e 54 somas. Por fim, calculados todos os elementos da equação, temos 5 produtos e 3 somas para determinar a derivada. Totalizando, deste modo, $46(j + 8) + 13$ operações.

Agora, vamos estabelecer o custo computacional do cálculo de $r_{i,j}$ e suas derivadas de primeira e segunda ordem, para os mesmos casos, considerando a construção de coordenadas conformes. Vamos começar determinando o custo de calcular os motores envolvidos na construção. Para cada novo motor são necessários, conforme estabelecido na conversão de coordenadas, 48 operações. Com isso temos um custo de $48(j - i - 1)$ operações para determinar os motores com índice de $i + 1$ até j . O cálculo das derivadas de primeira e segunda ordem não requer a determinação de nenhum novo elemento, isto significa que esse custo se manterá para todas as funções sendo apenas necessária a manipulação dos coeficientes dos motores, que serão agrupados adequadamente. Para o cálculo de $r_{i,j}$ são necessários 5 produtos, 3 somas e uma operação de raiz, ou seja, 9 operações adicionais, totalizando $48(j - i) - 39$ operações. Para a derivada de primeira ordem, no caso $i < k \leq j$ além das operações para construção de $r_{i,j}$ são necessários 20 produtos e 13 somas, conforme a Equação (6.3). Totalizando, assim, $48(j - 1) - 6$ operações.

Finalmente, no cálculo da derivada segunda, dada pela Equação (6.18), temos inclusos a derivada de primeira ordem com relação à φ_p e φ_k . Para determinar a derivada em φ_p temos $48(j - i) - 6$ operações, e isto já inclui o cálculo de $r_{i,j}$, então, para a

derivada em φ_k são necessários ainda 20 produtos e 13 somas. Para o cálculo de $\frac{\partial^2 r_{i,j}^2}{\partial \varphi_p \partial \varphi_k}$ são necessários, por uma análise similar à que resulta na Equação (6.3), mais 26 produtos e 21 somas. Para a determinação da derivada segunda, ainda são necessários 3 produtos e uma soma. Totalizando, assim, $48(j - i + 1) + 30$ operações. As informações a respeito do custo computacional são resumidas na Tabela 17.

Tabela 17 – Custo computacional estimado para o cálculo de $r_{i,j}$ e suas derivadas de primeira e segunda ordem.

Função	Custo com matrizes	Custo com álgebra conforme
$r_{i,j}$	$46j + 26$	$48(j - i) - 39$
$\frac{\partial r_{i,j}}{\partial \varphi_k}$	$46(j + 3) + 4$	$48(j - i) - 6$
$\frac{\partial^2 r_{i,j}}{\partial \varphi_p \partial \varphi_k}$	$46(j + 8) + 13$	$48(j - i + 1) + 30$

Podemos verificar que quanto maior o índice i , maior a diferença entre o cálculo das funções.

Para o teste das funções, foram gerados quatro conjuntos de coordenadas internas, seguindo os mesmos procedimentos utilizados para gerar os modelos anteriores. O primeiro conjunto com 10 pontos, o segundo com 30, o terceiro com 50 e o quarto com 100 pontos. Foram, então, calculadas as coordenadas cartesianas utilizando matrizes com translações encaixadas e, a partir de tais coordenadas, calculamos a distância euclidiana entre os pares de átomos, sendo estes os valores tomados como referência. Feito isso, foram calculadas as matrizes de distâncias a partir das coordenadas cartesianas e conformes. Por fim, calculamos a maior diferença absoluta entre os valores obtidos por cada modelo e os valores de referência. Na Tabela 18 é possível verificar a maior diferença absoluta para cada um dos conjuntos de pontos.

Tabela 18 – Maior diferença absoluta entre as distâncias geradas e a matriz de referência.

n.º de pontos	Modelo conforme	Modelo matricial
10	$3.55271 \cdot 10^{-15}$	$8.88178 \cdot 10^{-16}$
30	$1.59872 \cdot 10^{-14}$	$5.32907 \cdot 10^{-15}$
50	$5.32907 \cdot 10^{-14}$	$1.24345 \cdot 10^{-14}$
100	$1.56319 \cdot 10^{-14}$	$1.95399 \cdot 10^{-14}$

Fonte: Elaborada pelo autor.

Podemos perceber, com base nos valores apresentados na tabela, que o uso da álgebra conforme fornece distâncias tão precisas quanto o modelo já consolidado.

Para os testes da derivada, utilizamos apenas o conjunto de 10 pontos, visto que a quantidade de variáveis envolvidas é suficiente para uma boa avaliação da funcionalidade

dos métodos. Foram calculadas as derivadas das funções $r_{1,10}$, $r_{2,7}$ e $r_{3,6}$ com relação a cada um dos ângulos φ_k , com k variando de 1 a 10. Neste teste, verificamos a diferença absoluta entre as derivadas $\frac{\partial r_{i,j}}{\partial \varphi_k}$ do modelo descrito por Phillips, Rosen e Walke (1995) e o modelo conforme, apresentado neste texto. A Tabela 19 apresenta estas diferenças para cada derivada calculada. Os valores de $k \in \{1, 2, 3\}$ foram iguais a zero para todas as três funções e por esta razão foram removidos da tabela.

Tabela 19 – Diferença absoluta entre as derivadas $\frac{\partial r_{i,j}}{\partial \varphi_k}$ calculadas com matrizes e álgebra conforme.

	φ_4	φ_5	φ_6	φ_7	φ_8	φ_9	φ_{10}
$r_{1,10}$	$6.4 \cdot 10^{-16}$	$9.8 \cdot 10^{-15}$	0	$2.2 \cdot 10^{-15}$	$1.8 \cdot 10^{-14}$	$2.2 \cdot 10^{-15}$	$6.7 \cdot 10^{-16}$
$r_{2,7}$	$8.9 \cdot 10^{-16}$	$3.1 \cdot 10^{-15}$	$3.1 \cdot 10^{-15}$	$2.7 \cdot 10^{-15}$	0	0	0
$r_{3,6}$	$8.9 \cdot 10^{-16}$	$7.5 \cdot 10^{-17}$	$4.4 \cdot 10^{-16}$	0	0	0	0

Fonte: Elaborada pelo autor.

Em concordância com o cálculo de distâncias, o uso de coordenadas conformes no cálculo das derivadas de $r_{i,j}$ se mostram tão precisos quanto o uso de coordenadas cartesianas já consolidadas na literatura.

7 Considerações Finais

Ao longo dos capítulos deste texto, buscamos destrinchar os processos usuais de construção de coordenadas a fim de estabelecer um novo modelo de representação de proteínas. Nosso modelo, intitulado coordenadas conformes, provou-se tão preciso quanto as versões já estabelecidas e utilizadas. Além de incluir em sua composição as coordenadas cartesianas, as coordenadas conformes também codificam as distâncias a partir do produto interno. Isso foi particularmente útil ao trabalhar com as funções de distância.

Naturalmente, espelhando-se no caminho traçado para coordenadas cartesianas, percebemos a necessidade de atualizar coordenadas conformes já calculadas a partir de mudanças conformacionais, dadas em termos das coordenadas internas. Este processo de atualização é de fundamental importância, uma vez que, ao lidar com proteínas, tais mudanças são comuns, e a atualização evita a necessidade de recalcular completamente a estrutura. Como no processo de construção, os algoritmos de atualização mostraram-se tão precisos e robustos quanto os modelos já conhecidos, tendo reposta igualmente eficiente a perturbações nos ângulos de torção.

Como o uso de coordenadas conformes incluem duas novas coordenadas em cada ponto, é natural perceber um acréscimo no custo computacional, se comparado às coordenadas cartesianas. Contudo, a depender da escolha do posicionamento das translações, este acréscimo pode ser menor, tornando, ainda assim, o modelo viável. Isso indica que, se aperfeiçoadas, as coordenadas conformes pode ser extremamente competitivas com a representação por coordenadas cartesianas.

Entre novas possibilidades, obtidas ao se trabalhar no espaço conforme, destacou-se, indubitavelmente, o cálculo de funções de distância e suas derivadas de primeira e segunda ordem. As equações criadas na elaboração das coordenadas conformes, e as propriedades de codificação de distância por meio do produto interno, permitiram descrever tais funções de maneira elegante e sucinta. Com isso, sua implementação computacional tornou-se mais simples e direta, sendo possível calcular tanto as funções quanto suas derivadas sem a necessidade de determinar novos termos. Além disso, testes mostraram que tanto a função de distância quanto suas derivadas são tão precisas quanto às determinadas nos métodos convencionais.

Explorar novos caminhos podem incluir alguns percalços. O uso da álgebra geométrica e, especialmente, da álgebra geométrica conforme, apresentou uma série de dificuldades. Em geral, o produto geométrico tem propriedades bem estabelecidas quando se dá entre dois vetores. Infelizmente, quando lidamos com k -vetores, ou multivetores, estas propriedades tornam-se mais escassas. Além disso, a não comutatividade do produto pode

gerar uma série de dificuldades na simplificação de certas expressões. Isso foi contornado ao longo do processo, com novas identidades estabelecidas conforme a necessidade.

Ainda que muito tenha sido feito, há muito a se fazer. Um novo sistema de representação abre inúmeras possibilidades que devem ser exploradas. Por exemplo, as implementações computacionais, desses novos métodos, foram feitas na intenção de validar os resultados teóricos e ilustrá-los, e podem ainda ser aprimoradas afim de mostrar maior eficiência e menor custo computacional.

Referências

- BERG, J.; TYMOCZKO, J.; STRYER, L. *Bioquímica*. [S.l.]: Guanabara Koogan, 2008. ISBN 9788527713696.
- BOHM, H.-J.; STAHL, M. The use of scoring functions in drug discovery applications. *Reviews in computational chemistry*, Wiley Online Library, v. 18, p. 41–88, 2002.
- CHOI, V. On updating torsion angles of molecular conformations. *Journal of chemical information and modeling*, ACS Publications, v. 46, n. 1, p. 438–444, 2006.
- CHYS, P.; CHACÓN, P. Spinor product computations for protein conformations. *Journal of computational chemistry*, Wiley Online Library, v. 33, n. 21, p. 1717–1729, 2012.
- DHANIK, A.; KAVRAKI, L. E. Protein-ligand interactions: computational docking. *John Wiley Sons, Ltd. doi*, v. 10, n. 9780470015902, p. a0004105, 2012.
- DION, S. M.; PACCA, J. L. d. A.; MACHADO, N. J. Quaternions: sucessos e insucessos de um projeto de pesquisa. *Estudos avançados*, SciELO Brasil, v. 9, n. 25, p. 251–262, 1995.
- FERNANDES, L.; LAVOR, C.; OLIVEIRA, M. *Álgebra Geométrica e Aplicações*. [S.l.]: SBMAC, 2017. ISBN 978-85-8215-080-1.
- FLOWER, D. Rotational superposition: A review of methods. *Journal of molecular graphics & modelling*, v. 17, n. 3-4, p. 238–244, 1999.
- KANATANI, K. *Understanding Geometric Algebra: Hamilton, Grassmann, and Clifford for Computer Vision and Graphics*. [S.l.]: AK Peters/CRC Press, 2015.
- LOUNESTO, P. *Clifford Algebras and Spinors*. 2. ed. [S.l.]: Cambridge University Press, 2001. (London Mathematical Society Lecture Note Series).
- MACDONALD, A. *Linear and geometric algebra*. [S.l.]: Alan Macdonald, 2010.
- _____. *Vector and geometric calculus*. [S.l.]: CreateSpace Independent Publishing Platform, 2012. v. 1.
- MURRAY, G. Rotation about an arbitrary axis in 3 dimensions. *Online] http://inside.mines.edu*, 2013.
- NAMBA, A. M.; SILVA, V. B. da; SILVA, C. D. Dinâmica molecular: teoria e aplicações em planejamento de fármacos. *Eclética Química*, SciELO Brasil, v. 33, n. 4, p. 13–24, 2008.
- PHILLIPS, A. T.; ROSEN, J. B.; WALKE, V. H. Molecular structure determination by convex, global underestimation of local energy minima. *Global Minimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*, Citeseer, v. 23, p. 181–198, 1995.
- SELIG, J. *Geometrical methods in robotics*. [S.l.]: Springer Science & Business Media, 2013.

- SELIG, J. M. *Geometric fundamentals of robotics*. [S.l.]: Springer Science & Business Media, 2004.
- SELIG, J. M.; BAYRO-CORROCHANO, E. Rigid body dynamics using clifford algebra. *Advances in applied Clifford algebras*, Springer, v. 20, n. 1, p. 141–154, 2010.
- SEOK, C.; COUTSIAS, E. A. Efficiency of rotational operators for geometric manipulation of chain molecules. *Bulletin of the Korean Chemical Society*, Korean Chemical Society, v. 28, n. 10, p. 1705–1708, 2007.
- THOMPSON, H. B. Calculation of cartesian coordinates and their derivatives from internal molecular coordinates. *The Journal of Chemical Physics*, AIP, v. 47, n. 9, p. 3407–3410, 1967.
- TODESCHINI, R.; CONSONNI, V. *Molecular descriptors for chemoinformatics: volume I: alphabetical listing/volume II: appendices, references*. [S.l.]: John Wiley & Sons, 2009. v. 41.
- ZHANG, M.; KAVRAKI, L. E. A new method for fast and accurate derivation of molecular conformations. *Journal of Chemical Information and Computer Sciences*, ACS Publications, v. 42, n. 1, p. 64–70, 2002.