



Rafael Santos de Oliveira Alves

Álgebra de Clifford Aplicada ao Cálculo de Estruturas Moleculares

CAMPINAS
2013



UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA
E COMPUTAÇÃO CIENTÍFICA

Rafael Santos de Oliveira Alves

Álgebra de Clifford Aplicada ao Cálculo de Estruturas Moleculares

Orientador(a): Prof. Dr. Carlile Campos Lavor

Tese de doutorado apresentada ao Instituto de
Matemática, Estatística e Computação Científica da Unicamp para
obtenção do título de Doutor em Matemática Aplicada

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA TESE
DEFENDIDA PELO ALUNO RAFAEL SANTOS DE OLIVEIRA ALVES,
E ORIENTADA PELO PROF.DR CARLILE CAMPOS LAVOR

Assinatura do Orientador

Carlile Campos Lavor

CAMPINAS
2013

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

AL87a Alves, Rafael Santos de Oliveira, 1982-
Álgebra de Clifford aplicada ao cálculo de estruturas moleculares / Rafael Santos de Oliveira Alves. – Campinas, SP : [s.n.], 2013.

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

1. Clifford, Álgebra de. 2. Geometria molecular. 3. Geometria de distâncias. I. Lavor, Carlile Campos, 1968-. II. Universidade Estadual de Campinas. Instituto de Matemática, Estatística e Computação Científica. III. Título.

Informações para Biblioteca Digital

Título em inglês: Clifford Algebras applied to molecular structure calculations

Palavras-chave em inglês:

Clifford Algebras

Molecular structure

Distance geometry

Área de concentração: Matemática Aplicada

Titulação: Doutor em Matemática Aplicada

Banca examinadora:

Carlile Campos Lavor [Orientador]

Waldyr Alves Rodrigues Junior

Antonio Mucherino

Antonio Murilo Santos Macedo

Leandro Augusto Frata Fernandes

Data de defesa: 25-02-2013

Programa de Pós-Graduação: Matemática Aplicada

Tese de Doutorado defendida em 25 de fevereiro de 2013 e aprovada

Pela Banca Examinadora composta pelos Profs. Drs.

Carlile Campos L

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

Waldyr

Prof(a). Dr(a). **WALDIR ALVES RODRIGUES JUNIOR**

Antonio Mucherino

Prof(a). Dr(a). **ANTONIO MUCHERINO**

Antonio Murilo S. Macedo

Prof(a). Dr(a). **ANTONIO MURILO SANTOS MACEDO**

Leandro Augusto Frata Fernandes

Prof(a). Dr(a). **LEANDRO AUGUSTO FRATA FERNANDES**

Agradecimentos

Em primeiro lugar agradeço aos meus pais, Rui e Guida, pelo apoio e compreensão durante todo o período, me proporcionando toda a tranquilidade para que o trabalho fosse realizado. Agradeço também o apoio de meus irmãos, Nelson e Eduardo, e de minhas cunhadas, Renata e Renata, sempre presentes durante toda a jornada. Vocês tornaram possível a realização deste trabalho do início ao fim.

À minha companheira Mariana Sacrini, pela dedicação, compreensão e companheirismo em todos os momentos. E não apenas nos últimos quatro, mas já há quase sete anos sempre ao meu lado me dando todo o apoio possível. Obrigado por ter sido fundamental para que tudo desse certo.

Agradeço a todos os meus amigos, simplesmente pela sua amizade nesses últimos quatro anos. Cito especialmente os amigos que estão ou estiveram comigo na Unicamp e no Círculo Militar de São Paulo, que puderam estar mais próximos, mas agradeço a todos que também não fazem parte destes grupos, e são igualmente importantes.

Aos funcionários do IMECC, que sempre foram solícitos e eficientes. Agradeço especialmente à Livia, ao Ednaldo e principalmente à Tania, todos da Secretaria de Pós Graduação, aos quais dei bastante trabalho e sempre me orientaram e ajudaram da melhor maneira possível. Ao Professor Dr. Aurélio Ribeiro Leite de Oliveira, enquanto Coordenador de Pós-Graduação da Matemática Aplicada e posteriormente como Coordenador Geral de Pós-Graduação do IMECC, pela grande ajuda no decorrer do curso e pela confiança. E aos demais professores do IMECC que de alguma forma contribuíram para que o trabalho fosse concluído.

Ao amigo e orientador Professor Dr. Carlile Campos Lavor, com quem tenho o prazer de trabalhar desde 2006, a partir do Mestrado, um enorme agradecimento. Por acreditar em mim e no trabalho do início ao fim, até em momentos em que eu mesmo tinha dúvidas, me dando todo

o respaldo e a confiança necessários para seguir em frente. Agradeço pelo desafio de explorar novas áreas e desenvolver novas técnicas, por me proporcionar uma nova experiência acadêmica e pessoal através de um estágio acadêmico no exterior. Período fundamental para a conclusão deste trabalho.

Agradeço à Universidade de Rennes I, França, especialmente ao IRISA, por ter me recebido durante o segundo semestre de 2012. Agradeço ao Professor Dr. Antonio Mucherino, que se tornou um amigo, pela oportunidade e pela supervisão em Rennes, por toda a ajuda durante este período, desde a adaptação à cidade e à universidade, até o desenvolvimento efetivo do trabalho. Agradeço muito pelo esforço e pelo tempo dedicado, fundamentais para a conclusão da parte principal deste trabalho.

Ao Engenheiro de Softwares Dr. Andrea Cassioli, da École Polytechnique de Paris, registro também um grande agradecimento. Por ter passado um período em Rennes trabalhando incansavelmente comigo no processo de implementação de nosso algoritmo, e estar continuamente dedicado a este trabalho. Tem sido importantíssimo no desenvolvimento do algoritmo e na obtenção dos resultados computacionais. Faço ainda um agradecimento ao Engenheiro de Computação da Unicamp, Thiago Godoi, pela colaboração para que eu me familiariza-se com a programação em C. Agradeço também ao Professor Dr. Dietmar Hildenbrand da Universidade de Tecnologia de Darmstadt, Alemanha, pela grande ajuda e incentivo no início do trabalho com o modelo conforme das álgebras de Clifford, estando sempre disposto a colaborar desde o primeiro contato por e-mail, em 2009.

Por fim, agradeço ao CNPq pelo apoio financeiro, que permitiu dedicar-me exclusivamente ao desenvolvimento deste trabalho.

Campinas,
25 de fevereiro de 2013.

Rafael Santos de Oliveira Alves

Abstract

The Molecular Distance Geometry Problem (MDGP) consists in finding a three-dimensional embedding of simple, weighted, undirected graph such that the weight in the edges correspond to the inter-atomic distances of a molecule. This is a continuous search problem which can be discretized under some assumptions, yielding the Discretized MDGP (DMDGP), which is solved by a Branch and Prune (BP) algorithm using information about the distances among some atoms of the molecule. If the distances are given by a set of lower and upper bounds, a new problem arises: the interval DMDGP (iDMDGP). From a geometric interpretation of this problem, we propose a new approach, using Clifford Algebras, in order to improve the BP efficiency and treat algebraically the issues related to interval distances.

Resumo

O Problema de Geometria de Distâncias Moleculares (PGDM) consiste em encontrar uma imersão tridimensional de um grafo simples, não orientado, de forma que o peso nas arestas corresponda às distâncias inter-atômicas de uma molécula. Este é um problema de busca em um espaço contínuo, mas que pode ser discretizado sob algumas exigências, dando origem ao PGDM discretizado (PGDMD), que é solucionado usando informações sobre distâncias entre alguns átomos da molécula através de um algoritmo Branch and Prune (BP). Caso as distâncias sejam dadas por um conjunto de limites inferiores e superiores, temos um novo problema: o PGDMD intervalar (iPGDMD). A partir da interpretação geométrica deste último, propomos uma nova abordagem utilizando a Álgebra de Clifford a fim de tornar o algoritmo BP mais eficiente e de poder tratar algebricamente os problemas relacionados ao tratamento das distâncias intervalares.

Sumário

Agradecimentos	vi
Abstract	viii
Resumo	ix
1 Introdução	1
2 O Problema de Geometria de Distâncias Moleculares	4
2.1 PGDM	4
2.2 PDGDM	6
2.3 Discretização do PGDM	7
2.4 Posição Espacial dos Átomos	9
2.4.1 Complexidade	12
2.5 Algoritmo <i>Branch and Prune</i> (BP)	12
2.5.1 Estrutura do Algoritmo	12
2.6 Resultados Computacionais	14
2.7 PDGDM <i>Intervalar</i>	15
2.7.1 Definição do <i>i</i> PDGDM	16
2.7.2 Instâncias do <i>i</i> PDGDM	16

2.7.3	Algoritmo <i>iBP</i>	18
3	Introdução à Álgebra de Clifford	20
3.1	Álgebras de Clifford Cl_2 e Cl_3 e o Produto Geométrico	20
3.1.1	Cl_2 e o <i>Bivetor</i>	21
3.1.2	Álgebra de Clifford Cl_3	22
3.2	Produtos na Álgebra de Clifford	24
3.2.1	Produto Geométrico	25
3.2.2	Produto Exterior	27
3.2.2.1	A Álgebra Exterior $(\wedge \mathbb{R}^3, \wedge)$	28
3.2.3	Produtos Internos: Contrações	28
3.2.3.1	Interpretação Geométrica	30
3.2.4	Dualização	32
3.2.5	Transformações: Reflexão e Rotação em Cl_3	33
3.3	Modelo Projetivo	36
3.3.1	Elementos da Geometria em $Cl_{3,1}$	36
3.4	Modelo Conforme	37
3.4.1	Base do Espaço Conforme	39
3.4.2	Pontos	40
3.4.3	Esferas	41
3.4.4	Círculo	42
3.4.5	Planos	42
3.4.6	Retas	43
3.4.7	Par de Pontos	44

4	AC Aplicada ao PDGDM	49
4.1	Ferramentas Computacionais	49
4.2	Interseção de Esferas	50
4.3	PDGDM: Distâncias Exatas	52
4.3.1	Três Distâncias Exatas	52
4.3.2	Distância Exata Adicional	54
4.4	iPDGDM: Distâncias Intervalares	55
4.4.1	Duas Distâncias Exatas, Uma Intervalar	55
4.4.2	Formação e Descrição dos Arcos	56
4.4.3	Aplicação de R	59
4.4.4	Adição de uma Distância Intervalar	61
4.4.4.1	Novas Distâncias Intervalares	65
4.5	Resultados Computacionais	66
4.5.1	Implementação	67
4.5.2	Função Clifford	69
4.5.3	Resultados e Conclusões	71
5	Conclusão e Trabalhos Futuros	75
	Referências Bibliográficas	77

Capítulo 1

Introdução

Este trabalho é sobre uma aplicação da Álgebra de Clifford [33] à Geometria de Distâncias [21]. Problemas de geometria de distâncias relacionados à moléculas consistem em se determinar a posição espacial de cada átomo, a partir da estrutura de um grafo cujos pontos representam estes átomos e as arestas representam os pares de átomos cujas distâncias são conhecidas. O peso sobre cada aresta se refere às distâncias inter-atômicas. A estrutura deste grafo e outras informações sobre distâncias, obtidas através de experimentos de Ressonância Magnética Nuclear (RMN), constituem os dados disponíveis para a formulação do problema de se determinar a estrutura tridimensional de uma molécula, chamado de Problema de Geometria de Distâncias Moleculares (PGDM) [29]. Uma subclasse das instâncias do PGDM, que atende à suposições adicionais acerca de sua estrutura, pode ser discretizada e o problema relacionado à estas instâncias é chamado de Problema Discreto da Geometria de Distâncias Moleculares (PDGDM) [30]. Como veremos, esta discretização sugere uma interpretação geométrica do problema através da interseção de esferas, especialmente entre três esferas. Uma vez que a interseção entre três esferas, em geral, se dá em no máximo dois pontos, que representam as possíveis posições de um átomo, o espaço de soluções pode ser descrito por uma árvore binária, o que será visto no Capítulo 2.

O Capítulo 2 trata desta classe de problemas e está dividido em duas partes. Na primeira, consideramos o PDGDM, cujas distâncias disponíveis são exatas. Na segunda, consideramos o problema intervalar, o iPDGDM [31], em que parte das distâncias é dada por um conjunto de limites inferiores e superiores. O problema ao qual aplicamos os conceitos da Álgebra de Clifford é o intervalar. Um algoritmo Branch-and-Prune intervalar (iBP) é usado na resolução

Introdução

do iPDGDM. O tratamento de distâncias intervalares pelo iBP pode ocasionar problemas. Isto porque para a construção do espaço de busca os pontos são gerados através de distâncias coletadas dentro de cada intervalo e, em geral, o intervalo não possui apenas distâncias factíveis, o que implica a geração de pontos infactíveis. Um número pequeno de distâncias escolhidas em um intervalo pode ocasionar uma perda significativa de pontos factíveis, podendo inclusive conter apenas pontos infactíveis. Além disso, um número maior de distâncias pode ocasionar um grande crescimento no espaço de busca, fazendo com o que o algoritmo tenha que percorrer uma grande quantidade de pontos, dos quais muitos são infactíveis. Com o objetivo de superar estes problemas, propomos um novo método para a resolução do iPDGDM. Esta é a principal contribuição deste trabalho, o algoritmo C-iBP, que se origina da integração de uma nova função, que chamamos de `Clifford`, ao iBP original.

A função `Clifford` foi criada com base nos conceitos da Álgebra de Clifford (AC) [15, 33], apresentados no Capítulo 3, especialmente nas características do Modelo Conforme [22, 40]. Estes conceitos permitem lidarmos com objetos da geometria de maneira intuitiva sem uma preocupação direta com um sistema de coordenadas em suas operações. Esferas do \mathbb{R}^3 são representadas por vetores no espaço conforme e podem ser consideradas os elementos básicos deste modelo. Os produtos da AC são usados no cálculo de interseções entre objetos, o que torna intuitiva a interpretação de círculos como uma das possibilidades de interseção de duas esferas. De fato, esta é uma forma de definir o círculo no modelo conforme. A interseção de esferas ainda nos leva a um importante elemento chamado de *par de pontos*. O par de pontos é o resultado da interseção de três esferas e é o principal objeto de nossa abordagem ao iPDGDM. No C-iBP, os pontos no espaço de busca são resultados de interseções entre três esferas e pertencem, portanto, a um par de pontos. O procedimento de construção deste espaço de busca é descrito no Capítulo 4.

O Capítulo 4 trata das contribuições apresentadas neste trabalho, através da aplicação da AC ao iPDGDM. A interseção de um círculo com duas esferas de mesmo centro e raios distintos é tratada como a interseção de um círculo com uma casca esférica. Isto porque os pontos que buscamos estão exatamente na casca esférica gerada por estas esferas. Esta interseção, em geral, produz dois arcos na circunferência, e são estes arcos que contém os pontos factíveis. Descrevemos os pontos destes arcos em função de uma de suas extremidades, com a aplicação de um rotor, operador responsável pelas rotações no espaço conforme. Outra abordagem para descrição de um arco de circunferência pode ser encontrada em [9]. O ângulo entre as extremidades é o ângulo máximo de rotação. A principal característica do C-iBP é que este ângulo

Introdução

máximo muda de acordo com a factibilidade do arco. Em outras palavras, de acordo com informações disponíveis sobre as distâncias, em alguns casos estes arcos podem ser reduzidos. Sendo assim, dizemos que o ângulo máximo de rotação é o maior ângulo para o qual a rotação de uma extremidade ao longo do arco gera um ponto ainda factível, o que permite que apenas pontos factíveis sejam computados. Ao contrário do iBP, o C-iBP descarta as regiões infactíveis antes de gerar os pontos para o espaço de busca. Com isso buscamos reduzir o número de pontos necessários para que uma solução seja encontrada. A primeira versão do algoritmo apresenta resultados promissores, mostrados na Seção 4.5. O desenvolvimento do algoritmo segue em busca de novas versões com número de operações e tempo de processamento ainda mais competitivos.

Capítulo 2

O Problema de Geometria de Distâncias Moleculares

Neste capítulo, apresentamos as classes de problemas estudados durante este trabalho, além dos algoritmos utilizados para resolvê-los. Inicialmente, definimos o PGDM (*Problema de Geometria de Distâncias Moleculares*) [29], uma classe de problemas de geometria de distâncias (PGD) aplicada ao cálculo de estruturas moleculares, em seguida o PDGDM (Problema Discreto de Geometria de Distâncias em Moléculas) [30], uma discretização do PGDM, que lida com um subconjunto de instâncias do problema original. Apresentamos o PDGDM exato [30], isto é, onde todas distâncias conhecidas são exatas, para depois definirmos o PDGDM intervalar (*i*PDGDM) [31], em que as distâncias são fornecidas através de intervalos. Os algoritmos apresentados no decorrer do texto, o *Branch and Prune* (BP) [37] e o BP intervalar (*i*BP) [31], são aplicados à proteínas, mais precisamente à cadeia principal de átomos destas proteínas.

2.1 PGDM

A função de uma molécula é determinada pela sua estrutura química, isto é, pelos átomos que a compõem e como eles se ligam, e pela sua estrutura tridimensional [12]. Determinar esta estrutura em \mathbb{R}^3 , chamada de conformação, é um problema que envolve métodos matemáticos e análise química. Algumas informações sobre a estrutura são obtidas com o uso de técnicas de Ressonância Magnética Nuclear (RMN) [21], capazes de fornecer a distância entre pares de

átomos separados por uma distância menor que um valor entre 5 e 6 Å [50]. O problema de encontrar as posições dos átomos em \mathbb{R}^3 , dado um conjunto de distâncias entre eles, é definido por um grafo $G = (V, A, d)$ como um PGD, onde V é o conjunto de átomos e A é o conjunto dos pares de átomos $\{u, v\}$ para os quais conhecemos a distância d_{uv} .

Definição 1. *Problema de Geometria de Distâncias Moleculares (PGDM):* dado um grafo simples, ponderado e não orientado $G = (V, A, d)$, com $d : A \rightarrow \mathbb{R}_+$, existe uma imersão $x : V \rightarrow \mathbb{R}^3$, tal que $\|x_u - x_v\| = d_{uv}$, para todo $\{u, v\} \in A$? “ $\|\cdot\|$ ” representa a norma euclidiana.

O PGDM é um problema **NP**-difícil [48], a menos que todas as distâncias inter atômicas sejam conhecidas. Neste caso, ele é resolvido em tempo linear [13].

Este problema é usualmente formulado como um problema de otimização contínua:

$$\min g(x),$$

para

$$g(x) = \sum_{\{u,v\} \in A} (\|x_u - x_v\|^2 - d_{uv}^2)^2, \quad (2.1)$$

onde x é solução se, e somente se $g(x) = 0$.

Para formalizar o PGDM da forma como ele é abordado aqui, são feitas três suposições que o caracterizam como um problema de decisão. São elas:

- consideramos um conjunto de distâncias exatas, ao invés de aproximadas;
- não ocorre nenhum erro de medida;
- a imersão ótima tridimensional do grafo não sofreu influência de um termo de minimização da energia potencial na função objetivo.

Os dois primeiros itens referem-se à erros que provém das distâncias medidas através da RMN. Erros sistemáticos em cada medida são tratados introduzindo-se limites para cada distância [36]. Em [37], é apresentado um método que lida com esta variação, e medidas completamente erradas são tratadas com o uso de um Código Corretor de Erros (CCE) proposto em [5]. Para o método apresentado aqui, de [30], o CCE pode ser usado como um passo de pré processamento.

Esta teoria tem aplicações, principalmente, em conformação molecular e proteômica, localização de redes sensoriais, e na construção e rigidez de grafos. Essas aplicações têm em comum a geometria de distâncias euclidianas. Um índice de contribuições para a geometria de distâncias com o uso de grafos e redes sensoriais, além de uma série de abordagens ao PGDM é apresentado em [30].

Uma importante limitação dos algoritmos aplicados ao PGDM é a precisão da solução, isto porque várias conformações produzem na função objetivo valores muito próximos de zero. Algoritmos combinatórios têm se mostrado capazes de produzir valores mais precisos, o que o tornaria mais eficiente para discriminar os valores da função objetivo. Isso serviu de motivação para o trabalho com esta classe de algoritmos.

Muitos algoritmos são testados em proteínas, que consistem de uma cadeia principal de átomos e cadeias secundárias. Neste trabalho, consideraremos apenas a cadeia principal. Para cada aminoácido que as compõem, consideramos os átomos N , C_α e C , e esta estrutura nos permite formular o PGDM de maneira *discreta*, dando origem ao PDGDM. Este problema aplica-se a um subconjunto de instâncias do PGDM para as quais uma abordagem discreta é viável. Este conjunto contém a maioria das cadeias principais de proteínas [30].

2.2 PDGDM

Nesta seção, apresentamos o PDGDM, algumas de suas propriedades, exigências e sua interpretação geométrica.

Quando uma ordem é definida dentro de V , e u , v são respectivamente o i -ésimo e o j -ésimo índices, podemos escrever $d_{i,j}$ para $d_{u,v}$.

Definição 2. *Problema Discreto de Geometria de Distâncias Moleculares (PDGDM):* dado um grafo simples, ponderado e não orientado $G = (V, A, d)$ tal que exista uma ordem v_1, \dots, v_n em V satisfazendo:

1.

$$\forall i \in \{4, \dots, n\} \quad \forall j, k \in \{i-3, \dots, i\} \quad (\{j, k\} \in A);$$

2. é válida a desigualdade triangular:

$$\forall i \in \{2, \dots, n-1\} \quad d_{i-1,i+1} < d_{i-1,i} + d_{i,i+1};$$

Existe uma imersão $x : V \rightarrow \mathbb{R}^3$ tal que $\|x_u - x_v\| = d_{u,v} \forall \{u, v\} \in A$?

As distâncias entre átomos consecutivos $d_{i-1,i}$ para $2 \leq i \leq n$, são chamadas de *comprimentos de ligação*, e os ângulos $\theta_{i-2,i}$ entre os átomos v_{i-2} , v_{i-1} e v_i para $3 \leq i \leq n$ são chamados de *ângulos de ligação*. O conjunto A é particionado em dois conjuntos, H e F , onde $H = \{\{i, j\} \in A \mid |i - j| \leq 3\}$ e $F = A \setminus H^1$. Além disso, a ordem em V é definida por uma cadeia linear de átomos conectados por ligações covalentes.

Na Definição 2, a primeira afirmação exige que os comprimentos e ângulos de ligação, assim como as distâncias entre átomos separados por no máximo três posições sejam conhecidas. Grupos de 4 átomos consecutivos na cadeia principal de uma proteína costumam estar mais próximos uns aos outros do que o limiar de 6 Å que os experimentos de RMN são capazes de medir, o que torna essa afirmação aplicável a muitas proteínas [12, 50]. A segunda afirmação, em termos de uma desigualdade triangular, diz que nenhum ângulo de ligação pode ser um múltiplo de π . Não se conhece nenhuma proteína com ângulo de ligação igual a π , o que torna a segunda afirmação também aplicável a proteínas.

Para determinar se uma instância do PGDM é de fato um instância do PDGDM, é necessário encontrar uma ordenação que satisfaça as duas afirmações da Definição 2. Lavor *et al.* discutem o problema em relação à afirmação 1 em [28]. Já a afirmação 2 é satisfeita com probabilidade 1. O conjunto de triplas de valores que satisfazem a desigualdade triangular na igualdade tem medida nula no conjunto de todas estas triplas.

2.3 Discretização do PGDM

Além dos comprimentos e ângulos de ligação, para se descrever uma molécula de n átomos também é necessário conhecer os ângulos de torção (ou diedrais) $\omega_{i-3,i}$, $i \geq 4$. Este é o ângulo formado pelos vetores normais aos planos definidos por $i-3$, $i-2$, $i-1$ e $i-2$, $i-1$, i (Figura 2.3-1). É importante observar que na maioria dos casos assumimos que todos os ângulos

¹“\” é a operação de complemento de conjuntos.

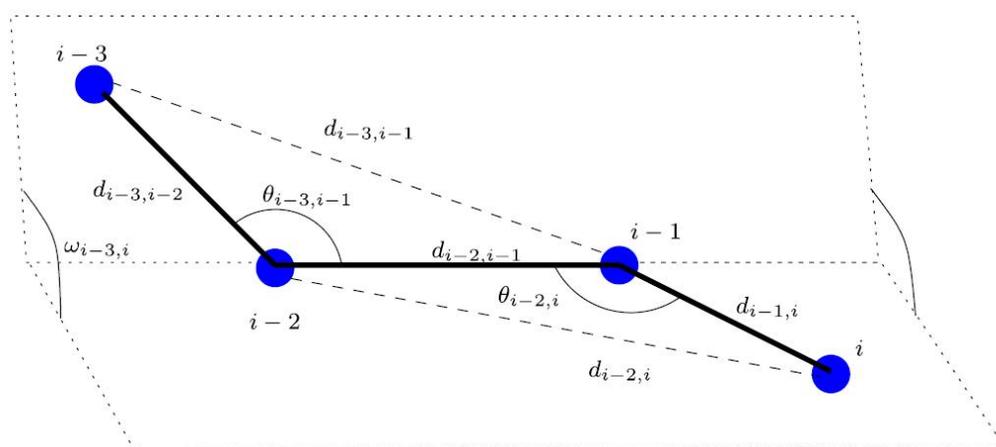


Figura 2.3-1. Comprimentos de ligação, ângulos de ligação e de torção.

e comprimentos de ligação são conhecidos. De forma que podemos fixar os 3 primeiros átomos da cadeia (Seção 2.4) e encontrar o 4^o com o auxílio de $\omega_{1,4}$. Para encontrarmos o 5^o, utilizamos $\omega_{1,4}$ e $\omega_{2,5}$, e assim por diante em toda a cadeia.

Geometricamente, na formulação discreta, a busca pelo átomo na posição i pode ser interpretada como uma interseção entre três esferas. Tais esferas são centradas nos átomos correspondentes às posições $i-3$, $i-2$ e $i-1$, e possuem raios $d_{i-3,i}$, $d_{i-2,i}$ e $d_{i-1,i}$, respectivamente. As afirmações da Definição 2 e o fato de dois átomos não ocuparem o mesmo lugar no espaço garantem que haja no máximo duas possibilidades para o i -ésimo átomo, pois a interseção, em geral, se dará em no máximo dois pontos (Figura 2.3-2). Esta interpretação motivou a abordagem deste problema através da Álgebra de Clifford.

Portanto, fixados os 3 primeiros átomos, há apenas 2 possibilidades para o 4^o, e cada uma delas nos fornece outras 2 para o 5^o átomo, e assim por diante, de modo que para o n -ésimo átomo teremos 2^{n-3} possibilidades, que conseqüentemente é o número de possíveis conformações de n átomos. Desta maneira, o espaço de soluções pode ser descrito por uma árvore binária (Figura 2.3-3). Algumas dessas possibilidades são ineficazes, de acordo com informações sobre distâncias adicionais, isto é, aquelas entre pares de átomos distantes mais de três posições.

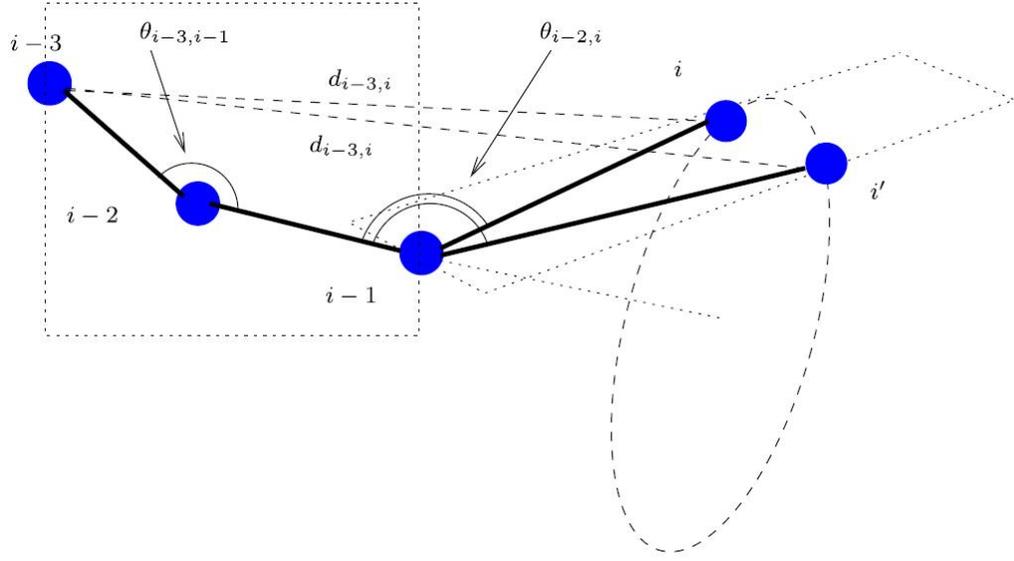


Figura 2.3-2. Há apenas 2 possibilidades para o átomo i .

2.4 Posição Espacial dos Átomos

Dados os comprimentos $d_{1,2}, \dots, d_{n-1,n}$ e os ângulos de ligação e torção, respectivamente $\theta_{1,3}, \dots, \theta_{n-2,n}$ e $\omega_{1,4}, \dots, \omega_{n-3,n}$, sabemos de [43] que a posição espacial $(x_{i_1}, x_{i_2}, x_{i_3})$ de cada átomo i pode ser obtida através de (2.2).

$$\begin{bmatrix} x_{i_1} \\ x_{i_2} \\ x_{i_3} \\ 1 \end{bmatrix} = B_1 B_2 B_3 \dots B_i \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \quad i \in 4, \dots, n, \quad (2.2)$$

onde $B_1 = I_4$,

$$B_2 = \begin{bmatrix} -1 & 0 & 0 & -d_{1,2} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, B_3 = \begin{bmatrix} -\cos \theta_{1,3} & \text{sen } \theta_{1,3} & 0 & -d_{2,3} \cos \theta_{1,3} \\ \text{sen } \theta_{1,3} & -\cos \theta_{1,3} & 0 & d_{2,3} \text{sen } \theta_{1,3} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

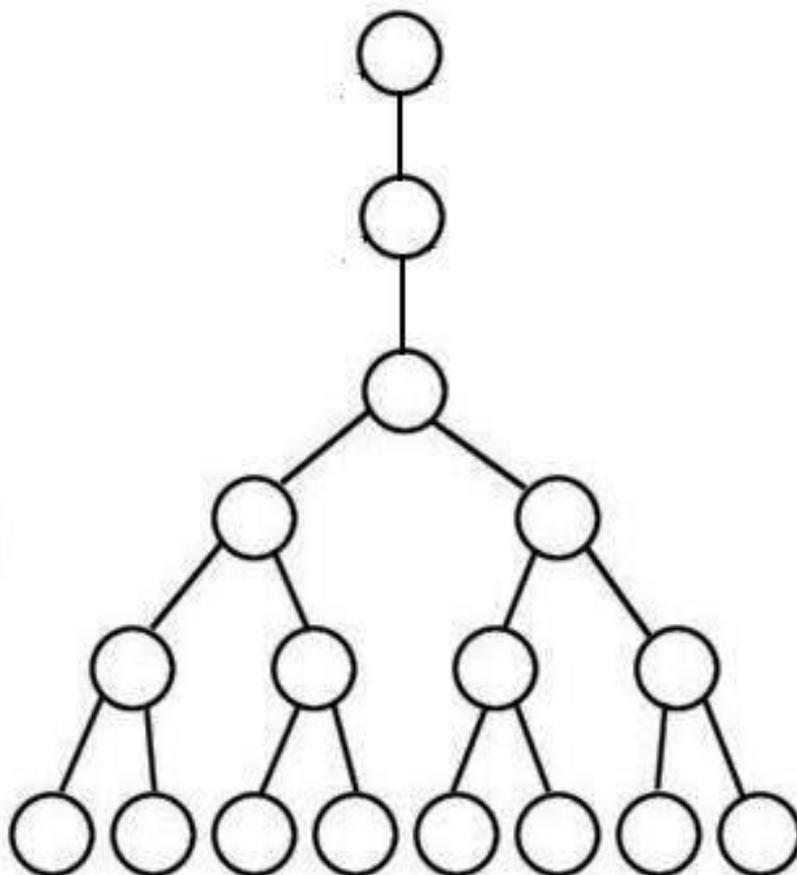


Figura 2.3-3. Exemplo de uma árvore binária que representa o número de possibilidades para cada átomo, a partir do posicionamento dos três primeiros.

e

$$B_i = \begin{bmatrix} -\cos \theta_{i-2,i} & \text{sen } \theta_{i-2,i} & 0 & -d_{i-1,i} \cos \theta_{i-2,i} \\ \text{sen } \theta_{i-2,i} \cos \omega_{i-3,i} & -\cos \theta_{i-2,i} \cos \omega_{i-3,i} & -\text{sen } \omega_{i-3,i} & d_{i-1,i} \text{sen } \theta_{i-2,i} \cos \omega_{i-3,i} \\ \text{sen } \theta_{i-2,i} \text{sen } \omega_{i-3,i} & -\cos \theta_{i-2,i} \text{sen } \omega_{i-3,i} & \cos \omega_{i-3,i} & d_{i-1,i} \text{sen } \theta_{i-2,i} \text{sen } \omega_{i-3,i} \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (2.4)$$

para $i \in \{4, \dots, n\}$. Ou seja, as coordenadas cartesianas de todos os átomos são obtidas através dos valores de $\cos \omega_{i-3,i}$ e $\text{sen } \omega_{i-3,i}$.

Para determinarmos completamente B_2 e B_3 , precisamos apenas de $d_{1,2}$, $d_{2,3}$ e $\theta_{1,3}$, que, como dissemos, são dados conhecidos para o PDGDM, de forma que os 3 primeiros átomos da

cadeia são fixados nas posições a seguir:

$$\begin{aligned} x_1 &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \\ x_2 &= \begin{pmatrix} -d_{1,2} \\ 0 \\ 0 \end{pmatrix}; \\ x_3 &= \begin{pmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} \\ d_{2,3} \sin \theta_{1,3} \\ 0 \end{pmatrix}. \end{aligned}$$

Do fato de todas as distâncias entre átomos $i - 3$, $i - 2$, $i - 1$ e i serem conhecidas, os ângulos de torção $\cos \omega_{i-3,i}$ podem ser calculados para todo $i \in \{4, \dots, n\}$, através de (2.5), expressão baseada na lei dos cossenos para ângulos de torção (diedrais) [44],

$$\cos \omega_{i-3,i} = \frac{d_{i-3,i-2}^2 + d_{i-2,i}^2 - 2d_{i-3,i-2}d_{i-2,i} \cos \theta_{i-2,i} \cos \theta_{i-1,i+1} - d_{i-3,i}^2}{2d_{i-3,i-2}d_{i-2,i} \sin \theta_{i-2,i} \sin \theta_{i-1,i+1}}. \quad (2.5)$$

Dessa forma, é possível encontrar o valor de $\cos \omega_{1,4}$ a fim de posicionar o quarto átomo, de modo que $\sin \omega_{1,4} = \pm \sqrt{1 - \cos^2 \omega_{1,4}}$. Estas duas possibilidades para $\sin \omega_{1,4}$ implicam que teremos no máximo duas possibilidades para o quarto átomo, x_4 e x'_4 :

$$\begin{aligned} x_4 &= \begin{bmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} - d_{3,4} \cos \theta_{1,3} \cos \theta_{2,4} + d_{3,4} \sin \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{2,3} \sin \theta_{1,3} - d_{3,4} \sin \theta_{1,3} \cos \theta_{2,4} - d_{3,4} \cos \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{3,4} \sin \theta_{2,4} (\sqrt{1 - \cos^2 \omega_{1,4}}) \end{bmatrix}, \\ x'_4 &= \begin{bmatrix} -d_{1,2} + d_{2,3} \cos \theta_{1,3} - d_{3,4} \cos \theta_{1,3} \cos \theta_{2,4} + d_{3,4} \sin \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{2,3} \sin \theta_{1,3} - d_{3,4} \sin \theta_{1,3} \cos \theta_{2,4} - d_{3,4} \cos \theta_{1,3} \sin \theta_{2,4} \cos \omega_{1,4} \\ d_{3,4} \sin \theta_{2,4} (-\sqrt{1 - \cos^2 \omega_{1,4}}) \end{bmatrix}. \end{aligned}$$

A única diferença entre x_4 e x'_4 é o sinal da última coordenada. Para cada solução parcial de i átomos, existem no máximo 2 possibilidades para o átomo $i + 1$, de forma que o número total de soluções para uma molécula de n átomos é finito, e no máximo de 2^{n-3} , a menos de rotações e translações. Este fato está demonstrado em [30].

2.4.1 Complexidade

Existe uma simetria nas conformações que resolvem o PDGDM em relação ao plano determinado pelos três primeiros átomos, daí o fato de x_4 e x'_4 só se diferenciarem por um sinal em uma coordenada. Esta propriedade pode reduzir pela metade o custo computacional da resolução. Prova-se também que o PDGDM é um problema NP-difícil, [30]

2.5 Algoritmo *Branch and Prune* (BP)

Nesta seção, apresentamos o algoritmo BP, proposto por C. Lavor, L. Liberti, N. Maculan e A. Mucherino para a solução do PDGDM [39].

A ação do BP é descrita com base na árvore binária da Figura 2.3-3. A cada passo, o algoritmo determina duas possibilidades (nós) para o posicionamento de um átomo, x_i e x'_i , porém, uma delas ou as duas podem ser infactíveis de acordo, por exemplo, com informações sobre distâncias adicionais. A busca se ramifica por todos os nós factíveis da árvore. Quando um nó é infactível, os caminhos que se utilizam dele são descartados.

Um dos testes de factibilidade é dado por: para todo par $\{i, j\} \in F$ com $j < i$ verifica-se a desigualdade $||x_j - x_i|| - d_{j,i} < \epsilon$, onde $\epsilon > 0$ é uma tolerância dada e $|\cdot|$ indica o módulo em \mathbb{R} . Se a desigualdade não for satisfeita, o nó i é descartado. Apesar de simples, este teste é efetivo. A estrutura do algoritmo se completa com os testes de poda, que podem ser de natureza geométrica ou físico-química. Outra importante característica é que, no pior caso, de tempo exponencial, o BP encontrará todas as soluções incongruentes para a instância dada.

2.5.1 Estrutura do Algoritmo

Seja G o grafo que representa a árvore de busca. G é inicializado nos nós $\{1, 2, 3, 4\}$. Isto porque os 3 primeiros átomos são fixados e devido à simetria de soluções em relação ao plano determinado por eles, podemos fixar também o 4º átomo em uma de suas possíveis posições. No i -ésimo nível do grafo, isto é no momento em que o átomo i é colocado, armazena-se:

- a posição $x_i \in \mathbb{R}^3$ do átomo i ;

Algoritmo 1: BP.

```
1 BranchAndPrune( $G, v, i$ )
2 se ( $i \leq n - 1$ ) então
3   // CALCULAR AS POSSÍVEIS POSIÇÕES PARA O  $i$ -ÉSIMO ÁTOMO:
4   calcula as matrizes de torção  $B_i, B'_i$  via (2.4);
5   recupera a matriz de torção acumulada  $Q_{i-1}$  de  $P(v)$ ;
6   calcula  $Q_i = Q_{i-1}B_i, Q'_i = Q_{i-1}B'_i$  e  $x_i, x'_i$  de  $Q_i y, Q'_i y$ ;
7   seja  $\lambda = 1, \rho = 1$ ;
8   // APLICAR TESTE DE FACTIBILIDADE:
9   se ( $x_i$  é factível) então
10    cria um nó  $z$ , armazena  $Q_i$  e  $x_i$  em  $z$ , e seja  $P(z) = v$  e  $L(v) = z$ ;
11    defina  $G \leftarrow G \cup \{z\}$ ;
12    BranchAndPrune( $G, z, i + 1$ );
13  senão
14    defina  $L(v) = \text{PODA}$ ;
15  fim
16  se ( $x'_i$  é factível) então
17    cria um nó  $z'$ , armazena  $Q_i$  e  $x_i$  em  $z'$ , seja  $P(z) = v$  e  $R(v) = z'$ ;
18    defina  $G \leftarrow G \cup \{z'\}$ ;
19    BranchAndPrune( $G, z', i + 1$ );
20  senão
21    defina  $R(v) = \text{PODA}$ ;
22  fim
23 senão
24   // NÍVEL  $n$  ALCANÇADO, UMA SOLUÇÃO FOI ENCONTRADA:
25   solução armazenada nos nós nivelados de  $n$  a 1, saída dada pelo caminho inverso;
26 fim
```

- o produto acumulado das matrizes de torção $Q_i = \prod_{j=1}^i B_j$;
- um ponteiro para o nó de origem $P(i)$;
- ponteiros para os subnós $L(i)$ e $R(i)$;

O algoritmo BP está representado pelo algoritmo 1, para uma molécula de n átomos.

A execução do algoritmo gera uma árvore binária em que, a cada iteração, 2 novos nós são computados, representando as duas posições possíveis para o átomo correspondente. Durante a construção da árvore, são realizados os testes de factibilidade, que podem ser encontrados em [38], de modo que os ramos com nós infactíveis sejam podados. Este processo reduz rapi-

damente o tamanho da árvore diminuindo consideravelmente o tempo de solução do problema.

Além disso, o algoritmo BP se mostrou capaz de produzir soluções bastante precisas para instâncias relacionadas a conformação de proteínas. Uma forma de constatarmos tal fato se dá através do cálculo de uma função de penalização chamada *LDE* (*Largest Distance Error*), dada por (2.6)

$$LDE(X) = \frac{1}{|A|} \sum_{i,j} \frac{||x_i - x_j|| - d_{i,j}}{d_{i,j}}, \quad (2.6)$$

onde $X = \{x_1, x_2, \dots, x_n\}$ é uma conformação de uma molécula de n átomos e $|A|$ é o número de distâncias conhecidas (o número de elementos do conjunto A na Definição 1).

A seguir, apresentamos alguns resultados computacionais obtidos com o software *MD-jeep* [39], sem a preocupação com a comparação entre este e outros métodos (estas informações estão detalhadas também em [39]).

2.6 Resultados Computacionais

A Tabela 2.1 apresenta os resultados do MD-jeep para algumas instâncias geradas artificialmente, através da extração da cadeia principal de átomos de moléculas de proteínas. Tais moléculas são encontradas no PDB (*Protein Data Bank*) [6], um banco de dados disponível na internet voltado exclusivamente para a conformação tridimensional de proteínas. Esta técnica é bastante utilizada [7, 30, 53], e consiste em calcular todas as distâncias possíveis entre átomos, mantendo aquelas que sejam menores que 6 Å, a fim de simular a obtenção dos dados através de RMN.

A primeira coluna contém o nome de cada molécula tal como no PDB, seguida do número de átomos, quantidade de distâncias conhecidas, número de soluções encontradas, LDE da melhor solução e o tempo de execução. A eficiência do MD-jeep pode ser constatada pelos valores de LDE sempre muito próximos de 0.

A Figura 2.6-4 é da conformação da melhor solução para a instância *1rgs* de 792 átomos, mostrada na Tabela 2.1, obtida pelo software RasMol [49].

instância	n	$ A $	# Sol	melhor LDE	tempo (s)
1crn	138	846	2	5.79e-14	0
1hoe	222	1259	2	7.26e-14	0
1jk2	270	1816	8	5.63e-14	0
1a70	291	1628	2	3.25e-13	0
1rgs	792	4936	8	1.55e-13	0.05
1n4w	1610	10920	2	2.58e-13	0.04
1mqq	2032	13016	8	5.40e-13	0.08
1rwh	2265	14057	2	4.49e-14	0.1
3b34	2790	19563	4	4.91e-12	0.15
2e7z	2907	27706	2	1.22e-12	0.17

Tabela 2.1. Resultados obtidos com processador Intel Core 2 Duo T8300 2.1Gz e 4Gb de memória RAM, operando Linux.

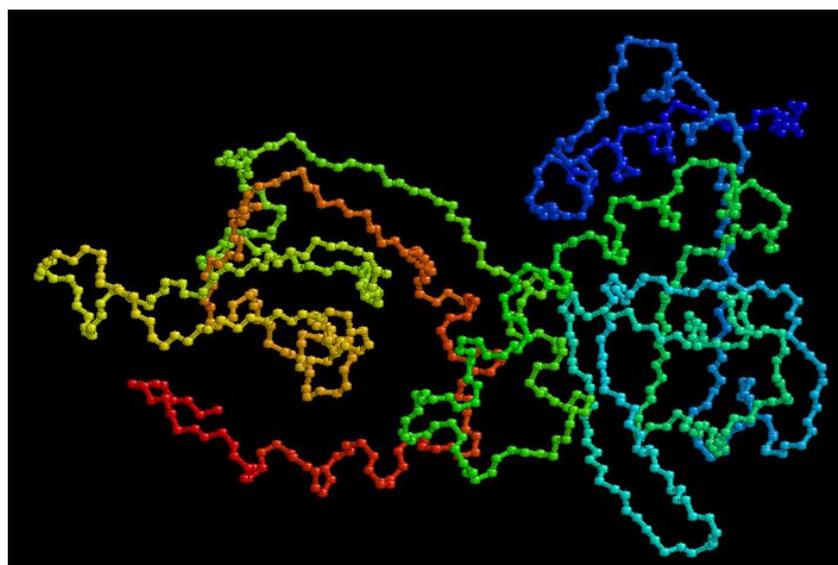


Figura 2.6-4. Representação do traçado da cadeia principal de átomos da molécula *1rgs*. As cores representam cadeias de aminoácidos.

2.7 PDGDM Intervalar

O problema foi descrito até aqui considerando o caso exato. Nesta Seção, destacamos o caso intervalar, em que parte das distâncias exatas serão substituídas por conjuntos de limites inferiores e superiores. Para este caso, a discretização ainda é possível e um algoritmo BP intervalar (*iBP*) é proposto para a resolução do problema discretizado.

As instâncias do *iPDGDM* devem satisfazer o conjunto de suposições dadas na Seção (2.7.1). Se isto não acontece imediatamente, uma instância pode passar por um processo de

reordenação de seus átomos, a fim de atender, então, às hipóteses do problema.

2.7.1 Definição do *i*PDGDM

A definição do problema é a mesma utilizada para o PDGDM, mas desta vez as informações sobre as distâncias são divididas de acordo com três subconjuntos de A : A' , A'' e F , definidos por:

- A' consiste no conjunto de todos os pares de átomos separados por no máximo duas ligações covalentes. As distâncias interatômicas neste conjunto são exatas.
- A'' consiste nos pares de átomos separados por exatamente três ligações covalentes. Para estes pares é possível calcular limites inferiores e superiores para as distâncias, que formam o que chamamos de distâncias intervalares. Os possíveis valores destas distâncias serão representados por um conjunto discretizado de D valores dentro de cada intervalo.
- F é o conjunto de distâncias que podem ser estimadas pela RMN. A partir de agora, estas distâncias também são representadas por intervalos. Repetimos que estes experimentos são capazes apenas de estimar distâncias entre átomos mais próximos do que um limiar entre 5 e 6 Å.

As distâncias correspondentes ao conjunto F podem conter erros de medida. Por esse motivo, a discretização do espaço da busca é feita com base apenas nos nós correspondentes à $A' \cup A''$, sendo que as distâncias em F são usadas para a verificação da factibilidade de imersões parciais. Desta forma, o espaço discretizado de busca independe completamente de dados experimentais de RMN.

2.7.2 Instâncias do *i*PDGDM

Dado um grafo G , a instância realacionada a ele pode não satisfazer as hipóteses do *i*PDGDM. Em [31], uma reordenação na sequência de átomos é proposta, de modo que a nova sequência satisfaça as hipóteses necessárias. Nesta nova sequência, os átomos podem aparecer duplicados. Porém, a distância entre cópias do mesmo átomo é igual a zero, e isto aumenta o número de

distâncias exatas que são considerados. Embora estas repetições impliquem que a nova sequência seja maior que a original, cópias de um átomo já posicionado só têm uma possibilidade de posicionamento, não permitindo uma ramificação da árvore em relação a este ponto. Logo, o aumento na sequência não se reflete em um aumento da árvore obtida da discretização.

Seja $G = (V, A, d)$ um grafo associado a uma instância de um PGD e $V' = V \cup \{0\}$. O conjunto A é particionado em A' e A'' , definidos na Seção 2.7.1.

Definição 3. Uma *reordenação* é um sequência $r : \mathbb{N} \rightarrow V'$ de comprimento $|r| \in \mathbb{N}$, com $r_i = 0, \forall i > |r|$, tal que:

- $G[\{r_1, r_2, r_3\}]$ é uma clique²;
- $\forall i \in \{4, \dots, |r|\}$, os conjuntos $\{r_{i-2}, r_i\}$ e $\{r_{i-1}, r_i\}$ são pares em A' ;
- $\forall i \in \{4, \dots, |r|\}$, o conjunto $\{r_{i-3}, r_i\}$ satisfaz ou $r_{i-3} = r_i$ ou é um par $A' \cup A''$.

Como dissemos, na prática, a reordenação gera uma cadeia principal de proteínas virtualmente mais longa que a original, cuja estrutura é derivada do grafo original, a fim de obtermos uma instância que satisfaça as hipóteses do problema. As duas primeiras propriedades da reordenação garantem que há pelo menos dois antecessores adjacentes à um átomo r_i , cujas distâncias a r_i são exatas, para $i = \{3, 4, \dots, |r|\}$. Se $\{r_{i-3}, r_i\} \in A'$ ou $r_{i-3} = r_i$, de acordo com a discretização, r_i possui duas possibilidades de posicionamento. E caso $\{r_{i-3}, r_i\} \in A''$, teremos a situação descrita pela Figura 2.7-5, em que r_i pode estar em qualquer ponto de dois arcos simétricos de um circunferência. Neste caso, a discretização ocorre da maneira que descrevemos na definição de A'' , na Seção 2.7.1. Uma quantidade D , de valores dentro do intervalo correspondente à distância entre r_{i-3} e r_i , é determinada, e cada um destes valores produz um ponto sobre os arcos.

Desta forma, uma **reordenação** corresponde à uma instância do PDGDM, com distâncias intervalares. A construção de um reordenação, a partir de uma instância que originalmente não era uma instância do PDGDM é detalhada em [31].

²Uma clique é um subgrafo completo, ou seja, um subgrafo com arestas entre quaisquer par de pontos.

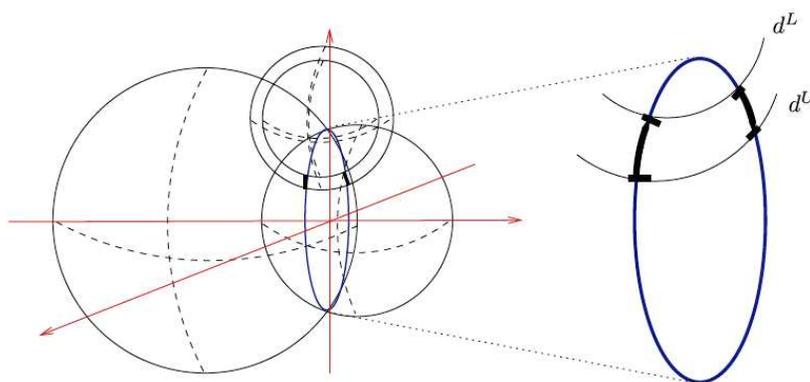


Figura 2.7-5. O círculo na interseção de duas esferas, é intersectado por uma casca esférica, produzindo dois arcos.

2.7.3 Algoritmo *iBP*

O algoritmo *iBP* é uma extensão, com a integração de distâncias intervalares, do algoritmo BP já apresentado, e foi o primeiro algoritmo exato proposto para instâncias do PDGDM, que considera distâncias intervalares. Os dados de entrada são: o índice i da reordenação, cuja imagem r_j indexa o átomo que está sendo posicionado; a reordenação r ; a função de distâncias inter atômicas d ; e um inteiro D , que é o número de valores escolhidos para a discretização dos intervalos.

No passo 4, que se inicia na linha 11 do algoritmo 2, testa-se a factibilidade de todas as posições possíveis para o átomo r_i . A recursividade do algoritmo *iBP* gera a estrutura de uma árvore de busca, em que cada nó possui um número de subnós igual a b . Os nós no nível $|r|$ correspondem à imersões.

Alguns resultados computacionais deste algoritmo são encontrados em [31], e no Capítulo 4.

Algoritmo 2: iBP.

```
1 iBP( $j, r, d, D$ )
2 se ( $r_j$  é um átomo duplicado) então
3   copia as coordenadas da cópia anterior de  $r_j$  em  $x_{r_j}^1$ 
4   iBP( $j + 1, r, d, D$ );
5 senão
6   se ( $d(r_{j-3}, r_j)$  é exata) então
7      $b = 2$ ;
8   senão
9      $b = 2D$ ;
10  fim
11  para  $k \in \{1, \dots, b\}$  faça
12    calcula a  $k$ -ésima posição  $x_{r_j}^k$  para o  $r_j$ -ésimo átomo;
13    verifica a factibilidade de  $x_{r_j}^k$  utilizando informações em  $F$  (Seção 2.7.1);
14    se ( $x_{r_j}^k$  é factível) então
15      se ( $j = |r|$ ) então
16        uma solução  $x$  foi encontrada;
17      senão
18        iBP( $j + 1, r, d, D$ );
19      fim
20    fim
21  fim
22 fim
```

Capítulo 3

Introdução à Álgebra de Clifford

Neste capítulo, apresentamos os conceitos da Álgebra de Clifford (AC) necessários para o desenvolvimento do trabalho. O capítulo se divide da seguinte forma: no início são apresentadas as Álgebras de Clifford de duas e três dimensões sobre os reais, além de uma introdução ao produto geométrico. Em seguida são definidos os produtos que agem numa AC, além de outras importantes operações. As transformações como reflexão e rotação são apresentadas na sequência e, finalmente, são definidos os modelos homogêneos da AC, com ênfase ao modelo conforme. Este modelo será aplicado ao problema de geometria de distâncias tratado neste trabalho. Abordagens mais completas sobre a Álgebra de Clifford podem ser encontradas em [15, 22, 23, 33, 40, 51, 52]. Um tratamento histórico sobre o desenvolvimento da AC é visto, dentre outras referências, em [33].

3.1 Álgebras de Clifford Cl_2 e Cl_3 e o Produto Geométrico

Nesta seção, são definidas as Álgebras de Clifford Cl_2 e Cl_3 e seus elementos, os multivetores, bem como é introduzido o produto geométrico, ou de Clifford.

3.1.1 Cl_2 e o Bivetor

Consideremos o espaço \mathbb{R}^2 , com a norma euclidiana, e sua base canônica $\{e_1, e_2\}$. Definimos um novo produto entre vetores, a partir da seguinte exigência sobre $v \in \mathbb{R}^2$:

$$v^2 = \|v\|^2, \quad (3.1)$$

de forma que a norma seja preservada. Para determinar as propriedades deste produto, basta representar v como $v_1e_1 + v_2e_2$ em (3.1). Assim, temos que:

$$\begin{aligned} (v_1e_1 + v_2e_2)(v_1e_1 + v_2e_2) &= v_1^2 + v_2^2 \\ v_1^2e_1^2 + v_1v_2e_1e_2 + v_2v_1e_2e_1 + v_2^2e_2^2 &= v_1^2 + v_2^2, \end{aligned}$$

o que é satisfeito se

$$e_1^2 = e_2^2 = 1 \quad \text{e} \quad e_1e_2 = -e_2e_1. \quad (3.2)$$

As propriedades em (3.2) introduzem um novo elemento, chamado de *bivetor*, resultado deste produto entre dois vetores distintos e ortogonais. O bivetor e_1e_2 , Figura 3.1-1 pode ser escrito simplesmente como e_{12} . Sua orientação é determinada pelo sinal, de modo que $-e_{12}$

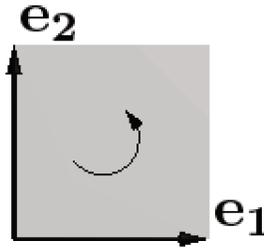


Figura 3.1-1. Bivetor e_1e_2 .

representa a mesma área com a orientação inversa. Um bivetor define um subespaço orientado bidimensional, enquanto um vetor define um subespaço orientado unidimensional.

Este produto, que será tratado em detalhes pela Seção 3.2.1 e que foi definido para vetores de \mathbb{R}^2 , é chamado de *produto de Clifford* ou *produto geométrico*. De (3.2), o produto geométrico

entre u e $v \in \mathbb{R}^2$ é dado por:

$$\begin{aligned} uv &= (u_1e_1 + u_2e_2)(v_1e_1 + v_2e_2) \\ &= u_1v_1 + u_2v_2 + (u_1v_2 - u_2v_1)e_{12}, \end{aligned} \tag{3.3}$$

a soma de um escalar, equivalente ao produto escalar entre u e v , e um bivector.

A área de um bivector é determinada pelo seu coeficiente, em (3.3), por exemplo, a área do bivector é $|u_1v_2 - u_2v_1|$.

Os vetores e_1 e e_2 , o bivector e_{12} , além de um escalar, formam uma base para o espaço multivetorial de 4 dimensões Cl_2 , Tabela 3.1. A álgebra definida sobre este espaço, com uma forma bilinear definida pelo produto interno de \mathbb{R}^2 , é a *Álgebra de Clifford Cl_2* . Um elemento desta álgebra é chamado de *multivetor*. Um multivetor arbitrário de Cl_2 é dado pela combinação linear dos elementos de sua base, (3.4).

escalar	1
vetores	e_1, e_2
bivector	e_{12}

Tabela 3.1. Base de Cl_2

$$u = u_0 + u_1e_1 + u_2e_2 + u_3e_{12}, \tag{3.4}$$

com $e_1^2 = e_2^2 = 1$ e $e_{12}^2 = -1$.

3.1.2 Álgebra de Clifford Cl_3

A Álgebra de Clifford Cl_3 é definida sobre \mathbb{R}^3 de forma análoga, com o uso de uma forma bilinear dada pelo produto interno de \mathbb{R}^3 . Uma base para Cl_3 é dada pela Tabela 3.2, e as regras para o produto geométrico são listadas em (3.6).

escalar	1
vetores	e_1, e_2, e_3
bivectores	e_{12}, e_{13}, e_{23}
trivector	e_{123}

Tabela 3.2. Base de Cl_3

Um multivetor arbitrário de Cl_3 é dado pela combinação linear dos 8 elementos da Tabela 3.2:

$$v = v_0 + v_1e_1 + v_2e_2 + v_3e_3 + v_4e_{12} + v_5e_{13} + v_6e_{23} + v_7e_{123}, \quad (3.5)$$

e para $i \neq j = 1, 2, 3$, temos

$$\begin{aligned} e_i^2 &= 1 \\ e_{ij}^2 &= -1 \\ e_{123}^2 &= -1 \end{aligned} \quad (3.6)$$

Um novo elemento é introduzido em Cl_3 , o *trivetor*. Este trivetor é obtido da multiplicação entre 3 vetores distintos e ortogonais de \mathbb{R}^3 , e resulta em um elemento orientado de volume. O trivetor e_{123} , Figura 3.1-2, é um subespaço orientado tridimensional. Sua orientação pode ser determinada pela regra da mão direita ou da mão esquerda, detalhes sobre a orientação do trivetor podem ser encontrados em [15, Capítulo 2].

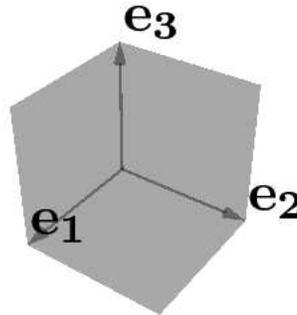


Figura 3.1-2. Trivetor e_{123} .

A multiplicação de dois vetores de \mathbb{R}^3 , como em Cl_2 , é a soma do produto escalar de \mathbb{R}^3 com um bivetor, de fato:

$$\begin{aligned} ab &= (a_1e_1 + a_2e_2 + a_3e_3)(b_1e_1 + b_2e_2 + b_3e_3) \\ &= (a_1b_1 + a_2b_2 + a_3b_3) + (a_1b_2 - a_2b_1)e_{12} + (a_1b_3 - a_3b_1)e_{13} + (a_2b_3 - a_3b_2)e_{23}. \end{aligned}$$

Estes resultados são estendidos à \mathbb{R}^n , construindo a Álgebra de Clifford Cl_n . Os subespaços que compõem a base de Cl_n , através da base canônica de \mathbb{R}^n , estão na Tabela 3.3. A Tabela

Grau	Elementos da base	Multiplicidade
0 (escalar)	1	$\binom{n}{0} = 1$
1 (vetores)	e_1, e_2, \dots, e_n	$\binom{n}{1} = n$
2 (bivetores)	e_{12}, e_{13}, e_{23}	$\binom{n}{2}$
3 (trivetores)	$e_{123}, e_{124}, \dots, e_{n-2, n-1, n}$	$\binom{n}{3}$
\vdots	\vdots	\vdots
n (pseudoescalar)	$e_{123\dots n}$	$\binom{n}{n} = 1$

Tabela 3.3. Base de Cl_n

Cl_n é formada por todos os subespaços de dimensão menor ou igual a n , obtidos de todas as combinações dos elementos e_1, e_2, \dots, e_n . O número de subespaços de dimensão k é dado por $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, e o número total de subespaços é igual a 2^n . De forma que um multivetor arbitrário de Cl_n é escrito como uma combinação linear de 2^n elementos.

O único subespaço de Cl_n com dimensão n é chamado de pseudoescalar, e é representado por I . Para Cl_3 , por exemplo, temos $I = e_{123}$.

Os subespaços que compõem uma Álgebra de Clifford são também chamados de *blades da base*. Um subespaço de dimensão k é dito um k -blade. Os escalares são representados por 0-blades, vetores por 1-blades e assim por diante. Um k -blade sempre pode ser representado por um produto exterior entre k vetores linearmente independentes. Uma combinação linear de k -blades é chamada de k -vetor.

3.2 Produtos na Álgebra de Clifford

Nesta seção, são apresentados alguns dos produtos definidos numa Álgebra de Clifford: geométrico, exterior e as contrações à esquerda e à direita. Destacamos suas propriedades, aplicações e interpretação geométrica.

3.2.1 Produto Geométrico

O produto geométrico (ou de Clifford), já introduzido para vetores, é um produto linear e invertível. É o produto fundamental da álgebra de Clifford, e os demais podem ser deduzidos a partir dele. Os axiomas do produto geométrico são listados abaixo:

$$\begin{aligned} e_i^2 &= \pm 1 \\ e_i e_j &= -e_j e_i \\ \alpha e_i &= e_i \alpha, \end{aligned} \tag{3.7}$$

para $i \neq j = 1, 2, \dots, n$ e $\alpha \in \mathbb{R}$. Para vetores, podemos generalizar os resultados obtidos para Cl_2 e Cl_3 . Dados dois vetores a e b de \mathbb{R}^n , temos

$$ab = a \cdot b + a \wedge b, \tag{3.8}$$

onde \cdot representa o conhecido produto escalar entre vetores, e \wedge representa um produto exterior, este último será revisto na Seção 3.2.2.

O produto entre multivetores é uma combinação linear dos produtos entre os elementos da base do espaço multivetorial (blades da base), de forma que é suficiente entendermos o produto geométrico apenas entre estes elementos. Para isso, usaremos o exemplo em (3.9):

$$\begin{aligned} e_{1234}e_{2356} &= e_{12342356} \\ &= e_{12234356}(-1)^2 \\ &= e_{12233456}(-1)^3 \\ &= -e_{1456}. \end{aligned} \tag{3.9}$$

O resultado apresenta apenas os índices que pertenciam à somente um dos termos, no caso acima: (1, 4) ao primeiro, (5, 6) ao segundo termo. As permutações nos índices determinam o sinal do produto, segundo os Axiomas em (3.7). O expoente de -1 em cada expressão indica o número de permutações até aquele momento.

Para um produto entre multivetores de Cl_2 , definimos uma tabela de multiplicação para as blades (Tabela 3.4).

O produto geométrico entre multivetores é ainda distributivo e associativo:

	e_1	e_2	e_{12}
e_1	1	e_{12}	e_2
e_2	$-e_{12}$	1	$-e_1$
e_{12}	$-e_2$	e_1	-1

Tabela 3.4. Produto Geométrico em Cl_2 .

- $A(B + C) = AB + AC$
- $(AB)C = A(BC)$,

em geral $AB \neq BA$.

Dois multivetores são multiplicados elemento a elemento, portanto para

$$a = a_0 + a_1e_1 + a_2e_2 + a_3e_{12} \quad e \quad b = b_0 + b_1e_1 + b_2e_2 + b_3e_{12}$$

temos

$$\begin{aligned} ab = & a_0b_0 + a_0b_1e_1 + a_0b_2e_2 + a_0b_3e_{12} + \\ & a_1b_0e_1 + a_1b_1e_1^2 + a_1b_2e_{12} + a_1b_3e_1e_{12} + \\ & a_2b_0e_2 + a_2b_1e_2e_1 + a_2b_2e_2^2 + a_2b_3e_2e_{12} + \\ & a_3b_0e_{12} + a_3b_1e_{12}e_1 + a_3b_2e_{12}e_2 + a_3b_3e_{12}^2 \end{aligned}$$

e com o auxílio da Tabela 3.4

$$\begin{aligned} ab = & (a_0b_0 + a_1b_1 + a_2b_2 - a_3b_3) + \\ & (a_0b_1 + a_1b_0 - a_2b_3 + a_3b_2)e_1 + \\ & (a_0b_2 + a_1b_3 + a_2b_0 - a_3b_1)e_2 + \\ & (a_0b_3 + a_1b_2 - a_2b_1 + a_3b_0)e_{12}. \end{aligned}$$

As regras para o produto entre elementos da base são gerais, portanto os resultados apresentados para Cl_2 são estendidos à Cl_n .

A seguir serão apresentados outros produtos definidos em uma Álgebra de Clifford, úteis no contexto desta tese.

3.2.2 Produto Exterior

O produto exterior, denotado por “ \wedge ”, é definido para vetores de \mathbb{R}^n a partir do produto geométrico. De (3.8), o produto exterior $a \wedge b$ é definido por:

$$a \wedge b = \frac{1}{2}(ab - ba). \tag{3.10}$$

A partir das propriedades do produto geométrico é possível, portanto, deduzir as propriedades do produto exterior. Dos axiomas em (3.7) e (3.10), para vetores do \mathbb{R}^n e $\alpha \in \mathbb{R}$ temos:

$$\begin{aligned} u \wedge v &= -v \wedge u \\ \alpha u \wedge u &= 0, \end{aligned} \tag{3.11}$$

isto é, o produto exterior é *anticomutativo*. Além disso, é nulo para vetores linearmente dependentes. O produto exterior, como o produto geométrico, é associativo, linear e distributivo.

O produto exterior entre dois vetores, gera um subespaço bidimensional, ou seja, um 2-blade, Figura 3.2-3. O produto de um 2-blade e um vetor, por sua vez, gera um espaço tridimensional, ou um 3-blade, Figura 3.2-4.

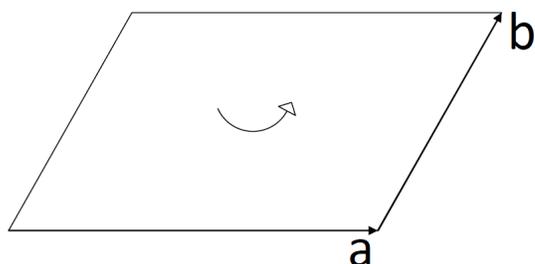


Figura 3.2-3. bivetor $a \wedge b$

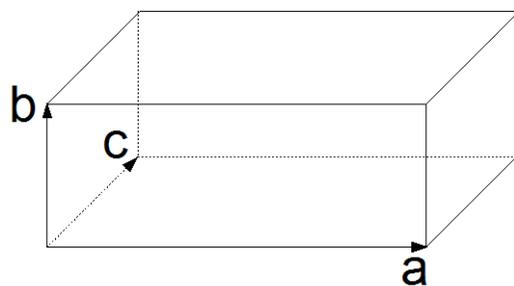


Figura 3.2-4. trivetor $a \wedge b \wedge c$

De forma geral, uma k -blade genérica é o produto exterior de k vetores. O produto exterior entre tais elementos tem dimensão igual a soma do grau de cada elemento. Ou seja, para A_k e B_l de graus k e l , respectivamente, onde $0 \leq k, l \leq n$, temos

$$A_k \wedge B_l = (AB)_{k+l}.$$

escalar	1
vetores	e_1, e_2, e_3
bivetores	$e_1 \wedge e_2, e_2 \wedge e_3, e_1 \wedge e_3$
pseudoescalar	$e_1 \wedge e_2 \wedge e_3$

Tabela 3.5. Base de $(\wedge \mathbb{R}^3, \wedge)$.

Se $k + l > n$, teremos $(AB)_{k+l} = 0$. Pois esta condição indica que os subespaços A_k e B_l são linearmente dependentes.

Uma vez que o produto exterior também é linear e distributivo, mais uma vez é suficiente entendermos o produto entre elementos da base, para compreendermos o produto entre multi-vetores.

3.2.2.1 A Álgebra Exterior $(\wedge \mathbb{R}^3, \wedge)$

Os subespaços $\wedge^0 \mathbb{R}^3, \wedge^1 \mathbb{R}^3, \wedge^2 \mathbb{R}^3$ e $\wedge^3 \mathbb{R}^3$ e o produto exterior determinam o espaço multi-vetorial $\wedge \mathbb{R}^3$, onde

$$\wedge \mathbb{R}^3 = \wedge^0 \mathbb{R}^3 \oplus \wedge^1 \mathbb{R}^3 \oplus \wedge^2 \mathbb{R}^3 \oplus \wedge^3 \mathbb{R}^3.$$

O espaço $\wedge \mathbb{R}^3$ equipado com o produto exterior define a álgebra exterior $(\wedge \mathbb{R}^3, \wedge)$ [33, 52].

A base para $(\wedge \mathbb{R}^3, \wedge)$ é dada pela Tabela 3.5.

O produto geométrico, introduzido por Clifford na álgebra exterior, é invertível, ao contrário do produto exterior. Dessa forma, sua definição garante que a norma preservada no produto de vetores, isto é, $|ab| = |a||b|$, para $a, b \in \mathbb{R}^3$, enquanto para o produto exterior, temos $|a \wedge b| \leq |a||b|$. Esta característica permite, por exemplo, que operações com o produto geométrico representem rotações ou reflexões em \mathbb{R}^3 , como veremos na Seção 3.2.5.

3.2.3 Produtos Internos: Contrações

O produto geométrico entre dois vetores é a soma do produto escalar com o produto exterior entre eles. Tratemos então do caso em que um dos vetores é substituído por um bivector. Neste caso, o resultado é a soma entre um vetor e um trivector, como mostra (3.12) para $a \in \mathbb{R}^3$ e

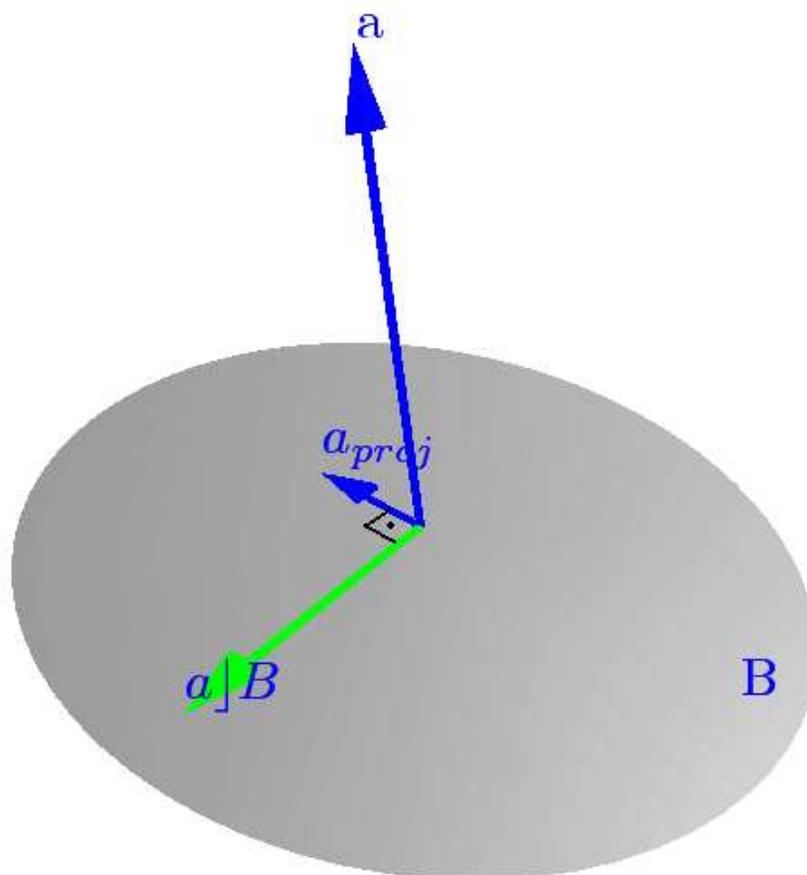


Figura 3.2-5. Contração à esquerda entre o vetor a e o bivetor B .

$$B \in \wedge^2 \mathbb{R}^3:$$

$$\begin{aligned} aB &= (a_1e_1 + a_2e_2 + a_3e_3)(B_1e_{12} + B_2e_{13} + B_3e_{23}) \\ &= -(a_2B_1 + a_3B_2)e_1 + (a_1B_1 - a_3B_3)e_2 + (a_1B_2 + a_2B_3)e_3 \\ &\quad + (a_1B_3 - a_2B_2 + a_3B_1)e_{123}. \end{aligned} \tag{3.12}$$

Em (3.12), o trivetor é dado exatamente pelo produto exterior $a \wedge B$, e o vetor é dado por uma operação chamada de *contração à esquerda* entre a e B , denotada por $a|B$. Neste caso, podemos escrever

$$aB = a|B + a \wedge B.$$

O vetor $a|B$ está sobre o plano definido por B e é perpendicular à a , Figura 3.2-5. Esta afirmação será demonstrada adiante, para o caso mais geral, na Seção 3.2.3.1.

A contração à esquerda é uma operação da forma $\rfloor : \bigwedge^k \mathbb{R}^n \times \bigwedge^l \mathbb{R}^n \longrightarrow \bigwedge^{l-k} \mathbb{R}^n$ e pode ser definida pelas seguintes propriedades, [15, 33]:

$$x \rfloor y = x \cdot y \quad (3.13)$$

$$x \rfloor (U \wedge V) = (x \rfloor U) \wedge V + (-1)^{gr(U)} U \wedge (x \rfloor V) \quad (3.14)$$

$$(U \wedge V) \rfloor W = U \rfloor (V \rfloor W), \quad (3.15)$$

onde $x, y \in \mathbb{R}^3$, $U, V, W \in \bigwedge \mathbb{R}^3$ e $gr(U)$ é o grau de U . A contração à esquerda ainda possui outras propriedades, em relação a escalares:

$$\alpha \rfloor A = \alpha A$$

$$A \rfloor \alpha = 0, \text{ se } gr(A) > 0.$$

A segunda propriedade para escalares pode ser generalizada. A contração $A \rfloor B$ é nula se $gr(A) > gr(B)$ ou se A e B forem ortogonais. De fato, o grau de $A \rfloor B$ é $gr(B) - gr(A)$. Demonstrações de propriedades das contrações podem ser encontradas nos apêndices de [15].

Esta seção foi dedicada à contração à esquerda, pois é esta operação que será usada ao longo do trabalho como produto interno. Em alguns momentos, não havendo riscos de confusão, nos referiremos a esta operação apenas por “ \cdot ”. A contração à direita é relacionada à contração à esquerda pela expressão (3.16).

$$B \llbracket A = (-1)^{gr(A)(gr(B)+1)} A \rfloor B, \quad (3.16)$$

onde $gr(B) \geq gr(A)$.

3.2.3.1 Interpretação Geométrica

Alguns autores se referem à operação $A \rfloor B$ como “a projeção de A em B sendo retirado de B ”, ou seja, $A \rfloor B$ seria o subespaço B sem o que pertence à A . De fato, esta operação resulta em um subespaço de B , ortogonal ao subespaço A , com norma proporcional às normas de A e de B e ao cosseno do ângulo entre A e sua projeção sobre B . Aqui, provamos as duas primeiras afirmações.

Primeiro: $A \rfloor B \subset B$. Para $gr(B) = l$, $gr(A) = k$ e $l \geq k$, utilizando as propriedades da

contração à esquerda, temos que

$$A]B = (A' \wedge a)]B = A'](a]B),$$

onde

$$a]B = a](b_1 \wedge b_2 \wedge \dots \wedge b_k) = \sum_{i=1}^l b_1 \wedge b_2 \wedge \dots \wedge \underbrace{(a]b_i)}_{\in \mathbb{R}} \wedge \dots \wedge b_l,$$

portanto, $a]B \subset B$. Se o processo é repetido para A' , teremos, analogamente, uma contração de um elemento A'' e um elemento de B . Como $l \geq k$, podemos repetir o processo até que reste apenas um escalar $A^{(k)}$ multiplicado por um elemento de B . Logo, $A]B \subset B$.

Segundo: $A]B$ é ortogonal à A . De fato, seja $a \subset A$. Portanto $a \wedge A = 0$. Agora, para $a](A]B)$, temos

$$a](A]B) = (a \wedge A)]B = 0.$$

Logo, a é perpendicular à $A]B$, e como a é um vetor arbitrário de A , temos o resultado.

A demonstração da afirmação acerca da norma é encontrada em [15], Apêndice B.

Por último, falamos da não associatividade das contrações, novamente fazendo referência à contração à esquerda. Os produtos $A](B]C)$ e $(A]B)]C$, em geral, nem mesmo possuem o mesmo grau. O grau do primeiro é $gr(A) - (gr(B) - gr(C))$ enquanto do segundo é $gr(B) - gr(A) - gr(C)$. Para $A](B]C)$, temos um subespaço perpendicular a A contido no subespaço gerado por $B]C$, que por sua vez é perpendicular a B e está contido em C . Logo, $A](B]C)$ é um subespaço contido em C perpendicular a A e a B , confirmando o que diz a propriedade (3.15).

Para a contração $(A]B)]C$, utilizando os mesmos termos, dizemos que é um subespaço de C que é perpendicular ao subespaço $A]B$, que por sua vez está contido em B e é perpendicular a A . Para este caso, é exigido que A esteja contido em C , e assim temos que

$$(A]B)]C = A \wedge (B]C), \text{ para } A \subseteq C. \quad (3.17)$$

O lado direito de 3.17 pode ser interpretado como: retira-se de C a projeção de B em C e em seguida A é recolocado em C . Esta reconstrução de A está relacionada à exigência de que A estivesse em C a priori. A demonstração de 3.17 pode ser encontrada em detalhes em [15].

A próxima Seção trata da *Dualização*, operação que será muito utilizada adiante e é definida a partir dos produtos definidos até aqui.

3.2.4 Dualização

Para definirmos a dualização, outras definições são necessárias, a começar pela operação de *reversão*. A reversão é representada por “ \sim ”. Para um multivetor arbitrário v de Cl_3 , dado por (3.5), temos

$$\begin{aligned}\tilde{v} &= v_0 + v_1e_1 + v_2e_2 + v_3e_3 + v_4e_{21} + v_5e_{31} + v_6e_{32} + v_7e_{321} \\ &= v_0 + v_1e_1 + v_2e_2 + v_3e_3 - v_4e_{12} - v_5e_{13} - v_6e_{23} - v_7e_{123}.\end{aligned}$$

Para um k -vetor (ou k -blade) $A_{(k)} = (a_1 \wedge a_2 \wedge \dots \wedge a_k)$ em $\bigwedge^k \mathbb{R}^n$, temos que

$$\tilde{A}_{(k)} = (a_k \wedge a_{k-1} \wedge \dots \wedge a_1). \quad (3.18)$$

O efeito neste caso, é o de reverter a ordem dos vetores no produto exterior.

A reversão é usada, por exemplo, no cálculo do elemento inverso de subespaços (blades) A_k . O produto geométrico é invertível, porém nem sempre um multivetor possui inverso. Mas no caso de blades de norma não nula, o inverso existe e é dado por

$$A^{-1} = \frac{\tilde{A}}{\|A\|^2},$$

para uma blade A , onde $\|A\|^2 = A\tilde{A}$. De fato, $AA^{-1} = A\frac{\tilde{A}}{A\tilde{A}} = \frac{\|A\|^2}{\|A\|^2} = 1$. Para vetores, o inverso é dado simplesmente por

$$a^{-1} = \frac{a}{\|a\|^2},$$

para $a \in \mathbb{R}^n$, já que $aa^{-1} = a\frac{a}{\|a\|^2} = \frac{\|a\|^2}{\|a\|^2} = 1$.

O *dual* de um elemento A é representado por A^* , e obtido de sua contração à esquerda com o elemento inverso do pseudoescalar (I^{-1}). No caso de k -vetores, a dualização é uma operação de $\bigwedge^k \mathbb{R}^n$ para $\bigwedge^{n-k} \mathbb{R}^n$, dada por

$$A_k^* = A_k \rfloor I^{-1}$$

Em Cl_3 , temos que o dual de um vetor é um plano, determinado por um bivector, Figura 3.2-6. O plano A gerado pelo dual de b , de acordo com a definição de contração à esquerda é o

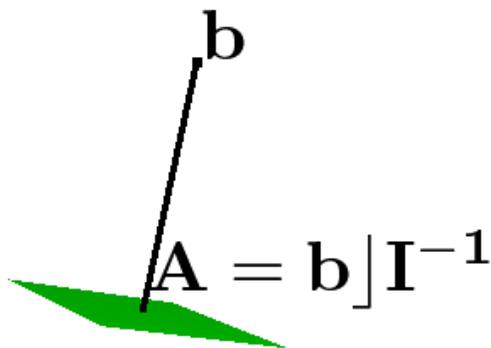


Figura 3.2-6. Dualização de um vetor em \mathbb{R}^3 .

complemento ortogonal de b em \mathbb{R}^3 .

Por último, é importante salientar que a operação $(A^*)^*$ não é necessariamente igual a A . Isto é, o dual do dual não retorna, em geral, ao elemento original. A operação que reverte o dual é a contração à esquerda entre o dual e o pseudoescalar. Temos:

$$A^* = A | I^{-1}$$

$$(A^*)^{-*} = A^* | I = A.$$

3.2.5 Transformações: Reflexão e Rotação em Cl_3

Inicialmente, suponhamos que desejamos refletir um vetor de $a \in \mathbb{R}^2$ em relação a $b \in \mathbb{R}^2$, obtendo um vetor a' .

Da Figura 3.2-7, temos que $a' = a_{||} - a_{\perp}$, onde $a_{||}$ é a componente de a paralela à b e a_{\perp} , a

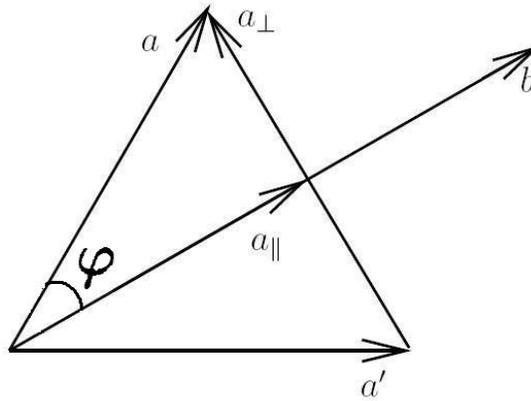


Figura 3.2-7. Reflexão de um vetor em relação a outro no plano.

componente perpendicular. A componente paralela é dada por

$$\begin{aligned} a_{\parallel} &= |a| \cos(\phi) \frac{b}{|b|} \\ &= |a||b| \cos(\phi) \frac{b}{|b|^2} \\ &= (a \cdot b) b^{-1}. \end{aligned}$$

A componente perpendicular é dada por

$$\begin{aligned} a_{\perp} &= a - a_{\parallel} \\ &= a - (a \cdot b) b^{-1} \\ &= (ab - a \cdot b) b^{-1} \\ &= (a \wedge b) b^{-1}. \end{aligned}$$

Logo,

$$\begin{aligned} a' &= (a \cdot b) b^{-1} - (a \wedge b) b^{-1} \\ &= (a \cdot b - a \wedge b) b^{-1} \\ &= bab^{-1}. \end{aligned}$$

Agora, em \mathbb{R}^3 , considere um plano π com vetor normal n , onde $|n| = 1$. A reflexão de um vetor a em relação a este plano é mostrada na Figura 3.2-8.

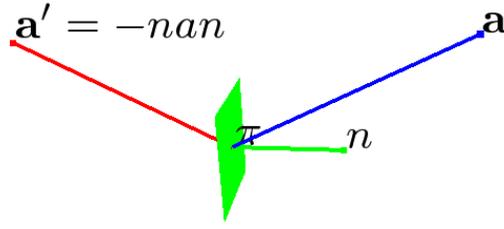


Figura 3.2-8. Reflexão de um vetor sobre um plano.

Seja $a' = a_{\parallel} - a_{\perp}$, onde a_{\parallel} e a_{\perp} são, respectivamente, as componentes de a paralela e perpendicular ao plano π . Em relação ao vetor n , esta relação é invertida, e usando o caso da reflexão em \mathbb{R}^2 , temos

$$\begin{aligned} a_{\perp} &= (a \cdot n)n \\ a_{\parallel} &= (a \wedge n)n. \end{aligned}$$

Logo, temos $a' = -nan$. Se repetirmos este processo em relação a outro plano V com vetor normal unitário v , teremos

$$a'' = -va'v = (vn)a(nv).$$

Chamando $R = vn$, temos $\tilde{R} = nv$, e portanto a'' pode ser escrito como $Ra\tilde{R}$. A composição de duas reflexões gera uma rotação. Neste caso, a'' representa uma rotação de a de duas vezes o ângulo entre n e v em relação ao plano também definido por n e v . O multivetor R é chamado de rotor. Para as reflexões, a própria expressão dos planos poderia ter sido utilizada, neste caso teríamos $a' = \pi a \pi$ e $a'' = (\alpha\pi)a(\pi\alpha)$.

O rotor é o produto geométrico entre dois vetores e, portanto, igual à soma entre um escalar e um bivector. No caso de \mathbb{R}^3 , a forma geral do Rotor é dada exatamente por um *quatérnio* [33], (3.19).

$$R = r_0 + r_1e_{12} + r_2e_{13} + r_3e_{23}. \tag{3.19}$$

O Rotor pode ainda ser representado pelas formas exponencial e trigonométrica.

$$R = e^{-\frac{1}{2}\rho\phi} \tag{3.20}$$

$$R = \cos\left(\frac{\phi}{2}\right) - \rho\text{sen}\left(\frac{\phi}{2}\right), \tag{3.21}$$

$$\tag{3.22}$$

onde ϕ é o ângulo de rotação e ρ é um bivector obtido da dualização do eixo de rotação.

Nas próximas seções apresentamos dois modelos geométricos que nos auxiliarão na representação dos elementos da geometria, que foi uma das motivações ao uso da Álgebra de Clifford.

3.3 Modelo Projetivo

Para definirmos o modelo projetivo, utilizaremos $Cl_{3,1}$. O índice p em $Cl_{p,q}$, é o número de elementos da base do espaço vetorial com quadrado igual a 1, e q o número de elementos da base com quadrado igual a -1. Para o modelo conforme, próxima seção, esta notação é mantida.

A coordenada adicional, em relação ao modelo euclidiano, é representada por e_- . Este vetor, como já foi dito possui a seguinte característica: $e_-^2 = -1$. O uso desta coordenada homogênea induz a métrica de Minkowski [32].

A álgebra $Cl_{3,1}$ tem dimensão $2^4 = 16$ e sua base é dada pela Tabela 3.6.

escalar	1
vetores	e_-, e_1, e_2, e_3
bivetores	$e_{12}, e_{13}, e_{23}, e_{-1}, e_{-2}, e_{-3}$
trivetores	$e_{123}, e_{-12}, e_{-13}, e_{-23}$
pseudoescalar	$I = e_{-123}$

Tabela 3.6. Base de Cl_3

Neste modelo, que representa a imersão de um espaço de três dimensões em um espaço com quatro dimensões¹, elementos geométricos tais como pontos, retas e planos são construídos de maneira intuitiva e representados de maneira simples por blades.

3.3.1 Elementos da Geometria em $Cl_{3,1}$

No espaço projetivo, um ponto é dado pela soma de sua representação em \mathbb{R}^3 e da coordenada homogênea por

$$X = x + e_-.$$

¹a imersão pode ser feita também de um espaço n -dimensional para um espaço $n + 1$ -dimensional

Se fizermos $x = 0$, percebemos que e_- representa a origem deste espaço homogêneo.

Uma reta também pode ser representada neste modelo, através do produto exterior entre dois pontos:

$$\begin{aligned} L &= X_1 \wedge X_2 \\ &= (x_1 + e_-) \wedge (x_2 + e_-) \\ &= x_1 \wedge x_2 + (x_1 - x_2)e_-. \end{aligned}$$

E o produto exterior entre três pontos, define um plano:

$$\begin{aligned} \Pi &= X_1 \wedge X_2 \wedge X_3 \\ &= (x_1 + e_-) \wedge (x_2 + e_-) \wedge (x_3 + e_-) \\ &= x_1 \wedge x_2 \wedge x_3 + (x_1 - x_2) \wedge (x_1 - x_3)e_-. \end{aligned}$$

Mais detalhes sobre o modelo projetivo, da maneira como foi apresentado aqui, podem ser encontrados em [20, 45].

Este modelo é limitado à representação de elementos planos. A representação de uma esfera será possível no modelo *conforme*, apresentado a seguir.

3.4 Modelo Conforme

Para este modelo, utilizaremos a álgebra $Cl_{4,1}$. Esta álgebra é também conhecida por Álgebra Geométrica Conforme (AGC). Este poderoso modelo da geometria euclidiana representa a principal ferramenta para nossa abordagem ao PGDM. A motivação para o uso dessa teoria foi a facilidade para representar elementos geométricos, especialmente círculos e esferas, e as transformações sobre estes. Mais detalhes sobre a AGC podem ser encontrados em [14, 15, 20, 32, 40, 42, 51].

Se para o modelo projetivo faz-se uma imersão de um espaço n -dimensional em um espaço $n+1$ -dimensional, na AGC a imersão é feita em um espaço $n+2$ -dimensional. Isto porque no processo de interpretação de blades enquanto elementos da geometria, duas dimensões são “perdidas”.

Seguindo [20], o processo de interpretação ocorre da seguinte maneira para o caso de $n = 3$: as blades no espaço 5-dimensional intersectam um hiperplano. Esta interseção gera um elemento plano no espaço 4-dimensional, que por sua vez se intersecta com um parabolóide tridimensional, chamado de horosfera. Esta última interseção gera uma cônica que é projetada sobre um hiperplano, ortogonal ao eixo de simetria da horosfera, no espaço tridimensional. Estas projeções representam pontos, retas, planos, círculos e esferas. A Figura 3.4-9 ilustra a situação para um círculo.

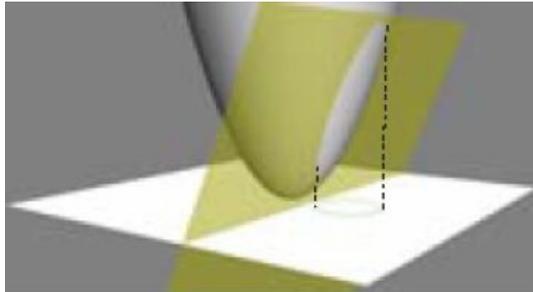


Figura 3.4-9. Interpretação geométrica de uma blade (imagem retirada de [20]).

Antes de seguirmos com as representações de elementos geométricos, devemos definir os elementos adicionais à \mathbb{R}^n . Assim como no espaço projetivo, é usada uma coordenada homogênea e_- , com a propriedade $e_-^2 = -1$, o que novamente nos remete ao espaço de Minkowski. Além deste, um outro vetor, representado por e_+ , completa a base ortonormal $\{e_-, e_+\}$ de $\mathbb{R}^{1,1}$. O espaço $\mathbb{R}^{4,1}$, que é o que nos interessa, é dado por $\mathbb{R}^3 \oplus \mathbb{R}^{1,1}$. Uma base deste espaço é dada por

$$\{e_1, e_2, e_3, e_-, e_+\}.$$

As propriedades dos novos elementos são listadas a seguir:

$$e_+^2 = 1 \tag{3.23}$$

$$e_-^2 = -1$$

$$e_+ \cdot e_- = 0.$$

A partir destes vetores, uma nova base pode ser definida. A base $\{e_0, e_\infty\}$ formada por um vetor que representa a origem e um vetor que representa um ponto no infinito, é chamada de base nula, pois seus elementos têm quadrados iguais a zero. Eles são definidos em função de

$\{e_-, e_+\}$, através de (3.24) e (3.25):

$$e_0 = \frac{1}{2}(e_- - e_+) \text{ (representa a origem do espaço 3D);} \quad (3.24)$$

$$e_\infty = e_- + e_+ \text{ (representa o infinito).} \quad (3.25)$$

Temos portanto,

$$e_0^2 = 0 \text{ e } e_\infty^2 = 0.$$

Outras propriedades desta base são dadas por (3.26), (3.27) e (3.28):

$$\begin{aligned} e_\infty \cdot e_0 &= (e_- + e_+) \cdot \frac{1}{2}(e_- - e_+) & (3.26) \\ &= \frac{1}{2}(e_- \cdot e_- - e_- \cdot e_+ + e_+ \cdot e_- - e_+ \cdot e_+) \\ &= \frac{1}{2}(-1 + 0 + 0 - 1) = -1 \end{aligned}$$

$$e_\infty e_0 = e_\infty \cdot e_0 + E = E - 1 \quad (3.27)$$

$$e_0 e_\infty = e_0 \cdot e_\infty + e_0 \wedge e_\infty = -E - 1, \quad (3.28)$$

para $E = e_\infty \wedge e_0$. Mais detalhes e abordagens diferentes acerca da imersão de \mathbb{R}^n em $\mathbb{R}^{n+1,1}$ podem ser encontrados em [32], [42], [47] e [46].

3.4.1 Base do Espaço Conforme

Para representarmos uma base do espaço conforme 5-dimensional utilizaremos e_∞ e e_0 , obtendo

$$\{e_1, e_2, e_3, e_0, e_\infty\}.$$

A álgebra $Cl_{4,1}$ associada possui $2^5 = 32$ blades de base, listados na tabela 3.7. Agora, a partir desta base, podemos descrever elementos básicos da geometria, começando pelos pontos. A partir da próxima seção, utilizaremos as representações diretas e duais dos elementos de acordo

escalar	1
vetores	$e_1, e_2, e_3, e_0, e_\infty$
bivetores	$e_1 \wedge e_2, e_2 \wedge e_3, e_1 \wedge e_3, e_1 \wedge e_\infty,$ $e_1 \wedge e_0, e_2 \wedge e_\infty, e_2 \wedge e_0, e_3 \wedge e_\infty,$ $e_3 \wedge e_0, e_0 \wedge e_\infty$
trivetores	$e_1 \wedge e_2 \wedge e_3, e_1 \wedge e_2 \wedge e_0, e_1 \wedge e_2 \wedge e_\infty,$ $e_1 \wedge e_3 \wedge e_0, e_1 \wedge e_3 \wedge e_\infty, e_2 \wedge e_3 \wedge e_0,$ $e_2 \wedge e_3 \wedge e_\infty, e_1 \wedge e_0 \wedge e_\infty, e_2 \wedge e_0 \wedge e_\infty,$ $e_3 \wedge e_0 \wedge e_\infty$
quadrivetores	$e_1 \wedge e_2 \wedge e_3 \wedge e_0, e_1 \wedge e_2 \wedge e_3 \wedge e_\infty,$ $e_1 \wedge e_2 \wedge e_0 \wedge e_\infty, e_1 \wedge e_3 \wedge e_0 \wedge e_\infty,$ $e_2 \wedge e_3 \wedge e_0 \wedge e_\infty$
pseudoescalar	$e_1 \wedge e_2 \wedge e_3 \wedge e_\infty \wedge e_0$

Tabela 3.7. Blades (subespaços) da AGC para $n = 3$

com [4, 10, 24, 27, 42, 45–47, 54].

3.4.2 Pontos

O uso da métrica de Minkowski tem como objetivo a geração de vetores nulos. A projeção de pontos de \mathbb{R}^3 sobre $\mathbb{R}^{4,1}$ estão sobre um cone, que é o *cone nulo* do espaço do Minkowski [32, 45]. Os pontos X neste cone possuem a propriedade $X^2 = 0$ e por isso são chamados vetores nulos.

Um ponto deste cone está relacionado ao seu representante em \mathbb{R}^3 através de (3.29):

$$X = x + \frac{1}{2}x^2e_\infty + e_0, \tag{3.29}$$

onde $x \in \mathbb{R}^3$.

Antes de passarmos para o próximo elemento, veremos mais um uso da contração à esquerda, que sem possibilidade de confusão, será representada aqui simplesmente por “ \cdot ”. A contração é usada para calcular, de maneira eficiente, a distância entre 2 pontos. De fato, dados x e y vetores de \mathbb{R}^3 , calculemos a contração entre os elementos X e Y correspondentes:

$$X \cdot Y = (x + \frac{1}{2}x^2e_\infty + e_0) \cdot (y + \frac{1}{2}y^2e_\infty + e_0). \tag{3.30}$$

Como a contração é distributiva e $x \cdot e_\infty = 0$, $y \cdot e_\infty = 0$, $x \cdot e_0 = 0$ e $y \cdot e_0 = 0$, temos,

$$\begin{aligned} X \cdot Y &= x \cdot y + \left(\frac{1}{2}x^2e_\infty\right) \cdot e_0 + e_0 \cdot \left(\frac{1}{2}y^2e_\infty\right) \\ &= -\frac{1}{2}(x^2 + y^2 - 2x \cdot y), \text{ pois } e_\infty \cdot e_0 = e_0 \cdot e_\infty = -1, \\ &= -\frac{1}{2}(x - y)^2. \end{aligned} \tag{3.31}$$

Logo, $(x - y)^2 = -2(X \cdot Y)$. Isto é, o quadrado da distância euclidiana entre dois pontos é a contração à esquerda entre os pontos associados a eles em $Cl_{4,1}$, multiplicado por -2 .

3.4.3 Esferas

A esfera é o elemento básico do espaço conforme. Da representação da esfera, é possível representar os demais elementos. A expressão que representa a esfera no espaço conforme é dada por

$$S = c - \frac{1}{2}(c^2 - \rho^2)e_\infty + e_0,$$

onde $c \in \mathbb{R}^3$ é o centro da esfera e $\rho \in \mathbb{R}$ o raio.

Vamos mostrar que esta expressão realmente representa uma esfera. Uma esfera de centro $c \in \mathbb{R}^3$ e raio ρ , em \mathbb{R}^3 , é descrita pela equação $(x - c)^2 = \rho^2$. De (3.31), sabemos que

$$X \cdot U = -\frac{1}{2}(x - c)^2 = -\frac{1}{2}\rho^2,$$

e portanto

$$X \cdot U - (-1)\frac{\rho^2}{2} = 0. \tag{3.32}$$

Agora, usamos o fato de $X \cdot e_\infty = -1$ e reescrevemos (3.32):

$$X \cdot \left(U - \frac{1}{2}\rho^2e_\infty\right) = 0. \tag{3.33}$$

Então, para $S = U - \frac{1}{2}\rho^2e_\infty$, temos que

$$X \cdot S = 0 \Leftrightarrow x|(x - c)^2 = \rho^2.$$

Neste caso, os elementos X que satisfazem esta condição constituem um supespaço chamado *núcleo do produto interno*.

Uma esfera também pode ser representada pela sua forma dual, S^* . Nesta representação, a esfera é representada pelo produto exterior entre 4 pontos sobre ela:

$$S^* = X_1 \wedge X_2 \wedge X_3 \wedge X_4. \quad (3.34)$$

Um ponto X está sobre a esfera se $X \wedge S^* = 0$, de forma que S^* representa uma esfera e os pontos X que satisfazem $X \wedge S^* = 0$ consituem o chamado *espaço nulo do produto exterior*.

No decorrer do Capítulo 4, decidiremos qual representação utilizar. Esta escolha se dá de acordo com os dados disponíveis. Se for necessária uma troca de representação, utilizamos a operação dual. Ou seja, com os dados disponíveis para a representação direta (raio e centro), é possível calcular a representação dual. E o inverso também é possível, com o uso da reversão do dual. Isto é válido para qualquer elemento.

3.4.4 Círculo

Um círculo pode ser definido como a interseção de duas esferas. A operação que representa esta interseção é exatamente o produto exterior. De forma que se S_1 e S_2 se intersectam, teremos que $C = S_1 \wedge S_2$ define um círculo. A Figura 4.2-1 do início do próximo capítulo ilustra esta situação.

Caso não se tenha as esferas, o círculo também possui uma representação dual: o produto exterior de 3 pontos sobre ele, dado por:

$$C^* = X_1 \wedge X_2 \wedge X_3. \quad (3.35)$$

3.4.5 Planos

Um plano pode ser representado com a substituição de um dos pontos em (3.34) por e_∞ , de modo que (3.36) define um plano que passa por X_1 , X_2 e X_3 :

$$\pi^* = X_1 \wedge X_2 \wedge X_3 \wedge e_\infty. \quad (3.36)$$

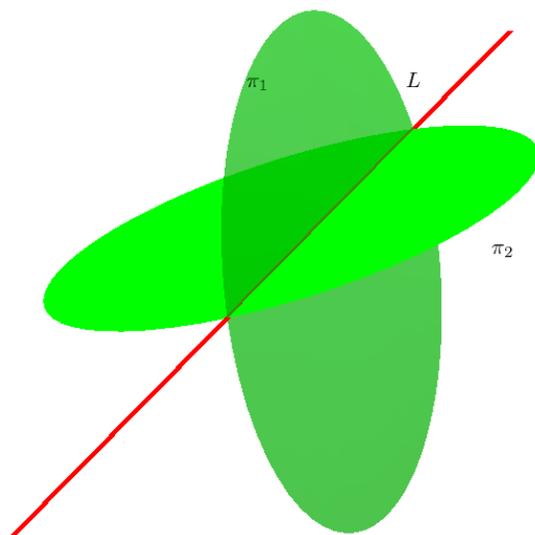


Figura 3.4-10. Interseção de dois planos

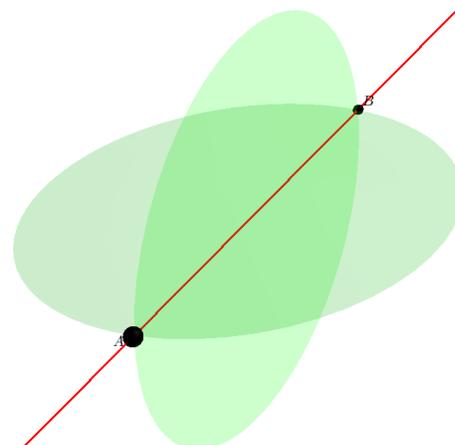


Figura 3.4-11. Reta determinada por A e B.

Esta é a representação dual de um plano e geometricamente pode ser interpretada, como uma esfera de raio infinito.

A representação direta, Eq. (3.37), é dada pela distância do plano à origem $d \in \mathbb{R}$ e pelo seu vetor normal n , como já mencionamos na Seção 3.3:

$$\pi = n + de_{\infty}. \tag{3.37}$$

3.4.6 Retas

As retas também possuem duas maneiras de serem representadas, e uma delas é a partir da interseção entre 2 planos. O produto exterior entre dois planos π_1 e π_2 que se intersectam, define uma reta L dada por:

$$L = \pi_1 \wedge \pi_2.$$

Assim como esferas, círculos e planos, a representação dual de uma reta é dada pelo produto exterior entre 2 pontos sobre ela e um ponto que representa o infinito. Geometricamente, uma reta pode ser interpretada com um círculo de raio infinito:

$$L^* = X_1 \wedge X_2 \wedge e_{\infty}.$$

3.4.7 Par de Pontos

Por último, o par de pontos. Como o nome diz, representa dois pontos, mas é um único elemento de $Cl_{4,1}$. O par de pontos (Pp) é obtido através da interseção de 3 esferas, representação direta, ou pelo produto exterior entre os dois pontos que o formam, representação dual $(Pp)^*$. Para cada caso, temos:

$$Pp = S_1 \wedge S_2 \wedge S_3; \quad (3.38)$$

e

$$(Pp)^* = X \wedge Y. \quad (3.39)$$

A Equação (3.38) é desenvolvida na seção 4.2. O Par de Pontos é um dos elementos mais utilizados em nosso trabalho. De uma expressão de um Par de Pontos em sua representação direta, não é intuitivo enxergar quais são os pontos que o determinam, para isso, é necessário extrair estes pontos. Tal extração é obtida através da fórmula (3.40), de [15]:

$$P_{\pm} = \frac{\pm \sqrt{(Pp)^* \lrcorner (Pp)^* + (Pp)^*}}{e_{\infty} \lrcorner (Pp)^*}. \quad (3.40)$$

A demonstração desta fórmula, não encontrada na literatura, é apresentada abaixo:

Prova. Durante toda esta demonstração usaremos “ \cdot ” para representar as contrações. Temos que $(Pp)^* = X \wedge Y$, então calcularemos $(X \wedge Y)^2$, já que $(Pp)^* \cdot (Pp)^* = (X \wedge Y) \cdot (X \wedge Y) = (X \wedge Y)^2$. Mas antes, mostraremos que $X^2 = 0$, como já citamos, mas sem prova. Um vetor no cone nulo do espaço conforme tem quadrado igual a zero, e por isso é chamado de vetor nulo, [32], [46]. Todos os cálculos a seguir se baseiam no fato de que para vetores do espaço conforme $\mathbb{R}^{n+1,1}$, o produto geométrico age da mesma forma que para vetores do \mathbb{R}^n , isto é, para $X, Y \in \mathbb{R}^{n+1,1}$, obtemos

$$XY = X \cdot Y + X \wedge Y.$$

Inicialmente, temos:

$$\begin{aligned}
 X^2 &= (x + \frac{1}{2}x^2e_\infty + e_0)(x + \frac{1}{2}x^2e_\infty + e_0) \\
 &= x^2 + (\frac{1}{2}x^2)xe_\infty + xe_0 + (\frac{1}{2}x^2)e_\infty x + (\frac{1}{2}x^2)^2e_\infty^2 + (\frac{1}{2}x^2)e_\infty e_0 + e_0x + (\frac{1}{2}x^2)e_0e_\infty + e_0^2 \\
 &= x^2 + (\frac{1}{2}x^2)e_\infty e_0 + (\frac{1}{2}x^2)e_0e_\infty \\
 &= x^2 + x^2\frac{-1+E}{2} + x^2\frac{-1-E}{2} \\
 &= \frac{x^2}{2}(2 - 1 + E - 1 - E) \\
 &= 0.
 \end{aligned} \tag{3.41}$$

Agora, calculamos $(X \wedge Y)^2$:

$$\begin{aligned}
 (X \wedge Y)^2 &= (\frac{1}{2}(XY - YX))^2 \\
 &= \frac{1}{4}((XYXY) + (XY^2X) + (YX^2Y)(YXYX)) \\
 &= \frac{1}{4}(X(YXY) + (YXY)X) \\
 &= \frac{1}{4}(X \cdot (YXY) + X \wedge (YXY) + (YXY) \cdot X + (YXY) \wedge X) \\
 &= \frac{1}{4}(2X \cdot (YXY)) \\
 &= \frac{1}{2}(X \cdot (YXY)).
 \end{aligned} \tag{3.42}$$

Antes de seguirmos, vamos calcular $X \cdot (YXY)$, separadamente:

$$\begin{aligned}
 YXY &= Y(XY) \\
 &= Y(X \cdot Y + X \wedge Y) \\
 &= Y(X \cdot Y) + Y(X \wedge Y) \\
 &= Y(X \cdot Y) + Y \cdot (X \wedge Y) + \underbrace{Y \wedge (X \wedge Y)}_{=0} \\
 &= Y(X \cdot Y) + (Y \cdot X)Y + \underbrace{(Y \cdot Y)}_0 X \\
 &= 2(X \cdot Y)Y,
 \end{aligned} \tag{3.43}$$

e portanto,

$$\begin{aligned}
 (X \wedge Y)^2 &= \frac{1}{2}(X \cdot (YXY)) \\
 &= \frac{1}{2}(2(X \cdot Y)^2) \\
 &= (X \cdot Y)^2.
 \end{aligned} \tag{3.44}$$

Substituindo $(Pp)^* = X \wedge Y$ em $\frac{\pm \sqrt{(Pp)^* \cdot (Pp)^*} + (Pp)^*}{e_\infty \cdot (Pp)^*}$, temos, para o numerador, que:

$$\begin{aligned}
 \pm \sqrt{(Pp)^* \cdot (Pp)^*} + (Pp)^* &= (X \wedge Y) \pm \sqrt{(X \wedge Y)^2} \\
 &= X \wedge Y \pm X \cdot Y,
 \end{aligned} \tag{3.45}$$

e para o denominador:

$$\begin{aligned}
 -e_\infty \cdot (Pp)^* &= -e_\infty(X \wedge Y) \\
 &= -(\underbrace{(e_\infty \cdot X)}_{-1}Y - \underbrace{(e_\infty \cdot Y)}_{-1}X) \\
 &= Y - X.
 \end{aligned} \tag{3.46}$$

Precisamos do inverso de $Y - X$. Para isso, calculamos $(Y - X)^2$:

$$\begin{aligned}
 (Y - X)^2 &= Y^2 + X^2 - XY - YX \\
 &= -X \wedge Y - X \cdot Y - Y \wedge X - Y \cdot X \\
 &= -2(X \cdot Y).
 \end{aligned} \tag{3.47}$$

Logo,

$$\frac{1}{Y - X} = \frac{Y - X}{-2(X \cdot Y)}. \tag{3.48}$$

Finalmente, temos que:

$$\frac{\pm \sqrt{(Pp)^* \cdot (Pp)^*} + (Pp)^*}{-e_\infty \cdot (Pp)^*} = \frac{(X \wedge Y \pm X \cdot Y)(Y - X)}{-2(X \cdot Y)} \tag{3.49}$$

e,

$$\begin{aligned}
 \frac{(X \wedge Y - X \cdot Y)(Y - X)}{-2(X \cdot Y)} &= \frac{-YX(Y - X)}{-2(X \cdot Y)} \\
 &= \frac{-YXY + YX^2}{-2(X \cdot Y)} \\
 &= \frac{YXY}{2(X \cdot Y)} \\
 &= \frac{Y(X \cdot Y + X \wedge Y)}{2(X \cdot Y)} \\
 &= \frac{Y(X \cdot Y)}{2(X \cdot Y)} + \frac{Y \wedge X \wedge Y + Y \cdot (X \wedge Y)}{2(X \cdot Y)} \\
 &= \frac{Y}{2} + \frac{(Y \cdot X)Y - (Y \cdot Y)X}{2(X \cdot Y)} \\
 &= 2\frac{Y}{2} \\
 &= Y.
 \end{aligned} \tag{3.50}$$

De forma análoga, para o sinal positivo, obtemos X . Esta fórmula nos fornece 2 pontos em $Cl_{4,1}$, dos quais sabemos as coordenadas correspondentes em \mathbb{R}^3 . Este processo é parte fundamental para nossa abordagem do PDGDM. Fechamos o capítulo apresentando a Tabela 3.8, com as representações de cada elemento.

Elemento	Repr. Direta	Dual
Ponto	$X = x + \frac{1}{2}x^2e_\infty + e_0$	
Esfera	$S = X - \frac{1}{2}\rho^2e_\infty$	$S^* = X_1 \wedge X_2 \wedge X_3 \wedge X_4$
Círculo	$C = S_1 \wedge S_2$	$C^* = X_1 \wedge X_2 \wedge X_3$
Plano	$\pi = n + \mathbf{d}e_\infty$	$\pi^* = X_1 \wedge X_2 \wedge X_3 \wedge e_\infty$
Reta	$L = \pi_1 \wedge \pi_2$	$L^* = X_1 \wedge X_2 \wedge e_\infty$
Par de Pontos	$Pp = S_1 \wedge S_2 \wedge S_3$	$(Pp)^* = X_1 \wedge X_2$

Tabela 3.8. Representações de cada elemento geométrico. Por convenção, letras minúsculas representam elementos de \mathbb{R}^3 , com excessão de \mathbf{d} , que é um escalar.

A observação da Tabela 3.8 deixa claro o papel do produto exterior em cada coluna. Na representação direta, é uma operação de interseção, na representação dual, é uma operação de construção. No Capítulo 4, apresentamos uma operação de interseção que age sobre os elementos dualizados.

No próximo capítulo, apresentamos como a Álgebra de Clifford pode ser utilizada no cál-

culo de estruturas moleculares.

Capítulo 4

AC Aplicada ao PDGDM

Neste capítulo, integramos os conceitos da Álgebra de Clifford à resolução do PDGDM e iPDGDM. A motivação para esta integração partiu inicialmente da interpretação geométrica do problema discretizado, como interseção de esferas. A Álgebra Geométrica Conforme lida de forma simples e intuitiva com estas interseções, o que tornou também o trabalho com distâncias intervalares promissor. Aplicações do Modelo Conforme, especialmente em computação gráfica e robótica, podem ser encontradas em [10, 11, 17, 24, 26, 27, 46, 47, 51].

Inicialmente, consideramos o problema exato, em que são conhecidas as posições de três antecessores de um átomo i e suas respectivas distâncias a i . Esta situação corresponde ao PDGDM. Em seguida, estudamos a situação encontrada no iPDGDM, em que apenas as distâncias de dois antecessores são conhecidas, além de um intervalo para a uma terceira distância a um antecessor. Para os dois casos, consideramos também a existência de distâncias adicionais, aquelas que determinam a factibilidade de um ponto. Para o iPDGDM, estas distâncias também são dadas por intervalos.

4.1 Ferramentas Computacionais

Antes da implementação da nossa abordagem baseada na Álgebra de Clifford ao algoritmo iBP, codificado em linguagem C , o trabalho foi desenvolvido com o uso de um software chamado CLUCalc [41]. A ligação entre o trabalho desenvolvido com o CLUCalc e o iBP, em outras

palavras, a tradução de um CLUscript para a linguagem C , foi realizada com o uso de outra poderosa ferramenta, o *Gaalop* [25].

O CLUCalc é um software autônomo desenvolvido por Christian Perwass para o trabalho com a AC, especialmente com o modelo conforme. O software se utiliza de três janelas: edição (entrada), saída e visualização. A linguagem utilizada para a criação dos scripts foi desenvolvida com o objetivo de se assemelhar à maneira como escrevemos no papel. Por exemplo, o produto exterior $A \wedge B$ é escrito simplesmente como $A^{\wedge}B$, dentre outros exemplos. A janela de visualização é totalmente interativa, permitindo ao usuário rotacionar objetos, ou ainda modificar em tempo real a saída de um script com a mudança de qualquer parâmetro, através de “sliders”. Dentre outras características, é possível ainda criar animações. Muitos exemplos de CLUscripts estão disponíveis para download em <http://www.gaalop.de/dhilden/>, página do professor Dietmar Hildenbrand, da Universidade de Darmstadt, Alemanha. Outras ferramentas são usadas para o trabalho com a AC, como o pacote GABLE [16, 35] para o MatLab, que não permite o trabalho com elementos dos espaços projetivo e conforme, ou o pacote CLIFFORD [1, 2], para o Maple, desenvolvido por Rafal Ablamowicz e Bertfried Fauser, atualmente na versão 13.3 de 2012, utilizado na criação de algoritmos baseados em AG. Há ainda o software autônomo CLICAL [34], segundo C. Perwass, a primeira implementação computacional de AG, realizada por um grupo liderado por Pertti Lounesto, em Helsinki. Há ainda os geradores de códigos em linguagem $C/C++$, como o Gaigen [19] e o Gaalop [25]. Este último, desenvolvido na Universidade de Darmstadt, traduz CLUscripts para a linguagem $C/C++$ e foi o escolhido para o desenvolvimento deste trabalho. Descrições de outras ferramentas computacionais podem ser encontradas em [18, 40].

4.2 Interseção de Esferas

Duas esferas em \mathbb{R}^3 têm um círculo como interseção (Figura 4.2-1), e uma terceira esfera intersecta este círculo em 2 pontos (Figura 4.2-2), respectivamente, casos tratados nas Seções 3.4.4 e 3.4.7.

Vamos tratar do caso mais geral, do *par de pontos*, dado pela interseção de três esferas. As informações disponíveis sobre as esferas serão centro e raio. Sendo assim, utilizaremos a

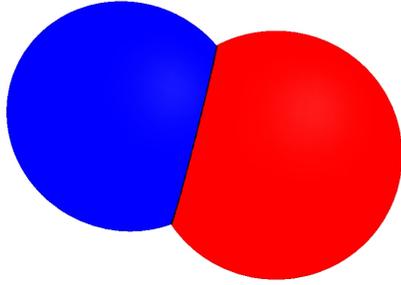


Figura 4.2-1. Interseção de duas esferas

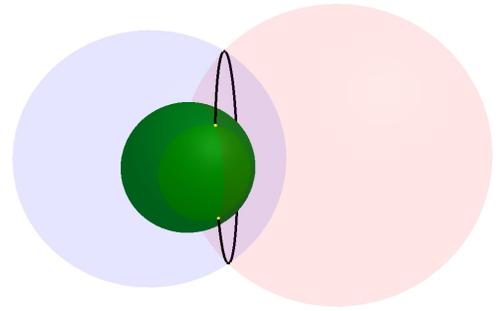


Figura 4.2-2. Interseção de três esferas

representação direta, segundo a Tabela 3.8, dada por (4.1).

$$S_i = X_i - \frac{1}{2}r_i^2 e_\infty, \quad (4.1)$$

onde $X_i \in Cl_{4,1}$ representa o centro e r_i representa o raio da esfera S_i . A equação para o par de pontos é obtida de (3.38):

$$\begin{aligned} Pp &= S_1 \wedge S_2 \wedge S_3 = \left(X - \frac{1}{2}r_1^2 e_\infty \right) \wedge \left(Y - \frac{1}{2}r_2^2 e_\infty \right) \wedge \left(Z - \frac{1}{2}r_3^2 e_\infty \right) \\ &= \left(x + \frac{1}{2}(x^2 - r_1^2)e_\infty + e_0 \right) \wedge \left(y + \frac{1}{2}(y^2 - r_2^2)e_\infty + e_0 \right) \wedge \left(z + \frac{1}{2}(z^2 - r_3^2)e_\infty + e_0 \right) \\ &= a_1 e_{23} \wedge e_\infty + a_2 e_{31} \wedge e_\infty + a_3 e_{12} \wedge e_\infty + b_1 e_{23} \wedge e_0 + b_2 e_{31} \wedge e_0 + b_3 e_{12} \wedge e_0 \\ &\quad + c_1 e_1 \wedge E + c_2 e_2 \wedge E + c_3 e_3 \wedge E + d e_{123}. \end{aligned} \quad (4.2)$$

Para $x = (x_1, x_2, x_3)$, $y = (y_1, y_2, y_3)$ e $z = (z_1, z_2, z_3)$, temos que os coeficientes de (4.2) são dados por:

$$\begin{aligned}
a_1 &= 0.5((x_2y_3 - x_3y_2)(z^2 - r_3^2) + (x_3z_2 - x_2z_3)(y^2 - r_2^2) + (y_2z_3 - y_3z_2)(x^2 - r_1^2)); \\
a_2 &= 0.5((x_3y_1 - x_1y_3)(z^2 - r_3^2) + (x_1z_3 - x_3z_1)(y^2 - r_2^2) + (y_3z_1 - y_1z_3)(x^2 - r_1^2)); \\
a_3 &= 0.5((x_1y_2 - x_2y_1)(z^2 - r_3^2) + (x_2z_1 - x_1z_2)(y^2 - r_2^2) + (y_1z_2 - y_2z_1)(x^2 - r_1^2)); \\
b_1 &= (x_2y_3) - (x_2z_3) - (x_3y_2) + (x_3z_2) + (y_2z_3) - (y_3z_2); \\
b_2 &= (x_3y_1) - (x_3z_1) - (x_1y_3) + (x_1z_3) + (y_3z_1) - (y_1z_3); \\
b_3 &= (x_1y_2) - (x_1z_2) - (x_2y_1) + (x_2z_1) + (y_1z_2) - (y_2z_1); \\
c_1 &= 0.5((x_1 - z_1)(y^2 - r_2^2) - (x_1 - y_1)(z^2 - r_3^2) - (y_1 - z_1)(x^2 - r_1^2)); \\
c_2 &= 0.5((x_2 - z_2)(y^2 - r_2^2) - (x_2 - y_2)(z^2 - r_3^2) - (y_2 - z_2)(x^2 - r_1^2)); \\
c_3 &= 0.5((x_3 - z_3)(y^2 - r_2^2) - (x_3 - y_3)(z^2 - r_3^2) - (y_3 - z_3)(x^2 - r_1^2)); \\
d &= (x_1y_2z_3) - (x_1y_3z_2) - (x_2y_1z_3) + (x_2y_3z_1) + (x_3y_1z_2) - (x_3y_2z_1).
\end{aligned}$$

A seguir, consideramos a ação da interseção de esferas, inicialmente sobre o PDGDM, e depois sobre o iPDGDM.

4.3 PDGDM: Distâncias Exatas

4.3.1 Três Distâncias Exatas

Neste caso, buscamos a posição de um átomo v_i a partir das posições de três de seus antecessores, além de suas respectivas distâncias a v_i . As posições dos átomos v_{i-1} , v_{i-2} e v_{i-3} serão centros de esferas e as distâncias d_{v_{i-3},v_i} , d_{v_{i-2},v_i} e d_{v_{i-1},v_i} , serão os raios destas esferas. Chamaremos tais esferas de S_1 , S_2 e S_3 . As possíveis posições para v_i são os dois pontos na interseção de S_1 , S_2 e S_3 (Figura 4.3-3).

Abaixo, apresentamos uma sequência de passos implementada no CLUCalc, para o problema com 3 distâncias exatas:

1. Configuramos o programa para o trabalho no espaço conforme e, de acordo com os dados disponíveis, definimos a representação *direta*.

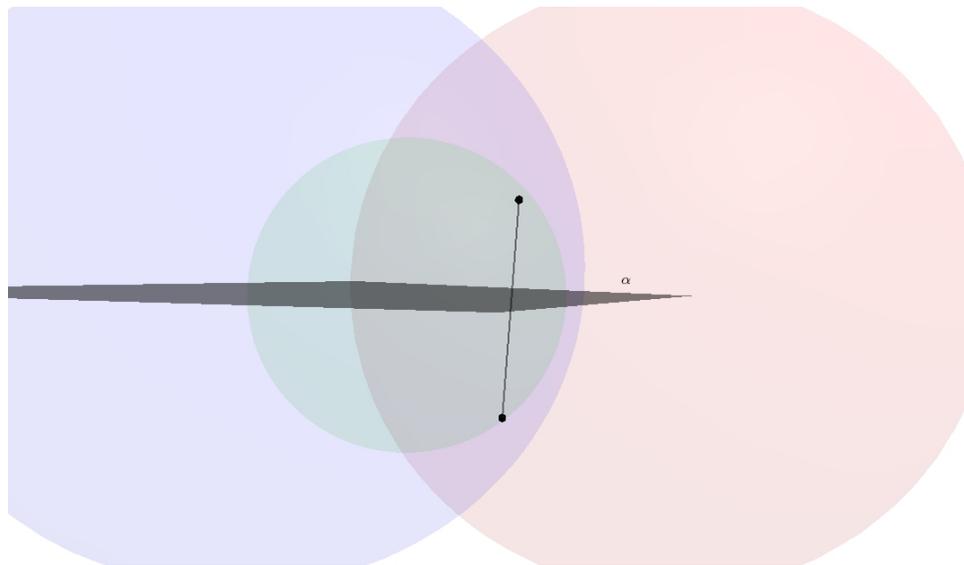


Figura 4.3-3. Pontos simétricos em relação a α , plano definido pelos centros das esferas.

2. Para este caso, usaremos 3 esferas S_1 , S_2 e S_3 , definidas por $S_i = X_i - \frac{1}{2}r_i^2 e_\infty$, $i = 1, 2, 3$.
3. Precisamos determinar os pontos de interseção entre as esferas. A Equação (4.2) fornece uma expressão para o par de pontos P_p que representa esta interseção.
4. Os pontos P_1 e P_2 são extraídos de P_p através de (3.40).

O Exemplo 1 foi executado pelo CLUCalc.

Exemplo 1. Dados:

- Pontos conhecidos: $v_{i-3} = (0, 0, 0)$, $v_{i-2} = (-4, 0, 0)$ e $v_{i-1} = (-3, -1.1133, 1.3268)$.
- Distâncias conhecidas: $d_{v_{i-3}, v_i} = 3.5615$, $d_{v_{i-2}, v_i} = 3.1102$ e $d_{v_{i-1}, v_i} = 1.3$.

A saída do programa nos fornece P_1 e P_2 (Tabela 4.1) e a visualização gerada pelo script é vista na Figura 4.3-4.

Exemplo 1 - Output:
$P_1 = -2.376e_1 - 0.9907e_2 + 2.461e_3 + 6.342e_\infty + e_0$
$P_2 = -2.376e_1 - 2.251e_2 + 1.403e_3 + 6.342e_\infty + e_0$

Tabela 4.1. Solução do Exemplo 1.

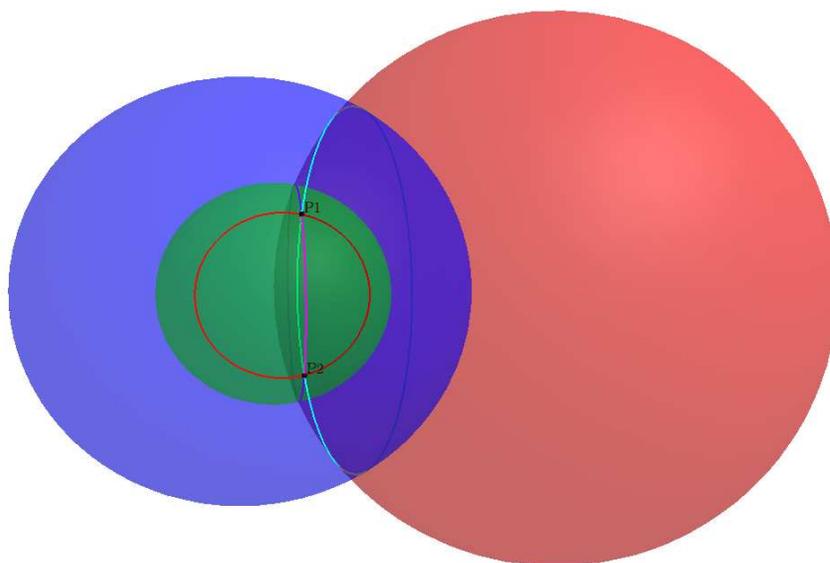


Figura 4.3-4. Solução do Exemplo 1. Os Pontos P_1 e P_2 são a interseção entre as 3 esferas.

As posições dos pontos P_1 e P_2 em \mathbb{R}^3 são automaticamente obtidas pelas suas coordenadas e_1 , e_2 e e_3 :

$$p_1 = (-2.376, -0.9907, 2.461)$$

$$p_2 = (-2.376, -2.251, 1.403).$$

4.3.2 Distância Exata Adicional

A factibilidade de um ponto pode ser verificada através de distâncias adicionais, aquelas, como já foi dito, entre átomos distantes mais de três posições na sequência. Se para um átomo v_i houver uma distância adicional, teremos quatro distâncias exatas, o que resulta em uma única possibilidade para seu posicionamento.

De fato, agora teremos quatro esferas e a resolução pode ser vista de duas formas. Simplesmente como a interseção entre quatro esferas $S_1 \wedge S_2 \wedge S_3 \wedge S_4$, ou em duas etapas. Em cada etapa, resolvemos um subproblema com três esferas e cada subproblema irá gerar um par de pontos. Para que as restrições sejam satisfeitas por um átomo, os pares precisam ter um ponto comum. A Figura 4.3-5 foi obtida adicionando uma esfera S_4 , de centro em $(-1.7183, -1.3373, 3.1292)$ e raio 0.9999, ao exemplo 1. Já a Tabela 4.2 mostra a saída gerada pelo software para este exemplo. Os pontos gerados pela adição de uma nova esfera, P_3 e P_4 , fazem com que o ponto P_2 ,

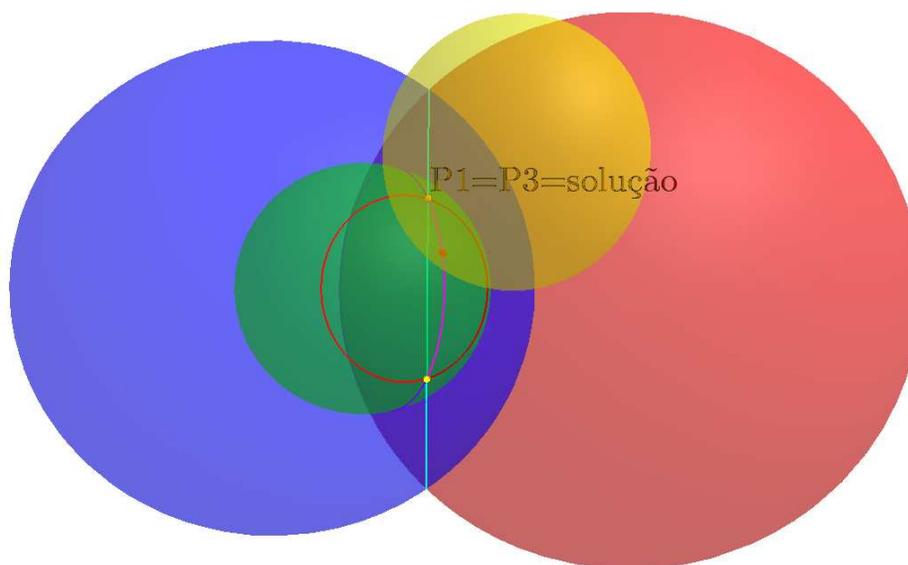


Figura 4.3-5. Exemplo com quatro distâncias exatas. São encontrados 2 pares de pontos, com um ponto em comum que é a solução do problema.

4 exatas - Output:
$P_1 = -2.376e_1 - 0.9907e_2 + 2.461e_3 + 6.342e_\infty + e_0$
$P_2 = -2.376e_1 - 2.251e_2 + 1.403e_3 + 6.342e_\infty + e_0$
$P_3 = -2.376e_1 - 0.9908e_2 + 2.461e_3 + 6.342e_\infty + e_0$
$P_4 = -2.247e_1 - 1.528e_2 + 2.302e_3 + 6.342e_\infty + e_0$

Tabela 4.2. Solução para 4 distâncias exatas.

encontrado na primeira resolução, seja descartado. Portanto, temos que P_1 (ou P_3) é a solução do problema.

O exemplo em duas etapas ilustra o caso em que a árvore é construída e a poda ocorre depois. O uso da interseção entre as quatro esferas, durante a construção da árvore, faz com que apenas pontos factíveis sejam calculados.

4.4 iPDGDM: Distâncias Intervalares

4.4.1 Duas Distâncias Exatas, Uma Intervalar

Neste caso, conhecemos apenas as distâncias exatas de dois átomos anteriores e uma distância intervalar para um terceiro antecessor. Sem perda de generalidade, assumimos que procuramos

v_4 e conhecemos d_{v_3, v_4} e d_{v_2, v_4} . Sabemos ainda que $d_{v_1, v_4} \in l$, onde l é um intervalo fechado em \mathbb{R} .

Para tratar este problema, precisamos de quatro esferas. Uma para cada distância conhecida, esferas Azul e Verde, e outras duas de mesmo centro, para a distância intervalar, as quais têm seus raios definidos pelos extremos do intervalo l , esferas Vermelhas (Figura 4.4-6).

Esta situação é equivalente à resolução de dois subproblemas análogos ao Exemplo 1. Um para cada extremo do intervalo l . Sejam S_2 e S_3 as esferas correspondentes a v_2 e v_3 , e S_1 e S_{1_b} as esferas correspondentes a v_1 . O primeiro subproblema envolve S_1 , S_2 e S_3 , e o segundo, S_{1_b} , S_2 e S_3 . A solução de cada um é um par de pontos. Portanto, temos quatro pontos sobre C , interseção entre S_2 e S_3 .

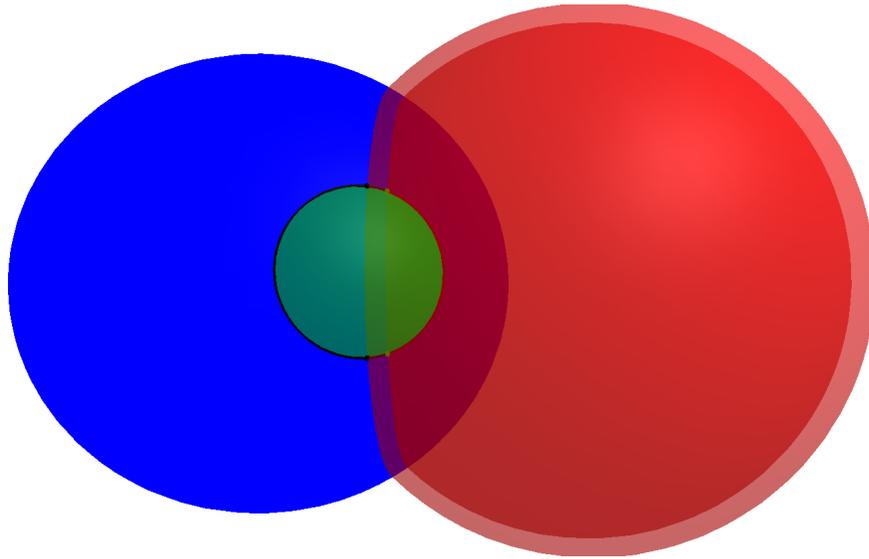


Figura 4.4-6. Cada uma das esferas vermelhas forma um par de pontos na interseção com S_2 e S_3 .

Estes pontos determinam arcos em C (Figura 4.4-7) e a solução está em um destes arcos.

4.4.2 Formação e Descrição dos Arcos

A circunferência C , dada por $C = S_2 \cap S_3$, é interceptada por S_1 em dois pontos, digamos P_1 e P_2 . Tais pontos, simétricos em relação ao plano definido por v_1 , v_2 e v_3 , serão extremidades de arcos distintos. Já S_{1_b} intercepta C em outros dois pontos, P_3 e P_4 , que possuem a mesma simetria e serão as outras extremidades dos arcos. Os arcos formados são, portanto, P_1P_3 , $Arco 1$ e P_2P_4 , $Arco 2$.

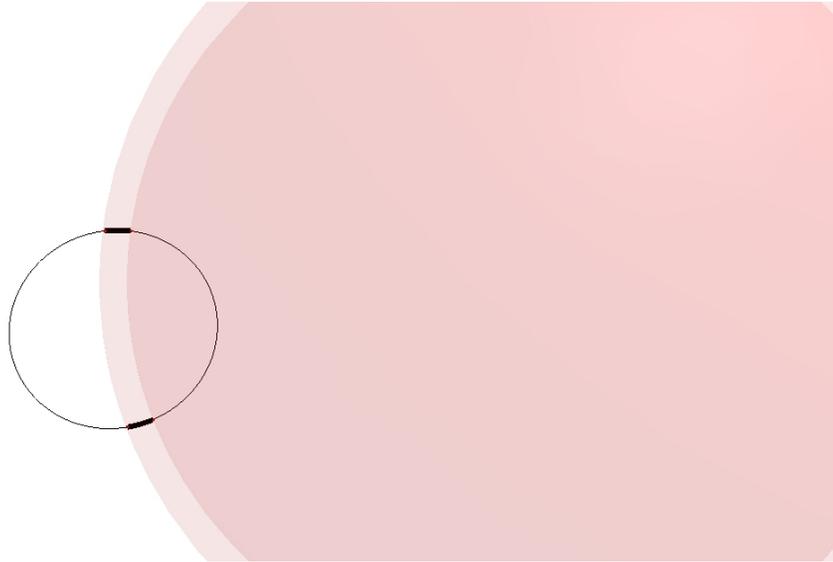


Figura 4.4-7. As 2 esferas que interceptam a circunferência são as esferas vermelhas da Figura 4.4-6, que representam a distância intervalar.

Se v_1 é o centro das esferas na Figura 4.4-7, é possível observar que qualquer esfera centrada em v_1 , com raio em l , interceptará C em pontos destes arcos. Visualmente, identificamos com facilidade as regiões que podem conter a solução do problema, mas queremos uma descrição algébrica de tais regiões. Do nosso conhecimento, não há uma descrição direta de um arco de circunferência através da Álgebra Geométrica Conforme. Desta forma, criamos uma maneira de descrevê-lo utilizando *rotores*. O método consiste em criar um rotor R que aja sobre arco. Ou seja, sobre o plano da circunferência com eixo de rotação passando pelo seu centro. Sua ação dependerá do ângulo relacionado ao arco correspondente. Tal valor representa o ângulo máximo de rotação.

Com R criado, definimos uma das extremidades de cada arco como o ponto de partida da rotação, e o ângulo de rotação como um valor entre 0 e o ângulo máximo. Vamos descrever o processo de obtenção dos ângulos de rotação e do eixo de rotação por meio da AGC.

Para este caso, o ângulo correspondente a cada arco é o mesmo (Figura 4.4-8). O que fazemos, na prática, é criar duas retas passando pelo centro de C e por cada extremidade do arco. Para isso precisamos determinar o centro de C . Isto pode ser feito através da interseção de uma reta que passa por este ponto (como o eixo de rotação) com o plano que contém C . O que pode ser feito através das Equações (4.3) e (4.4).

$$L = (C \wedge e_\infty) \rfloor I \quad (4.3)$$

e

$$\beta = (C^* \wedge e_\infty) \rfloor I. \tag{4.4}$$

Em (4.3) e (4.4), as expressões nos parênteses estão na forma dual, respectivamente de uma reta que passa pelo centro de C e de um plano que contém C . Por isso, calculamos a contração à esquerda com I e, assim, podemos utilizar o produto exterior para a interseção:

$$U = \beta \wedge L. \tag{4.5}$$

A seguir, calculamos o ângulo θ entre as retas determinadas por U e P_1 , que denotaremos por $\overleftrightarrow{UP_1}$, e por U e P_3 , denotada por $\overleftrightarrow{UP_3}$. Tal ângulo é dado por (4.6).

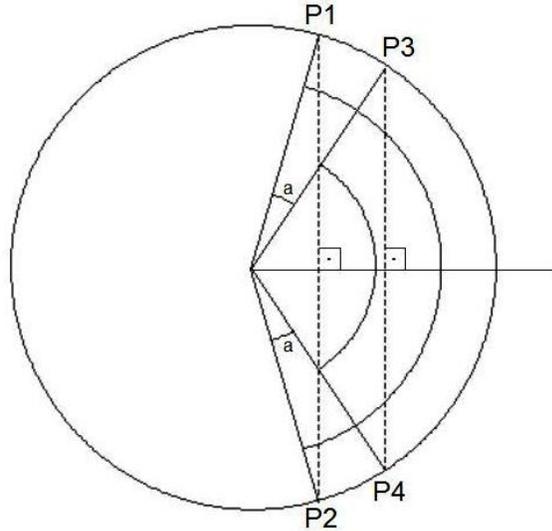


Figura 4.4-8. Os ângulos correspondentes aos arcos P_1P_3 e P_2P_4 são congruentes.

$$\theta = \arccos \left(\frac{\overleftrightarrow{UP_1} \cdot \overleftrightarrow{UP_3}}{|\overleftrightarrow{UP_1}| |\overleftrightarrow{UP_3}|} \right). \tag{4.6}$$

O eixo de rotação é obtido de L , a partir da Equação (4.3). E de fato, a expressão $C \wedge e_\infty$ representa uma reta. Vejamos a demonstração para este fato.

Prova. De acordo com a Tabela 3.8, uma reta pode ser determinada a partir de dois pontos. Sabemos que a reta que determina o eixo de rotação é ortogonal a circunferência C passando

pelo seu centro, e portanto, passa pelo centro das esferas que geraram a circunferência, que são pontos conhecidos, de modo que podemos definir tal reta como $L^* = x_{S_2} \wedge x_{S_3} \wedge e_\infty$, onde x_{S_2} e x_{S_3} são os centros de S_2 e S_3 , respectivamente. A equação abaixo mostra que $L^* = S_2 \wedge S_3 \wedge e_\infty$:

$$\begin{aligned} S_2 \wedge S_3 \wedge e_\infty &= (x_{S_2} - \frac{1}{2}r_2^2 e_\infty) \wedge (x_{S_3} - \frac{1}{2}r_3^2 e_\infty) \wedge e_\infty \\ &= x_{S_2} \wedge x_{S_3} \wedge e_\infty = L^*, \end{aligned} \quad (4.7)$$

pois $e_\infty \wedge e_\infty = 0$. Logo, $x_{S_2} \wedge x_{S_3} \wedge e_\infty = \underbrace{S_2 \wedge S_3}_{=C} \wedge e_\infty = C \wedge e_\infty$. ■

Em seguida, aplicamos o rotor R , definido por (3.21), em P_1 e em P_2 (veja (4.8)). Esta operação faz com que P_1 “ande” sobre o arco na direção de P_3 . O mesmo ocorrendo com P_4 em relação a P_2 , de modo que P_1 e P_4 percorrem todos os candidatos à solução. A seguir, mais detalhes sobre a ação do rotor.

4.4.3 Aplicação de R

Definimos R em função de um ângulo ϕ entre 0 e θ , de modo que $\phi = 0$ representa que não há rotação, o ponto permanece estático, e $\phi = \theta$ representa a maior rotação permitida. É importante lembrar que, para a realização da rotação, é necessário aplicar o rotor à esquerda e seu reverso à direita do ponto, como feito com P_1 em (4.8):

$$R(\phi)P_1\tilde{R}(\phi) = y, \text{ tal que } y \in \text{Arco } 1. \quad (4.8)$$

Quando $\phi = \theta$, temos $R(\theta)P_1\tilde{R}(\theta) = P_3$.

Para P_2 , ou usamos $-\phi$, pois a rotação ocorre no sentido contrário, ou simplesmente aplicamos o Rotor em P_4 . Esta última foi a opção que utilizamos para a implementação, pois assim o sentido de rotação é mantido: $P_1 \rightarrow P_3 \rightarrow P_4 \rightarrow P_2$.

Exemplo 2. Para este exemplo, usamos os mesmos dados do Exemplo 1, transformando uma distância exata em intervalar. Alguns passos são análogos aos do Exemplo 1:

1. Definir as variáveis no espaço conforme;
2. Definir as esferas S_1 , S_{1b} , S_2 e S_3 ;

3. Calcular círculo que contém a solução: $C = S_2 \wedge S_3$;

4. Calcular os pares de pontos

$$Pp_1 = S_1 \wedge S_2 \wedge S_3 \rightarrow P_1 \text{ e } P_2;$$

$$Pp_2 = S_{1_b} \wedge S_2 \wedge S_3 \rightarrow P_3 \text{ e } P_4;$$

5. Os pontos formam em C os arcos P_1P_3 e P_2P_4 .

6. Calcular o ângulo θ correspondente aos arcos e o eixo de rotação L ;

7. Aplicar o rotor $R(\phi)$ aos extremos dos arcos, onde $0 \leq \phi \leq \theta$;

8. Calcular as possíveis soluções:

$$X_1 = R(\phi)P_1\tilde{R}(\phi)$$

$$X_2 = R(\phi)P_4\tilde{R}(\phi).$$

- Pontos conhecidos: $v_{i-3} = (0, 0, 0)$, $v_{i-2} = (-4, 0, 0)$ e $v_{i-1} = (-3, -1.1133, 1.3268)$.
- Distâncias conhecidas: $3.5 \leq d_{i-3,i} \leq 3.7$, $d_{i-2,i} = 3.1102$ e $d_{i-1,i} = 1.3$.

Exemplo 2 - Output:
$P_1 = -2.32e_1 - 0.981e_2 + 2.43e_3 + 6.12e_\infty + e_0$
$P_2 = -2.32e_1 - 2.22e_2 + 1.39e_3 + 6.12e_\infty + e_0$
$P_3 = -2.5e_1 - 1.03e_2 + 2.52e_3 + 6.84e_\infty + e_0$
$P_4 = -2.5e_1 - 2.31e_2 + 1.45e_3 + 6.84e_\infty + e_0$
$\theta = 0.251 \text{ rad}$
$X_1 = -2.49e_1 - 1.02e_2 + 2.52e_3 + 6.78e_\infty + e_0$
$X_2 = -2.49e_1 - 2.3e_2 + 1.44e_3 + 6.78e_\infty + e_0$

Tabela 4.3. Este é um estado durante a rotação, onde X_1 e X_2 estão nos Arcos 1 e 2, respectivamente, e mudam a medida que a rotação acontece.

O script produz a Figura 4.4-9, que é análoga à Figura 4.4-6:

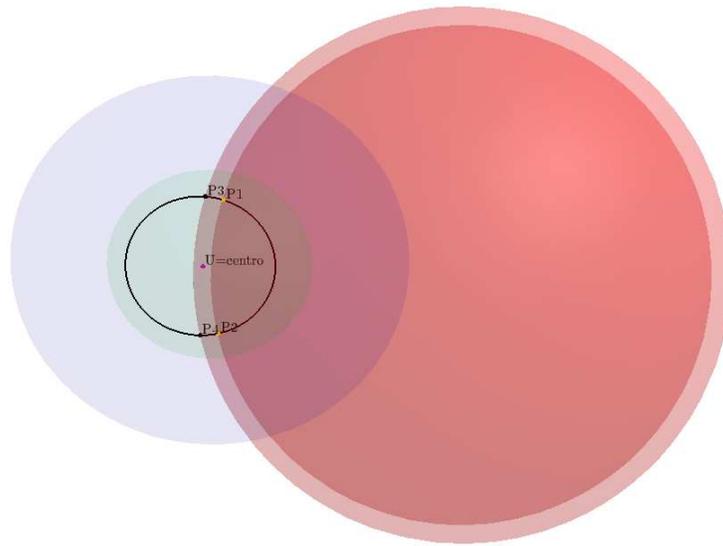


Figura 4.4-9. Exemplo 2. A esfera vermelha de menor raio produz P_1 e P_2 , a de maior raio produz, de forma correspondente, P_3 e P_4 . Essa correspondência gera os arcos P_1P_3 e P_2P_4 .

4.4.4 Adição de uma Distância Intervalar

Em alguns casos, podemos conhecer uma ou mais distâncias intervalares adicionais. Nesta seção, mostramos como uma distância intervalar adicional pode restringir a região de uma possível solução, em relação à situação apresentada na seção 4.4.1.

Se neste último a resolução se baseia na solução de dois subproblemas equivalentes ao Exemplo 1, agora trataremos de um problema que se baseia na resolução de dois subproblemas equivalentes ao Exemplo 2, o que sugere que tenhamos 4 arcos a serem analisados.

Podemos descrever brevemente este caso da seguinte forma: o primeiro subproblema utiliza as duas distâncias exatas e uma das intervalares gerando dois arcos, *Arcos* 1 e 2. Para o segundo subproblema as duas esferas geradas por distâncias exatas são mantidas enquanto a distância intervalar é substituída, produzindo outros dois arcos, *Arcos* 3 e 4 (Figuras 4.4-10 e 4.4-11).

As esferas que se originam das distâncias exatas geram a circunferência C que contém a solução, como nos casos anteriores. Os quatro arcos estão sobre C , e da Seção 4.4.1, podemos afirmar que o primeiro subproblema garante que a solução está sobre o *Arco* 1 ou o *Arco* 2. Porém, o segundo subproblema também garante a solução sobre o *Arco* 3 ou *Arco* 4. Já que a solução deve atender às restrições dos dois casos, isso nos leva a analisar apenas as *interseções* entre arcos gerados por subproblemas distintos. Desta forma, iremos, no melhor caso, descartar

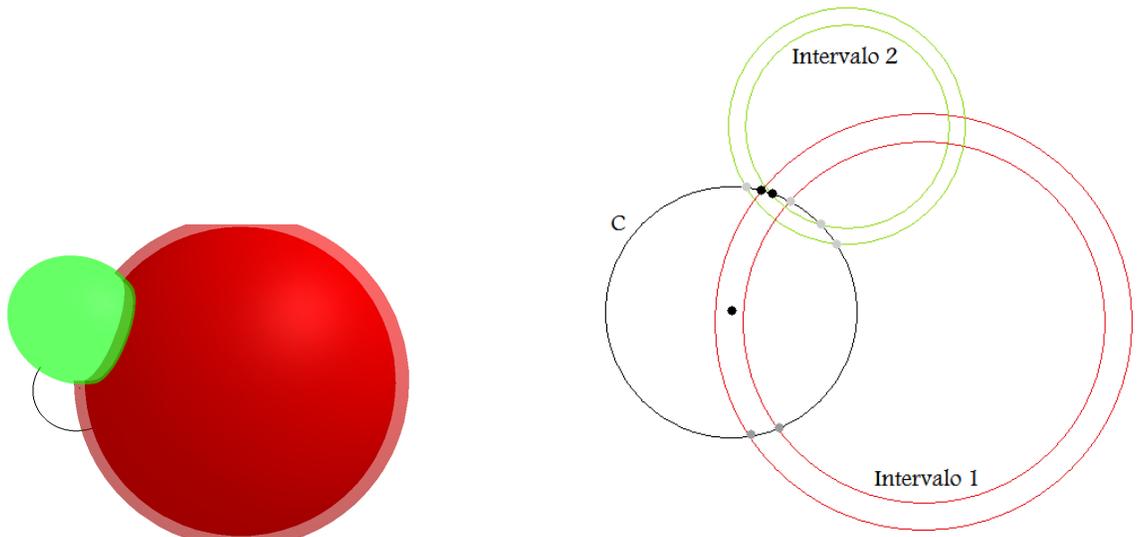


Figura 4.4-10. As esferas verdes e vermelhas representam a situação da Figura 4.4-10. Os círculos verde e vermelho representam a interseção entre as esferas correspondentes às distâncias exatas, respectivamente, às esferas verdes e vermelhas

até três arcos da solução e ainda reduzir o arco restante. Abaixo, ilustramos duas possibilidades de restrição da região factível.

A Figura 4.4-12 ilustra a situação após a resolução do primeiro subproblema. As Figuras 4.4-13 e 4.4-14 ilustram situações possíveis após a resolução do segundo subproblema. No primeiro caso, o arco que nos interessa após a adição de uma distância intervalar é P_3P_5 . Este arco é a interseção de P_1P_3 e P_5P_7 . O arco P_2P_4 é descartado, a exemplo do novo arco P_6P_8 . No segundo caso, a interseção entre P_1P_3 e P_5P_7 é o próprio arco P_1P_3 . Desta forma, nos concentramos apenas nesse arco.

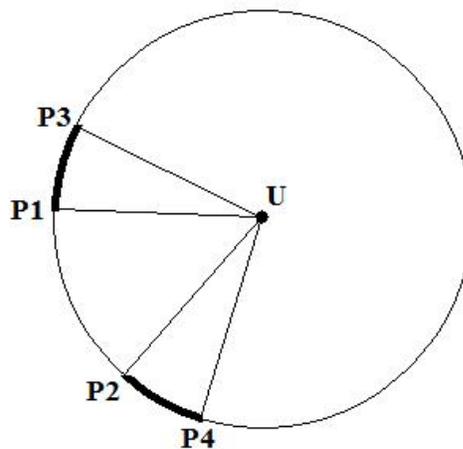


Figura 4.4-12. Situação após a resolução do primeiro subproblema

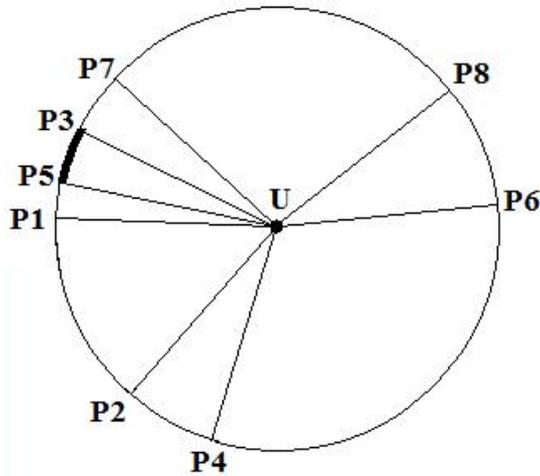


Figura 4.4-13. Segunda resolução (a)

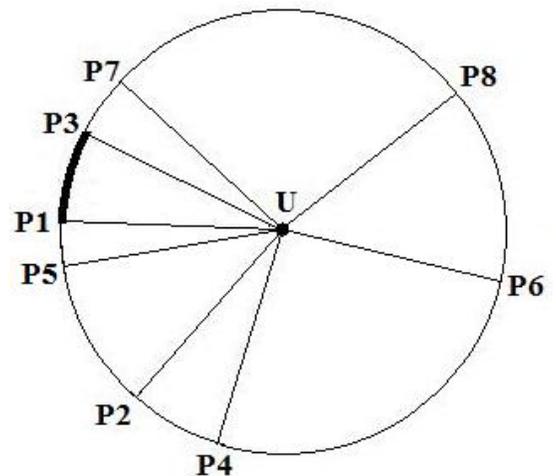


Figura 4.4-14. Segunda resolução (b)

No próximo exemplo, adicionamos uma distância intervalar ao Exemplo 2. A sequência de passos fica assim:

1. Definir as variáveis no espaço conforme;
2. Definir as esferas $S_1, S_{1_b}, S_2, S_3, S_4$ e S_{4_b} , através de $S_i = X_i - \frac{1}{2}r_i^2 e_\infty$;
3. Calcular o círculo que contém a solução: $C = S_2 \wedge S_3$;
4. Calcular os pares de pontos:

$$Pp_1 = S_1 \wedge S_2 \wedge S_3 \rightarrow P_1 \text{ e } P_2;$$

$$Pp_2 = S_{1_b} \wedge S_2 \wedge S_3 \rightarrow P_3 \text{ e } P_4;$$

$$Pp_3 = S_4 \wedge S_2 \wedge S_3 \rightarrow P_5 \text{ e } P_6;$$

$$Pp_4 = S_{4_b} \wedge S_2 \wedge S_3 \rightarrow P_7 \text{ e } P_8;$$

5. Os pontos formam em C os pares de arcos: P_1P_3 e P_2P_4 & P_5P_7 e P_6P_8 .
6. Buscar interseções entre arcos de pares diferentes;
7. Se a(s) interseção(s) diminuir(em) os arcos originais: calcular θ' , o ângulo entre os novos extremos;
8. Além do ângulo θ' , calcular o eixo de rotação L ;

9. Aplicar o rotor $R(\phi)$ aos extremos do arco, com $0 \leq \phi \leq \theta'$;
10. Solução $X = R(\phi)P_{inicial}\tilde{R}(\phi)$.

Exemplo 3. Dados:

- Pontos conhecidos: $v_{i-3} = (0, 0, 0)$, $v_{i-2} = (-4, 0, 0)$, $v_{i-1} = (-3, -1.1133, 1.3268)$ e $v_{i-j} = (-1.3720, -1.8260, 3.6596)$.
- Distâncias conhecidas: $3.5 \leq d_{i-3,i} \leq 3.7$, $d_{i-2,i} = 3.1102$, $d_{i-1,i} = 1.3$ e $1.772 \leq d_{i-j,i} \leq 1.8$.

A Tabela 4.4 mostra os oito pontos que constituem as extremidades dos arcos. A Figura 4.4-

Exemplo 3 - Output:
$P1 = -2.32e_1 - 0.981e_2 + 2.43e_3 + 6.12e_\infty + e_0$
$P2 = -2.32e_1 - 2.22e_2 + 1.39e_3 + 6.12e_\infty + e_0$
$P3 = -2.5e_1 - 1.03e_2 + 2.52e_3 + 6.84e_\infty + e_0$
$P4 = -2.5e_1 - 2.31e_2 + 1.45e_3 + 6.84e_\infty + e_0$
$P5 = -2.32e_1 - 0.98e_2 + 2.42e_3 + 6.1e_\infty + e_0$
$P6 = -2.27e_1 - 0.975e_2 + 2.39e_3 + 5.92e_\infty + e_0$
$P7 = -2.55e_1 - 1.05e_2 + 2.55e_3 + 7.05e_\infty + e_0$
$P8 = -2.06e_1 - 0.996e_2 + 2.22e_3 + 5.08e_\infty + e_0$
ângulo= 0.251rad
ângulo2= 0.329rad

Tabela 4.4. Solução do Exemplo 3.

15 mostra os arcos gerados pelas distâncias intervalares sobre a circunferência $C = S2 \wedge S3$, onde notamos que o arco que contém a solução é P_3P_5 . Logo, buscamos uma expressão que represente os pontos apenas deste arco.

O ângulo entre as retas $\overrightarrow{UP_3}$ e $\overrightarrow{UP_5}$ será o ângulo máximo de rotação. O rotor é o mesmo utilizado para descrever os pontos dos demais arcos, chamaremos de R . Para o Exemplo 3 temos:

$$\theta = \arccos \left(\frac{\overrightarrow{UP_3} \cdot \overrightarrow{UP_5}}{|\overrightarrow{UP_3}| |\overrightarrow{UP_5}|} \right) = 0.175rad,$$

Portanto, a solução é dada por pontos como y , onde

$$y = R(\phi)P_5\tilde{R}(\phi), \text{ para } 0 \leq \phi \leq 0.175. \quad (4.9)$$

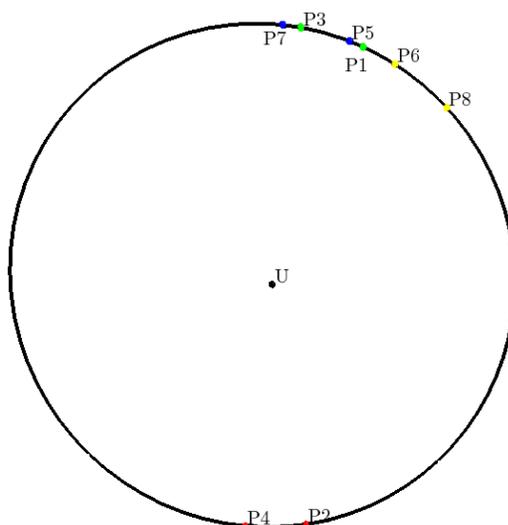


Figura 4.4-15. Os arcos P_1P_3 e P_5P_7 se intersectam criando o arco P_3P_5 , que contém a solução.

Utilizamos P_5 para manter o sentido de rotação.

Com a obtenção de um novo arco, resultado da interseção de outros dois arcos gerados por intervalos conhecidos, ocorre uma mudança nos extremos destes intervalos. Esta mudança, tal como a obtenção dos novos intervalos, é descrita a seguir. O descarte de arcos ainda resulta na poda de um ramo da árvore binária, que é uma das consequências que desejamos através desta nova abordagem.

4.4.4.1 Novas Distâncias Intervalares

Notamos que, com a obtenção de um novo arco, os intervalos que continham as distâncias que conhecíamos a priori também mudam. Para vermos como os novos intervalos são obtidos, tomemos como base o Exemplo 3, em que os arcos P_1P_3 e P_5P_7 se interceptam. Para o primeiro arco, podemos dizer que o ponto P_1 foi rotacionado, sobre o plano definido por C , até P_5 , e para o segundo dizemos que P_7 foi rotacionado, sobre o mesmo plano, até P_3 . Consideremos o 1º caso. P_5 foi obtido a partir da distância intervalar entre o átomo procurado x_i e o átomo x_{i-j} . A distância de x_{i-j} a P_5 determina um extremo do intervalo que contém a distância $d(x_{i-j}, x_i)$. P_1 foi obtido a partir da distância entre os átomos x_{i-4} e x_i e, analogamente, a distância entre x_{i-3} e P_1 determina um dos extremos do intervalo que contém a distância $d(x_{i-3}, x_i)$. Portanto, para descobrirmos o novo intervalo que contém $d(x_{i-3}, x_i)$, basta mantermos o extremo que gera P_3 e calcular a distância euclidiana entre o átomo x_{i-3} e o ponto P_5 . Este processo é repetido para

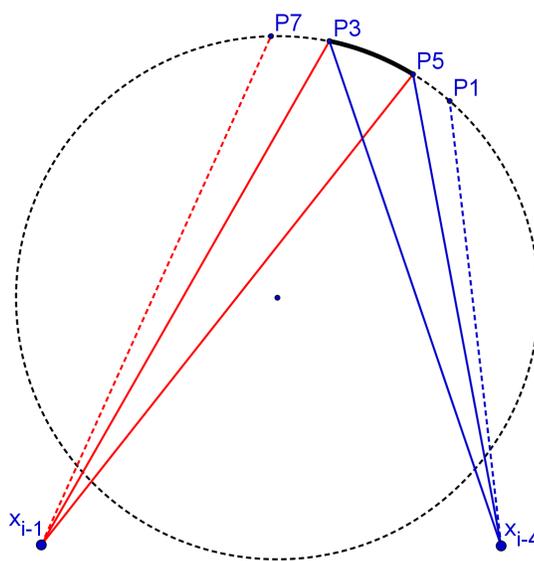


Figura 4.4-16. O segmento tracejado vermelho $\overline{x_{i-1}P_7}$ é substituído pelo segmento $\overline{x_{i-1}P_3}$, e o novo extremo do intervalo correspondente a distância entre os átomos x_{i-j} e x_i é $d(x_{i-j}, P_3)$. Analogamente, o segmento tracejado azul $\overline{x_{i-4}P_1}$ é substituído por $\overline{x_{i-4}P_5}$, de maneira que o extremo que foi modificado no intervalo correspondente a $d(x_{i-3}, x_i)$ seja agora igual a $d(x_{i-3}, P_5)$.

os pontos P_7 e P_3 a fim de encontrarmos também o novo intervalo que contém $d(x_{i-j}, x_i)$. A Figura 4.4-16 ilustra esta situação, sem a preocupação com a real posição dos átomos.

4.5 Resultados Computacionais

Chamaremos o novo algoritmo, com a abordagem da Álgebra de Clifford, de C-iBP. A integração desta abordagem ao iBP tem como objetivo diminuir o número de pontos calculados, para que a quantidade e a precisão de soluções possam ser aumentados mantendo o crescimento da árvore controlado.

No C-iBP, a poda é realizada antes da discretização, de forma que o algoritmo discretiza apenas regiões factíveis. Logo, em teoria, pode-se concluir que utilizando o mesmo número de pontos para a discretização de um arco, tanto para o iBP quanto para o C-iBP, há uma probabilidade maior de se obter soluções com o C-iBP. Se desejamos aumentar o número de soluções, e assim, a precisão destas soluções, aumentamos o número de pontos na discretização. Este aumento pode causar um crescimento muito grande no número de pontos calculados pelo iBP, dos quais muitos podem ser infactíveis. A seguir apresentamos os detalhes da implementação do algoritmo e em seguida alguns resultados computacionais.

4.5.1 Implementação

Nesta seção, descrevemos como ocorreu a implementação do C-iBP. Inicialmente as funções foram geradas com o CLUCalc, como nos exemplos das Seções 4.4 e 4.3 . O CLUCalc utiliza operadores da Álgebra de Clifford em seus cálculos e retorna multivetores. O objetivo foi criar uma função a ser inserida no iBP, modificando a fase de testes de factibilidade e discretização. Sendo assim, os cálculos realizados pelo CLUCalc deveriam ser traduzidos para a mesma linguagem do iBP, implementado em C.

Este processo foi realizado a partir do compilador *Gaalop*. CLUScripts baseados no modelo conforme 5D, são inseridos em sua interface. O software, por sua vez, gera automaticamente um código em C.

O modelo conforme possui uma base de 32 elementos, de forma que o Gaalop gera vetores de 32 coordenadas associados a cada multivetor. Ou seja, há uma identificação pré definida entre a base $\{1, e_1, e_2, e_3, e_\infty, e_0, e_1 \wedge e_2, \dots, I\}$ e as coordenadas $\{[0], [1], \dots, [31]\}$, e cada multivetor calculado pelo Gaalop, é representado em C como um vetor de 32 coordenadas. As coordenadas originais, do modelo conforme, aparecem como comentários no código gerado. De forma a otimizar o código, apenas as coordenadas não nulas são geradas.

Os exemplos abaixo ilustram este processo para o cálculo de um círculo. O primeiro é um exemplo numérico da interseção de duas esferas S_1 e S_2 de centros $(-4, 0, 0)$ e $(-3, -1.1133, 1.3268)$, e raios 3.1102 e 1.3, respectivamente. O segundo é um exemplo arbitrário, mais completo, em que os centros e as esferas também são gerados. Neste caso, o círculo tem sua representação em função das coordenadas das esferas.

Exemplo 4. Script:

```
a=3.1102;
b=1.3;
S1=VecN3(-4,0,0)-((a*a)/2)*einf;
S2=VecN3(-3,-1.1133,1.3268)-((b*b)/2)*einf;
?C=S1^S2;
```

Código gerado:

```
void calculate(float C[32])
```

```

C[6] = 4.4532f; // e1^e2
C[7] = -5.3072f; // e1^e3
C[8] = -11.129686f; // e1^einf
C[9] = -1.0f; // e1^e0
C[11] = 3.521733f; // e2^einf
C[12] = 1.1133f; // e2^e0
C[13] = -4.197104f; // e3^einf
C[14] = -1.3268f; // e3^e0
C[15] = -1.9915895f; // einf^e0

```

Exemplo 5. Script:

```

?X=VecN3 (x1, x2, x3);
?Y=VecN3 (y1, y2, y3);
?S1=X-((a*a)/2)*einf;
?S2=Y-((b*b)/2)*einf;
?C=S1^S2;

```

Código gerado:

```

void calculate(float a, float b, float x1, float x2, float x3,
float y1, float y2, float y3, float C[32], float S1[32], float S2[32],

X[1] = x1; // e1
X[2] = x2; // e2
X[3] = x3; // e3
X[4] = (x3 * x3 + x2 * x2 + x1 * x1) / 2.0f; // einf
X[5] = 1.0f; // e0
Y[1] = y1; // e1
Y[2] = y2; // e2
Y[3] = y3; // e3
Y[4] = (y3 * y3 + y2 * y2 + y1 * y1) / 2.0f; // einf
Y[5] = 1.0f; // e0

```

```

S1[1] = X[1]; // e1
S1[2] = X[2]; // e2
S1[3] = X[3]; // e3
S1[4] = X[4] - a * a / 2.0f; // einf
S1[5] = 1.0f; // e0
S2[1] = Y[1]; // e1
S2[2] = Y[2]; // e2
S2[3] = Y[3]; // e3
S2[4] = Y[4] - b * b / 2.0f; // einf
S2[5] = 1.0f; // e0
C[6] = S1[1] * S2[2] - S1[2] * S2[1]; // e1^e2
C[7] = S1[1] * S2[3] - S1[3] * S2[1]; // e1^e3
C[8] = S1[1] * S2[4] - S1[4] * S2[1]; // e1^einf
C[9] = S1[1] - S2[1]; // e1^e0
C[10] = S1[2] * S2[3] - S1[3] * S2[2]; // e2^e3
C[11] = S1[2] * S2[4] - S1[4] * S2[2]; // e2^einf
C[12] = S1[2] - S2[2]; // e2^e0
C[13] = S1[3] * S2[4] - S1[4] * S2[3]; // e3^einf
C[14] = S1[3] - S2[3]; // e3^e0
C[15] = S1[4] - S2[4]; // einf^e0

```

O comando *VecN3*, do CLUCalc, projeta um vetor de \mathbb{R}^3 no espaço conforme.

4.5.2 Função Clifford

A função *Clifford* funciona da seguinte forma: recebe como entrada 3 pontos consecutivos e suas respectivas distâncias ao próximo ponto, das quais uma é intervalar. Em um primeiro momento, calcula o círculo C que contém a solução, interseção das esferas que provém de distâncias exatas. Em seguida calcula as interseções com a casca esférica gerada pela distância intervalar, produzindo dois arcos em C .

Em um segundo momento, verifica-se a existência de distâncias intervalares adicionais à pontos já posicionados. Em caso afirmativo, novas interseções são computadas com o círculo C ,

gerando novos arcos. A função, então, calcula as interseções entre os arcos, como no Exemplo 3, descartando quaisquer pontos inactíveis. Logo, a discretização é feita apenas sobre regiões factíveis.

O processo de inserção desta função no código iBP foi realizado durante o estágio do autor no IRISA, Universidade de Rennes I (França), no segundo semestre de 2012, com a importante colaboração de Antonio Mucherino (IRISA) e Andrea Cassioli (LIX-École Polytechnique, Paris), dois dos responsáveis pelo desenvolvimento do iBP.

Neste processo, buscamos evitar que as vantagens oferecidas pela Álgebra de Clifford fossem perdidas durante a tradução da linguagem. O código segue em constante processo de melhoria nesta direção. Abaixo listamos alguns obstáculos enfrentados no desenvolvimento da primeira versão do código, e como foram e estão sendo tratados, com o objetivo de construirmos versões cada vez mais competitivas.

- Cascas esféricas de distâncias adicionais não intersectam o círculo que contém as soluções: a partir de um círculo C que contenha as soluções em arcos, distâncias adicionais geram esferas que podem intersectar os arcos e descartar pontos inactíveis. Porém, nem sempre estas interseções ocorrem, e ainda pode não ocorrer a interseção com o círculo. Na teoria sabemos que isso significa que curvas inteiras foram descartadas ou nada foi descartado, mas a implementação desta verificação não é simples. É necessário primeiramente testar quando há interseção, caso contrário o algoritmo realiza a operação e encontra pontos fora da circunferência e possivelmente soluções não factíveis. Uma solução para este problema foi determinar as distâncias máxima e mínima do átomo correspondente à esta distância intervalar, à cada um dos arcos da circunferência, e em seguida compará-las com o intervalo fornecido pelo problema. Este procedimento “conserta” as distâncias intervalares adicionais, porém, aumenta sensivelmente o número de operações do algoritmo, já que ele é realizado para cada distância adicional.
- Outra dificuldade enfrentada está na determinação da orientação da circunferência C . Atualmente, muitos testes são realizados para o algoritmo determinar as extremidades de um arco após o uso das distâncias intervalares.

Instâncias				C-iBP			iBP			
nome	n_{aa}	n_a	$ r $	Poda	D	Nós	Descarte	D	Nós	Descarte
hm30	4	18	28	FDD	4	13803	822	6	929	475
2jmy-s	5	26	42	FDD+FAT	3	5220	2152	4	3902	1855
2jmy-m	10	51	90	FDD+FAT	3	11172	4633	6	140270	75200
2jmy	15	77	134	FDD+FAT	3	19063	7951	12	47681	26242
2kxa	23	117	206	FDD+FAT	3	1275	396	8	7942	3937
2ppz	36	170	323	FDD+VDW	4	15618	6648	6	43073	18584

Tabela 4.5. Resultados computacionais.

4.5.3 Resultados e Conclusões

A Tabela 4.5 apresenta resultados de testes realizados com o C-iBP e o iBP. As informações nas colunas, referentes às instâncias e aos algoritmos, são listadas abaixo:

- n_{aa} e n_a : o número de aminoácidos e o número de átomos, respectivamente;
- $|r|$: o comprimento da *reordenação* da instância;
- Poda: dispositivos de poda utilizados para o descarte de pontos ineficazes;
- D : para o C-iBP é o número máximo, e para o iBP é o número exato de pontos usados na discretização, necessários para cada algoritmo encontrar ao menos uma solução;
- Nós: número de nós computados até a primeira solução;
- Descarte: número de nós descartados até a primeira solução.

Os dispositivos de poda são usados no iBP para o descarte de pontos ineficazes. O FDD (Factibilidade de Distâncias Diretas) verifica se as distâncias entre pontos já computados satisfazem as distâncias fornecidas pelos experimentos de RMN. Com base nestes mesmos experimentos, é possível obter informações sobre ângulos de torção. O FAT (Factibilidade dos Ângulos de Torção) verifica se os ângulos de torção de uma imersão parcial satisfazem estas informações. Detalhes sobre estes dispositivos são encontrados em [38]. Outro dispositivo utilizado foi o VDW, factibilidade de van der Waals. Um limite inferior para uma distância inter-atômica pode ser deduzido a partir do *raio de van der Waals* [8]. Se dois átomos não estão quimicamente ligados, a distância entre eles deve ser maior ou igual à soma de seus raios de van der Waals.

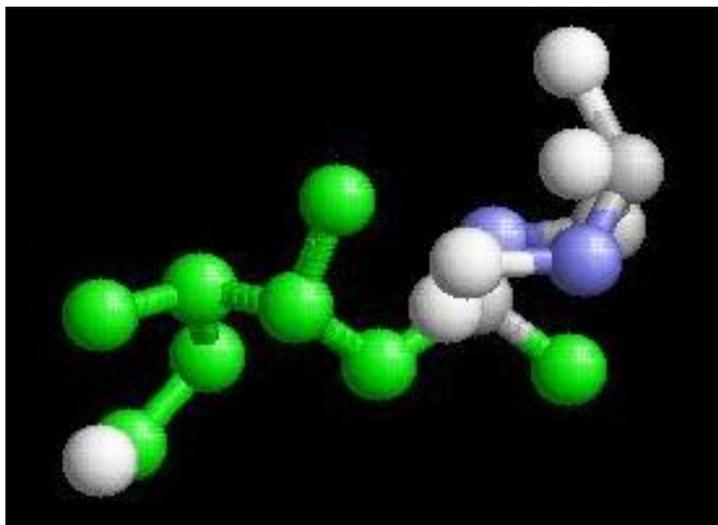


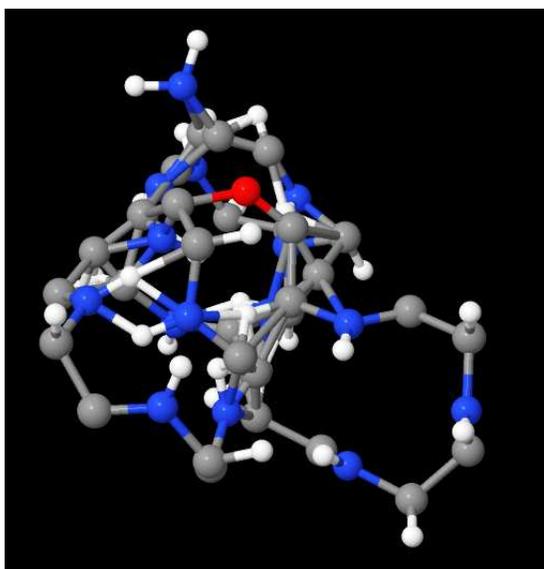
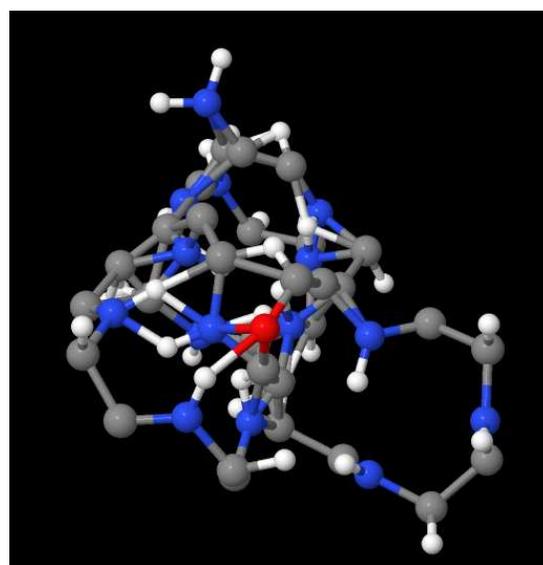
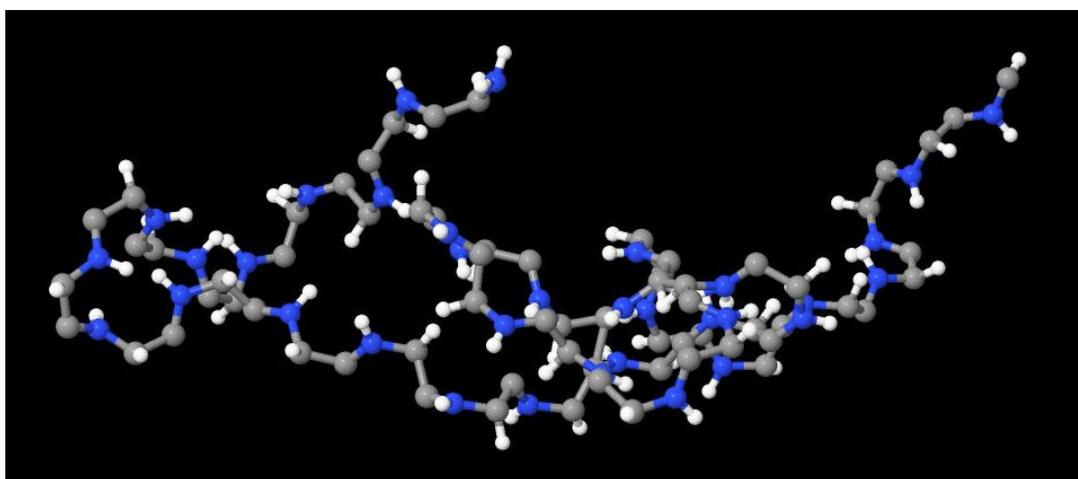
Figura 4.5-17. Molécula *hm30*.

O algoritmo C-iBP, em teoria, não gera pontos inactíveis. De forma que estes métodos não precisariam ser empregados. Porém, o algoritmo trabalha com tolerâncias nos cálculos dos ângulos referentes aos arcos, o que pode ocasionar a geração de pontos inactíveis. Logo, integramos os métodos FDD e FAT também ao C-iBP.

Neste momento, o principal parâmetro de comparação é D . O número de pontos necessários na discretização das curvas de modo a encontrar a primeira solução. No caso de C-iBP, pelo fato da discretização ser feita apenas sobre regiões factíveis, o número de pontos usados não é constante. Ele depende do tamanho da curva. Isto é, uma curva que sofreu um descarte maior de pontos, dado por distâncias intervalares adicionais, terá menos pontos em sua discretização. Isto evita que hajam muitos pontos em curvas muito pequenas. Um parâmetro que nos auxilia na determinação de D é ρ . Consideramos que pontos que distam menos do que esta tolerância, são considerados equivalentes. Para os testes utilizamos $\rho = 0.1 \text{ \AA}$.

A Figura 4.5-17 é de uma solução obtida com o C-iBP para a molécula *hm30*, de 18 átomos. As Figuras 4.5-18 e 4.5-19 se referem à duas soluções encontradas para a molécula *2jmy*, de 77 átomos. A Figura 4.5-20 é de uma solução para a molécula *2ppz*.

Podemos observar que em todos os testes, o número de pontos necessário para se encontrar uma solução é sempre menor no C-iBP. Em um teste para a instância *2jmy*, com $D = 4$, o iBP não encontrou nenhuma solução após pesquisar 2, 080, 184, 420 nós e descartar 1, 057, 771, 360. De fato só pôde encontrar a primeira solução com $D = 12$.

Figura 4.5-18. Molécula *2jmy*.Figura 4.5-19. Molécula *2jmy*.Figura 4.5-20. Molécula *2ppz*.

Uma consequência da diminuição em D é o cálculo de um menor número de nós. Pela Tabela 4.5 podemos observar que a medida que cresce o número de átomos, a diferença na quantidade de pontos computados é mais facilmente observada. Além disso, o algoritmo C-iBP permite soluções mais precisas e uma árvore de busca menor, pois podemos refinar a solução gerando apenas pontos factíveis. No caso do iBP, um refinamento equivalente exigiria um aumento maior de pontos discretizados, e portanto também um maior aumento no número de nós computados, já que parte dos nós gerados são infactíveis.

Estas conclusões condizem com as expectativas teóricas da abordagem pela Álgebra de Clifford. Porém, planejamos estender os testes para moléculas maiores a fim de nos certificar-

mos ainda mais sobre os resultados que esperávamos.

Capítulo 5

Conclusão e Trabalhos Futuros

Neste trabalho, apresentamos uma nova abordagem à resolução de uma classe de problemas de geometria de distâncias, chamada de iPDGDM. Através do modelo conforme das Álgebras de Clifford (AC), foi possível estabelecer um método promissor para tratar as distâncias intervalares, com base na interpretação geométrica do problema. Outra aplicação da Álgebra de Clifford é vista em [3], onde apresentamos uma nova abordagem ao algoritmo de Grover.

No iPDGDM, a busca por pontos se identifica com interseções de esferas no espaço, o que nos remete a uma aplicação direta do modelo conforme da AC, ao qual nos referimos também como Álgebra Geométrica Conforme (AGC).

Atualmente, um algoritmo Branch-and-Prune (iBP) é utilizado na busca por soluções do iPDGDM, e o iBP é um algoritmo eficiente. Porém, intuitivamente, é possível perceber onde este algoritmo pode ser aprimorado. Fato é que, em geral, ele calcula uma grande quantidade de pontos inactíveis. Logo, poderíamos imaginar um método que evita que tais pontos sejam calculados. É isto que o novo algoritmo Clifford-iBP (C-iBP) faz.

No C-iBP, criado pela inserção de uma função chamada `Clifford` no iBP, apenas pontos factíveis são gerados, pois a poda da região inactível ocorre previamente à discretização. Ao contrário do iBP, que discretiza as regiões e utiliza dispositivos de poda para eliminar os pontos inactíveis gerados.

A criação do algoritmo C-iBP se deve a um cuidadoso estudo sobre as teorias envolvidas e a uma adaptação ao uso de ferramentas computacionais, até o início do trabalho, desconhecidas

Conclusão e Trabalhos Futuros

do autor.

Como já foi dito, a inserção no iBP da função `Clifford`, criada de maneira independente, teve a colaboração de Andrea Cassioli e Antonio Mucherino, que fizeram parte do desenvolvimento do iBP.

Inserida a função, iniciou-se o trabalho de adaptação da função ao código, para que em seguida pudéssemos criar o algoritmo C-iBP. Resultados obtidos por este algoritmo são citados no Capítulo 4.

Nossas conclusões preliminares sobre o novo algoritmo dizem que são necessários menos pontos na discretização para o C-iBP encontrar a primeira solução. O número de pontos gerados pelo C-iBP é menor, da mesma forma que a árvore será menor. Além disso, podemos controlar o tamanho da árvore caso queiramos refinar a discretização. No iBP, este procedimento pode causar uma explosão combinatória.

Este algoritmo ainda precisa ser aprimorado para se tornar mais competitivo em suas próximas versões. Uma implementação mais robusta exige que os cálculos sejam cuidadosamente revistos para eliminar redundâncias ainda existentes e introduzir tolerâncias, para que a implementação se torne mais segura com respeito a problemas numéricos. Esta redundância em alguns cálculos e um grande número de testes realizados, a cada iteração, aumentam o tempo de processamento. Buscamos diminuir este número de testes com a implementação mais eficiente de algumas características da Álgebra de Clifford, especialmente no que diz respeito à orientação de rotações. No futuro, o objetivo é a criação de um algoritmo inteiramente baseado nas Álgebras de Clifford.

Referências Bibliográficas

- [1] R. Ablamowicz. Clifford algebra computations with Maple. In *Clifford (Geometric) Algebras*. 1996.
- [2] R. Ablamowicz and B. Fauser. Cliffordlib. <http://math.tntech.edu/rafal/cliff13/>. 2012.
- [3] R. Alves and C. Lavor. Clifford algebra applied to Grover’s algorithm. *Advances in Applied Clifford Algebras*, 20:477–488, 2010.
- [4] E. Bayro-Corrochano and L. E. Falcón. Geometric algebra of points, lines, planes and spheres for computer vision and robotics. *Robotica*, 23(6):755–770, 2005.
- [5] B. Berger, J. Kleinberg, and T. Leighton. Reconstructing a three-dimensional model with arbitrary errors. *J. ACM*, 46(2):212–235, 1999.
- [6] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, and P.E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.
- [7] P. Biswas, K.-C. Toh, and Y. Ye. A distributed SDP approach for large-scale noisy anchor-free graph realization with applications to molecular conformation. *SIAM J. Sci. Comput.*, 30(3):1251–1277, 2008.
- [8] A. Bondi. van der Waals volumes and radii. *The Journal of Physical Chemistry*, 68(3):441–451, 1964.
- [9] J. Cameron and J. Lasenby. Oriented conformal geometric algebra. *Advances in Applied Clifford Algebras*, 18:523–538, 2008.

- [10] O. Carbajal-Espinosa, A. Loukianov, and E. Bayro-Corrochano. Obstacle avoidance for a humanoid arm using conformal geometric algebra. In *10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2010.
- [11] O. Carbajal-Espinosa, A. Loukianov, and E. Bayro-Corrochano. Modeling and control of a humanoid arm using conformal geometric algebra and sliding modes. In *11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2011.
- [12] T. E. Creighton. *Proteins: Structures and Molecular Properties*. W. H. Freeman, second edition edition, 1992.
- [13] Q. Dong and Z. Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *Journal of Global Optimization*, 22:365–375, 2002.
- [14] C. Doran. Circle and sphere blending with conformal geometric algebra. <http://arxiv.org/abs/cs/0310017v1>, 2003.
- [15] L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science: An Object-Oriented Approach to Geometry (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., 2007.
- [16] L. Dorst, S. Mann, and T. Bouma. Gable: A Matlab tutorial for geometric algebra in <http://staff.science.uva.nl/leo/gable/tutorial.pdf>. 2002.
- [17] L.A.F. Fernandes and M. M de Oliveira. A general framework for subspace detection in unordered multidimensional data. *Pattern Recognition*, 45(9):3566 – 3579, 2012.
- [18] L.A.F. Fernandes and M.M. de Oliveira. Geometric algebra: A powerful tool for solving geometric problems in visual computing. In *Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing (Sibgrapi Tutorials)*, 2009.
- [19] D. Fontijne. Gaigen 2: a geometric algebra implementation generator. In *Proceedings of ACM GPCE*, 2006.
- [20] S. Franchini, G. Vassalo, and F. Sorbello. A brief introduction to Clifford algebra. Technical report, 2010.
- [21] T. Havel. Distance geometry. In *Encyclopedia of Nuclear Magnetic Resonance*. 1995.

- [22] D. Hestenes. *Geometric Calculus: a unified language for mathematics and physics*. D. Reidel Publishing Company, Holland, 1987.
- [23] D. Hestenes. *New Foundations for Classical Mechanics*. Kluwer Academic Publishers, Dordrecht, Holanda, 1993.
- [24] D. Hildenbrand. *Geometric Computing in Computer Graphics and Robotics using Conformal Geometric Algebra*. PhD thesis, Technischen Universität Darmstadt, 2006.
- [25] D. Hildenbrand. Gaalop. <http://www.gaalop.de>. 2012.
- [26] D. Hildenbrand, D. Fontijne, C Perwass, and L. Dorst. Geometric algebra and its application to computer graphics. Technical report, 2004.
- [27] D. Hildenbrand, J. Zamora, and E. Bayro-Corrochano. Inverse kinematics computation in computer graphics and robotics using conformal geometric algebra. *Advances in Applied Clifford Algebras*, 18:699–713, 2008.
- [28] C. Lavor, J. Lee, A. Lee-St. John, L. Liberti, A. Mucherino, and M. Sviridenko. Discretization orders for distance geometry problems. *Optimization Letters*, 6:783–796, 2012.
- [29] C. Lavor, L. Liberti, and N. Maculan. Molecular distance geometry problem. In *Encyclopedia of Optimization*. 2009.
- [30] C. Lavor, L. Liberti, N. Maculan, and A. Mucherino. The discretizable molecular distance geometry problem. *Computational Optimization and Applications*, 52(1):115–146, 2012.
- [31] C. Lavor, L. Liberti, and A. Mucherino. The interval Branch-and-Prune algorithm for the discretizable molecular distance geometry problem with inexact distances. *Journal of Global Optimization*, 2013 (a aparecer).
- [32] H. Li, D. Hestenes, and A. Rockwood. Geometric computing with Clifford algebras. chapter Generalized homogeneous coordinates for computational geometry. 2001.
- [33] P. Lounesto. *Clifford Algebras and Spinors*. Cambridge University Press, 2001.
- [34] P. Lounesto. CLICAL. <http://users.tkk.fi/~ppuska/mirror/Lounesto/CLICAL.htm>. 2002.
- [35] S. Mann, L. Dorst, and T. Bouma. The making of a geometric algebra package in Matlab in <http://www.cs.uwaterloo.ca/research/tr/1999/27/cs-99-27.pdf>. 1999.

- [36] J.J. Moré and Z. Wu. Distance geometry optimization for protein structures. *J. of Global Optimization*, 15(3):219–234, 1999.
- [37] A. Mucherino and C. Lavor. The branch and prune algorithm for the molecular distance geometry problem with inexact distances. In *Proceedings of the International Conference on Computational Biology*, 2009.
- [38] A. Mucherino, C. Lavor, T. Malliavin, L. Liberti, M. Nilges, and N. Maculan. Influence of pruning devices on the solution of molecular distance geometry problems. In *Experimental Algorithms*. 2011.
- [39] A. Mucherino, L. Liberti, and C. Lavor. MD-jeep: An implementation of a Branch and Prune algorithm for distance geometry problems. In *Mathematical Software ICMS*. 2010.
- [40] C Perwass. *Geometric Algebra with Applications in Engineering*. Springer-Verlag, 2009.
- [41] C Perwass. CLUCalc. <http://www.clucalc.info/>. 2010.
- [42] C Perwass and D. Hildenbrand. Aspects of geometric algebra in euclidean, projective and conformal space. Technical report, 2004.
- [43] A.T. Philips, J.B. Rosen, and V.H. Walken. Molecular structure determination by convex underestimation of local energy minima. In *Global Optimization of Nonconvex Energy Functions: Molecular Conformation and Protein Folding*. 1996.
- [44] A. Pogorelov. *Geometry*. Mir, Moscow, 1987.
- [45] B. Rosenhahn. *Pose Estimation Revisited*. PhD thesis, Institut für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel, 2003.
- [46] B. Rosenhahn, C. Perwass, and G. Sommer. Pose estimation of 3d free-form contours. *International Journal of Computer Vision*, 62:267–289, 2005.
- [47] B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra part i: The stratification of mathematical spaces. *Journal of Mathematical Imaging and Vision*, 22:27–48, 2005.
- [48] J.B. Saxe. Embeddability of weighted graphs in k-space is strongly np-hard. In *Proceedings of 17th Allerton Conference in Communications, Control and Computing*, 1979.
- [49] R Sayle. RasMol. <http://www.rasmol.org/>. 1992.

- [50] T. Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer, New York, 2002.
- [51] G. Sommer. *Geometric Computing with Clifford Algebras*. Springer-Verlag, 2001.
- [52] J. Vaz Jr. and R. Rocha Jr. *Algebras De Clifford e Espinores*. Livraria da Física, 2012.
- [53] D. Wu and Z. Wu. An updated geometric build-up algorithm for solving the molecular distance geometry problems with sparse distance data. *Journal of Global Optimization*, 37:661–673, 2007.
- [54] J. Zamora-Ezquivel. $g_{6,3}$ geometric algebra. In *Proceedings of the 9th International Conference on Clifford Algebras and their Applications in Mathematical Physics, Weimar*, 2011.