



UNICAMP

UNIVERSIDADE ESTADUAL DE
CAMPINAS

Instituto de Matemática, Estatística e
Computação Científica

ALYSSON MATOS DE SOUZA

Álgebra Geométrica Conforme aplicada à atualização de estruturas 3D de proteínas

Campinas

2023

Alysson Matos de Souza

Álgebra Geométrica Conforme aplicada à atualização de estruturas 3D de proteínas

Dissertação 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 Mestre em Matemática Aplicada.

Orientador: Carlile Campos Lavor

Este trabalho corresponde à versão final da Dissertação defendida pelo aluno Alysson Matos de Souza e orientada pelo Prof. Dr. Carlile Campos Lavor.

Campinas

2023

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

So89a Souza, Alysson, 1998-
Álgebra geométrica conforme aplicada à atualização de estruturas 3D de proteínas / Alysson Matos de Souza. – Campinas, SP : [s.n.], 2023.

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

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

Informações Complementares

Título em outro idioma: Conformal geometric algebra applied to updating 3D protein structures

Palavras-chave em inglês:

Geometric algebra

Proteins - Structure

Conformal space

Área de concentração: Matemática Aplicada

Titulação: Mestre em Matemática Aplicada

Banca examinadora:

Carlile Campos Lavor [Orientador]

Sueli Irene Rodrigues Costa

Rafael Santos de Oliveira Alves

Data de defesa: 14-03-2023

Programa de Pós-Graduação: Matemática Aplicada

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

- ORCID do autor: <https://orcid.org/0009-0003-2912-1797>

- Currículo Lattes do autor: <http://lattes.cnpq.br/9698236393366931>

**Dissertação de Mestrado defendida em 14 de março de 2023 e aprovada
pela banca examinadora composta pelos Profs. Drs.**

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

Prof(a). Dr(a). SUELI IRENE RODRIGUES COSTA

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

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.

Aos meus amados pais, José Maria e Maria Tereza, pelo incessante apoio.

Agradecimentos

Agradeço, em primeiro lugar, e não poderia ser diferente, à quem sempre esteve do meu lado, me apoiando e me dando todo o suporte necessário: meus queridos pais, José Maria e Maria Tereza. Sem eles, não estaria onde estou.

Um agradecimento especial e imenso ao professor Carlile Lavor, que me orientou neste trabalho, por todo suporte, acolhimento, atenção e paciência. Foi uma honra! Aproveito para estender o agradecimento a todos os professores do Imecc com quem tive o prazer de ter sido aluno.

Agradeço às prezadas amigas que o Imecc me trouxe, que compartilharam comigo momentos felizes, de compreensão e, sobretudo, companheirismo.

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

“Geometria sem álgebra é burra! Álgebra sem geometria é cega!” - David Hestenes

Resumo

Relacionadas com praticamente todas as funções de um sistema biológico, desde a atuação na defesa do organismo até o transporte de substâncias e a transmissão de impulsos nervosos, as proteínas são macromoléculas extremamente importantes para os organismos vivos. Compreender sua função passa intrinsecamente por conhecer seu formato tridimensional. Em diversas aplicações, devido à ocasião de mudanças conformacionais das moléculas de proteínas, é extremamente necessário realizar atualizações da estrutura 3D. Por esse motivo, esse trabalho tem o objetivo de apresentar e detalhar os principais métodos de atualização de tais estruturas. Para isso, é necessário entender o que mudanças nas coordenadas internas - um sistema bastante utilizado em Bioquímica - afetam as coordenadas cartesianas dos pontos que representam átomos. Além disso, também tratamos de um recente sistema de coordenadas que surge com a utilização da Álgebra Geométrica Conforme: as coordenadas conformes. Com ele, buscamos mostrar que os resultados obtidos com os métodos clássicos também são observados nessa nova abordagem.

Palavras-chave: estruturas 3D de proteínas. álgebra geométrica. modelo conforme.

Abstract

Related to practically all functions of a biological system, from acting in the body's defense to the transport of substances and the transmission of nerve impulses, proteins are extremely important macromolecules for living organisms. Understanding its function intrinsically involves knowing its three-dimensional format. In several applications, due to the occasion of conformational changes of protein molecules, it is extremely necessary to perform updates of the 3D structure. For this reason, this work aims to present and detail the main methods of updating such structures. For this, it is necessary to understand what changes in the internal coordinates - a system widely used in Biochemistry - affect the Cartesian coordinates of the points that represent atoms. In addition, we also deal with a recent coordinate system that arises with the use of Conformal Geometric Algebra: the conformal coordinates. With it, we seek to show that the results obtained with the classical methods are also observed in this new approach.

Keywords: 3D structures of proteins. geometric algebra. conformal model.

Lista de ilustrações

Figura 1 – Estrutura fundamental dos aminoácidos.	17
Figura 2 – Estruturas secundária e terciária de uma molécula de tirosina quinase LCK.	19
Figura 3 – Exemplo de duas moléculas de proteínas, cujas imagens foram extraídas do <i>Protein Data Bank</i> (PDB), grande base de dados de estruturas 3D de proteínas.	20
Figura 4 – Combinação de dois aminoácidos por meio de uma ligação peptídica, causando a perda de uma molécula de água.	20
Figura 5 – Ilustração das coordenadas internas de uma molécula, em que d é a distância de ligação, θ o ângulo de ligação e ψ o ângulo de torção. . . .	21
Figura 6 – Ilustração da circunferência que surge como lugar geométrico para o posicionamento de um átomo.	23
Figura 7 – Ilustração da mudança de posição de um átomo devido a uma alteração realizada no ângulo de torção associado.	23
Figura 8 – Ilustração da rotação e da translação no espaço.	25
Figura 9 – Rotação em relação ao eixo X no espaço.	26
Figura 10 – Sequência Z-X-Z usando os Ângulos de Euler. Na primeira imagem, uma rotação em torno do eixo Z por um ângulo α ; na segunda, em torno do eixo X' por um ângulo β ; por fim, em torno de Z' , com ângulo γ . . .	28
Figura 11 – Rotação em torno do eixo X.	30
Figura 12 – Rotação em torno do eixo Y.	31
Figura 13 – Rotação do vetor v por um ângulo θ em torno do eixo de rotação representado por u , que é unitário. Como resultado, temos o vetor v' . . .	32
Figura 14 – Destaque dos pontos anteriores a x_k , que não têm distâncias alteradas entre si, apesar das atualizações ilustradas em verde.	40
Figura 15 – Destaque dos pontos posteriores a x_{k-3} , que não têm distâncias alteradas entre si, apesar das atualizações realizadas, ilustradas em verde.	42
Figura 16 – Destaque dos pontos anteriores a x_{k-2} e posteriores a x_{k-1} , cujas distâncias são alteradas devido as atualizações, ilustradas em verde. . . .	43
Figura 17 – Interseção de duas esferas produzindo uma circunferência, lugar geométrico das possíveis atualizações do ponto x_i	43
Figura 18 – Disposição dos pontos e ligações representando uma pequena molécula.	44
Figura 19 – Vetor unitário u que servirá como eixo de rotação.	45
Figura 20 – Na imagem da esquerda, as possíveis localizações para o resultado da rotação. Na direita, uma rotação por meio de um ângulo ϕ_{k_1}	45
Figura 21 – Atualização do ponto x_{k_1} , representado por x'_{k_1} em verde, após a translação.	46

Figura 22 – Atualização dos pontos entre x_{k_1} e x_{k_1+1} , representados por x'_{k_1} e x'_{k_1+1} e ilustrados em verde.	46
Figura 23 – Ilustração do modelo com propagação direta, com as atualizações dos pontos em verde.	47
Figura 24 – Ilustração do modelo com propagação reversa, com as atualizações dos pontos em verde.	47
Figura 25 – Norma da diferença entre os ângulos de torção originais e atualizados ($ \psi' - \psi $) em função da norma das alterações nos ângulos de torção ($ \phi $). Em azul, os valores obtidos em cada simulação; em cinza, a reta identidade.	54
Figura 26 – Diferença entre as normas $ \psi'_i - \psi_i $ e $ \phi_i $	55
Figura 27 – Valores obtidos do RMSD na comparação das versões do Algoritmo 1 com as propagações reversa e direta.	56
Figura 28 – Valores obtidos do RMSD na comparação entre o Algoritmo 1 (com Propagação Reversa) e os Algoritmos 2, 3, 4 e 5, cada um com as duas versões de propagação de rotações.	57
Figura 29 – Diferenças entre as normas $ \psi'_i - \psi_i $ e $ \phi_i $ em cada uma das duplas de algoritmos.	58
Figura 30 – Interpretação geométrica de bivectores e trivetores gerados a partir dos vetores da base canônica.	66
Figura 31 – Vetores e_4 e e_5 dispostos ortogonalmente em relação aos vetores canônicos do $\mathbb{R}^3$, que é representado pela reta vermelha. Além disso, e_0 e e_∞ estão presentes como parte do hipercone formado, conhecido como cone nulo.	72
Figura 32 – Representação de como um ponto do $\mathbb{R}^3$ é localizado no Modelo Conforme. O lugar geométrico neste último para todos os pontos do $\mathbb{R}^3$ é uma orosfera, ilustrada como uma parábola, em vermelho. Também pode ser visto que a coordenada e_0 representa a origem desse espaço.	74
Figura 33 – Localização dos pontos X e λX , ambos representando $x \in \mathbb{R}^3$ no Espaço Conforme.	75
Figura 34 – Vetor unitário b calculado a partir dos pontos X_{k_i-1} e X_{k_i-2}	83
Figura 35 – Representação geométrica do vetor t	84
Figura 36 – Rotação do segmento que liga os pontos X_{k_i-2} e X_{k_i} em relação a um eixo que não passa pela origem, gerando X'_{k_i}	85
Figura 37 – Representação da reta que passa pelos pontos X_{k_i-1} e X_{k_i-2}	86
Figura 38 – Plano de rotação definido como o dual da reta r que passa pelos pontos X_{i-2} e X_{i-1} . Em destaque sobre o plano, a circunferência onde habitam as possíveis atualizações do ponto X_i	87
Figura 39 – Atualização das coordenadas de pontos com a aplicação do motor $M_{[i]}$	87

Figura 40 – Norma da diferença entre os ângulos de torção originais e atualizados ($ \psi' - \psi $) em função da norma das alterações nos ângulos de torção ($ \phi $). Em azul, os valores obtidos em cada simulação; em cinza, a reta identidade.	92
Figura 41 – Diferença entre as normas $ \psi'_i - \psi_i $ e $ \phi_i $	92
Figura 42 – Valores obtidos do RMSD na comparação do Algoritmo 8 com os Algoritmos 6 e 7.	93
Figura 43 – Quantidade de operações em função do número de pontos que precisam ser atualizados.	94
Figura 44 – Tempo de processamento em função do número de pontos que precisam ser atualizados.	96

Lista de Algoritmos

Algoritmo 1 – Atualização de coordenadas cartesianas usando Rotações Simples e Translações Finais.	49
Algoritmo 2 – Atualização de coordenadas cartesianas usando Rotações Simples e Translações Encaixadas.	51
Algoritmo 3 – Atualização de coordenadas cartesianas usando Quatérnios e Translações Finais	52
Algoritmo 4 – Atualização de coordenadas cartesianas usando Quatérnios e Translações Encaixadas	52
Algoritmo 5 – Atualização de coordenadas cartesianas usando Quatérnios nativamente e Translações Finais.	53
Algoritmo 6 – Atualização de coordenadas conformes usando Motores e Translações Encaixadas	84
Algoritmo 7 – Atualização de coordenadas conformes com Motores e Translações Encaixadas - versão 2	88
Algoritmo 8 – Atualização de coordenadas conformes usando Rotores e Transladores e Translações Finais	90

Sumário

Introdução	15
1 Estudo sobre a Geometria de Proteínas	17
1.1 Modelagem do problema	18
2 Movimentos Rígidos no Espaço	25
2.1 Rotações usando ângulos de Euler	26
2.2 Rotações em torno de um eixo arbitrário	29
2.3 Rotações com Quatérnios	32
2.4 Operador de um Movimento Rígido no Espaço Homogêneo	37
3 Atualização de Coordenadas Cartesianas	39
3.1 Alterações em ângulos de torção e distâncias entre pontos	39
3.2 Métodos de Atualização de Coordenadas Cartesianas	43
3.3 Testes Computacionais	54
3.4 Exemplo detalhado	57
4 Álgebra Geométrica Conforme	62
4.1 Álgebra Geométrica	62
4.1.1 Produtos Interno e Externo	63
4.1.2 Produto Geométrico	64
4.1.3 Rotores	68
4.2 Modelo Conforme	71
4.3 Álgebra Geométrica no Modelo Conforme	75
4.3.1 Rotores	76
4.3.2 Transladadores	77
4.3.3 Motores	79
5 Atualização de Coordenadas Conformes	81
5.1 Métodos de Atualização de Coordenadas Conformes	82
5.2 Testes Computacionais	91
5.3 Exemplo detalhado	96
6 Considerações Finais	103
 REFERÊNCIAS	 105

Introdução

Relacionadas com praticamente todas as funções de um sistema biológico, desde a atuação na defesa do organismo até o transporte de substâncias e a transmissão de impulsos nervosos, as proteínas são macromoléculas extremamente importantes para os organismos vivos. Compreender sua função passa intrinsecamente por conhecer seu formato tridimensional. Este, inclusive, é ponto de muito interesse da indústria farmacêutica, na qual a atualização da estrutura 3D de uma molécula de proteína é fundamental para o processo de desenvolvimento de novos medicamentos (LAVOR et al., 2017).

É comum em Bioquímica representar as posições relativas entre os átomos de uma proteína por meio das **coordenadas internas**, formadas por um conjunto de comprimentos de ligação (entre cada dupla de átomos consecutivos), ângulos de ligação (entre cada trio de átomos consecutivos) e ângulos de torção (a cada 4 átomos consecutivos). Apesar de não ser um sistema trivial, ele é mais adequado para lidar com mudanças conformacionais das moléculas de proteínas (ZHANG; KAVRAKI, 2002; CHOI, 2006), fato que está ligado diretamente à necessidade de se calcular atualizações das estruturas 3D.

Por outro lado, estão as tradicionais **coordenadas cartesianas**. Com elas, podemos representar os átomos de uma molécula de proteína por meio de pontos no espaço. Esse sistema também é útil em diversos processos de química computacional, como por exemplo, funções de energia e centros de massa (CAMARGO, 2021).

Outra alternativa para representar as coordenadas dos pontos que representam os átomos de uma proteína é fazer uso das **coordenadas conformes** (CAMARGO, 2021), que surgem quando trabalhamos com a Álgebra Geométrica Conforme (AGC). A Álgebra Geométrica, em si, é um ramo relativamente novo da Matemática que se constitui como uma poderosa ferramenta para solucionar problemas geométricos. Ao adotar o Modelo Conforme, trabalhamos os objetos matemáticos num espaço de 5 dimensões para que as interpretações geométricas sejam colhidas no $\mathbb{R}^3$. Para isso, duas novas coordenadas são incluídas, e_0 e e_∞ , que serão melhor explicadas posteriormente. Fato é que as coordenadas conformes encapsulam as coordenadas cartesianas, de modo que o espaço tridimensional aparece como um subconjunto desse outro. Isso estabelece uma forte relação entre os dois sistemas e é fácil o trânsito entre eles, tanto na ida, quanto na volta.

Por esse motivo, e porque a funcionalidade do sistema cartesiano também se encontra no conforme, temos o objetivo de estudar como as coordenadas conformes podem ser usadas para simular a atualização da estrutura 3D de proteínas devido às mudanças conformacionais resultantes de alterações em dados específicos das coordenadas internas.

Esse fato também inclui o objetivo de utilizar um pacote específico de Álgebra Geométrica Conforme, encontrado em (ORTIZ-DURAN; ARAGON, 2017), para tratar os elementos e ferramentas que surgem ao assumir essa abordagem.

Para que comparações sejam feitas entre o tratamento do problema com a AGC e como isso é feito classicamente, também é importante apresentar os principais métodos para atualização de coordenadas cartesianas a partir de alterações nas coordenadas internas.

No capítulo 1, apresentamos uma introdução ao tema das proteínas e suas estruturas 3D, bem como também explicamos melhor e modelamos o problema em questão.

No capítulo 2, há uma fundamentação teórica clássica a respeito de rotações e translações, muito importantes na manipulação das coordenadas cartesianas, de modo que cheguemos aos métodos de atualização de estruturas 3D com uma bagagem suficiente para o pleno entendimento.

No capítulo 3, trabalhamos com os métodos clássicos para atualização de estruturas 3D de proteínas, levando em conta as coordenadas cartesianas dos pontos que representam átomos. Nesse capítulo, também apresentamos alguns testes computacionais e um exemplo detalhado para uma aplicação hipotética.

Na sequência, o capítulo 4 fica encarregado de apresentar os fundamentos de Álgebra Geométrica que serão úteis para o tratamento do nosso problema.

Por fim, o capítulo 5 utiliza-se de todo o ferramental proposto pela Álgebra Geométrica Conforme para realizar o tratamento do problema de atualização de estruturas 3D de proteínas.

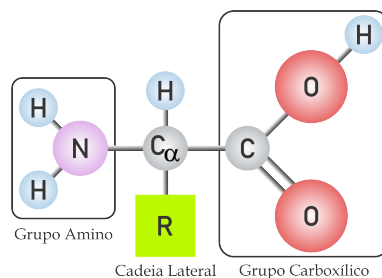
1 Estudo sobre a Geometria de Proteínas

Proteínas são macromoléculas extremamente importantes para os organismos vivos. Elas estão presentes em diversos processos biológicos essenciais, como no transporte de substâncias, na transmissão de impulsos nervosos, na defesa do organismo, na regulação do metabolismo, entre muitos outros, além de serem os principais constituintes de músculos, cabelos, unhas e diversos materiais biológicos (NELSON; COX, 2008).

O termo “proteína” se deriva da palavra grega “proteios”, que significa “o que está à frente”, ou “primeiro”, “primitivo”. Não à toa, são também conhecidas como as moléculas da vida. Mais precisamente, podemos dizer que as proteínas são macromoléculas originadas a partir da combinação de vários tipos de aminoácidos por meio de ligações peptídicas (NELSON; COX, 2008). Na composição de cada aminoácido estão os átomos, que se conectam por meio de ligações covalentes, nas quais há um compartilhamento de elétrons entre os pares.

Os aminoácidos, por sua vez, são moléculas pequenas compostas por um carbono central (também chamado carbono α e denotado por C_α), um átomo de hidrogênio (H), um grupo amino (H_2N), um ácido carboxílico ($COOH$) e uma cadeia lateral, geralmente denotada por R . É justamente a cadeia lateral, que é um grupo variável, que difere os 20 tipos de aminoácidos existentes. Por conta disso, eles compartilham de uma mesma estrutura fundamental, ilustrada na Figura 1. Mais informações e detalhes sobre aminoácidos e proteínas podem ser encontrados em (NELSON; COX, 2008).

Figura 1 – Estrutura fundamental dos aminoácidos.



Fonte: Elaborada pelo autor.

Já falamos sobre o quão importantes são as proteínas para os organismos vivos e seus processos biológicos. Esse fato já é notório há muito tempo, mas foi apenas na década de 50 do século XX que ficou claro que, além disso, há uma relação intrínseca entre a estrutura 3D de uma molécula em geral e a função que ela exerce. Fato esse que se estende, claro, às proteínas. Essa descoberta, inicialmente relacionada à estrutura 3D da molécula de DNA, rendeu o Prêmio Nobel de Medicina de 1962 aos cientistas James

Watson, Maurice Wilkins e Francis Crick ([LAVOR et al., 2017](#)).

Além da extrema importância das proteínas para os organismos vivos, há um motivo especial para que a elas dediquemos maior atenção do que às moléculas em geral. Os genes, que são um segmento de uma molécula de DNA, dão origem a um RNA mensageiro que sai do núcleo das células e orienta a produção de determinadas proteínas nos citoplasmas. O processo de transcrição dessas mensagens já é bem entendido, mas há muito a se investigar sobre o mecanismo para a formação da estrutura tridimensional de proteínas a partir de uma sequência de aminoácidos. O problema que estuda esse processo de enovelamento (ou dobramento) de proteínas é conhecido como *Protein Folding Problem* ([DILL et al., 2008](#)).

Ainda há outros problemas em que é extremamente importante determinar a estrutura 3D de proteínas, além de lidar com suas atualizações, entre eles: Dinâmica Molecular, em que se estudam métodos de simulação dos movimentos físicos dos átomos que a compõem; e Docking, que busca o acoplamento “perfeito” entre duas moléculas ([ZHANG; KAVRAKI, 2002](#)). Portanto, a atualização de estruturas 3D de proteínas é um assunto de muito interesse da indústria farmacêutica para o processo de desenvolvimento de novos medicamentos.

Em particular, por vislumbrar uma continuidade em pesquisas sobre o assunto, o presente trabalho também tem o objetivo de apresentar, de forma clara, os principais métodos para atualização das estruturas 3D, em que o entendimento e o domínio destes é essencial para problemas em que é necessário calcular várias atualizações a partir de uma mesma estrutura.

1.1 Modelagem do problema

Estudar a geometria de proteínas não é uma tarefa fácil, dada a complexidade de tudo que a envolve. Dessa forma, é justo estabelecer uma modelagem para o problema que facilite o tratamento matemático, sem que haja perda de suas características principais. Para isso, vamos introduzir algumas hipóteses e considerações e explicá-las, à medida em que também tornaremos mais claro o problema que será abordado no trabalho. Para começar:

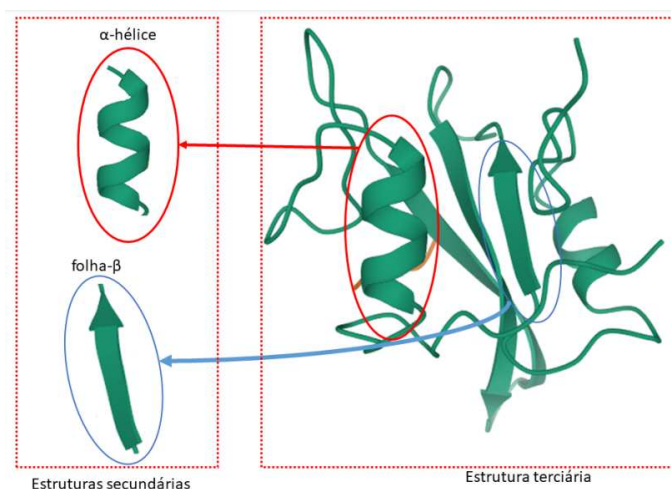
Consideração 1. *Os átomos serão representados por pontos no espaço.*

De fato, as proteínas são macromoléculas e, naturalmente, são constituídas por uma certa quantidade de átomos dispostos espacialmente. Pensando nisso, apesar de toda complexidade que envolve a figura de um átomo, faz sentido representá-los como pontos no espaço. Sendo assim, esse elemento será a base do modelo matemático que vamos tratar.

Consideração 2. *A cadeia principal é um grande indicativo da estrutura 3D de proteínas.*

A estrutura geral de uma proteína pode ser dividida em quatro partes: estruturas primária, secundária, terciária e quaternária. A estrutura primária corresponde à sequência de aminoácidos que compõem a proteína; a secundária está relacionada a padrões regulares que se repetem ao longo da estrutura molecular devido à atração entre átomos que não estão ligados covalentemente, ou seja, não interagem com troca de elétrons. Como resultado disso, surgem predominantemente duas conformações: as α -hélices, que assumem um formato espiral e que se assemelham a uma mola; e as folhas- β , que se parecem com setas e que geralmente se configuram lado a lado, mas em sentidos diferentes. A Figura 2 ilustra isso.

Figura 2 – Estruturas secundária e terciária de uma molécula de tirosina quinase LCK.



Fonte: (CAMARGO, 2021).

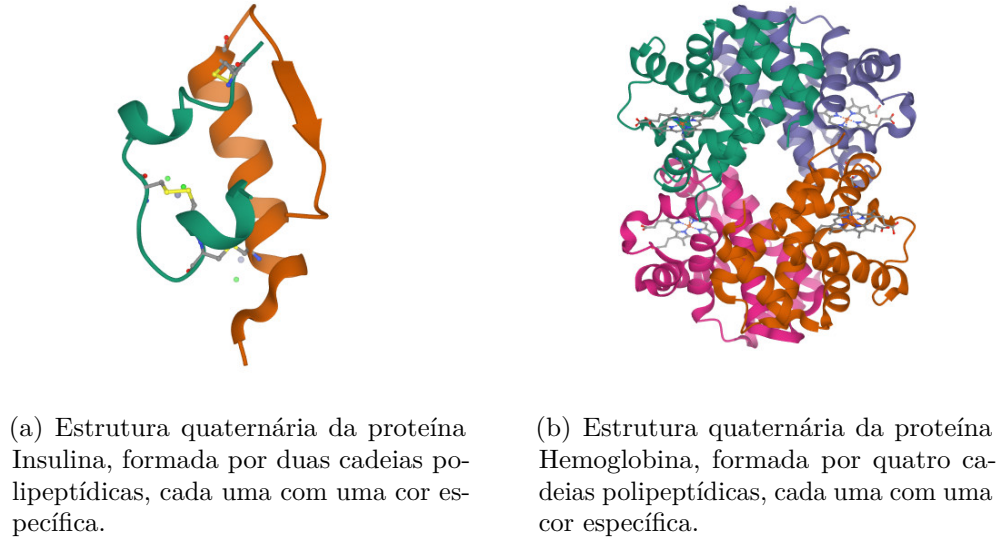
Também na Figura 2 é possível ver um exemplo da estrutura terciária. Esta é caracterizada pelo arranjo tridimensional que assume a cadeia polipeptídica da molécula de proteína, ou seja, o formato tridimensional resultado da composição de toda a sequência de aminoácidos presentes. Essa estrutura é a que mais importa, porque está diretamente relacionada à função que a proteína exerce.

Eventualmente, uma proteína pode ser constituída por duas ou mais cadeias polipeptídicas. Quando isso acontece, temos a estrutura quaternária. Dois exemplos estão na Figura 3.

Como dissemos, a estrutura que mais nos importa nesse trabalho é a terciária, que é o formato tridimensional que a molécula de proteína assume após a combinação de toda a sequência de aminoácidos. Essa combinação se por meio de ligações peptídicas, que podem ser vistas como um tipo específico de ligação covalente. Na verdade, são unidos os resíduos de aminoácidos, porque ocorre a perda de uma molécula de água quando um aminoácido é ligado a outro (NELSON; COX, 2008). Isso está ilustrado na Figura 4.

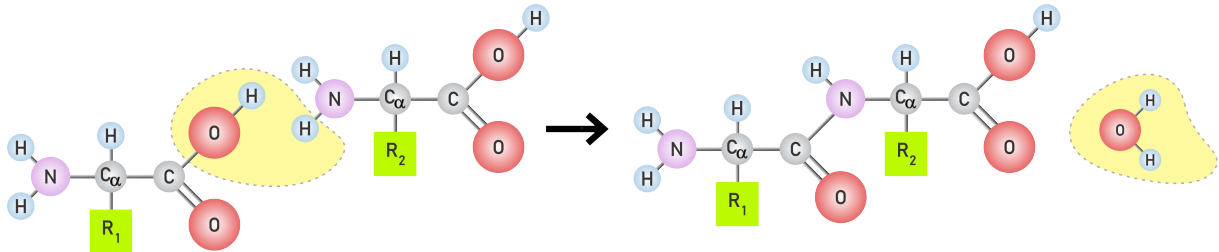
Note, também na Figura 4, que o resultado da ligação peptídica entre dois resíduos de aminoácidos é uma molécula em que se pode perceber um certo padrão: há uma

Figura 3 – Exemplo de duas moléculas de proteínas, cujas imagens foram extraídas do *Protein Data Bank* (PDB), grande base de dados de estruturas 3D de proteínas.



Fonte: (PDB ID: 1ZNI) e (PDB ID: 2QSP), respectivamente.

Figura 4 – Combinação de dois aminoácidos por meio de uma ligação peptídica, causando a perda de uma molécula de água.



Fonte: Elaborada pelo autor.

repetição na sequência dos átomos N , C_α e C . Esse padrão persiste se forem adicionados novos aminoácidos à estrutura. Esses átomos constituem justamente o que chamamos de cadeia principal de uma molécula de proteína, também conhecida como *backbone*. Na verdade, essa cadeia principal já dá uma boa estimativa da estrutura 3D da proteína. Por isso, a determinação dessa hipótese.

Consideração 3. *Existe um ordenamento dos átomos da cadeia principal.*

Do que foi abordado na hipótese anterior, pôde-se perceber que, na escala dos aminoácidos, há uma ordem dos átomos que compõem a cadeia principal: N , C_α e C . Mas além disso, pensando na proteína como um todo, vamos assumir uma ordem dos aminoácidos (e, conseqüentemente, de todos os átomos da cadeia principal da proteína). Dessa forma, como a base do tratamento matemático que vamos fazer são os pontos, criaremos uma

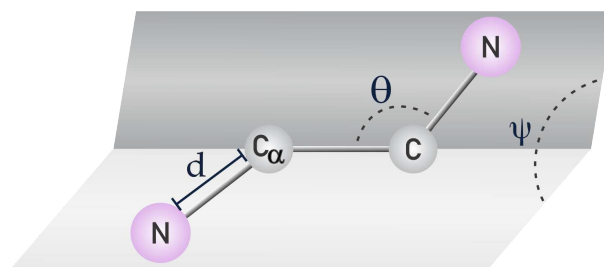
correspondência entre a sequência de átomos $N^1, C_\alpha^1, C^1, N^2, C_\alpha^2, C^2, \dots, N^{n/3}, C_\alpha^{n/3}, C^{n/3}$ e os pontos $x_1, x_2, \dots, x_n$, dispostos espacialmente.

Consideração 4. *A conversão de coordenadas internas para cartesianas (ou conformes) já está realizada.*

Em Bioquímica, é comum o uso de *coordenadas internas* para representar as conformações moleculares, isto é, as posições relativas entre os átomos. Por usar propriedades geométricas das ligações químicas, esse tipo de representação estabelece uma relação entre átomos subsequentes e, por isso, se torna mais adequado para descrever a estrutura 3D de proteínas. Além disso, derivações de uma conformação molecular são dadas naturalmente por esse sistema de representação.

As coordenadas internas de uma molécula são dadas por um conjunto de: *distâncias de ligação*, que são o comprimento das ligações covalentes (aquelas entre átomos consecutivos, em que há troca de elétrons); *ângulos de ligação (ou ângulos planos)*, formados por três átomos consecutivos; e *ângulos de torção*, formados por quatro átomos em sequência. A Figura 5 ilustra cada um desses componentes.

Figura 5 – Ilustração das coordenadas internas de uma molécula, em que d é a distância de ligação, θ o ângulo de ligação e ψ o ângulo de torção.



Fonte: Elaborada pelo autor.

Por outro lado, estão as tradicionais *coordenadas cartesianas*. Com elas, podemos representar os átomos de uma molécula por meio de pontos no $\mathbb{R}^3$. Isso é muito útil em algumas situações, como para lidar com centros de massa de moléculas, além de funções de energia e score de acoplamento, como relatado em (CHYS; CHAC6N, 2012; CAMARGO, 2021).

Além disso, existem as *coordenadas conformes*, menos conhecidas. A funcionalidade do sistema cartesiano se encontra também no conforme, pois passamos a adotar um espaço de 5 dimensões, de modo que o espaço tridimensional aparece como um subconjunto deste outro. Claro, isso estabelece uma forte relação entre esses dois sistemas e é fácil o trânsito entre eles, tanto na ida, quanto na volta.

Dada a importância de cada um dos sistemas de representação citados, é importante saber transitar entre eles. Como dito, entre coordenadas cartesianas e conformes,

não há dificuldade. Passar de qualquer uma delas para as coordenadas internas também não é difícil. Entretanto, o caminho inverso não é tão trivial. Apesar disso, vários estudos sobre a conversão de coordenadas internas já foram feitos nesse sentido, donde surgiram métodos eficazes para a solução desse problema. Exemplos são (THOMPSON, 1967; CHYS; CHACÓN, 2012; CAMARGO, 2021).

Um problema semelhante está na atualização da estrutura 3D de proteínas. Por mais que as coordenadas internas descrevam naturalmente as derivações de conformações moleculares, ainda é interessante convertê-las para as coordenadas cartesianas (ou conformes). Não é nada viável que a cada alteração nas coordenadas internas tenha-se que calcular novamente a estrutura 3D da proteína desde o começo, repetindo todo o processo de conversão. Esse problema é muito importante porque a existência de métodos eficientes para calcular derivações de conformações moleculares pode impactar nos processos de busca conformacional, de minimização de energia, além de serem úteis em aplicações em que é necessário calcular várias derivações de conformações de grandes moléculas de proteína (ZHANG; KAVRAKI, 2002).

Apesar de que métodos de atualização de coordenadas cartesianas e conformes também sejam conhecidos na literatura, como encontrado em (ZHANG; KAVRAKI, 2002; CHOI, 2006; CAMARGO, 2021), o presente trabalho foca nesse problema por visualizar eventuais próximos passos, dado que, em vários processos de química computacional, é importante e necessário calcular diversas vezes atualizações de uma mesma estrutura 3D de proteína.

Por todos esses motivos, vamos sempre assumir que a conversão de coordenadas internas para cartesianas (ou conformes) foi devidamente realizada e, assim, nos preocuparemos apenas com a atualização da estrutura 3D de proteínas.

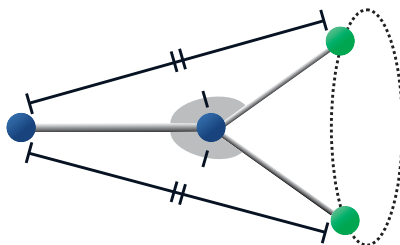
Hipótese 1. *Distâncias de ligação e ângulos de ligação serão considerados como dados a priori; a liberdade de variação está apenas nos ângulos de torção.*

Em geral, conhecimentos teóricos de química nos permitem estimar com precisão os comprimentos de ligações covalentes, além das distâncias entre átomos da cadeia principal não ligados covalentemente, mas com apenas um outro átomo entre eles. Essas ligações são bastante estáveis no sentido de que, apesar dos átomos não estarem totalmente estáticos, esses comprimentos não variam significativamente. Como consequência disso, os ângulos de ligação, formados por três átomos consecutivos, também podem ser estimados com precisão.

Dessa forma, dados três átomos consecutivos quaisquer, ainda que fixemos as coordenadas dos dois primeiros, não podemos localizar exatamente o terceiro nos valendo apenas das distâncias e ângulos de ligação porque, ao satisfazer esses dois critérios, o lugar geométrico para a localização do terceiro átomo será uma circunferência, conforme

ilustrado na Figura 6. A posição exata do ponto que representa esse átomo ainda vai depender do ângulo de torção, envolvendo a presença de mais um átomo.

Figura 6 – Ilustração da circunferência que surge como lugar geométrico para o posicionamento de um átomo.

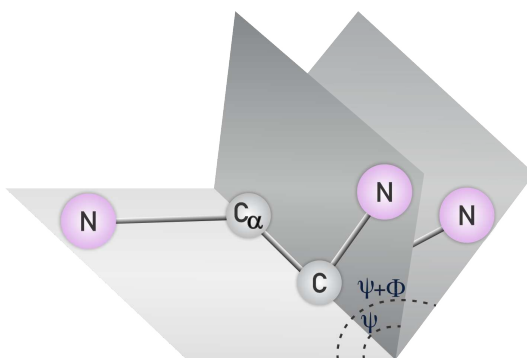


Fonte: Elaborada pelo autor.

O ângulo de torção, por sua vez, está mais suscetível a alterações porque está relacionado a 4 átomos consecutivos. A distância do primeiro para o quarto nessa sequência, diferentemente dos outros casos, não é dada por conhecimentos teóricos de química. Então, uma das alternativas para definir essa distância é recorrer a experimentos de RMN (Ressonância Magnética Nuclear), que contêm incertezas nos dados de saída. Como consequência, a determinação do ângulo de torção associado pode não ser tão exata, necessitando eventualmente de alterações.

Por conta disso tudo, vamos dar como fixas as distâncias entre átomos consecutivos e as medidas dos ângulos de ligação (ZHANG; KAVRAKI, 2002; LAVOR et al., 2017). Com isso, a liberdade de variação fica por conta dos ângulos de torção apenas, de modo que a Figura 7 ilustra como isso afeta a determinação da posição de um átomo.

Figura 7 – Ilustração da mudança de posição de um átomo devido a uma alteração realizada no ângulo de torção associado.



Fonte: Elaborada pelo autor.

Em (LAVOR et al., 2017), inclusive, ainda podemos encontrar valores relacionados a dados reais de moléculas. Quando formos simular aplicações dos métodos, vamos utilizá-las, tratando as distâncias de ligação iguais a $1,5\text{\AA}$, em que $\text{\AA}$ é a unidade de medida em escalas atômicas, e os ângulos de ligação iguais a $1,91$ radianos.

O estabelecimento de todas essas considerações e hipóteses nos permite delimitar o problema, mas ainda manter suas características mais importantes. Como consequência dessas definições, podemos resumi-lo com base nos dados de entrada que precisamos e nos dados de saída que queremos produzir:

- Dados de entrada:
 - quantidade n de átomos;
 - sequência de pontos $x_1, x_2, \dots, x_n$ que representam os átomos;
 - ângulos de torção que precisam ser alterados;
 - alterações para os respectivos ângulos de torção.
- Dados de saída:
 - pontos atualizados $x'_1, x'_2, \dots, x'_n$.

Cabe ressaltar que, como esse é o problema de atualização da estrutura, os dados de entrada estão relacionados aos dados de saída do problema de conversão de coordenadas. Com isso, precisamos das coordenadas cartesianas ou conformes da estrutura 3D original para, só então, atualizá-la.

2 Movimentos Rígidos no Espaço

No nosso problema de aplicação, rotações e translações são ferramentas fundamentais. Entendê-las claramente é de suma importância para saber o que fazem, como fazem e por que as usamos. A fim de cumprir esse objetivo, para chegar a um nível que seja útil para esse trabalho, o presente capítulo será destinado. Nele, estaremos sempre nos referindo à atuação no $\mathbb{R}^3$, a menos que se diga o contrário.

Por definição, uma rotação no espaço é uma transformação linear e ortogonal do $\mathbb{R}^3$ que preserva origem, distâncias e orientação. Geometricamente, é um deslocamento circular de um objeto em torno de um eixo, mantendo a origem fixa. Aqui, vamos adotar a convenção de entender a rotação como um giro no sentido anti-horário para ângulos de rotação positivos. Essa transformação pertence ao Grupo de Rotação 3D, representado simbolicamente por $SO(3)$ (JACOBSON, 1985). Uma ilustração da rotação está presente na Figura 8.

A translação no espaço, por sua vez, é uma transformação do $\mathbb{R}^3$ que não é linear e que pode ser vista como um deslocamento de um objeto resultante da adição de um vetor. Diferentemente da rotação, a translação não mantém a origem fixa e, portanto, preserva apenas distâncias e orientação. A translação também está ilustrada na Figura 8.

Figura 8 – Ilustração da rotação e da translação no espaço.



(a) Representação da rotação do vetor v em torno do eixo de rotação, gerando v' .

(b) Representação da translação do vetor v a partir do vetor de translação t , gerando v' .

Fonte: Elaborado pelo autor.

O nome do capítulo se justifica pelo fato de que falar em transformações do $\mathbb{R}^3$ que preservam distâncias e orientação é falar sobre Movimentos Rígidos no Espaço, já que estes podem ser entendidos como uma isometria que preserva orientação. A isometria é justamente uma transformação que preserva distâncias (LIMA, 2008), i.e., é uma função $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ tal que, para quaisquer $u, v \in \mathbb{R}^3$,

$$\|f(u) - f(v)\| = \|u - v\|, \quad (2.1)$$

em que, aqui e em todo o capítulo, estamos assumindo que $\|\cdot\|$ é a norma euclidiana.

Portanto, em geral, um movimento rígido no espaço é um deslocamento dado pela composição de uma rotação com uma translação (BOTTEMA, 1990). Analiticamente, tal movimento é representado pela expressão

$$f(v) = Rv + t, \quad (2.2)$$

em que $R \in \mathbb{R}^{3 \times 3}$ é uma matriz ortogonal, que representa a rotação, e $t \in \mathbb{R}^3$ é um vetor de translação (BOTTEMA, 1990). A ortogonalidade dessa matriz implica que R tem inversa e é dada por $R^{-1} = R^T$; além disso, vale que $\|f(u) - f(v)\| = \|Ru - Rv\| = \|u - v\|$. Desse modo, o movimento rígido dado pela expressão (2.2) de fato é uma isometria (LIMA, 2008).

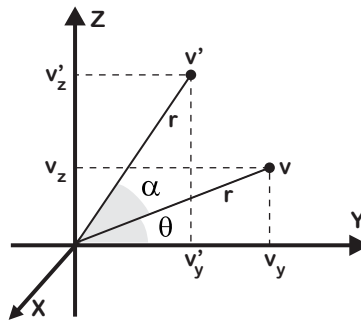
Até esse ponto, não há muito mais o que dizer sobre a translação além do fato de que a enxergamos como um deslocamento feito a partir da adição de um vetor. Mas sobre a rotação, sim, há muito a se dizer, porque podemos representá-la de algumas maneiras diferentes. Sobre isso se encarregam as próximas três seções deste capítulo.

2.1 Rotações usando ângulos de Euler

De um certo modo, podemos começar pensando na rotação no espaço como uma extensão da ideia de rotação no plano. Isso faz sentido mediante a fixação de um eixo coordenado, que será visto como eixo de rotação. A coordenada relacionada a ele permanecerá inalterada quando formos rotacionar um vetor, enquanto a rotação em si acontecerá nos outros dois eixos.

Por exemplo, suponha que fixamos o eixo coordenado X . Seja v um vetor do $\mathbb{R}^3$ e v' o resultado da rotação de v em torno do eixo de rotação X . Ilustramos isso na Figura 9. Além do vetor, foram representados os ângulos θ , medido do eixo Y até v , e α , de v até v' . A letra r representa a norma do vetor v (e claro, também de v').

Figura 9 – Rotação em relação ao eixo X no espaço.



Fonte: Elaborado pelo autor.

Vamos buscar uma forma de escrever v' em função de v . Por conhecimentos de trigonometria, sabemos que $v_y = r \cos(\theta)$ e $v_z = r \sin(\theta)$. Consequentemente, também

temos que $v'_y = r \cos(\theta + \alpha)$ e $v'_z = r \sin(\theta + \alpha)$. Fazendo uso das identidades trigonométricas

$$\begin{aligned}\sin(\theta + \alpha) &= \sin(\theta)\cos(\alpha) + \cos(\theta)\sin(\alpha) \\ \cos(\theta + \alpha) &= \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha),\end{aligned}$$

temos que

$$\begin{aligned}v'_y &= r \cos(\theta + \alpha) \\ &= r[\cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha)] \\ &= [r \cos(\theta)]\cos(\alpha) - [r \sin(\theta)]\sin(\alpha) \\ &= v_y \cos(\alpha) - v_z \sin(\alpha)\end{aligned}$$

e

$$\begin{aligned}v'_z &= r \sin(\theta + \alpha) \\ &= r[\sin(\theta)\cos(\alpha) + \cos(\theta)\sin(\alpha)] \\ &= [r \sin(\theta)]\cos(\alpha) + [r \cos(\theta)]\sin(\alpha) \\ &= v_y \sin(\alpha) + v_z \cos(\alpha).\end{aligned}$$

Agora, podemos escrever as coordenadas do vetor v' em função das coordenadas do vetor v . Fazendo isso, surge a matriz de rotação que os relaciona ([GOLDSTEIN; POOLE; SAFKO, 2002](#)):

$$\begin{pmatrix} v'_x \\ v'_y \\ v'_z \end{pmatrix} = \begin{pmatrix} v_x \\ v_y \cos(\alpha) - v_z \sin(\alpha) \\ v_y \sin(\alpha) + v_z \cos(\alpha) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}.$$

Por um processo análogo, mas fixando o eixo Y e depois também o eixo Z, completamos o conjunto das matrizes de rotação em relação a cada um dos eixos coordenados por meio dos ângulos α, β e γ , respectivamente. Vamos nos referir a essas matrizes como **rotações elementares**. Em ordem, temos:

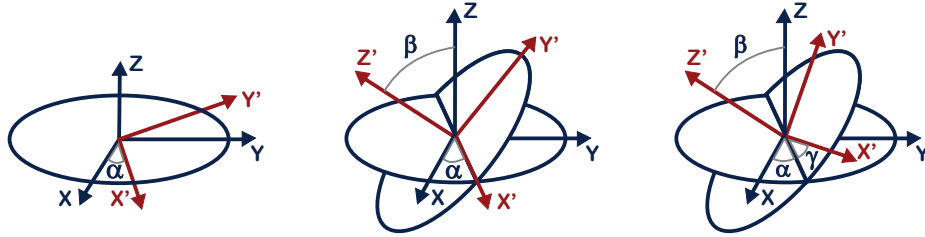
$$R_{\alpha,X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{pmatrix}, \quad R_{\beta,Y} = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \quad e \quad R_{\gamma,Z} = \begin{pmatrix} \cos(\gamma) & -\sin(\gamma) & 0 \\ \sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Nesse caso, os ângulos α, β e γ representam os conhecidos **Ângulos de Euler**. Herdando o nome do famoso matemático que os descreveu, eles constituem, com as matrizes acima, uma das formas mais básicas para se representar rotações no espaço. Dados três ângulos sucessivos, podemos construir uma matriz de rotação como produto de três rotações elementares. Também, qualquer rotação no espaço pode ser decomposta em função de três ângulos e das tais matrizes ([HENDERSON, 1977](#)).

Há de se ressaltar que não é única a ordem de composição das rotações usando os Ângulos de Euler! A notação X - Y - Z , por exemplo, representa rotações em torno do eixo X , depois Y e, por fim, Z . Entretanto, qualquer permutação dessa sequência que não repita o eixo de rotação consecutivamente representa uma forma de aplicar os Ângulos de Euler. X - X - Y , por exemplo, não representa. Nessa lógica, são 12 as possíveis sequências de rotações (HENDERSON, 1977). A depender da aplicação e da necessidade, uma ordem pode ser escolhida para maior utilidade.

Os Ângulos de Euler possuem uma vantagem de ser uma abordagem, de certa forma, intuitiva, porque é possível perceber a rotação como um todo decomposta em 3 rotações em torno dos eixos coordenados em determinada ordem. A Figura 10 ilustra um pouco disso.

Figura 10 – Sequência Z - X - Z usando os Ângulos de Euler. Na primeira imagem, uma rotação em torno do eixo Z por um ângulo α ; na segunda, em torno do eixo X' por um ângulo β ; por fim, em torno de Z' , com ângulo γ .



Fonte: Elaborado pelo autor.

Dessa forma, um vetor escrito como combinação linear da base canônica do $\mathbb{R}^3$, quando rotacionado, passa a ser escrito com os mesmos coeficientes daquela combinação linear, mas em relação aos vetores canônicos transformados. Abaixo, uma tradução algébrica disso:

$$\begin{aligned} (R_{\alpha,Z} \cdot R_{\beta,X} \cdot R_{\gamma,Z}) \cdot v &= R \cdot v \\ &= R \cdot [v_1(1, 0, 0)^T + v_2(0, 1, 0)^T + v_3(0, 0, 1)^T] \\ &= v_1[R \cdot (1, 0, 0)^T] + v_2[R \cdot (0, 1, 0)^T] + v_3[R \cdot (0, 0, 1)^T]. \end{aligned}$$

Entretanto, o uso dos Ângulos de Euler produz algumas inconveniências. Entre elas está o problema conhecido como *Gimbal Lock*, que causa um alinhamento entre dois eixos e reduz um grau de movimento para a rotação. Isso acontece quando o segundo Ângulo de Euler está em algum valor crítico (DIEBEL, 2006). Por exemplo, uma matriz de rotação construída como $R = R_{\alpha,Z} \cdot R_{0,X} \cdot R_{\beta,Z}$ tem a mesma atuação que $R_{\alpha+\beta,Z} \cdot R_{0,X} \cdot R_{0,Z}$.

Outro desconforto para o uso dos Ângulos de Euler surge quando queremos realizar uma rotação em torno de um eixo arbitrário que passa pela origem. De fato, as rotações elementares giram em torno dos eixos coordenados, mas o produto matricial delas

produz uma rotação que, por si só, gira em torno de algum eixo não especificado. Porém, não há uma relação clara entre esse eixo e os ângulos usados, bem como a sequência das rotações. Tentaremos contornar esse problema na próxima seção.

2.2 Rotações em torno de um eixo arbitrário

Vimos que, com os Ângulos de Euler, performamos uma composição de rotações elementares usando três ângulos dados. Na prática, quando fazemos essa composição, é como se estivéssemos realizando uma rotação em torno de um eixo arbitrário que passa pela origem, mas que, a priori, não conhecemos. Ou seja, não escolhemos o eixo de rotação; apenas definimos os ângulos e determinamos uma sequência de rotações elementares. Mas será possível definir previamente um eixo e um ângulo de rotação? Felizmente, sim! Para isso, não abandonaremos completamente as ideias da seção anterior, mas as adaptaremos.

A abordagem que vamos tratar agora se encarrega de realizar rotações em torno de um eixo arbitrário que passa pela origem. Para isso, precisamos definir um vetor unitário que seja colinear ao eixo de rotação. Então, sempre que dissermos que um vetor unitário é entendido como eixo de rotação, entenda-se que o primeiro foi tomado como um representante do segundo.

Primeiro, vamos mostrar a ideia central desse método e, em seguida, o explicaremos em mais detalhes. Com essa abordagem, uma rotação por um ângulo θ em torno de um eixo representado pelo vetor unitário u é descrita pelo seguinte procedimento:

- i) alinhar o vetor u ao eixo Z por meio de uma sequência de rotações apropriadas;
- ii) rotacionar em torno do eixo Z pelo ângulo θ ;
- iii) aplicar a inversa das rotações usadas no primeiro item.

Não necessariamente precisamos escolher o eixo Z para buscar esse alinhamento, mas essa é a versão mais usada. Além disso, todas essas rotações serão elementares.

Em geral, para tornar um eixo arbitrário coincidente com um eixo coordenado, basta rotacioná-lo em torno dos outros dois eixos coordenados. Por isso, atenderão ao objetivo do item i) uma composição de duas rotações: uma primeira em torno de X e uma outra em torno de Y. Ela será representada pelo produto $R_{\beta,Y} \cdot R_{\alpha,X}$. O item ii) será dado pela rotação $R_{\theta,Z}$, enquanto o terceiro item é igual a $(R_{\beta,Y} \cdot R_{\alpha,X})^{-1}$. Portanto, sabendo que a rotação é um operador ortogonal, podemos resumir esse processo em uma só matriz, denominada R :

$$\begin{aligned} R &= R_{\alpha,X}^{-1} \cdot R_{\beta,Y}^{-1} \cdot R_{\theta,Z} \cdot R_{\beta,Y} \cdot R_{\alpha,X} \\ &= R_{\alpha,X}^T \cdot R_{\beta,Y}^T \cdot R_{\theta,Z} \cdot R_{\beta,Y} \cdot R_{\alpha,X}. \end{aligned} \tag{2.3}$$

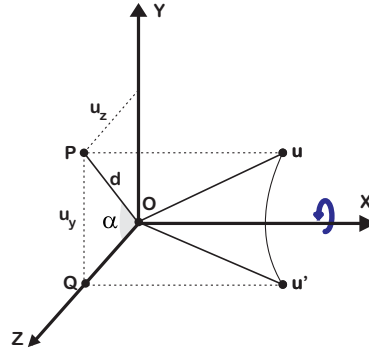
Note que a ordem dessas matrizes importa. A leitura desse produto da direita para a esquerda coincide exatamente com a sequência dos itens enunciados. Isso faz sentido, já que ao rotacionar um vetor v , estamos calculando o produto $R \cdot v$.

Seria interessante que a matriz R da equação (2.3) estivesse apenas em função do ângulo θ e da escolha de um vetor unitário u , visto como eixo de rotação. Para cumprir isso, as matrizes $R_{\beta,Y}$ e $R_{\alpha,X}$ podem ficar em função de u , sem que seja necessário calcular expressamente os valores de α e de β . Isso pode ser feito com conhecimentos de trigonometria no triângulo retângulo. Deixaremos isso mais claro.

Considere a figura abaixo. Queremos rotacionar u em torno do eixo X de modo que o vetor resultante, denominado u' , esteja localizado no plano XZ. Essa rotação é caracterizada por um ângulo α pertencente ao triângulo retângulo ΔOPQ representado na figura. Sabendo que $d = \sqrt{u_y^2 + u_z^2}$, temos que

$$\cos(\alpha) = \frac{u_z}{d} \quad e \quad \sen(\alpha) = \frac{u_y}{d}.$$

Figura 11 – Rotação em torno do eixo X.



Fonte: Elaborado pelo autor.

Note que essa primeira rotação está sendo executada no sentido anti-horário, como deve ser. Portanto, temos a matriz $R_{\alpha,X}$:

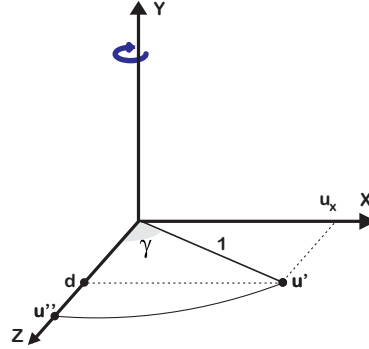
$$R_{\alpha,X} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & u_z/d & -u_y/d \\ 0 & u_y/d & u_z/d \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & u_z/\sqrt{u_y^2 + u_z^2} & -u_y/\sqrt{u_y^2 + u_z^2} \\ 0 & u_y/\sqrt{u_y^2 + u_z^2} & u_z/\sqrt{u_y^2 + u_z^2} \end{pmatrix}.$$

Feito isso, a situação resultante é a que está representada na figura abaixo. Analogamente, os valores do seno e do cosseno de γ são tais que

$$\cos(\gamma) = d \quad e \quad \sen(\gamma) = u_x.$$

Note que a rotação em torno do eixo Y pelo ângulo γ no sentido horário cumprirá o objetivo de levar u' até u'' . Entretanto, como adotamos a convenção de trabalhar com a

Figura 12 – Rotação em torno do eixo Y.



Fonte: Elaborado pelo autor.

rotação no sentido anti-horário, cumpriremos o mesmo objetivo ao fazer uma rotação nesse sentido em torno de Y com um ângulo $\beta = -\gamma$. Note que $\cos(\beta) = \cos(-\gamma) = \cos(\gamma) = d$ e $\sin(\beta) = \sin(-\gamma) = -\sin(\gamma) = -u_x$. Portanto, a matriz de rotação que usaremos em torno de Y é definida assim:

$$R_{\beta,Y} = \begin{pmatrix} d & 0 & -u_x \\ 0 & 1 & 0 \\ u_x & 0 & d \end{pmatrix} = \begin{pmatrix} \sqrt{u_y^2 + u_z^2} & 0 & -u_x \\ 0 & 1 & 0 \\ u_x & 0 & \sqrt{u_y^2 + u_z^2} \end{pmatrix}.$$

Da equação (2.3), já temos as matrizes $R_{\alpha,X}$ e $R_{\beta,Y}$ escritas em função de u . Além disso, as inversas dessas matrizes são dadas naturalmente pelo fato de que são ortogonais. Dessa forma, resta a matriz $R_{\theta,Z}$, que será dada por

$$R_{\theta,Z} = \begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

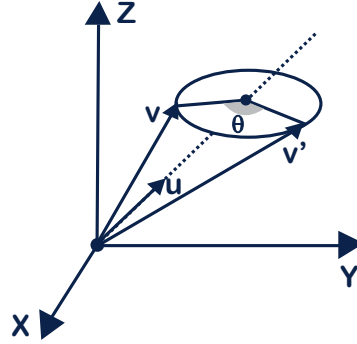
Finalmente, ao calcular o produto matricial da equação (2.3), com algumas simplificações e maiores detalhes que podem ser encontrados em (KOVÁCS, 2012; ROGERS; ADAMS, 1990), temos:

$$R = \begin{pmatrix} \cos\theta + u_x^2(1 - \cos\theta) & u_x u_y(1 - \cos\theta) - u_z \sin\theta & u_x u_z(1 - \cos\theta) + u_y \sin\theta \\ u_x u_y(1 - \cos\theta) + u_z \sin\theta & \cos\theta + u_y^2(1 - \cos\theta) & u_y u_z(1 - \cos\theta) - u_x \sin\theta \\ u_x u_z(1 - \cos\theta) - u_y \sin\theta & u_y u_z(1 - \cos\theta) + u_x \sin\theta & \cos\theta + u_z^2(1 - \cos\theta) \end{pmatrix}. \quad (2.4)$$

Dessa forma, se torna possível realizar rotações em torno de eixos arbitrários que passam pela origem. Na Figura 13, isso está ilustrado.

Na verdade, a matriz (2.4) é também uma consequência da **Fórmula de Rotação de Rodrigues** (RODRIGUES, 1840) que descreve a rotação de um vetor a em

Figura 13 – Rotação do vetor v por um ângulo θ em torno do eixo de rotação representado por u , que é unitário. Como resultado, temos o vetor v' .



Fonte: Elaborado pelo autor.

torno de u por um ângulo θ . A fórmula é a seguinte:

$$a' = a \cos \theta + (u \times a) \sin \theta + u(a \cdot u)(1 - \cos \theta). \quad (2.5)$$

De fato, fazendo $a' = (a'_x, a'_y, a'_z)$, $a = (a_x, a_y, a_z)$ e $u = (u_x, u_y, u_z)$, e com algumas simplificações, temos que

$$\begin{aligned} \begin{pmatrix} a'_x \\ a'_y \\ a'_z \end{pmatrix} &= \cos \theta \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + \sin \theta \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} + (1 - \cos \theta) \begin{pmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{pmatrix} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \\ &= \left[\cos \theta \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \sin \theta \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} + (1 - \cos \theta) \begin{pmatrix} u_x^2 & u_x u_y & u_x u_z \\ u_x u_y & u_y^2 & u_y u_z \\ u_x u_z & u_y u_z & u_z^2 \end{pmatrix} \right] \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}. \end{aligned}$$

Efetuada os produtos e as somas acima, encontramos novamente a matriz R de outrora. A fórmula (2.5) expressa de forma resumida e concisa a rotação de um vetor. Entretanto, a representação matricial será nosso foco pela facilidade em que expressa a composição de rotações, que é dada simplesmente pelo produto das matrizes.

Portanto, agora temos uma matriz que performa uma rotação em torno de um eixo arbitrário que passa pela origem. Para construí-la, somente um ângulo θ e um vetor unitário u são necessários. De imediato, um ganho intuitivo e prático: sabemos diretamente em relação a que eixo um vetor está sendo rotacionado.

2.3 Rotações com Quatérnios

Na seção anterior, fizemos uma adaptação à ideia da rotação com Ângulos de Euler para chegar a um método que performa rotações em torno de eixos arbitrários

passando pela origem. Mas essa não é a única maneira de realizar tal feito. Podemos usar uma nova entidade matemática que simplifica esse processo, tem um menor custo computacional e surgiu justamente para atender a demanda de realizar rotações tridimensionais: os quatérnios. Em toda essa seção, as propriedades e definições terão (KUIPERS, 1998) como embasamento principal.

Em meados do século XIX, quando os números complexos já eram utilizados para fazer rotações no plano, o matemático irlandês William Hamilton procurou obter um resultado análogo para o espaço a três dimensões. Hamilton percebeu que o elemento que procurava deveria ser uma quádrupla, ao invés de uma tripla como vinha tentando, e que deveria também abandonar a comutatividade para o produto.

Por ter surgido como uma extensão da ideia dos números complexos, os quatérnios herdam várias semelhanças com esses outros. A começar pela fórmula fundamental, em que são três as unidades imaginárias:

$$i^2 = j^2 = k^2 = ijk = -1. \quad (2.6)$$

Com esses novos elementos, temos um espaço vetorial de dimensão quatro que abriga os quatérnios, cuja base é $\{1, i, j, k\}$. Em relação aos números complexos, perdemos a comutatividade da multiplicação, como dito. Isso pode ser visto nas demais operações base envolvendo i, j e k , que são as seguintes:

$$\begin{array}{lll} jk = i, & ki = j, & ij = k, \\ kj = -i, & ik = -j, & ji = -k. \end{array} \quad (2.7)$$

O quatérnio, em si, também tem uma parte real e outra complexa (ou mais comumente dita, vetorial). Representado por uma quádrupla, como falado anteriormente, assume a forma

$$q = q_0 + q_1i + q_2j + q_3k, \quad (2.8)$$

em que q_0, q_1, q_2 e q_3 são escalares. Se $q_0 = 0$, resta apenas a parte vetorial e o que temos é justamente um vetor do $\mathbb{R}^3$. Ou seja, os quatérnios mapeiam todo o espaço tridimensional. Aos vetores do $\mathbb{R}^3$, nesse contexto, chamamos de quatérnios puros.

Há também outras representações para o quatérnio q que podem ser úteis de acordo com a conveniência. Duas delas são as seguintes,

$$q = (q_0, q_1, q_2, q_3) \quad e \quad q = q_0 + \mathbf{q},$$

em que $\mathbf{q}$ é a parte vetorial $q_1i + q_2j + q_3k$.

Ainda há alguns conceitos importantes que auxiliarão o desenvolvimento da ideia da rotação com quatérnios. Primeiro, dados $p = p_0 + p_1i + p_2j + p_3k$ e $q = q_0 + q_1i + q_2j + q_3k$,

a adição entre eles é dada pela soma de cada parte correspondente de p e q , i. e.,

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \quad (2.9)$$

O produto entre quatérnios não é tão trivial assim. Aplicando as propriedades distributivas e associativas e usando as operações base apresentadas acima, temos, de forma resumida, que

$$pq = (p_0 + \mathbf{p})(q_0 + \mathbf{q}) = p_0q_0 - \mathbf{p} \cdot \mathbf{q} + p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}, \quad (2.10)$$

em que $p_0q_0 - \mathbf{p} \cdot \mathbf{q}$ é a parte escalar e $p_0\mathbf{q} + q_0\mathbf{p} + \mathbf{p} \times \mathbf{q}$ a parte vetorial.

Além disso, a ideia do conjugado de um quatérnio também não é muito diferente em relação aos números complexos: apenas tomamos o inverso aditivo de cada coeficiente associado à parte vetorial. Portanto, temos que

$$q^* = q_0 - \mathbf{q} = q_0 - q_1i - q_2j - q_3k. \quad (2.11)$$

Por fim, a norma de um quatérnio, denotada por $|q|$ e por vezes chamada de comprimento de q , é dada por

$$|q| = \sqrt{q^*q} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}. \quad (2.12)$$

Como consequência disso, podemos calcular o inverso de um quatérnio observando o fato de que $|q|^2 = q^*q$. Ou seja,

$$q^{-1}qq^* = q^*qq^{-1} = q^* \implies q^{-1} = \frac{q^*}{|q|^2}. \quad (2.13)$$

Agora, já temos o suficiente para seguir em frente. Até aqui, definimos os quatérnios, bem como algumas operações relacionadas e conceitos importantes. Entretanto, não mostramos a relação dessa nova entidade com a rotação no espaço. É natural, então, surgir a seguinte questão: será possível associar um ângulo a um quatérnio, assim como associamos a uma matriz de rotação?

Sabemos que, para todo ângulo β , vale a identidade $\cos^2\beta + \sin^2\beta = 1$. Vamos buscar algo parecido com os quatérnios. Considere, então, um quatérnio unitário escrito como $q = q_0 + \mathbf{q}$. Como sua norma é 1, temos que $q_0^2 + |\mathbf{q}|^2 = 1$. Logo, deve haver algum β tal que $q_0^2 = \cos^2\beta$ e $|\mathbf{q}|^2 = \sin^2\beta$. Portanto, podemos reescrever q como

$$q = \cos\beta + \mathbf{u}\sin\beta,$$

em que $\mathbf{u} = \frac{\mathbf{q}}{|\mathbf{q}|} = \frac{\mathbf{q}}{\sin\beta}$ é o vetor unitário que representa a direção de q . Note também que substituindo β por $-\beta$, temos $\cos(-\beta) + \mathbf{u}\sin(-\beta) = \cos\beta - \mathbf{u}\sin\beta = q^*$.

Como queremos trabalhar com rotações de vetores do $\mathbb{R}^3$, vamos considerar também um vetor a desse espaço. De fato, ele é um quatérnio puro e pode ser escrito como $a = 0 + \mathbf{a}$. Se a ideia da rotação com quatérnios for inteiramente semelhante à rotação com matrizes, faz sentido pensar que o produto qa gera o vetor a' , resultado da rotação de a em torno de u por um ângulo β . Mas não é bem assim, pois a' não resulta, em geral, em um quatérnio puro, como deveríamos ter. Ou seja,

$$\begin{aligned} a' &= qa \\ &= (\cos\beta + \mathbf{u}\operatorname{sen}\beta)(0 + \mathbf{a}) \\ &= -(\mathbf{u} \cdot \mathbf{a})\operatorname{sen}\beta + \mathbf{a}\cos\beta + (\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta. \end{aligned}$$

Na tentativa de anular o termo escalar desse resultado, vamos tentar pensar na rotação com quatérnios como um produto triplo $qaq^{-1} = qa q^*$ e ver o que surge como consequência. Usando a regra para o produto entre quatérnios, temos que

$$\begin{aligned} a' &= qa q^* \\ &= [-(\mathbf{u} \cdot \mathbf{a})\operatorname{sen}\beta + \mathbf{a}\cos\beta + (\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta](\cos\beta - \mathbf{u}\operatorname{sen}\beta) \\ &= -(\mathbf{u} \cdot \mathbf{a})\operatorname{sen}\beta\cos\beta + [\mathbf{a}\cos\beta + \operatorname{sen}\beta(\mathbf{u} \times \mathbf{a})] \cdot (\mathbf{u}\operatorname{sen}\beta) + [(\mathbf{u} \cdot \mathbf{a})\operatorname{sen}\beta](\mathbf{u}\operatorname{sen}\beta) + \\ &\quad + \cos\beta[\mathbf{a}\cos\beta + \operatorname{sen}\beta(\mathbf{u} \times \mathbf{a})] - [\mathbf{a}\cos\beta + \operatorname{sen}\beta(\mathbf{u} \times \mathbf{a})] \times (\mathbf{u}\operatorname{sen}\beta) \\ &= -(\mathbf{u} \cdot \mathbf{a})\operatorname{sen}\beta\cos\beta + (\mathbf{a} \cdot \mathbf{u})\operatorname{sen}\beta\cos\beta + [(\mathbf{u} \times \mathbf{a}) \cdot \mathbf{u}]\operatorname{sen}^2\beta + [(\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta + \\ &\quad + \mathbf{a}\cos^2\beta + (\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta\cos\beta - (\mathbf{a} \times \mathbf{u})\operatorname{sen}\beta\cos\beta - [(\mathbf{u} \times \mathbf{a}) \times \mathbf{u}]\operatorname{sen}^2\beta. \end{aligned}$$

Eliminando termos que se anulam, além do fato de que $(\mathbf{u} \times \mathbf{a}) \cdot \mathbf{u} = 0$, temos que

$$\begin{aligned} a' &= [(\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta + \mathbf{a}\cos^2\beta + (\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta\cos\beta - (\mathbf{a} \times \mathbf{u})\operatorname{sen}\beta\cos\beta - [(\mathbf{u} \times \mathbf{a}) \times \mathbf{u}]\operatorname{sen}^2\beta \\ &= \mathbf{a}\cos^2\beta + [(\mathbf{u} \times \mathbf{a}) - (\mathbf{a} \times \mathbf{u})]\operatorname{sen}\beta\cos\beta + [(\mathbf{u} \cdot \mathbf{a})\mathbf{u} - (\mathbf{u} \times \mathbf{a}) \times \mathbf{u}]\operatorname{sen}^2\beta \\ &= \mathbf{a}\cos^2\beta + 2(\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta\cos\beta + [(\mathbf{u} \cdot \mathbf{a})\mathbf{u} - (\mathbf{u} \times \mathbf{a}) \times \mathbf{u}]\operatorname{sen}^2\beta. \end{aligned}$$

Como $(r \times s) \times t = (r \cdot t)s - (r \cdot s)t$, então

$$\begin{aligned} [(\mathbf{u} \cdot \mathbf{a})\mathbf{u} - (\mathbf{u} \times \mathbf{a}) \times \mathbf{u}]\operatorname{sen}^2\beta &= [(\mathbf{u} \cdot \mathbf{a})\mathbf{u} - (\mathbf{u} \cdot \mathbf{u})\mathbf{a} + (\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta \\ &= [-\mathbf{a} + 2(\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta. \end{aligned}$$

Logo, voltando ao a' , temos que

$$\begin{aligned} a' &= \mathbf{a}\cos^2\beta + 2(\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta\cos\beta + [-\mathbf{a} + 2(\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta \\ &= \mathbf{a}(\cos^2\beta - \operatorname{sen}^2\beta) + 2(\mathbf{u} \times \mathbf{a})\operatorname{sen}\beta\cos\beta + [(\mathbf{u} \cdot \mathbf{a})\mathbf{u}]\operatorname{sen}^2\beta. \end{aligned}$$

Para finalizar, vamos usar as três identidades trigonométricas a seguir:

$$\cos^2\beta - \operatorname{sen}^2\beta = \cos(2\beta), \quad \operatorname{sen}\beta\cos\beta = \frac{\operatorname{sen}(2\beta)}{2} \quad e \quad \operatorname{sen}^2\beta = \frac{1 - \cos(2\beta)}{2}.$$

Com essas simplificações, temos que (KUIPERS, 1998):

$$a' = a \cos(2\beta) + (\mathbf{u} \times \mathbf{a}) \sin(2\beta) + [(\mathbf{u} \cdot \mathbf{a})\mathbf{u}](1 - \cos(2\beta)). \quad (2.14)$$

Não temos aqui outra coisa senão a fórmula de Rodrigues novamente! O resultado deu um quatérnio puro (ou, simplesmente, um vetor), como queríamos, e pode-se verificar que a' e a têm a mesma norma. Isso tudo nos indica que realmente os quatérnios estão relacionados à rotação de um vetor. A diferença dessa fórmula para a (2.5) está apenas no ângulo. Mas isso não é um problema, já que podemos fazer $\beta = \frac{\theta}{2}$. Inclusive, essa é a razão pela qual um quatérnio escrito como

$$q = \cos \frac{\theta}{2} + u \sin \frac{\theta}{2} \quad (2.15)$$

é utilizado para realizar a rotação de um vetor a em torno de um vetor unitário u por um ângulo θ , e não $\frac{\theta}{2}$, como se poderia supor pela simples visualização da expressão acima.

Agora precisamos definir a representação matricial da rotação com os quatérnios. Seja $q = q_0 + q_1i + q_2j + q_3k$. Com os vetores qiq^* , qjq^* e qkq^* alocados em cada uma das colunas de forma respectiva, temos a seguinte matriz (KUIPERS, 1998):

$$R = 2 \begin{pmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1q_2 - q_0q_3 & q_1q_3 + q_0q_2 \\ q_1q_2 + q_0q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_2q_3 - q_0q_1 \\ q_1q_3 - q_0q_2 & q_2q_3 + q_0q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{pmatrix}. \quad (2.16)$$

Essa é mais uma matriz que representa a rotação de um vetor em torno de um eixo arbitrário que passa pela origem por meio de um ângulo dado. É natural surgir a pergunta: se na seção anterior já tínhamos uma matriz que fazia isso, por que desenvolver a rotação com quatérnios para obter o mesmo resultado?

Uma explicação para isso está no custo computacional das duas abordagens. Calcular a matriz da equação (2.4) implica a necessidade de se realizar 15 multiplicações e 10 somas, enquanto que para a matriz da equação (2.16), são necessárias 10 multiplicações e 10 somas. Além disso, é comum no nosso problema termos que compor rotações. O produto matricial entre matrizes 3×3 , como as da equação (2.4), custa 27 multiplicações e 18 somas. A matriz de (2.16) também é 3×3 , mas o uso dos quatérnios nos dispõe outra possibilidade menos custosa: calcular o produto entre quatérnios e, por fim, fazer a conversão do resultado para a versão matricial. Para isso, são necessárias 16 multiplicações e 12 somas. Dessa forma, o que com os quatérnios custa 48 operações, com a abordagem da seção anterior custa 70 operações.

2.4 Operador de um Movimento Rígido no Espaço Homogêneo

Ao longo de todo esse capítulo, à medida em que avançamos de seção em seção, fomos progredindo também em ganhos: primeiro, as rotações aconteciam apenas em torno dos eixos X, Y e Z ; depois, conseguimos realizar rotações em torno de eixos arbitrários que passam pela origem; o advento dos quatérnios também representou uma evolução. Mas ainda gostaríamos de outra vantagem que seria muito útil para o estudo do problema que vamos abordar: realizar rotações em torno de eixos arbitrários que não passam pela origem.

As rotações que estudamos nesse capítulo, por si só, não nos trazem essa possibilidade, já que atuam apenas em torno de eixos que passam pela origem. Por isso, voltamos o nosso foco para a translação. Ela será útil para levar eixos e segmentos até a origem, onde a rotação será efetivada, e ainda será responsável por retornar esses elementos às suas devidas localizações. Esse processo será melhor especificado no capítulo seguinte.

Fato é que utilizaremos bastante as translações juntamente das rotações. Tanto é que iniciamos este capítulo falando sobre movimentos rígidos, compostos por essas duas transformações. Inclusive, é fácil perceber que o movimento rígido dado pela equação (2.2) não é linear. Seria interessante que fosse, para termos a rotação e a translação num mesmo operador. Há uma forma de fazer isso: usando o chamado Espaço Homogêneo, apresentado em detalhes em (KANATANI, 2015), com uma dimensão a mais, no qual o espaço tridimensional se encontra imerso.

Nesse novo espaço, um vetor $v = (v_1, v_2, v_3) \in \mathbb{R}^3$ ganha uma coordenada extra e passa a ser representado por $V = (v_1, v_2, v_3, 1)$. Por consequência, as matrizes que formos usar devem estar em $\mathbb{R}^{4 \times 4}$. Além disso, vamos também adaptar a rotação e a translação para o Espaço Homogêneo. Esta última, dada por $T(v) = v + t$ no $\mathbb{R}^3$, agora é representada matricialmente por

$$\begin{pmatrix} 1 & 0 & 0 & t_1 \\ 0 & 1 & 0 & t_2 \\ 0 & 0 & 1 & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ 1 \end{pmatrix} = \begin{pmatrix} v_1 + t_1 \\ v_2 + t_2 \\ v_3 + t_3 \\ 1 \end{pmatrix}.$$

Fazendo algo semelhante com a rotação, o movimento rígido dado pela equação (2.2) passa a ser representado no Espaço Homogêneo por uma só matriz:

$$M = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

de modo que os coeficientes r_{ij} são as entradas da matriz de rotação R .

De fato, a atuação de M sobre um vetor homogêneo V produz uma rotação seguida de uma translação, pois

$$\begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ 1 \end{pmatrix} = \begin{pmatrix} r_{11}v_1 + r_{12}v_2 + r_{13}v_3 + t_1 \\ r_{21}v_1 + r_{22}v_2 + r_{23}v_3 + t_2 \\ r_{31}v_1 + r_{32}v_2 + r_{33}v_3 + t_3 \\ 1 \end{pmatrix}.$$

Optando por simplicidade, podemos também reescrever a matriz M de forma condensada:

$$M = \begin{pmatrix} R & t \\ \bar{0} & 1 \end{pmatrix},$$

em que $R \in \mathbb{R}^{3 \times 3}$ representa a rotação, $t \in \mathbb{R}^3$ o vetor de translação e $\bar{0} = (0, 0, 0)^T$.

Com essa nova representação matricial, conseguimos elegância e praticidade. Não será necessário calcular a rotação e a translação separadamente. Só precisamos nos atentar ao trânsito entre as coordenadas tridimensionais e quadridimensionais. Além disso, a composição de dois movimentos rígidos será descrito por um simples produto matricial.

Portanto, obtemos uma forma indireta de realizar rotações em torno de eixos arbitrários que não passam pela origem, já que a translação ainda está presente na construção desse operador.

3 Atualização de Coordenadas Cartesianas

Como visto anteriormente no capítulo 1, as coordenadas internas são bastante usadas em bioquímica para descrever mudanças conformacionais de uma molécula porque o fazem de uma maneira bastante natural. De fato, qualquer alteração angular ou nas distâncias entre ligações covalentes produzirá, como consequência, uma nova estrutura molecular. Em especial, vamos nos limitar ao caso em que apenas os ângulos de torção estão sujeitos a alterações.

Vamos partir do pressuposto de que, dado um conjunto de coordenadas internas, a conversão para coordenadas cartesianas foi devidamente realizada. Entretanto, por ocasião de mudanças conformacionais, alguns ângulos de torção têm de ser atualizados. Gostaríamos de ter novamente as coordenadas cartesianas da nova estrutura que se formou. Diante disso, há duas possibilidades: ou ignoramos a conversão de coordenadas de outrora para realizar uma nova ou buscamos formas de atualizar as coordenadas cartesianas que tínhamos antes da mudança conformacional. A segunda opção é mais interessante e é o que estudaremos nesse capítulo.

Os métodos de atualização de coordenadas cartesianas que apresentaremos aqui também podem ser encontrados em (CHOI, 2006; CAMARGO, 2021). Além de apresentar e discutir os principais métodos da literatura para o problema de atualização de coordenadas, nosso objetivo nesse capítulo é também descrever em detalhes cada uma de suas etapas.

Antes disso, é importante saber que efeito uma alteração em um ângulo de torção causa às distâncias entre átomos. Isso será útil até para validar os resultados que cada método produz.

3.1 Alterações em ângulos de torção e distâncias entre pontos

Da conversão de coordenadas internas para coordenadas cartesianas, sabemos que um ponto que representa um átomo é construído a partir da expressão

$$x_k = x_1 + d_2 R_{[2]} e_1 + d_3 R_{[3]} e_1 + \cdots + d_k R_{[k]} e_1, \quad (3.1)$$

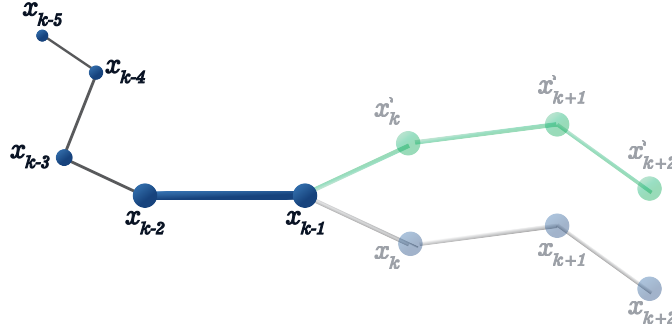
em que os termos d_i representam as distâncias de ligação entre átomos consecutivos e os ângulos de torção e ligação, representados por ψ_i e θ_i respectivamente, são usados para construir $R_{[i]} = R_2^{(\psi_2)} R_2^{(\theta_2)} R_3^{(\psi_3)} R_3^{(\theta_3)} \cdots R_i^{(\psi_i)} R_i^{(\theta_i)}$. Mais detalhes podem ser encontrados em (CAMARGO, 2021). Todas as matrizes R_i desse produto são rotações elementares: as que envolvem ângulos de torção atuam em torno do eixo X, enquanto que em torno do eixo Z são as que envolvem os ângulos de ligação.

Queremos saber o que acontece com as distâncias $r_{i,j} = \|x_j - x_i\|$, em que x_i e x_j não são necessariamente consecutivos, quando há uma alteração ϕ_k no ângulo de torção ψ_k , isto é, passamos a ter $\psi'_k = \psi_k + \phi_k$. Assim como no capítulo 2 de (CAMARGO, 2021), vamos analisar esse assunto em 3 casos. A opção por refazer as contas aqui é porque vamos detalhá-las um pouco mais e também porque isso será útil para uma reflexão importante posteriormente. Portanto,

1 - $i, j < k$:

Como a alteração acontece no ângulo ψ_k , os pontos x_i e x_j não precisam ser atualizados e, portanto, a distância entre eles permanece a mesma. A Figura 14 ilustra isso.

Figura 14 – Destaque dos pontos anteriores a x_k , que não têm distâncias alteradas entre si, apesar das atualizações ilustradas em verde.



Fonte: Elaborado pelo autor.

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

Primeiro, vamos mostrar que a distância $r_{i,j}$ não se altera quando $i, j \geq k$. De fato, os pontos x_i e x_j sofrem atualização devido a alteração no ângulo ψ_k e agora serão escritos como

$$\begin{aligned} x'_i &= x_1 + d_2 R_{[2]} e_1 + d_3 R_{[3]} e_1 + \cdots + d_k R'_{[k]} e_1 + \cdots + d_i R'_{[i]} e_1 \\ x'_j &= x_1 + d_2 R_{[2]} e_1 + d_3 R_{[3]} e_1 + \cdots + d_k R'_{[k]} e_1 + \cdots + d_j R'_{[j]} e_1, \end{aligned}$$

em que $R'_{[m]} = R_2^{(\psi_2)} R_2^{(\theta_2)} \cdots R_k^{(\psi_k + \phi)} R_k^{(\theta_k)} \cdots R_m^{(\psi_m)} R_m^{(\theta_m)}$.

Sem perda de generalidade, vamos considerar $i \leq j$ e, então, temos que

$$\begin{aligned} \|x'_j - x'_i\| &= \|(d_{i+1} R'_{[i+1]} + \cdots + d_j R'_{[j]}) e_1\| \\ &= \left\| R'_{[i+1]} \left(d_{i+1} + \sum_{p=i+2}^j d_p \left(\prod_{q=i+2}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) e_1 \right\|. \end{aligned}$$

Como a rotação não altera a norma de um vetor, então

$$\begin{aligned} \|x'_j - x'_i\| &= \left\| \left(d_{i+1} + \sum_{p=i+2}^j d_p \left(\prod_{q=i+2}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) e_1 \right\| \\ &= \|x_j - x_i\|, \end{aligned}$$

pois esse cálculo não envolve a alteração no ângulo de torção.

Resta provar, sem perda de generalidade, que $r_{i,j}$ não se altera quando $j \geq k$ e $i = k - 2$ ou $i = k - 1$. Em ambos os casos, apenas x_j é atualizado. Então, note que

$$\begin{aligned} \|x'_j - x_{k-2}\| &= \|(d_{k-1}R_{[k-1]} + d_kR'_{[k]} + \cdots + d_jR'_{[j]})e_1\| \\ &= \left\| R_{[k-1]} \left(d_{k-1} + R_k^{(\psi_k+\phi)} R_k^{(\theta)} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| \left(d_{k-1} + R_k^{(\psi_k+\phi)} R_k^{(\theta)} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\|. \end{aligned}$$

De fato, $R_k^{(\psi_k+\phi)} = R_k^{(\psi_k)} R_k^{(\phi)} = R_k^{(\phi)} R_k^{(\psi_k)}$, pois são rotações em torno do mesmo eixo. Além disso, $d_{k-1}R_k^{(\phi)}e_1 = d_{k-1}e_1$, pois e_1 é o eixo de rotação de $R_k^{(\phi)}$ e, portanto, não produz alteração em e_1 . Logo,

$$\begin{aligned} \|x'_j - x_{k-2}\| &= \left\| \left(d_{k-1} + R_k^{(\phi)} R_k^{(\psi_k)} R_k^{(\theta)} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| R_k^{(\phi)} \left(d_{k-1} + R_k^{(\psi_k)} R_k^{(\theta)} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| \left(d_{k-1} + R_k^{(\psi_k)} R_k^{(\theta)} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \|x_j - x_{k-2}\|, \end{aligned}$$

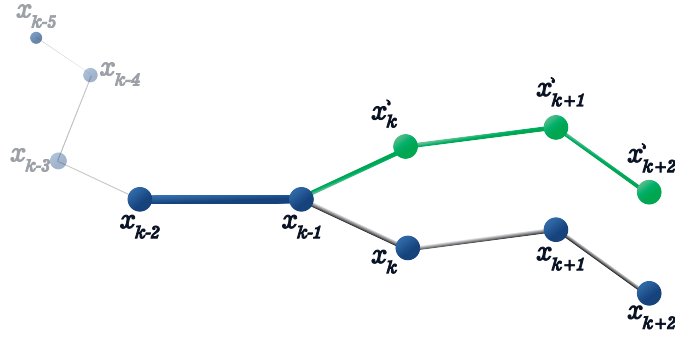
pois, novamente, não depende da alteração que foi imposta ao ângulo de torção ψ_k . Resultado semelhante encontramos quando fazemos o cálculo da distância $r_{k-1,j}$. Portanto, as distâncias $r_{i,j}$ não se alteram quando $i, j \geq k - 2$, como ilustra a Figura 15.

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

Analogamente aos outros itens, temos que:

$$\begin{aligned} \|x'_j - x_i\| &= \|(d_{i+1}R_{[i+1]} + \cdots + d_{k-2}R_{[k-2]} + \cdots + d_kR'_{[k]} + \cdots + d_jR'_{[j]})e_1\| \\ &= \left\| \left(d_{i+1}R_{[i+1]} + \cdots + R'_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| \left(d_{i+1}R_{[i+1]} + \cdots + R_k^{(\phi)} R_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\|. \end{aligned}$$

Figura 15 – Destaque dos pontos posteriores a x_{k-3} , que não têm distâncias alteradas entre si, apesar das atualizações realizadas, ilustradas em verde.



Fonte: Elaborado pelo autor.

Para que tenhamos $\|x'_j - x_i\| = \|x_j - x_i\|$, precisamos ter que

$$\begin{aligned} \|x'_j - x_i\| &= \left\| \left(d_{i+1}R_{[i+1]} + \cdots + R_k^{(\phi)}R_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| \left(d_{i+1}R_{[i+1]} + \cdots + R_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \|x_j - x_i\|. \end{aligned}$$

Para isso, precisaríamos ter que

$$\begin{aligned} \|x'_j - x_i\| &= \left\| \left(d_{i+1}R_{[i+1]} + \cdots + R_k^{(\phi)}R_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\| \\ &= \left\| R_k^{(\phi)} \left(d_{i+1}R_{[i+1]} + \cdots + R_{[k]} \left(d_k + \sum_{p=k+1}^j d_p \left(\prod_{q=k+1}^p R_q^{(\psi_q)} R_q^{(\theta_q)} \right) \right) \right) e_1 \right\|, \end{aligned}$$

porque com a justificativa de que a rotação não altera a norma de um vetor, poderíamos “eliminar” o termo $R_k^{(\phi)}$.

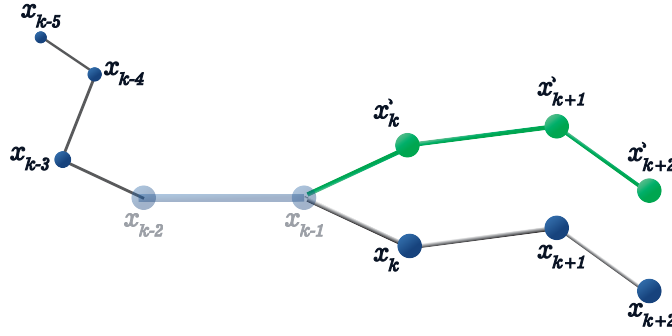
Mas não podemos garantir que $R_k^{(\phi)}(d_a R_{[a]}) = d_a R_{[a]}$, com $a \in [i+1, k-1]$. Logo, $\|x'_j - x_i\| \neq \|x_j - x_i\|$. Isso implica que as distâncias $r_{i,j} = \|x_j - x_i\|$, para $i \leq k-3$ e $j \geq k$, mudam devido a uma alteração no ângulo de torção ψ_k , como está ilustrado na Figura 16.

Portanto, podemos observar que um ponto com coordenada atualizada deve manter a mesma distância do ponto original para os seus dois antecessores. De outra forma, devemos ter essas duas igualdades satisfeitas:

$$\|x'_i - x_{i-1}\| = \|x_i - x_{i-1}\| \quad e \quad \|x'_i - x_{i-2}\| = \|x_i - x_{i-2}\|,$$

em que x'_i corresponde à nova localização de x_i . Um olhar atento a essas duas equações pode nos indicar o lugar geométrico onde habitam todas as possíveis atualizações de x_i .

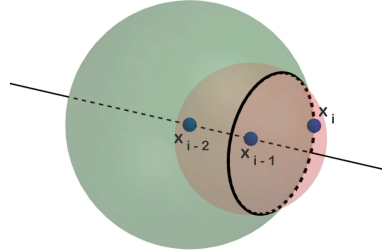
Figura 16 – Destaque dos pontos anteriores a x_{k-2} e posteriores a x_{k-1} , cujas distâncias são alteradas devido as atualizações, ilustradas em verde.



Fonte: Elaborado pelo autor.

Com apenas x'_i como incógnita, já que sabemos quem são x_i, x_{i-1} e x_{i-2} , as duas expressões acima representam esferas centradas em x_{i-1} e x_{i-2} , respectivamente. A primeira com raio igual a $\|x_i - x_{i-1}\|$, enquanto a segunda com raio $\|x_i - x_{i-2}\|$. De fato, x'_i pertence à interseção dessas duas esferas. Como consequência, as possíveis atualizações para o ponto x_i devem ser parte de uma circunferência cujo centro é interceptado perpendicularmente por uma reta que passa por x_{i-1} e x_{i-2} . Uma ilustração disso está na Figura 17.

Figura 17 – Interseção de duas esferas produzindo uma circunferência, lugar geométrico das possíveis atualizações do ponto x_i .



Fonte: Elaborado pelo autor.

Essas observações nos indicam a necessidade de usar os pontos x_{i-1} e x_{i-2} para criar o eixo pelo qual a rotação atua. Essa ideia é base dos métodos que vamos apresentar neste capítulo.

3.2 Métodos de Atualização de Coordenadas Cartesianas

Nessa seção, vamos apresentar um método no qual todos os pontos são atualizados usando matrizes que performam rotações em torno de eixos arbitrários que passam pela origem, além de translações apropriadas. No capítulo anterior, apresentamos duas formas para realizar rotações desse tipo: uma com uma matriz que é resultado de uma

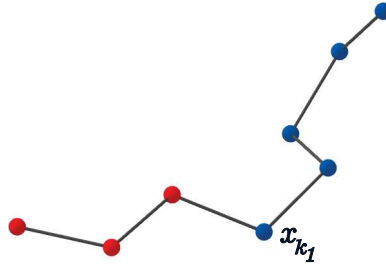
sequência de rotações que alinham o eixo de rotação a um eixo cartesiano; outra, usando os quatérnios.

Portanto, o mesmo método vale para esses dois casos, com pequenos ajustes no que refere à matriz de rotação usada. Logo, vamos explicá-lo de uma forma geral, para depois chegar ao específico. Na tentativa de fazer isso da forma mais clara possível, vamos tecer comentários a respeito de cada etapa à medida em que também ilustramos as consequências que elas produzem.

Antes de tudo, vamos denominar por grupo de átomos, e representá-los por G_i , todos os pontos que recebem a atuação da mesma rotação. De forma equivalente, se ψ_{k_i} e $\psi_{k_{i+1}}$ são ângulos de torção que devem ser alterados, então G_i é composto pelos pontos x_j , em que $k_i \leq j < k_{i+1}$. Além disso, vamos estabelecer que G_0 é o conjunto dos pontos que não precisam ser atualizados.

Seja x_{k_1} um ponto que representa o k_1 -ésimo átomo da cadeia principal de uma molécula, com $k_1 \geq 4$. Suponha que este seja o primeiro ponto que deve ter a posição atualizada em razão de uma alteração ϕ_{k_1} no ângulo de torção ψ_{k_1} . A Figura 18 ilustra um pequeno conjunto de pontos, em que x_{k_1} se encontra destacado. A ordem deve ser vista partindo do primeiro ponto vermelho, que coincide com a origem do sistema cartesiano.

Figura 18 – Disposição dos pontos e ligações representando uma pequena molécula.



Fonte: Elaborado pelo autor.

Para iniciar o processo de atualização da coordenada do ponto x_{k_1} , primeiro devemos estabelecer um eixo de rotação. Vimos anteriormente que ele deve ser construído a partir da ligação b_{k_1-2} , que conecta x_{k_1-2} e x_{k_1-1} . Para isso, representamos o eixo de rotação por um vetor unitário u construído a partir desses dois pontos. Assim,

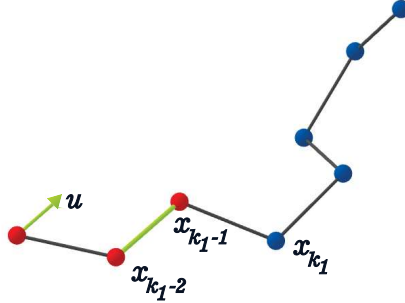
$$u = \frac{x_{k_1-1} - x_{k_1-2}}{\|x_{k_1-1} - x_{k_1-2}\|}. \quad (3.2)$$

É importante observar a ordem dos pontos. Se fizermos $x_{k_1-2} - x_{k_1-1}$, no final teremos $\psi'_{k_1} = \psi_{k_1} - \phi_{k_1}$, ao invés de $\psi'_{k_1} = \psi_{k_1} + \phi_{k_1}$.

Intuitivamente, como pode ser visto na Figura 19, temos uma ligação auxiliar que é criada como se tivéssemos trazido x_{k_1-2} para coincidir com a origem e x_{k_1-1} para

$x_{k_1-1} - x_{k_1-2}$. Depois disso, normalizamos esse vetor e o chamamos de u . Tendo feito isso, atendemos às exigências para a construção da matriz de rotação que precisamos.

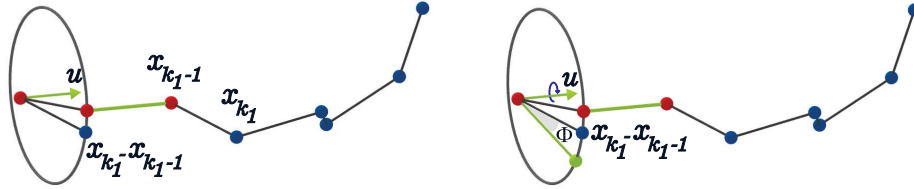
Figura 19 – Vetor unitário u que servirá como eixo de rotação.



Fonte: Elaborado pelo autor.

Seja R_1 a matriz criada. O próximo passo é aplicá-la. Como levamos o eixo de rotação à origem, devemos também transladar o ponto x_i por meio de $-x_{k_1-1}$. Portanto, a matriz R_1 deve ser aplicada à $x_{k_1} - x_{k_1-1}$. Veja, na Figura 20, a ilustração da rotação dessa ligação em torno do eixo representado por u .

Figura 20 – Na imagem da esquerda, as possíveis localizações para o resultado da rotação. Na direita, uma rotação por meio de um ângulo ϕ_{k_1} .



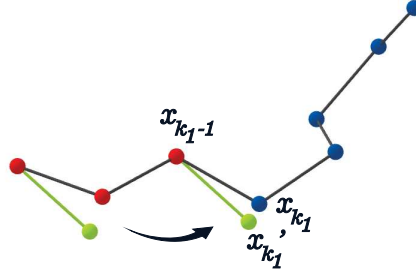
Fonte: Elaborado pelo autor.

Para finalizar, resta apenas uma translação para levar ao devido lugar o ponto atualizado. Note que o que temos que fazer é reposicionar essa atualização na estrutura que tínhamos. Nesse caso, usamos a translação inversa a que foi aplicada anteriormente, ou seja, $-(-x_{k_1-1}) = x_{k_1-1}$. Com isso, completamos o processo de atualização do ponto x_{k_1} . Na Figura 21, temos a representação dessa translação.

O que fizemos para atualizar o ponto x_{k_1} pode ser resumido com a expressão $x'_{k_1} = R_1(x_{k_1} - x_{k_1-1}) + x_{k_1-1}$. Ainda podemos generalizar esse processo e estender a ideia para a atualização de todos os pontos desse grupo de átomos. Se ψ_{k_2} é o próximo ângulo de torção que deve ser alterado, então a atualização dos pontos x_m , com $k_1 \leq m < k_2$, é dada por

$$x'_m = R_1(x_m - x_{m-1}) + x'_{m-1}. \quad (3.3)$$

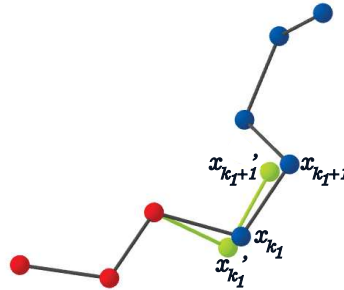
Figura 21 – Atualização do ponto x_{k_1} , representado por x'_{k_1} em verde, após a translação.



Fonte: Elaborado pelo autor.

Aqui, temos duas coisas a observar. Primeiro: a mesma rotação que é usada para atualizar o ponto x_{k_1} é aplicada também a todos os pontos que compõem G_1 . Isso permite a manutenção da estrutura formada pelos pontos desse grupo de átomos. Segundo: o último termo da equação (3.3) corresponde a uma translação em relação a x'_{m-1} . Com isso, fazemos com que um ponto atualizado seja devidamente anexado à nova estrutura que está se formando. Isso pode ser visto na Figura 22. Caso x_{m-1} pertença a G_0 , i.e., não precise sofrer atualização, então podemos considerar $x'_{m-1} = x_{m-1}$.

Figura 22 – Atualização dos pontos entre x_{k_1} e x_{k_1+1} , representados por x'_{k_1} e x'_{k_1+1} e ilustrados em verde.



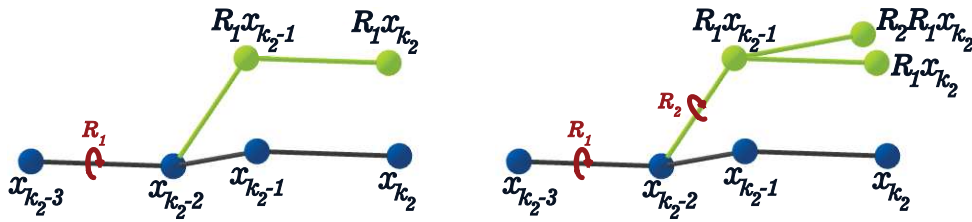
Fonte: Elaborado pelo autor.

Agora, vamos dirigir nossa atenção para uma próxima mudança num ângulo de torção, que acontece em ψ_{k_2} e é ocasionada por uma alteração ϕ_{k_2} . Para a atualização dos pontos que pertencem a G_2 , i.e., todos os x_m , com $k_2 \leq m < k_3$, o processo a ser seguido é semelhante ao já relatado. Entretanto, há uma diferença importante que precisa ser mencionada. Quando temos uma segunda alteração nos ângulos de torção, também geramos um vetor unitário que é entendido como eixo de rotação e, em seguida, construímos uma matriz R_2 . Só que esses pontos x_m não estão sujeitos apenas a essa rotação. Na verdade, a matriz R_1 também se aplica. Em geral, uma matriz R_i sempre é aplicada a todos os pontos a partir de x_{k_i} para que a estrutura formada por eles seja mantida, a menos da rotação aplicada.

Uma vantagem é que podemos compor R_1 e R_2 por meio do produto matricial e gerar uma nova matriz que performa as duas rotações ao mesmo tempo! Essa ideia, claro, se estende ao caso geral, com n matrizes dessas. Não tão claro assim é a ordem que deve ser escolhida para o produto. Sabemos que rotações em torno de um mesmo eixo comutam, mas em geral perdemos essa propriedade quando lidamos com rotações em torno de eixos distintos. Para ficar clara a questão da ordem das matrizes, vamos considerar novamente R_1 e R_2 .

Sabemos que x_{k_2} é o primeiro ponto que deve receber a aplicação da rotação R_2 . Consequentemente, x_{k_2-1} , que pertence ao grupo G_1 , sofre ação apenas de R_1 . Isso basta para afirmar que b_{k_2-2} , que liga x_{k_2-2} a x_{k_2-1} , foi atualizada. Acontece que essa ligação é justamente encarada como eixo de rotação de R_2 e usada para criar o vetor unitário necessário para esta matriz. Portanto, se usarmos a ordem R_2R_1 , devemos nos atentar a isso e construir R_2 a partir da ligação b'_{k_2-2} . Essa abordagem, geralmente descrita por um produto $R_n \cdots R_1$, é conhecida como **propagação direta**. Na Figura 23, uma ilustração do que foi dito.

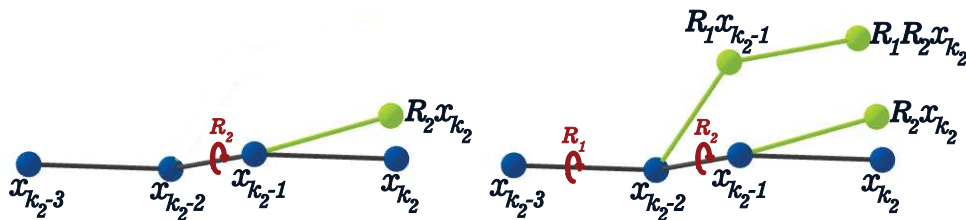
Figura 23 – Ilustração do modelo com propagação direta, com as atualizações dos pontos em verde.



Fonte: Elaborado pelo autor.

Por outro lado, a ordem R_1R_2 usa simplesmente b_{k_2-2} como base para construir o eixo de rotação de R_2 . Com isso, não há necessidade de buscar os pontos atualizados para construí-la. Essa abordagem, geralmente escrita como $R_1 \cdots R_n$, é conhecida como **propagação reversa** e está representada na ilustração da Figura 24.

Figura 24 – Ilustração do modelo com propagação reversa, com as atualizações dos pontos em verde.



Fonte: Elaborado pelo autor.

Ou seja, a principal diferença entre as duas abordagens é que, com a propagação direta, uma rotação altera o posicionamento dos pontos que serão usados para calcular os eixos das rotações seguintes, enquanto que a propagação reversa utiliza-se apenas dos pontos da estrutura original para construir todos os eixos de rotação. Isso indica uma maior suscetibilidade para acumulação de erros de arredondamento quando se usa a propagação direta. Menções a essas duas abordagens também podem ser encontradas em (CHYS; CHACÓN, 2012; CAMARGO, 2021).

Voltando à atualização das coordenadas dos pontos x_m , em que $k_2 \leq m < k_3$, isso se dá, com a propagação reversa, por meio do seguinte cálculo:

$$x'_m = R_1 R_2 (x_m - x_{m-1}) + x'_{m-1}.$$

Da mesma forma, se invertermos a ordem das matrizes R_i e alterarmos devidamente os eixos de rotação que forem necessários, temos a versão com a propagação direta.

O procedimento descrito até aqui segue analogamente até que não haja mais ângulos de torção para serem atualizados.

Essa forma de se aplicar as translações, em que, para cada ponto que precisa ter seu posicionamento atualizado, cria-se um vetor de ligação, aplica-se uma rotação e, por fim, uma translação, é o que (CAMARGO, 2021) chama de um modelo de atualização que faz uso de **Translações Finais**. Mais a frente, apresentaremos uma outra forma de se obter um resultado equivalente.

Mas antes, vamos partir para as especificidades de cada tipo de rotação que se aplica ao processo descrito acima, também observando cada abordagem de propagação. Primeiramente, vamos usar as matrizes de rotação apresentadas na seção 2 do capítulo anterior. Para criá-la, precisamos de um ângulo ϕ e do vetor unitário u , dado por (3.2). Portanto, temos que

$$R = \begin{pmatrix} \cos\phi + u_x^2(1 - \cos\phi) & u_x u_y(1 - \cos\phi) - u_z \sin\phi & u_x u_z(1 - \cos\phi) + u_y \sin\phi \\ u_x u_y(1 - \cos\phi) + u_z \sin\phi & \cos\phi + u_y^2(1 - \cos\phi) & u_y u_z(1 - \cos\phi) - u_x \sin\phi \\ u_x u_z(1 - \cos\phi) - u_y \sin\phi & u_y u_z(1 - \cos\phi) + u_x \sin\phi & \cos\phi + u_z^2(1 - \cos\phi) \end{pmatrix}.$$

Abaixo, o Algoritmo 1 utiliza-se dessa matriz. Adotamos o nome “Rotações Simples” para ir de acordo com a nomenclatura usada por (ZHANG; KAVRAKI, 2002; CHOI, 2006). Também representamos por Cc as coordenadas cartesianas e Cc' a sua atualização. Além disso, P representa as posições dos ângulos de torção que serão alterados e as respectivas alterações são dadas pelos ângulos que compõem o conjunto A .

Algoritmo 1 – Atualização de coordenadas cartesianas usando Rotações Simples e Translações Finais.

Entrada: $Cc = \{x_1, x_2, \dots, x_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$

Saída: $Cc' = \{x'_1, x'_2, \dots, x'_n\}$

1 **início**

2 $Cc' \leftarrow Cc$;

3 $R_{[0]} \leftarrow I_{3 \times 3}$;

4 **para** $i = 1, \dots, p$ **faça**

5 $u \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$;

6 Calcular a matriz R usando o ângulo ϕ_{k_i} e o vetor u como eixo;

7 $R_{[i]} \leftarrow R_{[i-1]}R$;

8 **para** $k_i \leq m < k_{i+1}$ **faça**

9 $x'_m \leftarrow R_{[i]}(x_m - x_{k_i-1}) + x'_{k_i-1}$

10 **fim**

11 **fim**

12 **fim**

O algoritmo acima está escrito baseado na propagação reversa de rotações. O equivalente a ele, levando em conta a propagação direta, é conseguido realizando as seguintes adaptações:

- Na linha 5 do Algoritmo 1, defina

$$u = \frac{x'_{k_i-1} - x'_{k_i-2}}{\|x'_{k_i-1} - x'_{k_i-2}\|};$$

- Na linha 7, faça

$$R_{[i]} = RR_{[i-1]}.$$

Assim como o Algoritmo 1, todos os outros neste capítulo serão escritos usando a propagação reversa de rotações. A adaptação para propagação direta, em cada um deles, será análoga ao que fizemos acima.

Vamos fazer uma breve análise sobre o custo computacional do Algoritmo 1 por meio da contagem de operações, desconsiderando as multiplicações por 1 e somas por 0, bem como o cálculo de operações trigonométricas, comuns a todos os métodos. Portanto, considere n_r o número de ângulos de torção alterados e n_a a quantidade de pontos que devem ser atualizados.

No que se refere à contagem de operações, as abordagens com propagação reversa e direta não têm nada diferente senão a ordem do produto entre as matrizes e os pontos usados para construir os eixos. Então, fazendo uma análise conjunta, temos que, para cada ângulo de torção que se altera, são necessárias: 3 somas e 3 divisões para calcular o eixo de rotação; 15 multiplicações e 10 somas para gerar a matriz R ; 27 multiplicações e

18 somas para calcular o produto das matrizes. Assim, temos ao todo 45 multiplicações e 31 somas. Além disso, cada atualização de um ponto custa 9 multiplicações e 12 somas. Portanto, o método de Rotações Simples com Translações Finais possui uma estimativa de custo computacional de $76 n_r + 21 n_a$ operações.

Uma outra versão para o Algoritmo 1 pode ser conseguida utilizando o Espaço Homogêneo. No final do capítulo passado, vimos que a utilização do Espaço Homogêneo permite a linearização da translação, de modo que ela pode ser descrita, juntamente com a rotação, em um único operador. Dessa forma, surge o modelo com **Translações Encaixadas**, apresentado em (THOMPSON, 1967) e assim denominado em (CAMARGO, 2021), em que utiliza-se de um único operador para o processo de atualização das coordenadas de todos os pontos de um mesmo grupo de átomos.

A adaptação para Translações Encaixadas do processo descrito até aqui começa pelo fato de que a expressão (3.3) pode ser reescrita como

$$x'_m = R_1(x_m - x_{k_1-1}) + x'_{k_1-1}. \quad (3.4)$$

Pode-se observar que, em geral, $x_m - x_{k_1-1}$ não corresponde a uma ligação covalente. Mas, mesmo assim, não há problema, pois ainda que esses pontos não sejam consecutivos, sabemos que a distância entre eles permanece fixa após uma alteração no ângulo de torção ψ_{k_1} , já que x_{k_1-1} é usado para construir o eixo de rotação de R_1 . Vimos isso no item 2 da análise feita na primeira seção deste capítulo.

Dessa forma, podemos simplificar o processo descrito pela equação (3.4), que indica a necessidade de realizar uma primeira translação por $-x_{k_1-1}$ a x_m , seguida de uma rotação desse segmento por R_1 , e depois outra translação, dessa vez por x_{k_1-1} ao segmento rotacionado. Matricialmente, e com auxílio do Espaço Homogêneo, isso é expresso da seguinte forma:

$$\begin{pmatrix} x'_m \\ 1 \end{pmatrix} = \left[\begin{pmatrix} I & x'_{k_1-1} \\ \bar{0} & 1 \end{pmatrix} \cdot \begin{pmatrix} R_1 & 0 \\ \bar{0} & 1 \end{pmatrix} \cdot \begin{pmatrix} I & -x_{k_1-1} \\ \bar{0} & 1 \end{pmatrix} \right] \cdot \begin{pmatrix} x_m \\ 1 \end{pmatrix},$$

no qual o produto dessas três matrizes entre os colchetes resulta em

$$M_1 = \begin{pmatrix} R_1 & x'_{k_1-1} - R_1 x_{k_1-1} \\ \bar{0} & 1 \end{pmatrix}.$$

Claro, essa ideia também se estende para o caso em que temos n matrizes de rotação compostas através do produto matricial. Para sintetizar o processo que envolve esse modelo, apresentamos o Algoritmo 2.

Algoritmo 2 – Atualização de coordenadas cartesianas usando Rotações Simples e Translações Encaixadas.

Entrada: $Cc = \{x_1, x_2, \dots, x_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$

Saída: $Cc' = \{x'_1, x'_2, \dots, x'_n\}$

1 **início**

2 $Cc' \leftarrow Cc$;

3 $R_{[0]} \leftarrow I_{3 \times 3}$;

4 **para** $i = 1, \dots, p$ **faça**

5 $u \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$;

6 Calcular a matriz R usando o ângulo ϕ_{k_i} e o vetor u como eixo;

7 $R_{[i]} \leftarrow R_{[i-1]} R$;

8 $t \leftarrow x'_{k_i-1} - R_{[i]} x_{k_i-1}$;

9 $M_i \leftarrow \begin{pmatrix} R_{[i]} & t \\ 0 & 1 \end{pmatrix}$;

10 **para** $k_i \leq m < k_{i+1}$ **faça**

11 $x'_m \leftarrow M_i x_m$;

12 **fim**

13 **fim**

14 **fim**

Novamente, com uma análise conjunta para as versões com cada uma das propagações de rotações, temos: 3 somas e 3 divisões para calcular o eixo de rotação; 15 multiplicações e 10 somas para gerar a matriz R ; 27 multiplicações e 18 somas para calcular o produto das matrizes; e 9 multiplicações e 9 somas para calcular o vetor de translação. Assim, temos ao todo 54 multiplicações e 40 somas. Além disso, cada atualização de um ponto custa 9 multiplicações e 9 somas. Portanto, uma estimativa de custo computacional de $94 n_r + 18 n_a$ operações.

Por outro lado, ao invés de usar as Rotações Simples, podemos obter os mesmos resultados usando a representação matricial da rotação com os quatérnios. Depois de calcular o vetor unitário u , construímos o quatérnio

$$q = \left(\cos \frac{\phi}{2}, \sin \frac{\phi}{2} u_x, \sin \frac{\phi}{2} u_y, \sin \frac{\phi}{2} u_z \right),$$

de modo que vamos considerar $q = (q_0, q_1, q_2, q_3)$.

Resta acrescentar que, ao invés de compor duas matrizes de rotação, vamos fazer o produto dos dois quatérnios primeiro para, só depois, construir a matriz que representa a rotação com quatérnios:

$$R = 2 \begin{pmatrix} q_0^2 + q_1^2 - \frac{1}{2} & q_1 q_2 - q_0 q_3 & q_1 q_3 + q_0 q_2 \\ q_1 q_2 + q_0 q_3 & q_0^2 + q_2^2 - \frac{1}{2} & q_2 q_3 - q_0 q_1 \\ q_1 q_3 - q_0 q_2 & q_2 q_3 + q_0 q_1 & q_0^2 + q_3^2 - \frac{1}{2} \end{pmatrix}.$$

O Algoritmos 3 e 4, respectivamente com Translações Finais e Translações Encaixadas, resumem os processos que envolvem a representação matricial da rotação com os quatérnios.

Algoritmo 3 – Atualização de coordenadas cartesianas usando Quatérnios e Translações Finais

Entrada: $Cc = \{x_1, x_2, \dots, x_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$
Saída: $Cc' = \{x'_1, x'_2, \dots, x'_n\}$

```

1 início
2    $Cc' \leftarrow Cc$ ;
3    $q_{[0]} \leftarrow (1, 0, 0, 0)$ ;
4   para  $i = 1, \dots, p$  faça
5      $u \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$ ;
6      $q \leftarrow (\cos \frac{\phi_{k_i}}{2}, \sin \frac{\phi_{k_i}}{2} u_x, \sin \frac{\phi_{k_i}}{2} u_y, \sin \frac{\phi_{k_i}}{2} u_z)$ ;
7      $q_{[i]} \leftarrow q_{[i-1]} * q$ ;
8     Calcular a matriz  $R$  a partir do quatérnio  $q_{[i]}$ ;
9     para  $k_i \leq m < k_{i+1}$  faça
10       $x'_m \leftarrow R(x_m - x_{k_i-1}) + x'_{k_i-1}$ 
11    fim
12  fim
13 fim

```

Algoritmo 4 – Atualização de coordenadas cartesianas usando Quatérnios e Translações Encaixadas

Entrada: $Cc = \{x_1, x_2, \dots, x_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$
Saída: $Cc' = \{x'_1, x'_2, \dots, x'_n\}$

```

1 início
2    $Cc' \leftarrow Cc$ ;
3    $q_{[0]} \leftarrow (1, 0, 0, 0)$ ;
4   para  $i = 1, \dots, p$  faça
5      $u \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$ ;
6      $q \leftarrow (\cos \frac{\phi_{k_i}}{2}, \sin \frac{\phi_{k_i}}{2} u_x, \sin \frac{\phi_{k_i}}{2} u_y, \sin \frac{\phi_{k_i}}{2} u_z)$ ;
7      $q_{[i]} \leftarrow q_{[i-1]} * q$ ;
8     Calcular a matriz  $R$  a partir do quatérnio  $q_{[i]}$ ;
9      $t \leftarrow x'_{k_i-1} - Rx_{k_i-1}$ ;
10     $M_i \leftarrow \begin{pmatrix} R & t \\ \bar{0} & 1 \end{pmatrix}$ ;
11    para  $k_i \leq m < k_{i+1}$  faça
12       $x'_m \leftarrow M_i x_m$ ;
13    fim
14  fim
15 fim

```

Para o Algoritmo 3, temos: 3 somas e 3 divisões para calcular o eixo de rotação; 3 multiplicações e 1 divisão para calcular o quatérnio; 16 multiplicações e 12 somas

para determinar o produto de dois quatérnios; 10 multiplicações e 10 somas para criar a representação matricial da rotação usando quatérnios. Logo, cada ângulo de torção alterado gera um custo de 33 multiplicações e 25 somas. Por outro lado, para cada ponto que deve ser atualizado, são necessárias 9 multiplicações e 12 somas. Portanto, essa abordagem gera um custo de $58 n_r + 21 n_a$ operações.

Já para o Algoritmo 4, são: 3 somas e 3 divisões para calcular o eixo de rotação; 3 multiplicações e 1 divisão para calcular o quatérnio; 16 multiplicações e 12 somas para determinar o produto de dois quatérnios; 10 multiplicações e 10 somas para criar a representação matricial da rotação usando quatérnios; e 9 multiplicações e 9 somas para calcular o vetor de translação. Logo, cada ângulo de torção alterado gera um custo de 42 multiplicações e 34 somas. Por outro lado, para cada ponto que deve ser atualizado, são necessárias 9 multiplicações e 9 somas. Dessa forma, temos um custo de $76 n_r + 18 n_a$ operações.

Nota-se um custo menor nessas abordagens envolvendo quatérnios quando comparadas com as Rotações Simples. Isso se deve ao fato de que compor quatérnios é mais barato computacionalmente do que compor as matrizes de rotação que lidamos.

A utilização das Translações Finais ainda nos abre a possibilidade de usar um método em que trabalhamos com os quatérnios de forma nativa, sem convertê-los em matrizes. Dessa forma, surge mais um algoritmo:

Algoritmo 5 – Atualização de coordenadas cartesianas usando Quatérnios nativamente e Translações Finais.

Entrada: $Cc = \{x_1, x_2, \dots, x_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$
Saída: $Cc' = \{x'_1, x'_2, \dots, x'_n\}$

1 **início**

2 $Cc' \leftarrow Cc$;

3 $q_{[0]} \leftarrow (1, 0, 0, 0)$;

4 **para** $i = 1, \dots, p$ **faça**

5 $u \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$;

6 $q \leftarrow (\cos \frac{\phi_{k_i}}{2}, \sin \frac{\phi_{k_i}}{2} u_x, \sin \frac{\phi_{k_i}}{2} u_y, \sin \frac{\phi_{k_i}}{2} u_z)$;

7 $q_{[i]} \leftarrow q_{[i-1]} * q$;

8 **para** $k_i \leq m < k_{i+1}$ **faça**

9 $t \leftarrow x_m - x_{k_i-1}$;

10 $l \leftarrow q_{[i]} * (0, t_1, t_2, t_3) * q_{[i]}^{-1}$;

11 $x'_m \leftarrow x_{k_i-1} + (l_2, l_3, l_4)$;

12 **fim**

13 **fim**

14 **fim**

Na contagem de operações, temos: 3 somas e 3 produtos para calcular u ; 4 produtos para calcular q ; 16 produtos e 12 somas para calcular $q_{[i]}$. Além disso, 3 somas

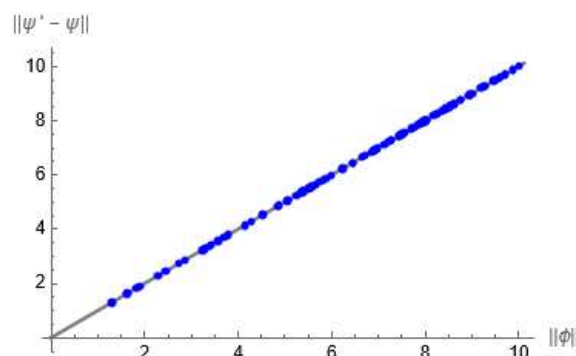
para calcular t ; 28 produtos e 20 somas para calcular l ; 3 somas para calcular x'_j . Com isso, ao todo temos $38n_r + 54n_a$ operações.

3.3 Testes Computacionais

Realizamos alguns testes computacionais com cada um dos algoritmos apresentados neste capítulo, com o auxílio do software *Wolfram Mathematica*. Uma primeira situação investigada é se as soluções obtidas com os métodos correspondem, experimentalmente, ao esperado. Uma das maneiras de se fazer isso é tomando os conjuntos de ângulos de torção originais (ψ) e atualizados (ψ'), juntamente com as alterações impostas (ϕ). A ideia é averiguar se realmente é observado experimentalmente que $\psi_i + \phi_i = \psi'_i$, para todo i , após o processamento do algoritmo. Pensando em facilitar a realização do experimento com um número satisfatório de situações, vamos investigar se de fato observamos que $\|\psi'_i - \psi_i\| = \|\phi_i\|$.

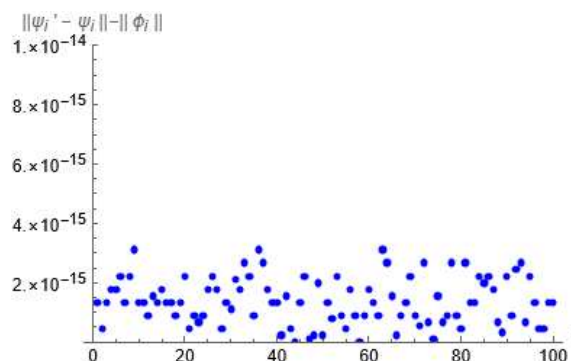
Para realizar esse experimento, consideramos uma molécula hipotética de proteína de 1000 átomos e impomos 100 diferentes alterações nos ângulos de torção, com a quantidade de ângulos alterados variando de 1% a 30% do total e o intervalo de valor dentro do intervalo de $[-1, 1]$. Para cada um dos 100 casos de alterações, obtemos as coordenadas dos pontos atualizados e, então, calculamos os ângulos de torção da estrutura gerada. O gráfico da Figura 25 apresenta os resultados utilizando o Algoritmo 1 com a Propagação Reversa.

Figura 25 – Norma da diferença entre os ângulos de torção originais e atualizados ($\|\psi' - \psi\|$) em função da norma das alterações nos ângulos de torção ($\|\phi\|$). Em azul, os valores obtidos em cada simulação; em cinza, a reta identidade.



Fonte: Elaborado pelo autor.

Como esperado, a igualdade $\|\psi'_i - \psi_i\| = \|\phi_i\|$ foi observada, porque os resultados de cada simulação estão aparentemente sobre a reta identidade, que representa a igualdade de valores nos dois eixos. Podemos ainda verificar a distância entre os valores obtidos em $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$ para averiguar o quão próximos estão. Na Figura 26, um gráfico que representa isso.

Figura 26 – Diferença entre as normas $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$.

Fonte: Elaborado pelo autor.

Com os resultados na ordem de 10^{-15} , podemos concluir que de fato o Algoritmo 1 realiza o que se propõe: as devidas alterações nos ângulos de torção.

Um caminho natural é realizar o mesmo procedimento para os demais algoritmos e verificar se eles também correspondem às expectativas no que diz respeito às soluções. Isso foi feito e os resultados encontrados foram semelhantes aos obtidos acima. Entretanto, tendo em vista outras possíveis análises a serem feitas, vamos usar uma métrica de similaridade de estruturas de proteínas conhecida como “RMSD”, do inglês *Root Mean Square Deviation* (raiz quadrada do desvio quadrático médio) para comparar os demais algoritmos com o Algoritmo 1.

O RMSD é uma das medidas mais utilizadas para comparar a similaridade entre duas estruturas sobrepostas. Encontrada, por exemplo, em (KUFAREVA; ABAGYAN, 2012), juntamente de outras métricas de similaridade estrutural, o RMSD retorna um valor, em Å, no intervalo $[0, \infty)$. Quanto mais próximo de 0, maior a similaridade entre as estruturas. O cálculo é feito por meio da fórmula

$$RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^n \delta_i^2}, \quad (3.5)$$

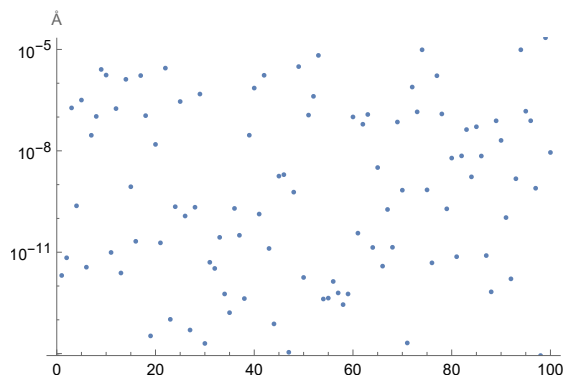
em que δ_i é a distância entre os átomos na posição i de cada uma das duas estruturas.

Para realizar o experimento, consideramos novamente uma molécula hipotética de proteína de 1000 átomos e vamos impor 100 distintas alterações em parte dos seus ângulos de torção, tal qual foi feito no experimento anterior. Como já realizamos testes sobre a precisão do Algoritmo 1, vamos tomá-lo como referência e comparar os demais a ele. A Figura 27 mostra os valores do RMSD calculados com os resultados obtidos com o Algoritmo 1, considerando as propagações reversa e direta.

Em seguida, apresentamos também as comparações entre o Algoritmo 1 e os Algoritmos 2, 3, 4 e 5, presentes na Figura 28.

De fato, os resultados presentes na Figura 28 indicam que os Algoritmos 2, 3,

Figura 27 – Valores obtidos do RMSD na comparação das versões do Algoritmo 1 com as propagações reversa e direta.



Fonte: Elaborado pelo autor.

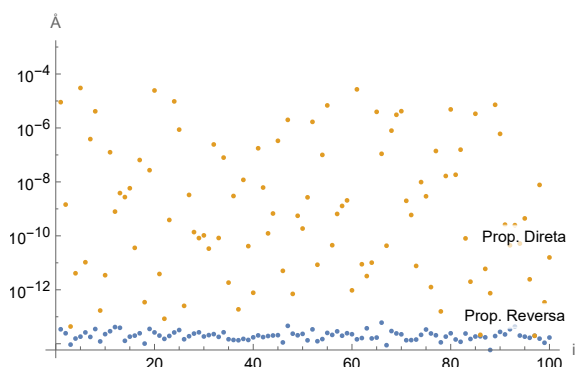
4 e 5, independentemente do tipo de propagação de rotações, produzem soluções próximas àquelas encontradas com o Algoritmo 1, como era esperado.

Pode-se notar ainda que as abordagens com a propagação direta apresentam maiores valores de RMSD na comparação com o Algoritmo 1 do que as abordagens com a propagação reversa. Isso significa que as estruturas resultantes dos algoritmos que envolvem a propagação direta são menos similares que do que os da reversa. De fato, esses valores são pequenos, mas abrem precedente para diferenças maiores, principalmente ao envolver proteínas com mais átomos e mais alterações em ângulos de torção. Essas diferenças entre as duas propagações eram esperadas, uma vez que a propagação direta se utiliza de pontos recém atualizados para construir os eixos de rotação, tornando-se mais suscetível à acumulação de erros de arredondamento. Por isso, daqui em diante trabalharemos apenas com as abordagens com propagação reversa.

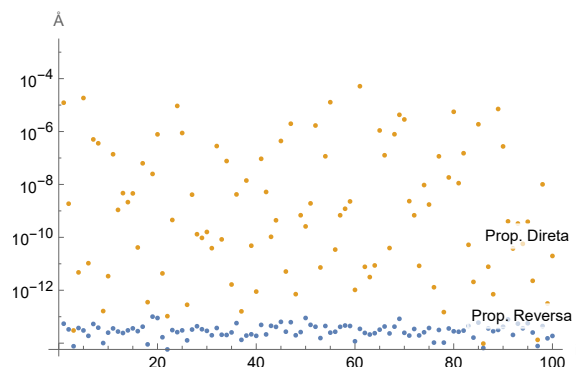
Por fim, também nos interessamos em investigar se há alguma diferença, em termos de precisão, entre os algoritmos que envolvem Translações Finais e Translações Encaixadas. Para isso, com os mesmos dados usados dos experimentos anteriores, buscamos ver se os Algoritmos 1 e 2 retornam soluções próximas, já que são baseados no mesmo tipo de procedimento, se diferindo apenas pelo fato de que um usa Translações Finais e o outro, não. O mesmo faremos para os Algoritmos 3 e 4. Assim como feito no primeiro experimento, queremos verificar o quão próximos estão os valores $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$ para cada dupla de algoritmos. Isso está apresentado no gráfico da Figura 29.

O que pode-se perceber com o resultado deste experimento é que não há diferenças significativas, em termos de precisão, entre a utilização das Translações Finais ou das Translações Encaixadas.

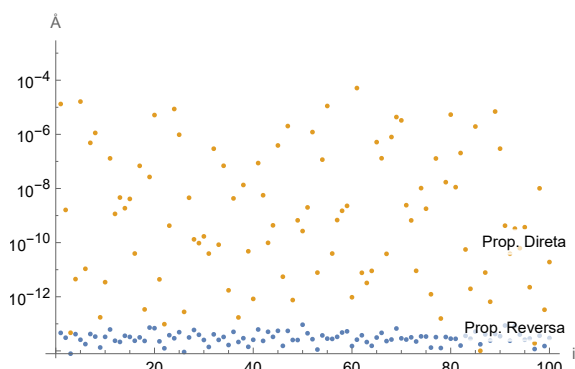
Figura 28 – Valores obtidos do RMSD na comparação entre o Algoritmo 1 (com Propagação Reversa) e os Algoritmos 2, 3, 4 e 5, cada um com as duas versões de propagação de rotações.



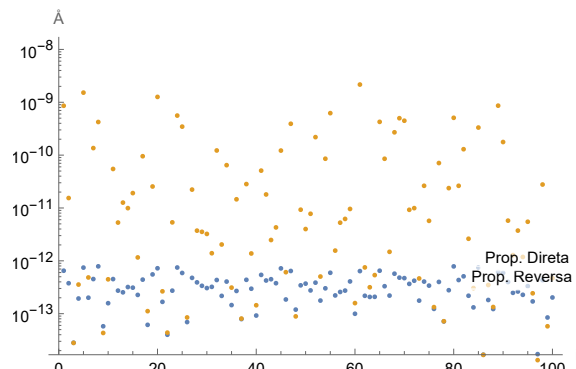
(a) Em azul, os valores de RMSD obtidos na comparação Alg. 1 x Alg. 2 (prop. reversa); em amarelo, Alg. 1 x Alg. 2 (prop. direta).



(b) Em azul, os valores de RMSD obtidos na comparação Alg. 1 x Alg. 3 (prop. reversa); em amarelo, Alg. 1 x Alg. 3 (prop. direta).



(c) Em azul, os valores de RMSD obtidos na comparação Alg. 1 x Alg. 4 (prop. reversa); em amarelo, Alg. 1 x Alg. 4 (prop. direta).



(d) Em azul, os valores de RMSD obtidos na comparação Alg. 1 x Alg. 5 (prop. reversa); em amarelo, Alg. 1 x Alg. 5 (prop. direta).

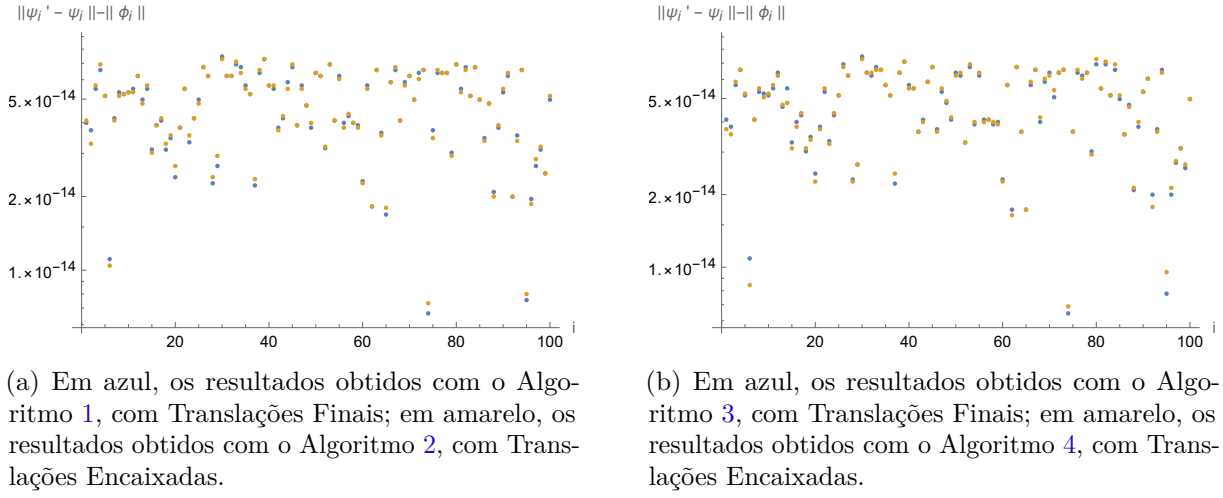
Fonte: Elaborado pelo autor.

3.4 Exemplo detalhado

Separamos essa seção para tratar um exemplo de atualização de coordenadas cartesianas para um caso hipotético em que temos uma molécula pequena, de 8 átomos. Seguimos todas as hipóteses elencadas no capítulo 1 para que a situação se assemelhe, apesar das limitações, ao que acontece na realidade. Além disso, o exemplo será apresentado com base nos Algoritmos 1 e 3 e, sempre que necessário, faremos distinção das diferenças nas etapas usando Rotações Simples e Quatérnios.

Partimos de um conjunto de coordenadas internas em que fixamos como 1.5 Å as distâncias de ligação d_i e 1.91 rad os ângulos de ligação θ_i . Os de torção ψ_i foram escolhidos aleatoriamente dentro do intervalo de $[-\pi, \pi]$. Também observamos que

Figura 29 – Diferenças entre as normas $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$ em cada uma das duplas de algoritmos.



Fonte: Elaborado pelo autor.

$d_1 = \theta_1 = \theta_2 = \psi_1 = \psi_2 = \psi_3 = 0$. Assim, os ângulos de torção escolhidos foram

$$\{0, 0, 0, -2.88422, -2.91555, -0.772891, 2.84816, -0.3755\}.$$

Então, a partir disso, fizemos a conversão de coordenadas, usando um dos métodos apresentados em (CAMARGO, 2021), e encontramos os seguintes pontos:

$$\begin{aligned} x_1 &= (0, 0, 0); & x_5 &= (-4.03, -2.78186, -0.0565105); \\ x_2 &= (-1.5, 0, 0); & x_6 &= (-3.08706, -3.84464, -0.537505); \\ x_3 &= (-1.9991, -1.41453, 0); & x_7 &= (-3.43632, -5.15192, 0.109821); \\ x_4 &= (-3.45517, -1.43003, -0.360054); & x_8 &= (-4.21596, -4.89468, 1.36521). \end{aligned}$$

Vamos supor agora que queremos impor duas alterações em ângulos de torção: $\phi_4 = 0.2$ rad e $\phi_6 = 0.4$ rad. Com isso, temos 3 grupos de átomos: G_0 , representado pelos pontos que não precisam ser atualizados (x_1, x_2 e x_3); G_1 , com os pontos que só recebem a aplicação de uma rotação (x_4 e x_5); e G_2 , composto por x_6, x_7 e x_8 , que recebem a atuação de duas rotações. Sobre a atualização de todos os pontos que compõem G_1 , devemos:

1. Determinar o vetor unitário u que será entendido como eixo de rotação:

$$u = \frac{x_3 - x_2}{1.5} = (-0.332736, -0.94302, 0).$$

2. Determinar a matriz de rotação R_1 usando ϕ_4 e u . Com Rotações Simples, fazemos a construção de acordo com (2.4). Com a representação matricial dos quatérnios,

dado por (2.16), primeiro encontramos $q = \left(\cos \frac{\phi_4}{2}, \sin \frac{\phi_4}{2} u_x, \sin \frac{\phi_4}{2} u_y, \sin \frac{\phi_4}{2} u_z \right) = (0.995004, -0.0332182, -0.0941449, 0)$. Em ambos os casos, temos

$$R_1 = \begin{pmatrix} 0.982273 & 0.00625465 & -0.187349 \\ 0.00625465 & 0.997793 & 0.0661045 \\ 0.187349 & -0.0661045 & 0.980067 \end{pmatrix}.$$

3. Calcular o vetor de translação $t = x'_3 - R_1 x_3$. Como x_3 não precisou ser atualizado, consideramos $x'_3 = x_3$. Depois disso, determinamos a matriz do Espaço Homogêneo que incorpora rotação e translação:

$$t = x'_3 - R_1 x_3 \implies M_1 = \begin{pmatrix} 0.982273 & 0.00625465 & -0.187349 & -0.0265898 \\ 0.00625465 & 0.997793 & 0.0661045 & 0.00938197 \\ 0.187349 & -0.0661045 & 0.980067 & 0.281024 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

4. Aplicar M_1 aos pontos x_4 e x_5 , obtendo

$$\begin{aligned} x'_4 &= M_1 \cdot x_4 = (-3.362, -1.46291, -0.624644); \\ x'_5 &= M_1 \cdot x_5 = (-3.99196, -2.79528, -0.345484). \end{aligned}$$

Finalizado o processo de atualização dos pontos do grupo G_1 , seguimos para as atualizações dos pontos que representam o grupo de átomos G_2 . Para isso, temos que:

1. Determinar o vetor unitário u que será entendido como eixo de rotação:

$$u = \frac{x_5 - x_4}{1.5} = (-0.383221, -0.901216, 0.202362).$$

2. Determinar a matriz de rotação R_2 usando ϕ_6 e u para o método de Rotações Simples. Para os Quatérnios, essa etapa é pulada. Logo,

$$R_2 = \begin{pmatrix} 0.932654 & -0.0515408 & -0.357072 \\ 0.106066 & 0.985175 & 0.134837 \\ 0.344829 & -0.16363 & 0.924294 \end{pmatrix}.$$

3. Compor todas as matrizes de rotação calculadas até aqui por meio do produto matricial, caso o método seja com Rotações Simples. Já com os Quatérnios, depois de encontrar $q = (0.980067, -0.0761343, -0.179044, 0.0402032)$ a partir de ϕ_6 e u , o compomos com o quatérnio encontrado anteriormente. Só depois disso, criamos a matriz de acordo com (2.16). Em ambos os casos, temos

$$R_{[2]} = \begin{pmatrix} 0.852181 & -0.0138093 & -0.523064 \\ 0.13446 & 0.971861 & 0.193406 \\ 0.505675 & -0.235149 & 0.830059 \end{pmatrix}.$$

4. Calcular o vetor de translação $t = x'_5 - R_{[2]}x_5$. Depois disso, determinamos a matriz do Espaço Homogêneo que incorpora rotação e translação:

$$t = x'_5 - R_{[2]}x_5 \implies M_2 = \begin{pmatrix} 0.852181 & -0.0138093 & -0.523064 & -0.625648 \\ 0.13446 & 0.971861 & 0.193406 & 0.461107 \\ 0.505675 & -0.235149 & 0.830059 & 1.08514 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

5. Aplicar M_2 aos pontos x_6, x_7 e x_8 . Com isso, obtemos as atualizações destes:

$$x'_6 = M_2 \cdot x_6 = (-2.92214, -3.79439, -0.0180029);$$

$$x'_7 = M_2 \cdot x_7 = (-3.54032, -4.98665, 0.650106);$$

$$x'_8 = M_2 \cdot x_8 = (-4.86491, -4.59868, 1.23742).$$

Dessa forma, obtivemos o mesmo resultado para a atualização dos pontos dados, tanto com as Rotações Simples, quanto com os Quatérnios. Isso tudo é um bom indício de que o processo atendeu às exigências pedidas, mas ainda há outras a verificar.

Vamos começar analisando as matrizes de distâncias para cada conjunto de coordenadas: as dadas inicialmente e as atualizadas. Nessas matrizes, vamos poder ter acesso às distâncias entre cada par de pontos. Como matrizes de distâncias são simétricas, vamos considerar apenas a parte triangular superior. Logo, para os pontos dados inicialmente, temos

$$D_1 = \begin{pmatrix} 0 & 1.5 & 2.44894 & 3.7567 & 4.89723 & 4.95985 & 6.19376 & 6.60272 \\ & 0 & 1.5 & 2.44894 & 3.76069 & 4.19391 & 5.50488 & 5.76178 \\ & & 0 & 1.5 & 2.44894 & 2.71624 & 4.00571 & 4.34622 \\ & & & 0 & 1.5 & 2.44894 & 3.75148 & 3.9445 \\ & & & & 0 & 1.5 & 2.44894 & 2.5534 \\ & & & & & 0 & 1.5 & 2.44894 \\ & & & & & & 0 & 1.5 \\ & & & & & & & 0 \end{pmatrix}.$$

Já para as coordenadas atualizadas:

$$D_2 = \begin{pmatrix} 0 & 1.5 & 2.44894 & 3.71932 & 4.88556 & 4.78922 & 6.15006 & 6.80782 \\ & 0 & 1.5 & 2.44894 & 3.76069 & 4.05219 & 5.42699 & 5.8311 \\ & & 0 & 1.5 & 2.44894 & 2.55266 & 3.94437 & 4.45902 \\ & & & 0 & 1.5 & 2.44894 & 3.75148 & 3.9445 \\ & & & & 0 & 1.5 & 2.44894 & 2.5534 \\ & & & & & 0 & 1.5 & 2.44894 \\ & & & & & & 0 & 1.5 \\ & & & & & & & 0 \end{pmatrix}.$$

Podemos ver claramente que as distâncias $r_{i-1,i}$, para $2 \leq i \leq n$, e $r_{i-2,i}$, para $3 \leq i \leq n$, permaneceram as mesmas. Podemos ter mais informações do que mudou de uma matriz para outra se fizermos a subtração a seguir. Logo,

$$D_1 - D_2 = \begin{pmatrix} 0 & 0 & 0 & 0.0373875 & 0.0116641 & 0.170626 & 0.0437022 & -0.205095 \\ & 0 & 0 & 0 & 0 & 0.141726 & 0.0778854 & -0.0693173 \\ & & 0 & 0 & 0 & 0.163586 & 0.0613447 & -0.112795 \\ & & & 0 & 2.22 * 10^{-16} & -4.44 * 10^{-16} & 0 & 4.44 * 10^{-16} \\ & & & & 0 & -4.44 * 10^{-16} & -4.44 * 10^{-16} & 0 \\ & & & & & 0 & 0 & 4.44 * 10^{-16} \\ & & & & & & 0 & 0 \\ & & & & & & & 0 \end{pmatrix}.$$

A menos de pequenos resíduos da ordem de 10^{-16} , o resultado atendeu o que foi previsto no início deste capítulo, quando do estudo das alterações em distâncias devido a uma alteração num ângulo de torção ψ_k . Vimos que $r_{i,j}$ não se altera quando $i, j < k$ e quando $i, j \geq k - 2$, de modo que só mudam as distâncias $r_{i,j}$ para $i \leq k - 3$ e $j \geq k$.

Por fim, uma última comparação. Relembremos que os ângulos de torção originais eram os seguintes

$$\{0, 0, 0, -2.88422, -2.91555, -0.772891, 2.84816, -0.3755\},$$

enquanto que os ângulos de torção calculados a partir das coordenadas atualizadas são

$$\{0, 0, 0, -2.68422, -2.91555, -0.372891, 2.84816, -0.3755\}.$$

Isso atende ao que queríamos, que era ter $\psi'_4 = \psi_4 + 0.2$ rad e $\psi'_6 = \psi_6 + 0.4$ rad.

4 Álgebra Geométrica Conforme

No capítulo passado, estávamos interessados em estudar alguns métodos clássicos para a atualização da estrutura 3D de moléculas, em que a manipulação das coordenadas cartesianas dos pontos que representam átomos era feita por meio do uso de matrizes de rotação e de quatérnios. Por conta disso, no capítulo 2 foi feita uma fundamentação sobre alguns aspectos teóricos para cumprir esse objetivo. A partir de agora, a proposta é semelhante, mas com o auxílio da Álgebra Geométrica. Dessa forma, o presente capítulo será destinado a outra breve fundamentação. Inicialmente, vamos abordar a versão euclidiana espacial da Álgebra Geométrica para, só depois, adotar o Modelo Conforme.

4.1 Álgebra Geométrica

A Álgebra Geométrica é um ramo relativamente novo da Matemática que se constitui como um poderoso sistema ao englobar, sob a mesma estrutura, algumas ideias matemáticas importantes, como os quatérnios, a Álgebra de Grassmann, entre outros, oferecendo uma linguagem universal para solucionar problemas geométricos.

O surgimento da Álgebra Geométrica veio como consequência das ideias de Hermann Grassmann, matemático alemão do século XIX que, dentre outras coisas, definiu a noção de *produto interno*, relacionando dois vetores a um escalar, e *produto externo*, gerando segmentos de planos orientados a partir de dois vetores linearmente independentes.

Duas vertentes surgem, ainda no século XIX, a partir das ideias de Grassmann: o Cálculo Vetorial, desenvolvido de forma independente pelo americano Josiah Gibbs e pelo inglês Oliver Heaviside; e a Álgebra Geométrica, também chamada Álgebra de Clifford, cuja alcunha se deve ao seu fundador, o inglês William Clifford. Por um lado, o Cálculo Vetorial ganhou grande notoriedade na comunidade científica no final do século XIX, apesar de utilizar a ideia de espaços vetoriais ao invés de espaços multivetoriais, como sugeriam as ideias de Grassmann; por outro lado, motivada pelo falecimento do seu idealizador aos 34 anos (e apenas um ano após o início do desenvolvimento da teoria), a Álgebra Geométrica não teve grandes avanços por quase um século e ficou, de certa forma, praticamente esquecida na comunidade científica. Foi quando o físico americano David Hestenes retomou o desenvolvimento da teoria, marcado pela publicação do livro “Space-Time Algebra” (HESTENES, 1966), e passou a atuar de forma ativa na disseminação da Álgebra Geométrica pela comunidade científica. Um outro trabalho notório nesse sentido foi (HESTENES, 1987), em que Hestenes utilizou a Álgebra Geométrica para a reescrita da mecânica clássica. Desde então, a Álgebra Geométrica se tornou uma

área emergente da Matemática, apresentando-se como uma alternativa de abordagem em diversas aplicações, como em robótica, computação gráfica e geometria molecular. Esses e outros fatos detalhados podem ser encontrados em (FERNANDES; LAVOR; OLIVEIRA, 2017).

4.1.1 Produtos Interno e Externo

Entre as contribuições de Grassmann que foram incorporadas à Álgebra Geométrica, estão os produtos interno e externo. Já dissemos que eles associam cada par de vetores a, respectivamente, escalares e segmentos de planos orientados. Agora, vamos abordar um pouco mais a ideia deles, o tanto quanto for preciso para seguirmos em frente. Mais detalhes podem ser encontrados em (KANATANI, 2015; FERNANDES; LAVOR; OLIVEIRA, 2017).

Em geral, um produto interno entre vetores (denotado por $\cdot$) retorna um valor escalar que depende da métrica assumida. Independentemente dessa escolha, esse produto atende às seguintes propriedades, para quaisquer $a, b, c \in \mathbb{R}^3$ e $\lambda \in \mathbb{R}$:

$$\begin{aligned} \text{comutatividade} : \quad & a \cdot b = b \cdot a, \\ \text{distributividade} : \quad & a \cdot (b + c) = a \cdot b + a \cdot c, \\ \text{homogeneidade} : \quad & a \cdot (\lambda b) = \lambda(a \cdot b). \end{aligned}$$

É comum ver uma quarta propriedade ser incorporada à essa definição: a positividade, de modo que $a \cdot a \geq 0$, em que a igualdade é assegurada quando a é um vetor nulo. Quando essa propriedade é inclusa, temos uma métrica Euclidiana. Entretanto, ressaltamos que estamos tratando de um produto interno mais geral, em que a positividade não precisa, necessariamente, ser satisfeita. Em (FERNANDES; LAVOR; OLIVEIRA, 2017), encontramos referências a métricas pseudo-Euclidianas, enquanto que em (KANATANI, 2015) diz que, ao perder a positividade, temos uma métrica não-Euclidiana.

Apesar dessa distinção, vamos inicialmente adotar um produto interno que observa justamente a positividade: o produto interno canônico (ou usual). Vamos fazer essa escolha porque queremos apresentar a versão euclidiana espacial da Álgebra Geométrica. A atuação desse produto nos vetores e_1, e_2 e e_3 é então definida pelo *delta de Kronecker*:

$$e_i \cdot e_j = \delta_{ij} = \begin{cases} 1, & \text{se } i = j, \\ 0, & \text{se } i \neq j. \end{cases} \quad (4.1)$$

Por outro lado, o produto externo não depende de métrica. Denotado pelo

símbolo $\wedge$, possui as seguintes propriedades, para quaisquer $a, b, c \in \mathbb{R}^3$ e $\lambda \in \mathbb{R}$:

$$\text{anticomutatividade : } a \wedge b = -b \wedge a,$$

$$\text{distributividade : } a \wedge (b + c) = a \wedge b + a \wedge c,$$

$$\text{homogeneidade : } a \wedge (\lambda b) = \lambda(a \wedge b)$$

$$\text{associatividade : } a \wedge (b \wedge c) = (a \wedge b) \wedge c.$$

Como consequência disso, a atuação do produto externo na base canônica do $\mathbb{R}^3$ se dá pelas seguintes relações (KANATANI, 2015):

$$e_1 \wedge e_1 = e_2 \wedge e_2 = e_3 \wedge e_3 = 0. \quad (4.2)$$

$$e_2 \wedge e_3 = -e_3 \wedge e_2, \quad e_3 \wedge e_1 = -e_1 \wedge e_3, \quad e_1 \wedge e_2 = -e_2 \wedge e_1. \quad (4.3)$$

Pode-se perceber que a propriedade anticomutativa desempenha um forte papel ao se utilizar esse produto. De fato, as relações de (4.3) são apenas um caso particular dessa propriedade e as igualdades de (4.2) são uma consequência. Note que se $a \wedge a = -a \wedge a$, então $a \wedge a = 0$. Em particular, isso se reflete nos vetores de base.

4.1.2 Produto Geométrico

O mais importante e poderoso produto da Álgebra Geométrica é o *produto geométrico* (ou produto de Clifford). Ele é a base criativa para vários elementos que não se limitam apenas a escalares e segmentos de planos orientados, além de ser usado em operações muito úteis, como em rotações e translações. Além do mais, vários outros produtos da Álgebra Geométrica podem ser escritos em função dele.

Com o produto geométrico, possuímos as propriedades associativa e distributiva, mas, em geral, não temos a comutatividade. Inclusive, apesar da anticomutatividade também ser observada em alguns casos, ela não vale geralmente. Um pouco disso está representado em (4.4) e (4.5), em que mostramos como o produto geométrico atua nos vetores canônicos do $\mathbb{R}^3$ (KANATANI, 2015). Não há nenhum símbolo específico para denotá-lo: basta a justaposição dos termos e isso já indica a sua atuação.

$$e_1^2 = e_2^2 = e_3^2 = 1. \quad (4.4)$$

$$e_2 e_3 = -e_3 e_2, \quad e_3 e_1 = -e_1 e_3, \quad e_1 e_2 = -e_2 e_1. \quad (4.5)$$

Quando for conveniente e não houver risco de confusão, vamos representar o produto geométrico $e_i e_j$ por e_{ij} .

As igualdades em (4.4) são consequência do fato de que, se $v = \{x_1, x_2, x_3\} \in \mathbb{R}^3$, então $v^2 = \|v\|^2 = x_1^2 + x_2^2 + x_3^2$. Dada a habitualidade dos produtos e_i^2 , com $i = 1, 2, 3$, é natural procurar entender também a natureza dos elementos $e_i e_j$ presentes em (4.5), em que $i \neq j$. Usando as propriedades do produto geométrico, temos que (LAVOR et al., 2017)

$$(e_i e_j)^2 = (e_i e_j)(e_i e_j) = -(e_j e_i)(e_i e_j) = -e_j(e_i e_i)e_j = -e_j e_j = -1.$$

Como o valor resultante é negativo, é fato que $e_i e_j$ não é um escalar. Pelo mesmo motivo, se fosse um vetor, isso também não aconteceria, pois o valor deveria corresponder ao quadrado da sua norma. Por ser distinto dos escalares e dos vetores e ser escrito como produto geométrico de dois vetores, chamaremos $e_i e_j$ de *bivetor*. O produto entre bivectores quaisquer retorna sempre um escalar.

Analogamente, temos que (LAVOR et al., 2017):

$$(e_1 e_2 e_3)^2 = (e_1 e_2 e_3)(e_1 e_2 e_3) = -e_1 e_2 e_1 e_3 e_2 e_3 = -(e_1 e_1)(e_2 e_2)(e_3 e_3) = -1.$$

Apesar do resultado também ser -1 , há uma distinção entre esse elemento e os anteriores. Se $e_1 e_2 e_3$ fosse um bivetor, então o produto geométrico entre ele e um outro bivetor resultaria em um escalar. Entretanto, sem perda de generalidade, temos que

$$(e_1 e_2 e_3)(e_2 e_3) = -e_1 e_2 e_2 e_3 e_3 = -e_1.$$

Uma vez que o produto geométrico entre $e_1 e_2 e_3$ e um bivetor resulta em um vetor, e seguindo os argumentos anteriores, concluímos que esse elemento não é um escalar, nem um vetor e nem um bivetor: o chamaremos, então, de *trivetor*. Naturalmente, em espaços de dimensões maiores, existem também os quadrvectores, os quintivetores e quantos mais, desde que suas dimensionalidades não excedam a do espaço em questão.

É interessante ver que, com a Álgebra Geométrica, passamos a trabalhar com um espaço multivetorial, de modo que a base é composta pelos elementos de cada porção k -vetorial (FERNANDES; LAVOR; OLIVEIRA, 2017):

$$\left\{ \underbrace{1}_{\text{Escalaes}}, \underbrace{e_1, e_2, e_3}_{\text{Espaço Vetorial}}, \underbrace{e_2 e_3, e_3 e_1, e_1 e_2}_{\text{Espaço Bivectorial}}, \underbrace{e_1 e_2 e_3}_{\text{Espaço Trivectorial}} \right\}.$$

De fato, a quantidade de elementos em cada porção k -vetorial é dada por $\binom{3}{k}$, de modo que o total é dado por 2^3 . Como $\binom{3}{k} = \binom{3}{3-k}$, podemos pensar numa relação entre as porções k e $(3-k)$ -vetoriais. Na verdade, uma é o complemento ortogonal (ou a representação dual) da outra na versão 3D da Álgebra Geométrica. Por consequência disso, geralmente chamamos $e_1 e_2 e_3$ de *pseudoscalar* unitário (FERNANDES; LAVOR; OLIVEIRA, 2017).

Figura 30 – Interpretação geométrica de bivetores e trivetores gerados a partir dos vetores da base canônica.



(a) Bivector gerado pelo produto geométrico entre os vetores e_1 e e_2 , interpretado como um segmento de plano orientado.

(b) Trivector gerado pelo produto geométrico entre os vetores e_1, e_2 e e_3 , interpretado como um volume orientado.

Fonte: Elaborado pelo autor.

Uma vez que possuímos essa base multivetorial, qualquer elemento da Álgebra Geométrica pode ser escrito como

$$m = a_0 + a_1e_1 + a_2e_2 + a_3e_3 + a_4e_2e_3 + a_5e_3e_1 + a_6e_1e_2 + a_7e_1e_2e_3, \quad (4.6)$$

a depender dos escalares a_i , para $i = 0, 1, \dots, 7$. Ou seja, um elemento genérico pode ser visto como uma combinação linear entre escalares, vetores, bivetores e trivetores. Conceber a interpretação geométrica de algo que é a soma de, por exemplo, um vetor com um trivector desafia a nossa capacidade de imaginação do abstrato, mas isso acontece e é natural na Álgebra Geométrica. Claro que isso implica que, como relaciona objetos de naturezas distintas, essa adição não deve ser vista de uma forma habitual, como a adição entre escalares ou entre vetores.

Agora, note que há algumas semelhanças entre a forma como os produtos interno, externo e geométrico atuam nos vetores da base canônica do $\mathbb{R}^3$. A equação (4.1) se assemelha a (4.4), assim como (4.3) é parecida com a equação (4.5). Isso não é à toa. Esses produtos estão relacionados e mostraremos o porquê.

Considere os vetores $a = (a_1, a_2, a_3) = a_1e_1 + a_2e_2 + a_3e_3$ e $b = (b_1, b_2, b_3) = b_1e_1 + b_2e_2 + b_3e_3$. Observando a equação (4.1) e as propriedades do produto interno, temos que

$$\begin{aligned} a \cdot b &= (a_1e_1 + a_2e_2 + a_3e_3) \cdot (b_1e_1 + b_2e_2 + b_3e_3) \\ &= a_1b_1(e_1 \cdot e_1) + a_1b_2(e_1 \cdot e_2) + a_1b_3(e_1 \cdot e_3) + a_2b_1(e_2 \cdot e_1) + \\ &\quad + a_2b_2(e_2 \cdot e_2) + a_2b_3(e_2 \cdot e_3) + a_3b_1(e_3 \cdot e_1) + a_3b_2(e_3 \cdot e_2) + a_3b_3(e_3 \cdot e_3) \\ &= a_1b_1 + a_2b_2 + a_3b_3. \end{aligned} \quad (4.7)$$

Do mesmo modo, sob as regras de (4.2) e (4.3) e as propriedades do produto

externo, temos que

$$\begin{aligned}
 a \wedge b &= (a_1 e_1 + a_2 e_2 + a_3 e_3) \wedge (b_1 e_1 + b_2 e_2 + b_3 e_3) \\
 &= a_1 b_1 e_1 \wedge e_1 + a_1 b_2 e_1 \wedge e_2 + a_1 b_3 e_1 \wedge e_3 + a_2 b_1 e_2 \wedge e_1 + \\
 &\quad + a_2 b_2 e_2 \wedge e_2 + a_2 b_3 e_2 \wedge e_3 + a_3 b_1 e_3 \wedge e_1 + a_3 b_2 e_3 \wedge e_2 + a_3 b_3 e_3 \wedge e_3 \\
 &= (a_2 b_3 - b_3 a_2) e_2 \wedge e_3 + (b_3 a_1 - a_1 b_3) e_3 \wedge e_1 + (a_1 b_2 - b_2 a_1) e_1 \wedge e_2.
 \end{aligned} \tag{4.8}$$

Já o produto geométrico ab resulta em

$$\begin{aligned}
 ab &= (a_1 e_1 + a_2 e_2 + a_3 e_3)(b_1 e_1 + b_2 e_2 + b_3 e_3) \\
 &= a_1 b_1 e_1 e_1 + a_1 b_2 e_1 e_2 + a_1 b_3 e_1 e_3 + a_2 b_1 e_2 e_1 + \\
 &\quad + a_2 b_2 e_2 e_2 + a_2 b_3 e_2 e_3 + a_3 b_1 e_3 e_1 + a_3 b_2 e_3 e_2 + a_3 b_3 e_3 e_3 \\
 &= a_1 b_1 + a_2 b_2 + a_3 b_3 + (a_2 b_3 - b_3 a_2) e_2 e_3 + (a_3 b_1 - b_1 a_3) e_3 e_1 + (a_1 b_2 - b_2 a_1) e_1 e_2.
 \end{aligned} \tag{4.9}$$

Ao observar a presença dos escalares $a_i b_i$ na equação (4.9), para $i = 1, 2, 3$, dá para perceber claramente que o produto interno, da equação (4.7), está incluso no cálculo do produto geométrico. Ao considerarmos $e_i e_j = e_i \wedge e_j$, para $i \neq j$, também podemos concluir que o resultado do produto externo de a por b em (4.8) constitui a outra parte do resultado de ab . Na verdade, essa última consideração não é uma causa, e sim uma consequência da seguinte definição, conhecida como equação fundamental da Álgebra Geométrica (KANATANI, 2015):

$$ab = a \cdot b + a \wedge b, \tag{4.10}$$

para $a, b \in \mathbb{R}^3$.

Note que, quando os vetores a e b são ortogonais, o produto interno é anulado e o produto geométrico se confunde com o produto externo. Por isso que, para $i \neq j$, temos $e_i e_j = e_i \wedge e_j$ e, então, as equações (4.3) e (4.5) se tornam equivalentes. Por outro lado, se a e b são paralelos, então o produto externo é cancelado e o produto geométrico é igual ao produto interno. Assim, a equação (4.4) é equivalente ao caso $i = j$ da equação (4.1).

Portanto, foi por meio da equação (4.10) que o William Clifford unificou as ideias de produto interno e produto externo apresentadas pouco antes por Hermann Grassmann. Estes dois últimos, inclusive, podem ser escritos em função do produto geométrico. Levando em conta que

$$ba = a_1 b_1 + a_2 b_2 + a_3 b_3 + (a_3 b_2 - b_2 a_3) e_2 e_3 + (a_1 b_3 - b_3 a_1) e_3 e_1 + (a_2 b_1 - b_1 a_2) e_1 e_2,$$

temos

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

4.1.3 Rotores

Já vimos que o produto interno entre dois vetores retorna um valor escalar que caracteriza a relação métrica entre eles e que o produto externo, a partir de dois vetores linearmente independentes, gera um segmento de plano orientado. Semelhantemente, também é interessante investigar a natureza do elemento resultante do produto geométrico entre dois vetores. Faremos isso.

Antes, suponha dois multivetores que, em relação àquele da equação (4.6), possuam apenas as porções $(2k)$ -dimensionais, para $k = 0, 1$. Portanto, os escrevemos como

$$\begin{aligned} P_1 &= a_0 + a_1 e_2 e_3 + a_2 e_3 e_1 + a_3 e_1 e_2 \\ P_2 &= b_0 + b_1 e_2 e_3 + b_2 e_3 e_1 + b_3 e_1 e_2. \end{aligned}$$

Um olhar atento aos elementos

$$\begin{aligned} P_1 + P_2 &= (a_0 + a_1 e_2 e_3 + a_2 e_3 e_1 + a_3 e_1 e_2) + (b_0 + b_1 e_2 e_3 + b_2 e_3 e_1 + b_3 e_1 e_2) \\ &= a_0 + b_0 + (a_1 + b_1) e_2 e_3 + (a_2 + b_2) e_3 e_1 + (a_3 + b_3) e_1 e_2 \end{aligned}$$

e

$$\begin{aligned} P_1 P_2 &= (a_0 + a_1 e_2 e_3 + a_2 e_3 e_1 + a_3 e_1 e_2)(b_0 + b_1 e_2 e_3 + b_2 e_3 e_1 + b_3 e_1 e_2) \\ &= a_0 b_0 - a_1 b_1 - a_2 b_2 - a_3 b_3 + (a_3 b_0 + a_2 b_1 - a_1 b_2 + a_0 b_3) e_2 e_3 + \\ &\quad + (a_3 b_1 - a_2 b_0 - a_0 b_2 - a_1 b_3) e_3 e_1 + (a_1 b_0 + a_0 b_1 + a_3 b_2 - a_2 b_3) e_1 e_2 \end{aligned}$$

nos permite concluir que a soma e o produto geométrico entre P_1 e P_2 , além do produto deles por um escalar, resulta sempre num multivetor com a mesma característica: possuir apenas as porções $(2k)$ -dimensionais, para $k = 0, 1$. Isso faz com que, segundo (KANATANI, 2015), o conjunto desses multivetores constitua uma subálgebra da Álgebra Geométrica.

O destaque especial que estamos destinando a esses elementos se explica pelo fato de que, se fizermos

$$i = e_3 e_2 \quad (= -e_2 e_3), \quad j = e_1 e_3 \quad (= -e_3 e_1), \quad k = e_2 e_1 \quad (= -e_1 e_2),$$

não temos outra coisa senão a Álgebra dos Quatérnios como parte da Álgebra Geométrica. Todas aquelas propriedades sobre quatérnios apresentadas em (2.6) e (2.7) são satisfeitas mediante o uso do produto geométrico. Sem perda de generalidade, note que de fato

$$\begin{aligned} jk &= (e_1 e_3)(e_2 e_1) = e_1 e_1 e_3 e_2 = e_3 e_2 = i \\ kj &= (e_2 e_1)(e_1 e_3) = e_2 e_3 = -i \end{aligned}$$

e

$$i^2 = (e_3 e_2)(e_3 e_2) = -e_3 e_2 e_2 e_3 = -1.$$

Dessa forma, um quatérnio q , escrito como $q_0 + q_1i + q_2j + q_3k$, é representado na Álgebra Geométrica por

$$Q = q_0 + q_1e_3e_2 + q_2e_1e_3 + q_3e_2e_1,$$

ressaltando sua vocação para se trabalhar com rotações, que é assunto de extremo interesse nesse trabalho.

Ao olhar o resultado da equação (4.9), pode-se perceber que o produto geométrico de a por b , em que $a, b \in \mathbb{R}^3$, retorna um multivetor apenas com as porções $(2k)$ -dimensionais, para $k = 0, 1$, tal qual os quatérnios na Álgebra Geométrica. Isso sugere uma relação entre esses elementos.

Considere $a, b \in \mathbb{R}^3$, com a imposição de que sejam unitários. Então,

$$\begin{aligned} ba &= (b_1e_1 + b_2e_2 + b_3e_3)(a_1e_1 + a_2e_2 + a_3e_3) \\ &= \underbrace{a_1b_1 + a_2b_2 + a_3b_3}_{c_0} + \underbrace{(a_3b_2 - a_2b_3)}_{c_1}e_2e_3 + \underbrace{(a_1b_3 - a_3b_1)}_{c_2}e_3e_1 + \underbrace{(a_2b_1 - a_1b_2)}_{c_3}e_1e_2 \\ &= c_0 + c_1e_2e_3 + c_2e_3e_1 + c_3e_1e_2 \\ &= c_0 - c_1\underbrace{e_3e_2}_i - c_2\underbrace{e_1e_3}_j - c_3\underbrace{e_2e_1}_k. \end{aligned}$$

Dessa forma, temos um quatérnio unitário como resultado do produto geométrico entre dois vetores também unitários. Assim, $R = ba$, chamado de *rotor*, é o elemento na Álgebra Geométrica responsável por realizar rotações.

Entretanto, essa expressão não indica claramente qual o ângulo usado, nem em relação a qual eixo essa rotação atua. Mas note que, por a e b serem unitários, existe um ângulo θ tal que $\cos\left(\frac{\theta}{2}\right) = c_0$. Com isso, temos

$$\begin{aligned} ba &= \cos\left(\frac{\theta}{2}\right) - c_1e_3e_2 - c_2e_1e_3 - c_3e_2e_1 \\ &= \cos\left(\frac{\theta}{2}\right) - (c_1e_3e_2 + c_2e_1e_3 + c_3e_2e_1). \end{aligned}$$

Fazendo $c = -c_1e_1 - c_2e_2 - c_3e_3$, temos que

$$\begin{aligned} c(e_1e_2e_3) &= (-c_1e_1 - c_2e_2 - c_3e_3)e_1e_2e_3 \\ &= -c_1e_1(e_1e_2e_3) - c_2e_2(e_1e_2e_3) - c_3e_3(e_1e_2e_3) \\ &= -c_1e_2e_3 - c_2e_3e_1 - c_3e_1e_2 \\ &= c_1e_3e_2 + c_2e_1e_3 + c_3e_2e_1. \end{aligned}$$

O termo $e_1e_2e_3$ é o pseudoescalar unitário da versão espacial da Álgebra Geométrica. Assim, podemos reescrever o produto geométrico entre b e a como (KANATANI,

2015):

$$\begin{aligned} ba &= \cos\left(\frac{\theta}{2}\right) - ce_1e_2e_3 \\ &= \cos\left(\frac{\theta}{2}\right) - \underbrace{\sin\left(\frac{\theta}{2}\right) \frac{c}{\sin\left(\frac{\theta}{2}\right)}}_{\text{eixo}} e_1e_2e_3 \end{aligned}$$

Note que na expressão acima conseguimos explicitar o ângulo e o vetor que representa o eixo de rotação. Isso indica que é possível usar uma fórmula para o rotor, equivalente a $R = ba$, em que podemos escolher o ângulo θ e o eixo de rotação. De fato, o rotor assim gerado estará relacionado a dois vetores a e b , mas não importa saber explicitamente quais são, porque geralmente o que precisamos saber é apenas qual o ângulo e o eixo de rotação usado. Assim, um rotor pode ser construído da seguinte forma:

$$R = \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right) \mathbf{b}e_1e_2e_3, \quad (4.11)$$

de modo que $\mathbf{b}$ é o vetor unitário que representa o eixo de rotação.

Lembremos que, com os quatérnios, o produto qvq^{-1} representava o resultado da rotação do vetor v . Semelhantemente, o termo RvR^{-1} , usando o produto geométrico, traduz a rotação de v pelo rotor R . Como R é ortogonal, então $R^{-1} = R^T = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) \mathbf{b}e_1e_2e_3$. De fato, usando as propriedades de distributividade e homogeneidade do produto geométrico, confirmamos a ortogonalidade do rotor:

$$\begin{aligned} RR^{-1} &= \left(\cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right) \mathbf{b}e_1e_2e_3\right) \left(\cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) \mathbf{b}e_1e_2e_3\right) \\ &= \cos^2\left(\frac{\theta}{2}\right) - \sin^2\left(\frac{\theta}{2}\right) \underbrace{(\mathbf{b}e_1e_2e_3)^2}_{-1} \\ &= \cos^2\left(\frac{\theta}{2}\right) + \sin^2\left(\frac{\theta}{2}\right) \\ &= 1. \end{aligned}$$

Por fim, podem haver situações em que precisaremos compor rotações. Anteriormente, quando representávamos as rotações por meio de matrizes, o simples produto matricial ocasionava na sua composição. Semelhantemente, isso é feito na Álgebra Geométrica por meio do produto geométrico. Note que, se um rotor R_1 é aplicado a v , que posteriormente recebe a atuação do rotor R_2 , então temos que

$$\begin{aligned} v' &= R_2(R_1vR_1^{-1})R_2^{-1} \\ &= (R_2R_1)v(R_1^{-1}R_2^{-1}) \\ &= (R_2R_1)v(R_2R_1)^{-1}. \end{aligned}$$

Dessa forma, $R = R_2R_1$ é também um rotor.

4.2 Modelo Conforme

Como nosso problema de aplicação está relacionado ao estudo da estrutura 3D de proteínas, nosso foco vai sempre estar em colher as interpretações geométricas no espaço tridimensional. Isso não significa, no entanto, que o espaço dos objetos matemáticos que vamos usar deva sempre ter a mesma dimensão do espaço físico. Também podemos trabalhá-los em espaços com uma dimensionalidade maior e nos valer de eventuais novas propriedades que possam surgir.

Em geral, ao espaço d -dimensional em que queremos colher as interpretações geométricas, chamamos de *espaço-base*; ao n -dimensional em que trabalhamos os objetos, para $n \geq d$, denominamos por *espaço de representação*. Quando, além de considerar isso, escolhemos uma métrica para este último, temos um *Modelo de Geometria*, que é utilizado para prover interpretação geométrica prática aos produtos e operações da Álgebra Geométrica (FERNANDES; LAVOR; OLIVEIRA, 2017). Na seção anterior, por exemplo, abordamos o Modelo Euclidiano, de modo que o espaço-base e o espaço de representação tinham a mesma dimensionalidade e a métrica escolhida era euclidiana, com o produto interno canônico.

Relembremos que no capítulo 2 trabalhamos com a ideia de isometria, escrita como um movimento rígido do tipo $f(v) = Rv + t$, em que R é uma matriz ortogonal representando uma rotação e t é um vetor de translação. Naquele mesmo capítulo, fizemos uso do Modelo Homogêneo para linearizar a translação e a combinar, num mesmo operador, com a rotação. Agora, queremos ir além e representar uma isometria do $\mathbb{R}^3$ por meio de um operador ortogonal. Ou seja, queremos que a equação

$$X \cdot Y = k||x - y||^2 \quad (4.12)$$

seja satisfeita (LAVOR et al., 2017; LAVOR; SOUZA; ARAGÓN, 2021), em que $x, y \in \mathbb{R}^3$ são representados no Modelo Homogêneo por X e Y , respectivamente, de modo que se $x = (x_1, x_2, x_3)$, então $X = x_1e_1 + x_2e_2 + x_3e_3 + e_4$. A coordenada e_4 é construída ortogonalmente aos vetores canônicos do $\mathbb{R}^3$.

Naturalmente, o produto interno que estamos considerando não é, necessariamente, o mesmo que usamos até aqui. Inclusive, deixaremos em aberto sua definição por enquanto. Independente disso, segue de (4.12) que $X \cdot X = 0$. Isso implica que, se quisermos seguir em frente na tentativa de representar uma isometria por um operador ortogonal, devemos abrir mão da positividade do produto interno. É o que faremos.

Segundo (LAVOR et al., 2017; LAVOR; SOUZA; ARAGÓN, 2021), isometrias do $\mathbb{R}^3$ não podem ser representadas por operadores ortogonais no Modelo Homogêneo, ainda que se abra mão da positividade do produto interno. Já que não funciona para o espaço a 4 dimensões, vamos tentar novamente adicionando mais uma coordenada, assim como feito nas referências citadas acima.

Para o espaço de 5 dimensões que estamos dispostos a trabalhar, vamos considerar novamente os vetores e_1, e_2 e e_3 e que, apenas entre eles, se tenha $e_i \cdot e_j = \delta_{ij}$. Além disso, vamos incluir e_4 e e_5 , ortonormais entre si e em relação aos demais. Assim, um ponto $x = (x_1, x_2, x_3)$ do $\mathbb{R}^3$ é escrito nesse outro espaço como

$$X = \underbrace{x_1 e_1 + x_2 e_2 + x_3 e_3}_x + x_4 e_4 + x_5 e_5 = x + x_4 e_4 + x_5 e_5.$$

Note que, como $X \cdot X = 0$, então

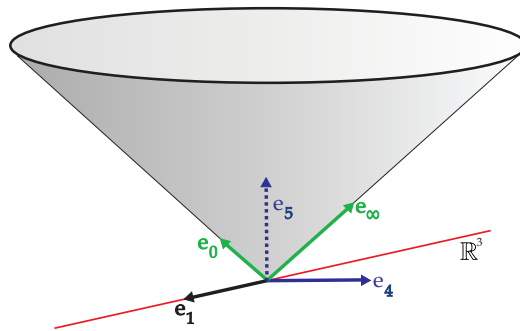
$$\begin{aligned} X \cdot X = 0 &\Rightarrow (x + x_4 e_4 + x_5 e_5) \cdot (x + x_4 e_4 + x_5 e_5) = 0 \\ &\Rightarrow x_4^2 (e_4 \cdot e_4) + x_5^2 (e_5 \cdot e_5) = -||x||^2, \end{aligned}$$

o que implica que ou $e_4 \cdot e_4$ ou $e_5 \cdot e_5$ (ou os dois) seja negativo. Faremos $e_4 \cdot e_4 = 1$ e $e_5 \cdot e_5 = -1$. Entretanto, não estamos satisfazendo, com isso, a relação dada por $X \cdot X = 0$. Mas ao definir (LAVOR et al., 2017)

$$e_0 = \frac{e_5 - e_4}{2} \quad e \quad e_\infty = e_5 + e_4,$$

temos $e_0 \cdot e_0 = 0$ e $e_\infty \cdot e_\infty = 0$. Uma interpretação geométrica desses novos elementos está presente na Figura 31. Como estamos tentando “visualizar” geometricamente um espaço de 5 dimensões, devemos sempre fazer um esforço de imaginação do abstrato e, para isso, geralmente abrimos mão de duas coordenadas do $\mathbb{R}^3$ na interpretação geométrica para poder enxergar as outras duas recém incluídas.

Figura 31 – Vetores e_4 e e_5 dispostos ortogonalmente em relação aos vetores canônicos do $\mathbb{R}^3$, que é representado pela reta vermelha. Além disso, e_0 e e_∞ estão presentes como parte do hipercone formado, conhecido como cone nulo.



Fonte: Baseado em (LAVOR; SOUZA; ARAGÓN, 2021), com algumas alterações feitas pelo autor.

Agora, considerando $X = x_0 e_0 + x + x_\infty e_\infty$ e $Y = y_0 e_0 + y + y_\infty e_\infty$, além do fato de que $e_0 \cdot e_\infty = \frac{e_5 - e_4}{2} \cdot (e_5 + e_4) = -1 = e_\infty \cdot e_0$, temos que

$$\begin{aligned} X \cdot Y &= (x_0 e_0 + x + x_\infty e_\infty) \cdot (y_0 e_0 + y + y_\infty e_\infty) \\ &= x \cdot y - (x_0 y_\infty + x_\infty y_0). \end{aligned}$$

Para o caso em que $X = Y$, de fato

$$X \cdot X = 0 \Rightarrow \|x\|^2 - 2x_0x_\infty = 0.$$

Como consequência, ao definir $x_0 = 1$, temos $x_\infty = \frac{1}{2}\|x\|^2$ e, então,

$$X = e_0 + x + \frac{1}{2}\|x\|^2 e_\infty \quad (4.13)$$

é a representação no espaço de 5 dimensões de um ponto $x \in \mathbb{R}^3$ qualquer. Resta saber se essa representação satisfaz a equação (4.12). Vejamos:

$$\begin{aligned} X \cdot Y &= \left(e_0 + x + \frac{1}{2}\|x\|^2 e_\infty\right) \cdot \left(e_0 + y + \frac{1}{2}\|y\|^2 e_\infty\right) \\ &= x \cdot y - \left(\frac{\|x\|^2}{2} + \frac{\|y\|^2}{2}\right) \\ &= -\frac{1}{2}[x \cdot x - 2(x \cdot y) + y \cdot y] \\ &= -\frac{1}{2}[(x - y) \cdot (x - y)] \\ &= -\frac{1}{2}\|x - y\|^2. \end{aligned} \quad (4.14)$$

Então está feito! Com a condição (4.12) satisfeita, de modo que $k = -\frac{1}{2}$, garantimos a representação de isometrias por meio de operadores ortogonais no espaço a 5 dimensões, como queríamos. Inclusive, $X \cdot Y = -\frac{1}{2}\|x - y\|^2$ é justamente o produto interno deste novo espaço, que denominamos por Modelo (ou Espaço) Conforme. Estamos assumindo uma métrica pseudo-euclidiana, já que perdemos a positividade do produto interno. Inclusive, definido assim, esse produto é semi-negativo.

A representação do produto interno no Modelo Conforme também é dada pelo produto matricial (LAVOR; SOUZA; ARAGÓN, 2021):

$$X \cdot Y = \begin{bmatrix} 1 & x & \frac{\|x\|^2}{2} \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 \\ 0 & I & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ y \\ \frac{\|y\|^2}{2} \end{bmatrix},$$

em que a matriz central corresponde à matriz de métrica associada, consequência da tabela de multiplicação entre cada par de vetores e_i e I é a identidade de ordem 3.

Voltando a equação (4.13), note que é fácil calcular o representante de $x \in \mathbb{R}^3$ no Espaço Conforme. Por outro lado, é mais fácil ainda encontrar o ponto que corresponde a X no $\mathbb{R}^3$: precisamos apenas descartar as coordenadas e_0 e e_∞ .

Ainda na equação (4.13), note que se $x = 0$, então $X = e_0$. Isso indica que a origem do $\mathbb{R}^3$ é representada pela coordenada e_0 (LAVOR et al., 2017).

Além disso, vale a homogeneidade no Modelo Conforme. Isso quer dizer que X e λX , para $\lambda \in \mathbb{R} - \{0\}$, representam o mesmo ponto nesse espaço. Com base nisso, considere os seguintes elementos, em que escolhemos $\lambda = \frac{1}{\|x\|^2/2}$:

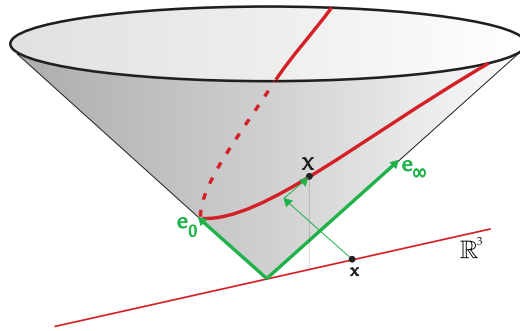
$$X = e_0 + x + \frac{1}{2}\|x\|^2 e_\infty \quad e \quad \frac{X}{\|x\|^2/2} = \frac{e_0}{\|x\|^2/2} + \frac{x}{\|x\|^2/2} + e_\infty.$$

De fato, fazer $\|x\|^2$ tender ao infinito implica numa tendência de $\frac{X}{\|x\|^2/2}$ a e_∞ . Por consequência, o mesmo acontece com X , graças à homogeneidade. Dessa forma, e_∞ é a representação do infinito do $\mathbb{R}^3$ (LAVOR et al., 2017).

É interessante também tentar entender a interpretação geométrica de um ponto no Espaço Conforme. Segundo (LAVOR; SOUZA; ARAGÓN, 2021), os pontos do $\mathbb{R}^3$ são identificados no Modelo Conforme como parte do que é chamado de horosfera, dada pela interseção do cone nulo com um plano que passa por e_0 e é ortogonal ao hiperplano gerado por e_0 e e_∞ .

Mas voltando nossa atenção para a expressão de um ponto $x \in \mathbb{R}^3$ no Modelo Conforme, dada por $X = e_0 + x + \frac{\|x\|^2}{2}e_\infty$, note que o x é primeiro transladado por e_0 e então é alocado sobre a horosfera por meio de outra translação, dada por $\frac{\|x\|^2}{2}e_\infty$. Isso está ilustrado na Figura 32.

Figura 32 – Representação de como um ponto do $\mathbb{R}^3$ é localizado no Modelo Conforme. O lugar geométrico neste último para todos os pontos do $\mathbb{R}^3$ é uma horosfera, ilustrada como uma parábola, em vermelho. Também pode ser visto que a coordenada e_0 representa a origem desse espaço.

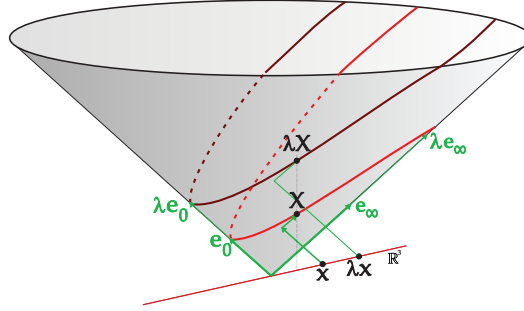


Fonte: Baseado em (LAVOR; SOUZA; ARAGÓN, 2021), com algumas alterações feitas pelo autor.

Como consequência dessa ideia de construção dos pontos no Modelo Conforme, podemos pensar no sentido geométrico da homogeneidade nesse espaço. Assim como acontece com $X = e_0 + x + \frac{\|x\|^2}{2}e_\infty$, podemos localizar x no Espaço Conforme por meio da translação de λx por λe_0 , seguida de outra translação, por $\lambda \frac{\|x\|^2}{2}e_\infty$. Dessa forma, temos

$\lambda X = \lambda e_0 + \lambda x + \lambda \frac{\|x\|^2}{2} e_\infty$ localizado numa orosfera que surge como uma interseção do cone nulo com um hiperplano que passa por λe_0 e é gerado por λe_0 e λe_∞ . Uma ilustração disso está na Figura 33.

Figura 33 – Localização dos pontos X e λX , ambos representando $x \in \mathbb{R}^3$ no Espaço Conforme.



Fonte: Elaborado pelo autor.

4.3 Álgebra Geométrica no Modelo Conforme

Introduzido na Álgebra Geométrica na última década do século XX ([HESTENES; LI; ROCKWOOD, 1999](#)), o Modelo Conforme é ideal para se trabalhar com Geometria Euclidiana. Munido de um produto interno pseudo-euclidiano, os objetos matemáticos são trabalhados no espaço de dimensão 5 para prover interpretações geométricas tridimensionais.

No início deste capítulo, definimos regras de atuação do produto geométrico sobre os vetores da base canônica do $\mathbb{R}^3$. Mas, no Modelo Conforme, inserimos duas novas coordenadas. Então, vamos estender a elas as regras do produto geométrico, lançando as bases, assim, para o que chamaremos de Álgebra Geométrica Conforme (AGC). Dessa forma, temos que

$$e_0^2 = e_\infty^2 = 0, \quad (4.15)$$

$$e_0 e_\infty + e_\infty e_0 = -2, \quad (4.16)$$

$$e_i e_0 = -e_0 e_i \quad (4.17)$$

$$e_i e_\infty = -e_\infty e_i, \quad (4.18)$$

para $i = 1, 2, 3$ ([KANATANI, 2015](#); [LAVOR et al., 2017](#)). Quando o produto geométrico envolve apenas os vetores e_1, e_2 e e_3 , continuam valendo as regras de (4.4) e (4.5).

Agora que estamos trabalhando com a versão conforme da Álgebra Geométrica, é natural questionar se o que definimos anteriormente também vale aqui, além verificar novos ganhos por ter subido de dimensão.

Primeiro, vamos verificar se a equação fundamental da Álgebra Geométrica é também observada no Modelo Conforme. Ao desenvolver o produto geométrico XY , em que $X = e_0 + x + \frac{\|x\|^2}{2}e_\infty$ e $Y = e_0 + y + \frac{\|y\|^2}{2}e_\infty$, além de observar as propriedades acima, temos

$$\begin{aligned} XY &= \left(e_0 + x + \frac{\|x\|^2}{2}e_\infty \right) \left(e_0 + y + \frac{\|y\|^2}{2}e_\infty \right) \\ &= (x - y)e_0 + xy + \frac{\|y\|^2}{2}e_0e_\infty + \frac{\|x\|^2}{2}e_\infty e_0 + \left(\frac{\|y\|^2}{2}x - \frac{\|x\|^2}{2}y \right) e_\infty. \end{aligned}$$

Analogamente,

$$YX = (y - x)e_0 + yx + \frac{\|x\|^2}{2}e_0e_\infty + \frac{\|y\|^2}{2}e_\infty e_0 + \left(\frac{\|x\|^2}{2}y - \frac{\|y\|^2}{2}x \right) e_\infty.$$

Então,

$$\begin{aligned} \frac{1}{2}(XY + YX) &= \frac{1}{2} \left[xy + yx + \left(\frac{\|x\|^2}{2} + \frac{\|y\|^2}{2} \right) \underbrace{(e_0e_\infty + e_\infty e_0)}_{-2} \right] \\ &= \frac{1}{2} [xy + yx - (\|x\|^2 + \|y\|^2)] \\ &= \frac{1}{2} \underbrace{(xy + yx)}_{x \cdot y} - \frac{1}{2}(\|x\|^2 + \|y\|^2) \\ &= -\frac{1}{2}[x \cdot x - 2(x \cdot y) + y \cdot y] \\ &= -\frac{1}{2}[(x - y) \cdot (x - y)] \\ &= -\frac{1}{2}\|x - y\|^2. \end{aligned}$$

De fato, a equação que relaciona os produtos interno e geométrico permanece válida na AGC. Como o produto externo não depende de métrica, nada muda e também a relação entre ele e o produto geométrico se mantém. Isso implica que a equação fundamental da Álgebra Geométrica é também válida no Modelo Conforme:

$$XY = X \cdot Y + X \wedge Y. \quad (4.19)$$

4.3.1 Rotores

Devido a importância que rotações desempenham no nosso problema de aplicação, também é importante verificar como os rotores definidos na seção anterior se comportam na AGC, dada a inclusão de duas novas coordenadas.

Vimos que um rotor pode ser construído na Álgebra Geométrica por meio do produto geométrico entre vetores a e b , de modo que $R = ba$. Como a rotação com rotores

se dá pelo produto $R(\cdots)R^{-1}$, note que

$$\begin{aligned} Re_0R^{-1} &= (ba)e_0(ba)^{-1} = bae_0a^{-1}b^{-1} = e_0baa^{-1}b^{-1} = e_0 \\ Re_\infty R^{-1} &= (ba)e_\infty(ba)^{-1} = bae_\infty a^{-1}b^{-1} = e_\infty baa^{-1}b^{-1} = e_\infty, \end{aligned}$$

desde que a e b sejam unitários (KANATANI, 2015). Com isso, rotações cujo eixo passa pela origem não alteram as coordenadas e_0 e e_∞ . Assim, o rotor R aplicado a um ponto X do Modelo Conforme gera o seguinte elemento:

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

O penúltimo termo implica no último porque a rotação preserva normas.

Como as coordenadas e_0 e e_∞ permanecem inalteradas mediante a ação do rotor R , rotacionar X do Modelo Conforme pode demandar o mesmo custo computacional que rotacionar o seu representante x no $\mathbb{R}^3$. Como visto na expressão acima, isso pode ser feito rotacionando $x \in \mathbb{R}^3$ e, só depois, levando-o ao Espaço Conforme. Por esses motivos, os rotores, definidos na seção anterior, não precisam ser adaptados para atuarem no Modelo Conforme.

4.3.2 Transladadores

No capítulo 2, falamos sobre translações e, no capítulo seguinte, apresentamos métodos para resolução do nosso problema de aplicação em que a translação era essencial. Porém, no presente capítulo ainda não tratamos o assunto.

Na versão euclidiana da Álgebra Geométrica, em que os espaço-base e de representação eram os mesmos, não havia a necessidade de uma atenção maior em relação à translação porque ela acontecia de uma forma bastante natural e habitual. Se $x = x_1e_1 + x_2e_2 + x_3e_3$ é transladado por $t = t_1e_1 + t_2e_2 + t_3e_3$, então o resultado disso é igual a

$$x + t = (x_1 + t_1)e_1 + (x_2 + t_2)e_2 + (x_3 + t_3)e_3.$$

Mas, no Modelo Conforme, isso não é trivial. Se $X = e_0 + x + \frac{\|x\|^2}{2}e_\infty$ e $T = e_0 + t + \frac{\|t\|^2}{2}e_\infty$, então

$$X + T = 2e_0 + (x + t) + \frac{\|x\|^2 + \|t\|^2}{2}e_\infty.$$

Por mais que no Espaço Conforme valha a homogeneidade, permitindo que o coeficiente de e_0 seja qualquer $\lambda \in \mathbb{R} - \{0\}$ ao garantir que X e λX correspondam ao mesmo ponto, o resultado de $X + T$ foge à regra e não representa a translação de X por T . Isso indica a necessidade de definir um operador que seja capaz de realizar translações na AGC. De fato, se isso for feito, então o resultado da translação de X por T deve ser igual a $e_0 + (x + t) + \frac{\|x + t\|^2}{2}e_\infty$.

Em (KANATANI, 2015) encontramos referência a esse operador, denominado *transladador*, que é definido como

$$T_t = 1 - \frac{1}{2}te_\infty, \quad (4.20)$$

em que $t \in \mathbb{R}^3$ representa o vetor pelo qual se realiza a translação. Ao usar $-t$ ao invés de t , temos

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

O formato geral de um transladador é dado por

$$\begin{aligned} T_t &= 1 - \frac{1}{2}(t_1e_1 + t_2e_2 + t_3e_3)e_\infty \\ &= 1 - \frac{t_1}{2}e_1e_\infty - \frac{t_2}{2}e_2e_\infty - \frac{t_3}{2}e_3e_\infty. \end{aligned}$$

Pelo fato de termos chegado ao Modelo Conforme a partir da imposição de que as isometrias deveriam ser escritas por meio de um operador ortogonal, e lembrando que a representamos por meio de um movimento rígido do tipo $f(v) = Rv + t$, em que R é uma rotação e t um vetor de translação, faz sentido pensar que esse último é ortogonal na AGC. De fato,

$$T_t T_t^{-1} = \left(1 - \frac{1}{2}te_\infty\right) \left(1 + \frac{1}{2}te_\infty\right) = 1 - \frac{1}{4}te_\infty te_\infty = 1.$$

Ainda segundo (KANATANI, 2015), o transladador T_t atua sobre e_0 , e_∞ e $x \in \mathbb{R}^3$ da seguinte forma:

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

Portanto, T_t aplicado ao ponto $X = e_0 + x + \frac{\|x\|^2}{2}e_\infty$ produz, como resultado,

o elemento

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

como esperávamos. Portanto, na AGC temos um operador ortogonal para representar uma translação.

4.3.3 Motores

Na primeira fundamentação teórica deste trabalho, em que falamos sobre movimentos rígidos no espaço, trabalhamos com isometrias do tipo $f(v) = Rv + t$, em que R era uma matriz ortogonal que representava uma rotação e t um vetor de translação. Com o advento da AGC, podemos obter um resultado equivalente combinando transladores e rotores. Segundo (KANATANI, 2015),

$$M = T_t R \quad (4.21)$$

é um movimento rígido no Espaço Conforme, de forma que sua computação é dada pelo produto geométrico $M(\cdots)M^{-1}$. O formato desse elemento é dado por

$$M = a_0 + a_1 e_2 e_3 + a_2 e_3 e_1 + a_3 e_1 e_2 + a_4 e_1 e_\infty + a_5 e_2 e_\infty + a_6 e_3 e_\infty + a_7 e_1 e_2 e_3 e_\infty.$$

De fato, por ser construído como produto geométrico entre um translador e um rotor, o motor é um operador que herda a ortogonalidade desses outros dois. Com isso, vale a igualdade $M^{-1} = M^T$, em que

$$M^T = a_0 - a_1 e_2 e_3 - a_2 e_3 e_1 - a_3 e_1 e_2 - a_4 e_1 e_\infty - a_5 e_2 e_\infty - a_6 e_3 e_\infty + a_7 e_1 e_2 e_3 e_\infty.$$

É interessante que o produto geométrico é inversível. Como consequência, há uma fórmula para o cálculo do inverso de multivetores, em geral. Como todos os operadores que estamos tratando na AGC são ortogonais, e dada a facilidade que temos para calcular o inverso desses elementos, não abordaremos tal fórmula, que pode ser encontrada em (FERNANDES; LAVOR; OLIVEIRA, 2017). Entretanto, ela faz uso de uma operação chamada *reversão*, que explica o fato de que, na expressão acima, todos os coeficientes, exceto a_0 e a_7 , foram multiplicados por -1 .

Agora, relembremos que os métodos de atualização de coordenadas cartesianas, apresentados no capítulo anterior, consistiam em transladar um ponto x por meio de um vetor $-t$, aplicar uma rotação e o resultado disso era transladado novamente, dessa vez por t . Isso pode ser resumido na AGC pelo produto geométrico ([CAMARGO, 2021](#)):

$$\begin{aligned} T_t(R(T_{-t}XT_{-t}^{-1})R^{-1})T_t^{-1} &= T_t(R(T_t^{-1}XT_t)R^{-1})T_t^{-1} \\ &= (T_tRT_t^{-1})X(T_tR^{-1}T_t^{-1}) \\ &= (T_tRT_t^{-1})X(T_tRT_t^{-1})^{-1}. \end{aligned}$$

O surgimento do elemento $T_tRT_t^{-1}$ de um lado do produto e o seu inverso do outro nos faz questionar a natureza do mesmo. Para entender isso, considere o seguinte:

$$T_tRT_t^{-1} = T_tRT_{-t}R^{-1}R = T_t(RT_{-t}R^{-1})R.$$

Entretanto,

$$\begin{aligned} RT_{-t}R^{-1} &= R\left(1 + \frac{1}{2}te_{\infty}\right)R^{-1} \\ &= RR^{-1} + \frac{1}{2}Rte_{\infty}R^{-1} \\ &= 1 + \frac{1}{2}RtR^{-1}Re_{\infty}R^{-1} \\ &= 1 + \frac{1}{2}RtR^{-1}e_{\infty} \\ &= T_{-RtR^{-1}}. \end{aligned}$$

Dessa forma, $T_tRT_t^{-1}$ pode ser reescrito como

$$\begin{aligned} T_tRT_t^{-1} &= T_t(RT_{-t}R^{-1})R \\ &= T_tT_{-RtR^{-1}}R \\ &= T_{t-RtR^{-1}}R, \end{aligned}$$

que nada mais é do que um motor. Portanto, esse novo elemento que surge na AGC nos permite realizar rotações em torno de eixos arbitrários e que não passam, necessariamente, pela origem. De fato, sua construção se dá pelo produto geométrico entre um translador e um rotor, mas o resultado de sua aplicação a um ponto já retorna a atualização direta deste último, no sentido de que não tem necessidade de ir à origem e retornar.

5 Atualização de Coordenadas Conformes

Este é mais um capítulo sobre métodos de atualização de coordenadas dos pontos que representam as localizações dos átomos de uma molécula devido a alterações em ângulos de torção. Entretanto, agora vamos utilizar todo o ferramental disposto pela Álgebra Geométrica Conforme (AGC). Boa parte da ideia dos métodos apresentados anteriormente será mantida, porém com alguns ajustes e aperfeiçoamentos. Além disso, vamos sempre considerar a propagação reversa na aplicação das rotações.

Para trabalhar com o Modelo Conforme, faremos uso do pacote *CGAlgebra* incorporado ao software *Wolfram Mathematica*. Ele é muito útil para realizar cálculos diretamente com elementos e produtos da Álgebra Geométrica Conforme. Em (ORTIZ-DURAN; ARAGON, 2017) pode ser encontrado um tutorial de uso e o link para baixá-lo.

Os métodos que vamos apresentar nesse capítulo são pensados para serem usados com os elementos da AGC, explorando todo o poder de representação e interpretação geométrica que essa nova ferramenta nos dispõe. Entretanto, vamos sempre fazer comentários de como implementar esses métodos à luz da Álgebra Linear, representando matricialmente os rotores e os transladores e sem usar os produtos da AGC. Apesar de não ser o propósito principal, pensamos que isso pode ser útil para mostrar que o novo caminho proposto funciona e que, se quisermos, podemos também buscar seus representantes para trabalhar num ambiente um pouco mais habitual.

Para começar, relembremos que um ponto $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ é escrito no Espaço Conforme como

$$X = e_0 + x_1 e_1 + x_2 e_2 + x_3 e_3 + \frac{1}{2} \|x\|^2 e_\infty.$$

Como queremos trabalhar uma versão matricial dos elementos da AGC, é útil estabelecer uma outra forma de representação de X :

$$X = \left(1, x_1, x_2, x_3, \frac{1}{2} \|x\|^2\right).$$

Se a conversão das coordenadas cartesianas para conformes é simples, a volta é ainda mais fácil! A única coisa que precisamos fazer é considerar a parte tridimensional, nos desfazendo das coordenadas de e_0 e e_∞ .

Uma vez que o trânsito entre as representações cartesianas e conformes de um ponto é feito de uma maneira bastante simples, vamos considerar que alterações nos ângulos de torção impactam diretamente as coordenadas conformes, sem necessidade de um passo intermediário por essa outra representação. Para isso, devemos partir do pressuposto de que a conversão de coordenadas internas para conformes foi devidamente realizada.

5.1 Métodos de Atualização de Coordenadas Conformes

Nos processos descritos no capítulo 3, vimos que é extremamente necessário realizar rotações em torno de eixos arbitrários que não passam pela origem. A solução para isso sempre foi transladar segmentos até a origem, porque só conseguíamos rotacionar em torno de eixos que passavam por lá. Depois disso, outra translação era aplicada para que, assim, o ponto representante de um átomo fosse anexado corretamente à estrutura atualizada. No capítulo anterior, vimos que esse processo pode ser descrito na AGC como

$$(T_t R T_t^{-1}) X (T_t R T_t^{-1})^{-1},$$

em que t é um vetor de translação. Também vimos que $T_t R T_t^{-1}$ é representado por um operador ortogonal, chamado de motor e representado por M , de modo que

$$M = T_v R, \quad (5.1)$$

em que $v = t - R t R^{-1}$.

Assim, com o auxílio desse operador, torna-se possível, de uma forma direta, rotacionar em torno de eixos arbitrários que não passam, necessariamente, pela origem. Vamos buscar ainda uma simplificação para o produto geométrico $T_v R$. Note que

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

O primeiro termo da expressão acima representa a parte rotacional do motor, enquanto que o segundo está relacionado com o deslocamento da rotação. Em relação a ele, considerando $R = \cos\left(\frac{\theta}{2}\right) - \text{sen}\left(\frac{\theta}{2}\right) \mathbf{b} e_{123}$ e fazendo $c = \cos\left(\frac{\theta}{2}\right)$ e $s = \text{sen}\left(\frac{\theta}{2}\right)$, temos que

$$\begin{aligned} (t R - R t) e_\infty &= (t(c - s \mathbf{b} e_{123}) - (c - s \mathbf{b} e_{123})t) e_\infty \\ &= (ct - s t \mathbf{b} e_{123} - ct + s \underbrace{\mathbf{b} e_{123} t}_{t e_{123}}) e_\infty \\ &= -s t \mathbf{b} e_{123} e_\infty - ct + s \mathbf{b} t e_{123} e_\infty \\ &= -s \underbrace{(t \mathbf{b} - \mathbf{b} t)}_{2t \wedge \mathbf{b}} e_{123} e_\infty \\ &= -2s(t \wedge \mathbf{b}) e_{123} e_\infty. \end{aligned}$$

Com isso, podemos reescrever o motor M :

$$\begin{aligned}
 M &= R - \frac{1}{2}(tR - Rt)e_\infty \\
 &= R - \frac{1}{2}(-2s(t \wedge \mathbf{b})e_{123\infty}) \\
 &= c - s\mathbf{b}e_{123} + s(t \wedge \mathbf{b}) \underbrace{e_{123\infty}}_{-e_{\infty 123}} \\
 &= c - s\mathbf{b}e_{123} - s(t \wedge \mathbf{b})e_{\infty 123} \\
 &= c - s(\mathbf{b} + (t \wedge \mathbf{b})e_\infty)e_{123} \\
 &= \cos\left(\frac{\theta}{2}\right) - \sin\left(\frac{\theta}{2}\right)(\mathbf{b} + (t \wedge \mathbf{b})e_\infty)e_{123}.
 \end{aligned} \tag{5.2}$$

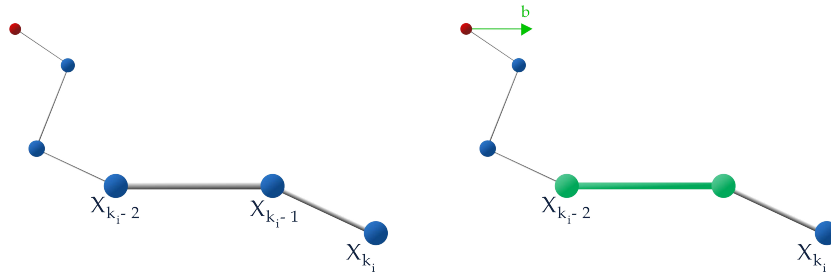
Esse resultado também é encontrado em (CAMARGO, 2021), mas partindo de $T_t R T_t^{-1}$ e não de $T_v R$. Ter essa simplificação é importante porque nos permite evitar operações desnecessárias. Falaremos sobre isso mais a frente.

Seja X_{k_i} , com $k_i \geq 4$, o primeiro ponto do grupo de átomos G_i , que são aqueles que devem ter as coordenadas conformes atualizadas devido a uma alteração ϕ_{k_i} no ângulo de torção ψ_{k_i} . A primeira coisa a se fazer é criar um eixo de rotação que passa pela origem e é construído a partir dos pontos X_{k_i-1} e X_{k_i-2} . Podemos buscar os representantes deles no $\mathbb{R}^3$, extraindo suas partes tridimensionais, e fazer

$$\mathbf{b} = \frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}, \tag{5.3}$$

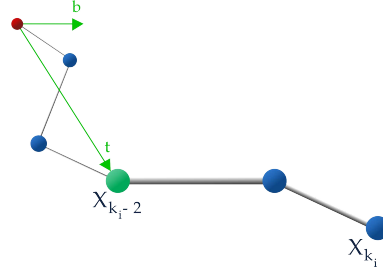
de modo que $\mathbf{b} = b_1e_1 + b_2e_2 + b_3e_3$ para os devidos b_1, b_2 e b_3 . O vetor b está representado na Figura 34.

Figura 34 – Vetor unitário b calculado a partir dos pontos X_{k_i-1} e X_{k_i-2} .



Fonte: Elaborado pelo autor.

Em seguida, precisamos calcular o motor responsável pela atualização das coordenadas conformes de todos os pontos de G_i . Usando o vetor unitário $\mathbf{b}$ e o ângulo ϕ_{k_i} , nos resta definir o vetor t para calcular o motor da equação (5.2). Podemos tomá-lo como a parte tridimensional de X_{k_i-2} , ou seja, $t = x_{k_i-2}$, e assim faremos como os métodos do capítulo 3. O vetor b está representado na Figura 35.

Figura 35 – Representação geométrica do vetor t .

Fonte: Elaborado pelo autor.

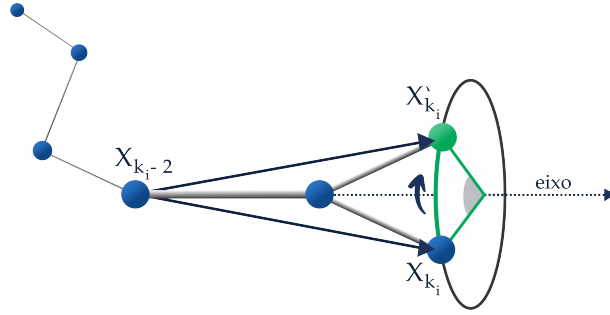
Depois disso, devemos realizar a composição, por meio do produto geométrico, desse motor com os outros já gerados (se houverem). O resultado disso será também um motor, que chamaremos de $M_{[i]}$.

Por fim, todos os pontos do grupo G_i terão suas coordenadas atualizadas por meio da aplicação de $M_{[i]}$, ou seja, realizando o produto geométrico $M_{[i]}X_mM_{[i]}^{-1}$, para $k_i \leq m < k_{i+1}$. Note que, assim, estamos usando o modelo de Translações Encaixadas, pois todos os pontos do grupo de átomos são atualizados por meio de um mesmo operador, que possui a rotação e a translação embutidas.

Intuitivamente, podemos enxergar esse processo como tomar o segmento que liga X_{k_i-2} a X_m e rotacioná-lo diretamente onde ele está, sem levá-lo à origem e voltar. Uma ilustração disso está na Figura 36. Esse método está resumido no Algoritmo 6, que também pode ser encontrado também em (CAMARGO, 2021), com algumas diferenças.

Algoritmo 6 – Atualização de coordenadas conformes usando Motores e Translações Encaixadas	
Entrada: $Co = \{X_1, X_2, \dots, X_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$	
Saída: $Co' = \{X'_1, X'_2, \dots, X'_n\}$	
1	início
2	$Co' \leftarrow Co$;
3	$M_{[0]} \leftarrow 1$;
4	para $i = 1, \dots, p$ faça
5	$\mathbf{b} \leftarrow \frac{x_{k_i-1} - x_{k_i-2}}{\ x_{k_i-1} - x_{k_i-2}\ }$;
6	$I \leftarrow (\mathbf{b} + (x_{k_i-2} \wedge \mathbf{b})e_\infty)e_{123}$;
7	$M \leftarrow \cos\left(\frac{\phi_{k_i}}{2}\right) - \sin\left(\frac{\phi_{k_i}}{2}\right)I$;
8	$M_{[i]} \leftarrow M_{[i-1]}M$;
9	para $k_i \leq m < k_{i+1}$ faça
10	$X'_m \leftarrow M_{[i]}X_mM_{[i]}^{-1}$;
11	fim
12	fim
13	fim

Figura 36 – Rotação do segmento que liga os pontos X_{k_i-2} e X_{k_i} em relação a um eixo que não passa pela origem, gerando X'_{k_i} .



Fonte: Elaborado pelo autor.

Para a contagem de operações, vamos novamente considerar n_r o número de ângulos de torção alterados e n_a a quantidade de pontos que devem ser atualizados. Para cada ângulo de torção que se altera, são necessárias: 3 somas e 3 divisões para calcular U ; 6 multiplicações e 3 somas para determinar I ; 6 multiplicações para calcular o motor R ; e 48 multiplicações e 40 somas para calcular o produto entre motores. Assim, temos ao todo 109 operações para cada ângulo de torção que se altera. Além disso, cada atualização de um ponto custa 42 multiplicações e 44 somas. Portanto, esse método possui uma estimativa de custo computacional de $109 n_r + 86 n_a$ operações.

Agora, vamos encontrar também a versão matricial do motor (5.2). Em geral, temos que um translador construído a partir de $t = t_1 e_1 + t_2 e_2 + t_3 e_3$ é escrito como $T_t = 1 - \frac{t_1}{2} e_{1\infty} - \frac{t_2}{2} e_{2\infty} - \frac{t_3}{2} e_{3\infty}$, enquanto que um rotor $R = \cos \frac{\theta}{2} - \text{sen} \frac{\theta}{2} \mathbf{b} e_{123}$ tem a forma $R = q_1 + q_2 e_{32} + q_3 e_{13} + q_4 e_{21}$. Então,

$$T_t R T_t^{-1} = q_1 + q_2 e_{32} + q_3 e_{13} + q_4 e_{21} + (q_3 t_3 - q_4 t_2) e_{1\infty} + (q_4 t_1 - q_2 t_3) e_{2\infty} + (q_2 t_2 - q_3 t_1) e_{3\infty}.$$

Por simplificação, considere $a_1 = q_3 t_3 - q_4 t_2$, $a_2 = q_4 t_1 - q_2 t_3$ e $a_3 = q_2 t_2 - q_3 t_1$. Fazendo essa substituição, aplicamos M a cada uma das coordenadas que compõem o Espaço Conforme, ou seja, fazemos o produto geométrico $M e_i M^{-1}$, para $i = 0, 1, 2, 3, \infty$, e cada resultado será alocado numa coluna. Como consequência, temos a seguinte matriz:

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -2a_1 q_1 - 2a_3 q_3 + 2a_2 q_4 & q_1^2 + q_2^2 - 1 & 2q_2 q_3 - 2q_1 q_4 & 2q_2 q_4 + 2q_1 q_3 & 0 \\ -2a_2 q_1 + 2a_3 q_2 - 2a_1 q_4 & 2q_2 q_3 + 2q_1 q_4 & q_1^2 + q_3^2 - 1 & -2q_1 q_2 + 2q_3 q_4 & 0 \\ -2a_3 q_1 - 2a_2 q_2 + 2a_1 q_3 & 2q_2 q_4 - 2q_1 q_3 & 2q_1 q_2 + 2q_3 q_4 & q_1^2 + q_4^2 - 1 & 0 \\ 2a_1^2 + 2a_2^2 + 2a_3^2 & -2a_1 q_1 + 2a_3 q_3 - 2a_2 q_4 & -2a_2 q_1 - 2a_3 q_2 + 2a_1 q_4 & -2a_3 q_1 + 2a_2 q_2 - 2a_1 q_3 & 1 \end{pmatrix}.$$

Portanto, se no Algoritmo 6, ao invés de usarmos os elementos e os produtos da AGC, quisermos usar uma versão matricial, devemos, para atualizar os pontos do grupo de átomos G_i ,

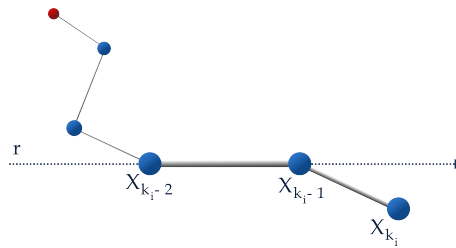
1. calcular b como $\frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$;
2. determinar $R = q_1 + q_2 e_{32} + q_3 e_{13} + q_{21}$;
3. calcular a matriz $\mathbf{M}$ a partir de R e $t = x_{k_i-2}$;
4. compor, por meio do produto matricial, todas as matrizes já construídas, gerando $\mathbf{M}_{[i]}$;
5. aplicar $\mathbf{M}_{[i]}$ aos pontos X_m , para $k_i \leq m < k_{i+1}$.

Ainda há uma outra forma de usar a AGC para encontrar o motor que realiza rotações em torno de um eixo arbitrário que não passa pela origem. No último algoritmo apresentado, o bivector que é utilizado para construir o motor é encontrado a partir de manipulações feitas com o produto geométrico T_v (ou $T_t R T_t^{-1}$). Agora, vamos encontrá-lo de outra forma, um pouco mais intuitiva e sem a necessidade de envolver a translação na sua construção.

Suponha novamente que queremos atualizar os pontos que pertencem ao grupo de átomos G_i , de modo que o primeiro ponto a ser atualizado é X_{k_i} . De fato, o eixo de rotação deve ser construído a partir de X_{k_i-1} e X_{k_i-2} . Então, assim faremos. Para isso, vamos construir uma reta que passa por esses dois pontos, representada na Figura 37 e dada na AGC por

$$r = X_{k_i-1} \wedge X_{k_i-2} \wedge e_\infty.$$

Figura 37 – Representação da reta que passa pelos pontos X_{k_i-1} e X_{k_i-2} .



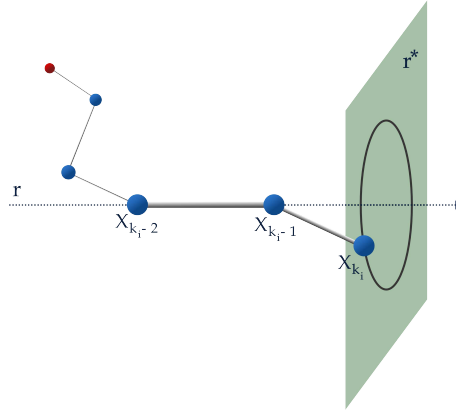
Fonte: Elaborado pelo autor.

Como vimos no capítulo 3, a circunferência que contém todas as possíveis atualizações de X_{k_i} habita em um plano que é perpendicular à reta que passa pelos dois pontos imediatamente anteriores. Já temos a reta r . O plano perpendicular a ela pode ser encontrado calculando o seu *dual*. Na Álgebra Geométrica, essa operação quando aplicada a um elemento k -dimensional, sempre retorna o seu “equivalente” de dimensão $(n - k)$, em que n é a dimensão do espaço como um todo. Como a reta tem 3 dimensões na AGC, o seu dual é um bivector, que pode ser visto como um plano de rotação no qual o motor

atua. Como precisamos de um bivector unitário, apenas o dividimos pela sua norma (que no pacote CGAlgebra pode ser calculado usando a função *Magnitude*). Dessa forma,

$$I = \frac{(X_{k_i-1} \wedge X_{k_i-2} \wedge e_\infty)^*}{\|(X_{k_i-1} \wedge X_{k_i-2} \wedge e_\infty)^*\|}.$$

Figura 38 – Plano de rotação definido como o dual da reta r que passa pelos pontos X_{i-2} e X_{i-1} . Em destaque sobre o plano, a circunferência onde habitam as possíveis atualizações do ponto X_i .



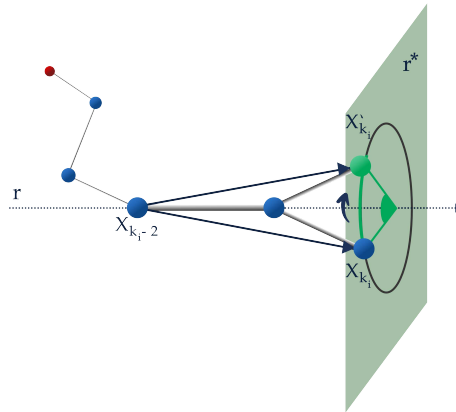
Fonte: Elaborado pelo autor.

Depois disso, calculamos o motor como

$$M = \cos\left(\frac{\phi_{k_i}}{2}\right) - \sin\left(\frac{\phi_{k_i}}{2}\right) I.$$

Após realizarmos a composição desse motor com os outros já definidos, gerando $M_{[i]}$, o aplicamos a todos os pontos do grupo G_i , ou seja, realizamos o produto geométrico $M_{[i]}X_mM_{[i]}^{-1}$, para $k_i \leq m < k_{i+1}$. Isso está ilustrado na Figura 39.

Figura 39 – Atualização das coordenadas de pontos com a aplicação do motor $M_{[i]}$.



Fonte: Elaborado pelo autor.

Essa versão é descrita resumidamente no Algoritmo 7.

Algoritmo 7 – Atualização de coordenadas conformes com Motores e Translações Encaixadas - versão 2	
Entrada: $Co = \{X_1, X_2, \dots, X_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$	
Saída: $Co' = \{X'_1, X'_2, \dots, X'_n\}$	
1	início
2	$Co' \leftarrow Co$;
3	$R_{[0]} \leftarrow 1$;
4	para $i = 1, \dots, p$ faça
5	$I \leftarrow \frac{(X_{k_i-1} \wedge X_{k_i-2} \wedge e_\infty)^*}{\ (X_{k_i-1} \wedge X_{k_i-2} \wedge e_\infty)^*\ }$;
6	$M \leftarrow \cos\left(\frac{\phi_{k_i}}{2}\right) - \sin\left(\frac{\phi_{k_i}}{2}\right) I$;
7	$M_{[i]} \leftarrow M_{[i-1]} M$;
8	para $k_i \leq m < k_{i+1}$ faça
9	$X'_m \leftarrow M_{[i]} X_m M_{[i]}^{-1}$;
10	fim
11	fim
12	fim

Aqui, para cada ângulo de torção que se altera, são necessárias: 12 multiplicações e 6 somas para determinar I ; 6 multiplicações para calcular o motor M ; e 48 multiplicações e 44 somas para calcular o produto entre motores, totalizando 112 operações para cada ângulo de torção que se altera. Além disso, cada atualização de um ponto custa 42 multiplicações e 44 somas. Portanto, o custo computacional estimado para esse método é de $112 n_r + 86 n_a$ operações.

O motor usado é o mesmo do algoritmo anterior, se diferenciando apenas na forma de encontrá-lo. Com isso, a matriz que o representa é a mesma B acima e, consequentemente, a versão matricial desse algoritmo também é a mesma do algoritmo anterior.

Portanto, com o uso da AGC também temos o poder de realizar rotações em torno de eixos arbitrários que não passam necessariamente pela origem, evitando assim a aplicação de translações de ida e volta.

Note que, já que estamos usando motores construídos como $M = T_v R = T_t R T_t^{-1}$, temos que efetuar um produto geométrico entre dois elementos desse para cada ângulo de torção que é alterado e, para cada ponto que deve ser atualizado, devemos aplicar também esse operador. Acontece que os motores que trabalhamos têm a forma

$$M = m_1 + m_2 e_{23} + m_3 e_{31} + m_4 e_{12} + m_5 e_{1\infty} + m_6 e_{2\infty} + m_7 e_{3\infty},$$

o que faz com que seja alto o custo computacional para efetuar essas operações. Vamos tentar uma alternativa em que aplicamos a rotação e a translação separadamente. Dessa forma, também abriremos mão das Translações Encaixadas e usaremos as Translações Finais.

Seja novamente o grupo de átomos G_i , composto pelos pontos X_m , com $k_i \leq m < k_{i+1}$, que devem ser atualizados devido a uma alteração ϕ_{k_i} no ângulo de torção ψ_{k_i} . Para atualiza-los, precisamos definir um rotor com base num vetor unitário construído a partir $X_{k_{i-1}}$ e $X_{k_{i-2}}$. Então, novamente temos

$$\mathbf{b} = \frac{x_{k_{i-1}} - x_{k_{i-2}}}{\|x_{k_{i-1}} - x_{k_{i-2}}\|} = b_1 e_1 + b_2 e_2 + b_3 e_3.$$

Com isso, o rotor que queremos calcular, que realiza rotações em torno de um eixo arbitrário que passa pela origem, é definido como

$$R = \left(\frac{\phi_{k_i}}{2} \right) - \text{sen} \left(\frac{\phi_{k_i}}{2} \right) \mathbf{b} e_{123}. \quad (5.4)$$

Depois disso, definimos como $R_{[i]}$ o produto geométrico entre esse rotor e os outros já gerados (se houver). Feita essa composição, devemos agora nos atentar à atualização das coordenadas conformes de cada ponto do grupo G_i .

Agora, precisamos selecionar um segmento que liga dois átomos para aplicar a rotação. Assim como nos dois algoritmos anteriores, vamos considerar cada segmento indo de $X_{k_{i-2}}$ a X_m , em que esse último representa todos os pontos de G_i , uma vez que $k_i \leq m < k_i + 1$. Como o rotor (5.4) realiza rotações em torno de eixos que passam pela origem, definimos esses segmentos como

$$l = x_m - x_{k_{i-2}}. \quad (5.5)$$

Aplicando o rotor a esse elemento, temos como resultado $t = R_{[i]} l R_{[i]}^{-1} = t_1 e_1 + t_2 e_2 + t_3 e_3$. O próximo passo é criar um translador a partir de t . Esse é o motivo, inclusive, pelo qual definimos (5.5) descartando as coordenadas e_0 e e_∞ : para que t seja um vetor do $\mathbb{R}^3$, já que é assim que construímos transladores. Dessa forma, temos

$$T_t = 1 - \frac{1}{2} t e_\infty. \quad (5.6)$$

Feito isso, resta apenas aplicar o translador T_t ao ponto $X'_{k_{i-2}}$:

$$X'_m = T_t X'_{k_{i-2}} T_t^{-1}.$$

A interpretação desse processo é que, ao tomar o segmento que liga $X_{k_{i-2}}$ a X_m , o levamos até a origem, onde foi rotacionado, e em seguida transladamos por ele o ponto $X'_{k_{i-2}}$, gerando a atualização de X_m , devidamente anexada na estrutura atualizada. Esse método está descrito no Algoritmo 8. Em (CAMARGO, 2021) também o encontramos (com pequenas diferenças).

Algoritmo 8 – Atualização de coordenadas conformes usando Rotores e Transladores e Translações Finais	
Entrada: $Co = \{X_1, X_2, \dots, X_n\}$, $P = \{k_1, k_2, \dots, k_p\}$, $A = \{\phi_{k_1}, \phi_{k_2}, \dots, \phi_{k_p}\}$	
Saída: $Co' = \{X'_1, X'_2, \dots, X'_n\}$	
1	início
2	$Co' \leftarrow Co;$
3	$R_{[0]} \leftarrow 1;$
4	para $i = 1, \dots, p$ faça
5	$b \leftarrow \frac{x_{k_{i-1}} - x_{k_{i-2}}}{ x_{k_{i-1}} - x_{k_{i-2}} };$
6	$R \leftarrow Cos\left(\frac{\phi_{k_i}}{2}\right) - Sen\left(\frac{\phi_{k_i}}{2}\right) be_{123};$
7	$R_{[i]} \leftarrow R_{[i-1]}R;$
8	para $k_i \leq m < k_{i+1}$ faça
9	$l \leftarrow x_m - x_{k_{i-2}};$
10	$t \leftarrow R_{[i]}lR_{[i]}^{-1};$
11	$T \leftarrow 1 - \frac{1}{2}te_{\infty};$
12	$X'_m \leftarrow TX'_{k_{i-2}}T^{-1};$
13	fim
14	fim
15	fim

Para cada ângulo de torção que se altera, são necessárias: 3 somas e 3 divisões para calcular U ; 4 multiplicações para calcular o rotor R ; e 16 multiplicações e 12 somas para calcular o produto entre rotores. Assim, temos ao todo 38 operações para cada ângulo de torção que se altera. Além disso, precisamos de: 3 somas para calcular l ; 22 multiplicações e 16 somas para encontrar t ; 4 produtos e 5 somas para aplicar o translador, totalizando 26 multiplicações e 24 somas. Portanto, esse método possui uma estimativa de custo computacional de $38 n_r + 50 n_a$ operações.

Assim como foi feito anteriormente, vamos também exibir a versão matricial dos elementos que usamos na AGC. Para obter a matriz que representa o rotor da equação (5.4), fazemos o produto geométrico $Re_i R^{-1}$, em que $i = 0, 1, 2, 3, \infty$. Alocando cada resultado desse em uma coluna, temos a matriz

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 2q_1^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_1q_4 & 2q_2q_4 + 2q_1q_3 & 0 \\ 0 & 2q_2q_3 + 2q_1q_4 & 2q_1^2 + 2q_3^2 - 1 & 2q_3q_4 - 2q_1q_2 & 0 \\ 0 & 2q_2q_4 - 2q_1q_3 & 2q_3q_4 + 2q_1q_2 & 2q_1^2 + 2q_3^2 - 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

de modo que usamos a representação de um rotor R como $q_1 + q_2e_{32} + q_3e_{13} + q_{21}$. A definição da primeira e da última coluna vai de acordo com o fato de que $Re_0R^{-1} = e_0$ e $Re_{\infty}R^{-1} = e_{\infty}$, que vimos no capítulo anterior. Isso faz com que o rotor definido na versão

euclidiana da Álgebra Geométrica não só funcione na AGC, como também tenha o mesmo custo computacional na sua aplicação. O bloco 3×3 central é justamente a representação matricial dos quatérnios, pois também vimos que eles são representados por rotores na AGC.

Um translador também tem sua versão matricial. Da mesma forma, efetuamos o produto geométrico Te_iT^{-1} , em que $i = 0, 1, 2, 3, \infty$, temos a matriz associada a (5.6):

$$\mathbf{T}_t = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ t_1 & 1 & 0 & 0 & 0 \\ t_2 & 0 & 1 & 0 & 0 \\ t_3 & 0 & 0 & 1 & 0 \\ t^2/2 & t_1 & t_2 & t_3 & 1 \end{pmatrix}.$$

Com essa versão matricial do Algoritmo 8, para atualizar os pontos do grupo G_i devemos:

1. calcular $\mathbf{b}$ como $\frac{x_{k_i-1} - x_{k_i-2}}{\|x_{k_i-1} - x_{k_i-2}\|}$;
2. determinar $R = q_1 + q_2e_{32} + q_3e_{13} + q_{21}$;
3. calcular a matriz $\mathbf{R}$ a partir de R ;
4. compor, por meio do produto matricial, todas as matrizes já construídas, gerando $\mathbf{R}_{[i]}$;

Para cada ponto X_m , com $k_i \leq m < k_i + 1$:

5. definir um translador $T_{x_{k_i-2}}$ e aplicá-lo a X_m , gerando l ;
6. aplicar $\mathbf{R}$ a l , gerando t ;
7. definir o translador $\mathbf{T}_t$ e aplica-lo ao ponto X'_{k_i-2} .

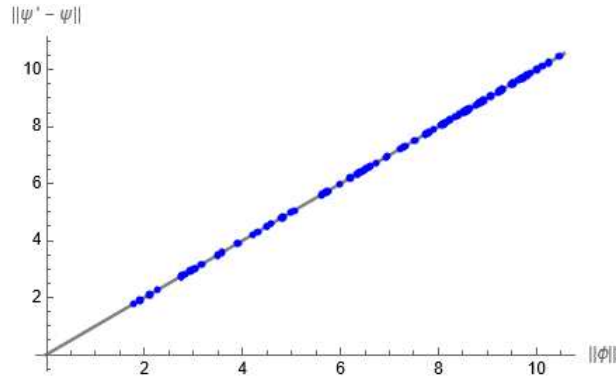
5.2 Testes Computacionais

Alguns experimentos foram feitos com o intuito de verificar a precisão dos métodos que envolvem a AGC, além de realizar comparações entre estes e aqueles apresentados no capítulo 3.

Para verificar se os resultados obtidos com os algoritmos da AGC correspondem ao esperado, realizamos um experimento, como feito anteriormente, em que consideramos uma molécula hipotética de proteína de 1000 átomos e impomos 100 diferentes alterações aleatórias nos ângulos de torção, com a quantidade alterada variando de 1% a 30% do total. A ideia, novamente, é averiguar se $\psi_i + \phi_i = \psi'_i$, para $i = 1, 2, \dots, 100$, é observado

experimentalmente. Então, para cada ϕ_i , calculamos as coordenadas conformes atualizadas e depois buscamos os ângulos de torção associados à estrutura atualizada. A Figura 40 mostra o gráfico encontrado.

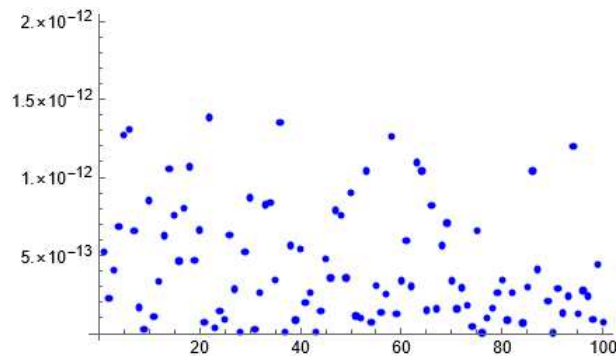
Figura 40 – Norma da diferença entre os ângulos de torção originais e atualizados ($\|\psi' - \psi\|$) em função da norma das alterações nos ângulos de torção ($\|\phi\|$). Em azul, os valores obtidos em cada simulação; em cinza, a reta identidade.



Fonte: Elaborado pelo autor.

De fato, o experimento acima, realizado utilizando o Algoritmo 8, indica que os resultados obtidos correspondem ao esperado, uma vez que os pontos azuis estão, aparentemente, sobre a reta identidade. Indo além, para verificar a distância entre os valores obtidos em $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$, fizemos um outro experimento, que está presente na Figura 41:

Figura 41 – Diferença entre as normas $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$.



Fonte: Elaborado pelo autor.

Apesar de a diferença entre as normas $\|\psi'_i - \psi_i\|$ e $\|\phi_i\|$ serem maiores do que aquelas encontradas no experimento análogo do capítulo 3, ainda são valores pequenos, de modo que podemos concluir que o Algoritmo 8 também cumpre o que se propõe.

Podemos fazer um procedimento análogo para os Algoritmos 6 e 7, mas vamos usar novamente o RMSD para verificar o quão próximas estão as soluções obtidas com os Algoritmos 6 e 7 em relação ao Algoritmo 8. Como aqui estamos tratando de coordenadas

conformes, precisamos adaptar a fórmula (3.5), sobre o cálculo do RMSD. Pela definição do produto interno conforme, presente em (4.14), temos que o quadrado da distância entre dois pontos $x, y \in \mathbb{R}^3$ está associado ao produto interno entre X e Y da seguinte forma:

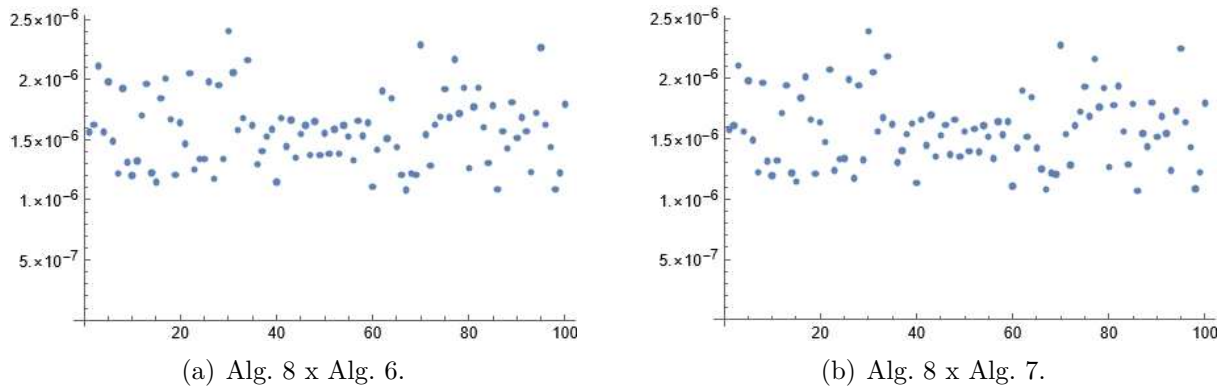
$$\|x - y\|^2 = -2(X \cdot Y).$$

Logo, a fórmula do RMSD é reescrita para coordenadas conformes como

$$RMSD = \sqrt{\frac{-2}{n} \sum_{i=1}^n X_i \cdot Y_i}. \quad (5.7)$$

Feito isso, consideramos novamente uma molécula hipotética de proteína de 1000 átomos e impomos 100 diferentes alterações nos ângulos de torção. As 100 estruturas atualizadas foram comparadas, via RMSD, entre os 3 algoritmos apresentados nesse capítulo. A Figura 42 apresenta o resultado.

Figura 42 – Valores obtidos do RMSD na comparação do Algoritmo 8 com os Algoritmos 6 e 7.



Fonte: Elaborado pelo autor.

De fato, os resultados obtidos aqui nos indicam que os métodos retratados nos Algoritmos 6 e 7 estão próximos àqueles encontrados com o Algoritmo 8, uma vez que o RMSD está próximo de 0, indicando uma boa similaridade entre as estruturas comparadas.

Agora, faremos uma breve análise sobre custo computacional. É importante ressaltar que os algoritmos da AGC foram implementados com a inclusão de algumas funções que buscam sua simplificação. Por exemplo: criamos uma função que retorna o resultado da aplicação de uma rotação R a um ponto do Modelo Conforme porque realizar o produto geométrico $R(\cdot \cdot)R^{-1}$ diretamente inclui operações que se repetem. O mesmo foi feito para os demais procedimentos necessários.

Trazendo também para a discussão os algoritmos apresentados no capítulo 3, a Tabela 1 compara o custo computacional em relação ao número de operações.

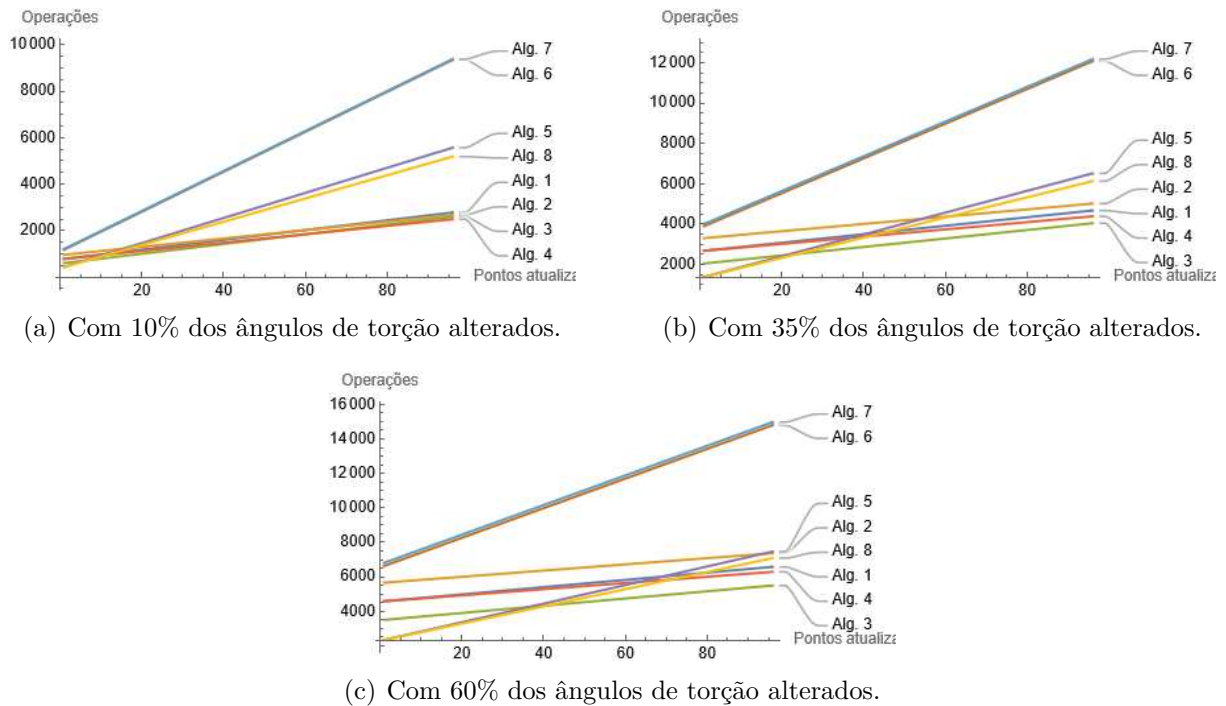
Tabela 1 – Contagem de operações para cada um dos algoritmos apresentados.

	Contagem de operações
Algoritmo 1 - (Rotações Simples e Translações Finais)	$76 n_r + 21 n_a$ operações
Algoritmo 2 - (Rotações Simples e Translações Encaixadas)	$94 n_r + 18 n_a$ operações
Algoritmo 3 - (Quatérnios e Translações Finais)	$58 n_r + 21 n_a$ operações
Algoritmo 4 - (Quatérnios e Translações Encaixadas)	$76 n_r + 18 n_a$ operações
Algoritmo 5 - (Quatérnios nativamente e Translações Finais)	$38 n_r + 54 n_a$ operações
Algoritmo 6 - (Motores e Translações Encaixadas, versão 1)	$109 n_r + 86 n_a$ operações
Algoritmo 7 - (Motores e Translações Encaixadas, versão 2)	$112 n_r + 86 n_a$ operações
Algoritmo 8 - (Rotores e Tradutores e Translações Finais)	$38 n_r + 50 n_a$ operações

Fonte: Elaborado pelo autor.

Como a contagem de operações foi feita considerando a quantidade de ângulos de torção alterados (n_r) e a quantidade de pontos que precisam ser atualizados (n_a) por consequência, decidimos por fixar 3 porcentagens de alterações de ângulos de torção e, a partir disso, calcular a quantidade de operações para cada número de pontos atualizados. Para isso, consideramos uma proteína de 100 átomos. Abaixo, a Figura 43 mostra os resultados obtidos, comparando todos os algoritmos desse trabalho.

Figura 43 – Quantidade de operações em função do número de pontos que precisam ser atualizados.



Fonte: Elaborado pelo autor.

No item a) da Figura 43, podemos observar que os Algoritmos 1, 2, 3 e 4, que envolvem o uso de matrizes, usam de um menor número de operações, enquanto os Algoritmos 6 e 7, com os Motores da AGC, ficam um pouco atrás dos demais. Num

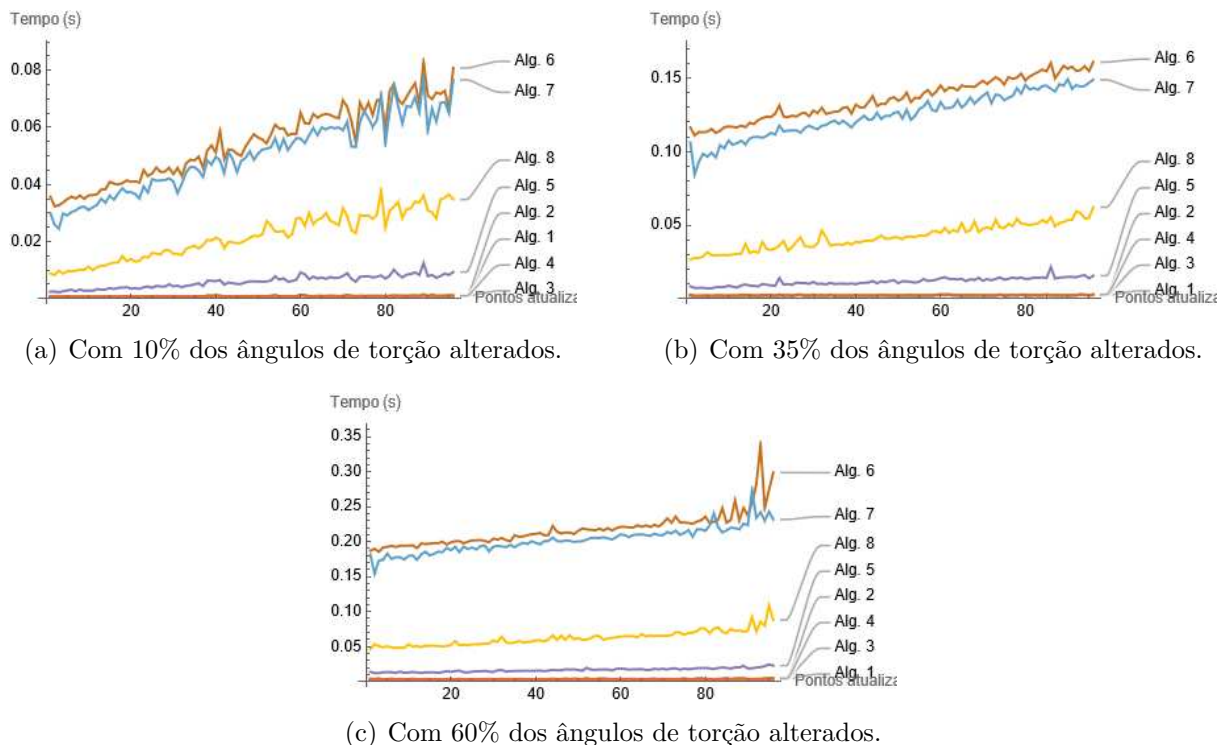
meio termo, estão os Algoritmos 5 e 8. Mas pode-se notar também que, ao aumentar a porcentagem de ângulos de torção alterados, estes últimos se mostram competitivos com os Algoritmos 1, 2, 3 e 4. Isso acontece porque os Algoritmos 5 e 8 necessitam de um número menor de operações por ângulo de torção alterado do que os demais. Em geral, os algoritmos com Translações Finais têm essa característica. Esses fatos indicam que há casos em que um algoritmo pode ser mais aconselhado do que o outro, a depender do número de ângulos de torção alterados e a quantidade de pontos que precisam ser atualizados, por consequência.

Por fim, alguns comentários sobre os Algoritmos 6 e 7. Podemos notar, tanto na Tabela 1, quanto na Figura 43, que os dois ficaram um pouco atrás dos demais. Na tabela, inclusive, fica evidenciado que eles envolvem mais operações tanto por ângulo de torção alterado, quanto por ponto que precisa ser atualizado. Por esses dois parâmetros apenas, seria natural descartá-los. Entretanto, como pode ser visto no capítulo 6 de (CAMARGO, 2021), os motores, elementos principais nos Algoritmos 6 e 7, são úteis no cálculo de distâncias interatômicas. Com eles, é possível calcular a distância entre dois pontos quaisquer usando apenas as coordenadas internas de uma proteína e, portanto, sem a necessidade de encontrar as coordenadas conformes dos respectivos pontos. Conhecer as distâncias interatômicas de uma proteína é muito importante em diversos processos de química computacional.

Por curiosidade, já que o nosso foco é mais teórico do que computacional, também decidimos verificar se a proporção entre os custos é observada na prática. Para isso, consideramos o tempo de processamento de cada um dos algoritmos para atualização de uma estrutura 3D de uma molécula hipotética de proteína de 100 átomos. Também vamos fixar 3 porcentagens de alterações nos ângulos de torção e, diferentemente do experimento anterior, teremos o tempo de processamento em função do número de pontos atualizados. Apresentamos o resultado do experimento na Figura 44.

As proporções entre os algoritmos, levando em conta os tempos de processamento, se assemelham, majoritariamente, àquelas encontradas com o experimento anterior, sobre a contagem de operações. Entretanto, as diferenças entre teoria e prática se devem, principalmente, ao fato de que usamos pacotes para trabalhar com os quatérnios e com a AGC, além de que tivemos que implementar esses algoritmos atribuindo algumas adequações e/ou funções criadas por nós, que eventualmente podem ser aprimoradas. Ainda assim, pode-se notar que a prática, exemplificada com os tempos de processamento, segue razoavelmente um padrão estabelecido pela teoria, na contagem de operações.

Figura 44 – Tempo de processamento em função do número de pontos que precisam ser atualizados.



Fonte: Elaborado pelo autor.

5.3 Exemplo detalhado

Assim como foi feito no capítulo 3, separamos essa seção para resolver um pequeno problema de atualização de coordenadas dos pontos que representam uma molécula hipotética de 8 átomos devido a alterações em ângulos de torção. Estamos assumindo as mesmas hipóteses de outrora sobre distâncias e ângulos de ligação, bem como os dados dos ângulos de torção originais e as alterações que queremos impor.

A diferença que vamos considerar aqui é que estamos partindo do pressuposto de que a conversão de coordenadas internas para conformes já foi devidamente realizada. Temos, então, os seguintes pontos relacionados aos átomos da estrutura original da nossa

pequena molécula:

$$X_1 = e[0]$$

$$X_2 = e[0] - 1.5e[1] + 1.125e[\infty]$$

$$X_3 = e[0] - 1.9991e[1] - 1.41453e[2] + 2.99866e[\infty]$$

$$X_4 = e[0] - 3.45517e[1] - 1.43003e[2] - 0.360054e[3] + 7.05641e[\infty]$$

$$X_5 = e[0] - 4.03e[1] - 2.78186e[2] - 0.0565105e[3] + 11.9914e[\infty]$$

$$X_6 = e[0] - 3.08706e[1] - 3.84464e[2] - 0.537505e[3] + 12.3e[\infty]$$

$$X_7 = e[0] - 3.43632e[1] - 5.15192e[2] + 0.109821e[3] + 19.1813e[\infty]$$

$$X_8 = e[0] - 4.21596e[1] - 4.89468e[2] + 1.36521e[3] + 21.798e[\infty]$$

Vamos adotar a representação de e_i por $e[i]$ ao longo desse seção. Fazemos isso para se assemelhar à forma como manipulamos tais objetos com o pacote *CGAlgebra*. Além disso, listamos na tabela abaixo algumas funções úteis para a resolução desse problema.

Tabela 2 – Funções usadas com o pacote *CGAlgebra*.

Função	Papel que desempenha
GeometricProduct[A, B]	Calcula o produto geométrico entre A e B.
OuterProduct[A, B]	Calcula o produto externo entre A e B.
Reversion[A]	Calcula o reverso do multivetor A.
MultivectorInverse[A]	Calcula o inverso do multivetor A, caso exista.
Dual[A]	Calcula o dual do multivetor A.
Grade[A, k]	Extraí a porção k-vetorial do multivetor A.

Vamos impor duas alterações em ângulos de torção: $\phi_4 = 0.2$ rad e $\phi_6 = 0.4$ rad. Com isso, temos 3 grupos de átomos: G_0 , representado pelos pontos que não precisam ser atualizados (X_1 , X_2 e X_3); G_1 , com os pontos que só recebem a aplicação de uma rotação (X_4 e X_5); e G_2 , composto por X_6 , X_7 e X_8 , que recebem a atuação de duas rotações.

Primeiro, vamos apresentar os resultados de cada etapa usando o pacote *CGAlgebra* (7), além dos resultados para o cálculo via versão matricial.

Portanto, sobre a atualização de todos os pontos que compõem G_1 , devemos:

1. Calcular o vetor unitário que será entendido como eixo de rotação:

$$\mathbf{b} = \frac{x_3 - x_2}{||x_3 - x_2||} = -0.332736e[1] - 0.94302e[2].$$

Para a versão matricial, consideramos

$$b = (-0.332736, -0.94302, 0).$$

2. Determinar o rotor a partir de $\mathbf{b}$ e $\phi_4 = 0.2$:

$$R_1 = \cos\left(\frac{\phi_1}{2}\right) - \sin\left(\frac{\phi_1}{2}\right) \mathbf{b}e_{123} = 0.995 - 0.0332e[3, 2] - 0.0941e[1, 3].$$

Para a versão matricial, tomamos $q = \left(\cos \frac{\phi_4}{2}, \sin \frac{\phi_4}{2} b_1, \sin \frac{\phi_4}{2} b_2, \sin \frac{\phi_4}{2} b_3 \right) = (0.995, -0.0332, -0.0941, 0)$ e criamos a seguinte matriz:

$$\mathbf{R}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.982273 & 0.00625465 & -0.187349 & 0 \\ 0 & 0.00625465 & 0.997793 & 0.0661045 & 0 \\ 0 & 0.187349 & -0.0661045 & 0.980067 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

3. Calcular os transladores a partir das ligações rotacionadas:

$$\begin{aligned} t_1 &= R_1(x_4 - x_3)R_1^{-1} \implies T_{t_1} = 1 + 0.6814e[1, \infty] + 0.0242e[2, \infty] + 0.3123e[3, \infty] \\ t_2 &= R_1(x_5 - x_4)R_1^{-1} \implies T_{t_2} = 1 + 0.3149e[1, \infty] + 0.6662e[2, \infty] - 0.1396e[3, \infty]. \end{aligned}$$

Na versão matricial, os transladores são escritos como:

$$\begin{aligned} \mathbf{T}_{t_1} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1.36289 & 1 & 0 & 0 & 0 \\ -0.0483767 & 0 & 1 & 0 & 0 \\ -0.624644 & 0 & 0 & 1 & 0 \\ 1.125 & -1.36289 & -0.0483767 & -0.624644 & 1 \end{pmatrix} \\ \mathbf{T}_{t_2} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.629966 & 1 & 0 & 0 & 0 \\ -1.33237 & 0 & 1 & 0 & 0 \\ 0.27916 & 0 & 0 & 1 & 0 \\ 1.125 & -0.629966 & -1.33237 & 0.27916 & 1 \end{pmatrix}. \end{aligned}$$

4. Calcular as atualizações dos pontos:

$$\begin{aligned} X'_4 &= T_{t_1}X'_3T_{t_1}^{-1} = e[0] - 3.362e[1] - 1.46291e[2] - 0.624644e[3] + 6.91665e[\infty] \\ X'_5 &= T_{t_2}X'_4T_{t_2}^{-1} = e[0] - 3.99196e[1] - 2.79528e[2] - 0.345484e[3] + 11.9344e[\infty]. \end{aligned}$$

Na versão matricial:

$$\begin{aligned} X'_4 &= \mathbf{T}_{t_1} \cdot X'_3 = (1, -3.362, -1.46291, -0.624644, 6.91665) \\ X'_5 &= \mathbf{T}_{t_2} \cdot X'_4 = (1., -3.99196, -2.79528, -0.345484, 11.9344). \end{aligned}$$

Para atualizar os pontos que pertencem a G_2 , devemos:

1. Calcular o vetor unitário que será entendido como eixo de rotação. Logo, temos

$$\mathbf{b} = -0.383221e[1] - 0.901216e[2] + 0.202362e[3].$$

Para a versão matricial,

$$b = (-0.383221, -0.901216, 0.202362).$$

2. Determinar o rotor a partir de $\mathbf{b}$ e $\phi_6 = 0.4$:

$$R_2 = 0.980067 - 0.0402e[1, 2] - 0.1790e[1, 3] + 0.0761e[2, 3].$$

Para a versão matricial, tomamos $q = \left(\cos \frac{\phi_6}{2}, \sin \frac{\phi_6}{2} b_1, \sin \frac{\phi_6}{2} b_2, \sin \frac{\phi_6}{2} b_3 \right) = (0.9801, -0.0761, -0.1790, 0.0402)$ e criamos a seguinte matriz:

$$\mathbf{R}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.932654 & -0.0515408 & -0.357072 & 0 \\ 0 & 0.106066 & 0.985175 & 0.134837 & 0 \\ 0 & 0.344829 & -0.16363 & 0.924294 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

3. Compor todos os rotores calculados até aqui. Então

$$R_{[2]} = R_1 R_2 = 0.9558 - 0.1121e[3, 2] - 0.2691e[1, 3] + 0.0388e[2, 1]$$

Na versão matricial, temos

$$\mathbf{R}_{[2]} = \mathbf{R}_1 \cdot \mathbf{R}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.852181 & -0.0138093 & -0.523064 & 0 \\ 0 & 0.13446 & 0.971861 & 0.193406 & 0 \\ 0 & 0.505675 & -0.235149 & 0.830059 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

4. Calcular os transladores a partir das ligações rotacionadas:

$$t_1 = R_{[2]}(x_6 - x_5)R_{[2]}^{-1} \implies T_{t_1} = 1 - 0.5349e[1, \infty] + 0.4995e[2, \infty] - 0.1637e[3, \infty]$$

$$t_2 = R_{[2]}(x_7 - x_6)R_{[2]}^{-1} \implies T_{t_2} = 1 + 0.3091e[1, \infty] + 0.5961e[2, \infty] - 0.3341e[3, \infty]$$

$$t_3 = R_{[2]}(x_8 - x_7)R_{[2]}^{-1} \implies T_{t_3} = 1 + 0.6623e[1, \infty] - 0.193e[2, \infty] - 0.2937e[3, \infty].$$

Na versão matricial, os transladores são escritos como:

$$\mathbf{T}_{t_1} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1.06983 & 1 & 0 & 0 & 0 \\ -0.999114 & 0 & 1 & 0 & 0 \\ 0.327481 & 0 & 0 & 1 & 0 \\ 1.125 & 1.06983 & -0.999114 & 0.327481 & 1 \end{pmatrix}$$

$$\mathbf{T}_{t_2} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -0.618177 & 1 & 0 & 0 & 0 \\ -1.19226 & 0 & 1 & 0 & 0 \\ 0.668109 & 0 & 0 & 1 & 0 \\ 1.125 & -0.618177 & -1.19226 & 0.668109 & 1 \end{pmatrix}$$

$$\mathbf{T}_{t_3} = \begin{pmatrix} 1. & 0 & 0 & 0 & 0 \\ -1.32459 & 1 & 0 & 0 & 0 \\ 0.387972 & 0 & 1 & 0 & 0 \\ 0.587313 & 0 & 0 & 1 & 0 \\ 1.125 & -1.32459 & 0.387972 & 0.587313 & 1 \end{pmatrix}.$$

5. Calcular as atualizações dos pontos:

$$X'_6 = T_{t_1} X'_5 T_{t_1}^{-1} = e[0] - 2.92214e[1] - 3.79439e[2] - 0.0180029e[3] + 11.4683e[\infty]$$

$$X'_7 = T_{t_2} X'_6 T_{t_2}^{-1} = e[0] - 3.54032e[1] - 4.98665e[2] + 0.650106e[3] + 18.9116e[\infty]$$

$$X'_8 = T_{t_3} X'_7 T_{t_3}^{-1} = e[0] - 4.86491e[1] - 4.59868e[2] + 1.23742e[3] + 23.1732e[\infty].$$

Na versão matricial:

$$X'_6 = \mathbf{T}_{t_1} \cdot X'_5 = (1, -2.92214, -3.79439, -0.0180029, 11.4683)$$

$$X'_7 = \mathbf{T}_{t_2} \cdot X'_6 = (1, -3.54032, -4.98665, 0.650106, 18.9116)$$

$$X'_8 = \mathbf{T}_{t_3} \cdot X'_7 = (1, -4.86491, -4.59868, 1.23742, 23.1732).$$

Agora vamos fazer o mesmo para os métodos cujo eixo de rotação não passa necessariamente pela origem. Para abreviar, vamos adotar apenas o método descrito no Algoritmo 6 e omitir a versão matricial. Então, para atualizar os pontos de G_1 , devemos:

1. Calcular o bivector unitário, a partir de X_3 e X_2 , que será entendido como plano de rotação. Logo, temos

$$I = \frac{(X_3 \wedge X_2 \wedge e_\infty)^*}{\text{Magnitude}[(X_3 \wedge X_2 \wedge e_\infty)^*]} = 0.94302e[1, 3] - 0.332736e[2, 3] + 1.41453e[3, \infty].$$

2. Determinar o motor a partir de I e $\phi_4 = 0.2$:

$$M_1 = \cos\left(\frac{\phi_1}{2}\right) - \sin\left(\frac{\phi_1}{2}\right) I = 0.995 - 0.0941e[1, 3] + 0.0332e[2, 3] - 0.1412e[3, \infty].$$

3. Calcular as atualizações dos pontos:

$$X'_4 = M_1 X_4 M_1^{-1} = e[0] - 3.362e[1] - 1.46291e[2] - 0.624644e[3] + 6.91665e[\infty]$$

$$X'_5 = M_1 X_5 M_1^{-1} = e[0] - 3.99196e[1] - 2.79528e[2] - 0.345484e[3] + 11.9344e[\infty].$$

Já para os pontos que pertencem a G_2 , devemos:

1. Calcular o bivector unitário, a partir de X_5 e X_4 , que será entendido como plano de rotação. Logo, temos

$$I = 0.2024e[1, 2] + 0.9012e[1, 3] - 0.3832e[2, 3] - 0.6139e[1, \infty] + 0.8372e[2, \infty] + 2.5658e[3, \infty].$$

2. Determinar o Motor a partir de I e $\phi_4 = 0.2$:

$$M_2 = 0.98 - 0.0402e[1, 2] - 0.1790e[1, 3] + 0.122e[1, \infty] + 0.07613e[2, 3] \\ - 0.1663e[2, \infty] - 0.5098e[3, \infty].$$

3. Compor todos os rotores construídos até aqui:

$$M_{[2]} = M_1 M_2 = 0.9558 - 0.0388e[1, 2] - 0.2691e[1, 3] + 0.1441e[1, \infty] + 0.1121e[2, 3] \\ - 0.1717e[2, \infty] - 0.6286e[3, \infty]$$

4. Calcular as atualizações dos pontos:

$$X'_6 = M_{[2]} X_6 M_{[2]}^{-1} = e[0] - 2.92214e[1] - 3.79439e[2] - 0.0180029e[3] + 11.4683e[\infty] \\ X'_7 = M_{[2]} X_7 M_{[2]}^{-1} = e[0] - 3.54032e[1] - 4.98665e[2] + 0.650106e[3] + 18.9116e[\infty] \\ X'_8 = M_{[2]} X_8 M_{[2]}^{-1} = e[0] - 4.86491e[1] - 4.59868e[2] + 1.23742e[3] + 23.1732e[\infty].$$

Fazendo a conversão de todos esses pontos atualizados para coordenadas cartesianas, notamos que o resultado foi o mesmo que o encontrado no capítulo 3 para o mesmo exemplo.

Mesmo assim, vamos calcular as matrizes de distâncias dos pontos originais e dos pontos atualizados. Como agora estamos trabalhando com coordenadas conformes, usamos o fato de que a distância entre os pontos X_i e X_j é dada na AGC por

$$r_{i,j} = \sqrt{-2(X_i \cdot X_j)},$$

fato decorrente da definição (4.14), em que $\cdot$ representa o produto interno do Modelo Conforme, que com o pacote *CGAlgebra* é representado pela função *InnerProduct*.

A diferença entre as matrizes de distâncias referidas é dada por

$$\begin{pmatrix} 0 & 0 & 0 & -0.0374 & -0.0117 & -0.1706 & -0.0437 & 0.2051 \\ 0 & 0 & 0 & 4.4409 \cdot 10^{-16} & -0.1417 & -0.0779 & 0.0693 \\ 0 & -2.8866 \cdot 10^{-16} & 4.4409 \cdot 10^{-16} & -0.1636 & -0.0613 & 0.1128 \\ 0 & 0 & -2.4424 \cdot 10^{-15} & -7.9936 \cdot 10^{-15} & 6.6613 \cdot 10^{-15} & -1.7763 \cdot 10^{-15} \\ 0 & 0 & 0 & -1.7763 \cdot 10^{-14} & 4.4409 \cdot 10^{-15} & 2.6645 \cdot 10^{-15} \\ 0 & 0 & 0 & 3.5527 \cdot 10^{-15} & -1.0214 \cdot 10^{-14} \\ 0 & 0 & 0 & -2.7311 \cdot 10^{-14} \\ 0 \end{pmatrix},$$

satisfazendo o fato de que $r_{i,j}$ não se altera quando $i, j < k$ e quando $i, j \geq k - 2$, de modo que só mudam as distâncias $r_{i,j}$ para $i \leq k - 3$ e $j \geq k$.

Além disso, os ângulos de torção calculados a partir dos pontos atualizados também satisfizeram as alterações $\phi_4 = 0.2$ rad e $\phi_6 = 0.4$ rad que impomos, pois os ângulos de torção originais são

$$\{0, 0, 0, -2.88422, -2.91555, -0.772891, 2.84816, -0.3755\}$$

e os que encontramos a partir da atualização das coordenadas são

$$\{0, 0, 0, -2.68422, -2.91555, -0.372891, 2.84816, -0.3755\}.$$

6 Considerações Finais

Calcular várias atualizações a partir de uma estrutura 3D de uma molécula de proteína é uma tarefa muito importante em diversas situações: Dinâmica Molecular, Docking, no estudo de Função da Energia Potencial, entre outras. Por tal importância, um dos nossos objetivos era dissecar os principais métodos de atualização de coordenadas de estruturas 3D de proteínas. Esse objetivo, naturalmente, incluía também uma fundamentação teórica sobre rotações e translações, tão importantes nos processos apresentados.

Não nos limitando apenas aos métodos que denominamos como “clássicos”, mais conhecidos e difundidos na literatura, buscamos também retratar uma outra perspectiva de abordagem para o problema de atualização de estruturas 3D de proteínas. Com a Álgebra Geométrica Conforme, pudemos notar simplificações e propriedades interessantes surgirem. Os quatérnios, tão importantes para codificar rotações, se mostraram, de uma forma muito natural, parte integrante dessa álgebra. A ortogonalidade de operadores, na presença dos rotores, transladores e motores, também é algo a se destacar.

Claro que algumas dificuldades surgiram. Ao incluir duas novas coordenadas, colher num espaço a 5 dimensões as interpretações geométricas do espaço tridimensional fatalmente indica um aumento no custo computacional. Exemplos são os Algoritmos 4 e 5 do último capítulo, utilizando motores. Mas não foi a regra! Por exemplo, a rotação com os rotores, como mostrado no capítulo 4, não requer custo adicional ao subir duas dimensões. Entretanto, no caso dos motores, um custo computacional maior nesse problema pode ser compensado por vantagens e simplificações no cálculo de distâncias interatômicas. Isso pode ser notado, por exemplo, no estudo sobre Função da Energia Potencial de proteínas, no qual são encontrados resultados nesse sentido em (CAMARGO, 2021).

Bastante influenciado pelos artigos (ZHANG; KAVRAKI, 2002; CHOI, 2006) e pela tese de doutorado (CAMARGO, 2021), o presente trabalho buscou fazer algumas distinções. Damos um enfoque considerável para fundamentações teóricas e detalhes no passo a passos dos métodos, bem como exemplos de aplicação, na tentativa de apresentá-los de forma clara. Mais significativamente, buscamos trabalhar os métodos da AGC com um pacote específico para Álgebra Geométrica. Isso é importante, porque permite que realizemos todos os processos sem a necessidade de conversão de elementos em matrizes. Ainda assim, buscamos apresentar, de forma resumida, a versão matricial dos algoritmos com a AGC, a fim de que o ambiente mais habitual da Álgebra Linear servisse também de afirmação da abordagem apresentada. Também trabalhamos com alguns experimentos para mostrar a eficiência e a precisão dos métodos em resolver o problema da atualização de estruturas 3D de proteínas, bem como comparamos por meio de uma métrica de

similaridade de estruturas e a partir dos custos computacionais dos respectivos algoritmos.

Por ser uma área relativamente nova da Matemática, a Álgebra Geométrica e a sua versão com o Modelo Conforme possuem um grande potencial de desenvolvimento, tanto em teoria, quanto em aplicações. Em especial, no problema de aplicação deste trabalho, os experimentos apresentados corroboram para essa afirmação, uma vez que os métodos com a AGC se mostraram tão precisos quanto os métodos denominados clássicos. É fato também que, por objetivarmos principalmente o funcionamento dos métodos com a AGC, estes ainda podem ser implementados de forma mais aprimorada, vislumbrando maior eficiência e menor custo computacional. Mas apesar disso, ficou evidente que a Álgebra Geométrica Conforme é uma boa e promissora alternativa na resolução do problema de atualização de estruturas 3D de proteínas.

Referências

- BOTTEMA, O. *Theoretical Kinematics*. 2. ed. [S.l.]: Dover Publications, 1990.
- CAMARGO, J. M. *Geometria de proteínas no espaço conforme*. Tese (Doutorado) — Universidade Estadual de Campinas, Campinas, 2021.
- 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.
- DIEBEL, J. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Technical Report, Stanford University*, 2006.
- DILL, K. A.; OZKAN, S. B.; SHELL, S.; WEIKL, T. R. The protein folding problem. *Annu Rev Biophys*, v. 37, p. 289–316, 2008.
- FERNANDES, L.; LAVOR, C.; OLIVEIRA, M. *Álgebra Geométrica e Aplicações*. [S.l.]: [S.l.]: SBMAC, 2017. ISBN 978-85-8215-080-1.
- GOLDSTEIN, H.; POOLE, C. P.; SAFKO, J. L. *Classical Mechanics*. 3. ed. [S.l.]: Addison-Wesley, 2002. ISBN 978-0-201-65702-9.
- HENDERSON, D. M. Euler angles, quaternions, and transformation matrices for space shuttle analysis. *NASA Technical Reports Server*, 1977.
- HESTENES, D. *Space-Time Algebra*. New York: Gordon and Breach, 1966.
- _____. *New foundations for classical mechanics*. Dordrecht, Netherlands: Reidel Publishing Company, 1987.
- HESTENES, D.; LI, H.; ROCKWOOD, A. *A unified algebraic framework for classical geometry*. In: Sommer, g. (ed.). *geometric computing with clifford algebra*. [S.l.]: Springer, 1999.
- JACOBSON, N. *Basic algebra*. 2. ed. New York: Dover Publications, 1985.
- KANATANI, K. *Understanding Geometric Algebra: Hamilton, Grassmann, and Clifford for Computer Vision and Graphics*. [S.l.]: AK Peters/CRC Press, 2015.
- KOVÁCS, E. Rotation about an arbitrary axis and reflection through an arbitrary plane. *Annales Mathematicae et Informaticae*, v. 40, n. 1, p. 175–186, 2012.
- KUFAREVA, I.; ABAGYAN, R. Methods of protein structure comparison. *Methods Mol Biol*, v. 857, p. 231–257, 2012.
- KUIPERS, J. B. *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace, and Virtual Reality*. 1. ed. [S.l.]: Princeton University Press, 1998.

- LAVOR, C.; MACULAN, N.; SOUZA, M.; ALVES, R. *Álgebra e Geometria no Cálculo de Estrutura Molecular*. [S.l.]: IMPA, 2017. ISBN 978-85-244-0432-0.
- LAVOR, C.; SOUZA, M.; ARAGÓN, J. L. Orthogonality of isometries in the conformal model of the 3d space. *Graphical Models*, v. 114, 2021. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1524070321000059>>.
- LIMA, E. *Álgebra Linear*. 7. ed. Rio de Janeiro: IMPA, 2008.
- NELSON, D.; COX, M. *Principles of Biochemistry*. [S.l.]: W.H. Freeman, New York, 2008.
- ORTIZ-DURAN, E. A.; ARAGON, J. L. Cgalgebra: a mathematica package for conformal geometric algebra. v.2.0. arXiv, 2017. Disponível em: <<https://arxiv.org/abs/1711.02513>>.
- RODRIGUES, O. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendants des causes qui peuvent les produire. *Journal de Mathématiques Pures et Appliquées*, v. 5, p. 380–440, 1840.
- ROGERS, D.; ADAMS, A. *Mathematical elements for computer graphics*. 2. ed. [S.l.]: McGraw-Hill publishing Company, 1990.
- THOMPSON, H. B. Calculation of cartesian coordinates and their derivatives from internal molecular coordinates. *The Journal of Chemical Physics*, v. 47, n. 9, p. 3407–3410, 1967.
- ZHANG, M.; KAVRAKI, L. A new method for fast and accurate derivation of molecular conformations. *Journal of chemical, information and modeling, ACS Publications*, v. 42, n. 1, p. 64–70, 2002.