

Rodrigo de Almeida Rosa

**Cálculo de grupos fundamentais de nuvens de pontos
através de técnicas de Análise Topológica de Dados
(TDA)**

São José do Rio Preto
2021

Rodrigo de Almeida Rosa

**Cálculo de grupos fundamentais de nuvens de pontos
através de técnicas de Análise Topológica de Dados
(TDA)**

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Orientador: Prof^ª. Dr^ª. Alice Kimie Miwa Libardi
Coorientador: Prof^ª. Dr^ª. Eliris Cristina Rizziolli

São José do Rio Preto
2021

R788c

Rosa, Rodrigo de Almeida

Cálculo de grupos fundamentais de nuvens de pontos através de técnicas de Análise Topológica de Dados (TDA) / Rodrigo de Almeida Rosa. -- São José do Rio Preto, 2021

95 p. : tabs., fotos

Dissertação (mestrado) - Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto

Orientadora: Alicie Kimie Miwa Libardi

Coorientadora: Eliris Cristina Rizzioli

1. Grupo fundamental. 2. GAP. 3. Análise Topológica de Dados. 4. Homologia de Persistência. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Biociências Letras e Ciências Exatas, São José do Rio Preto. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Rodrigo de Almeida Rosa

Cálculo de grupos fundamentais de nuvens de pontos através de técnicas de Análise Topológica de Dados (TDA)

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Matemática, junto ao Programa de Pós-Graduação em Matemática, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Financiadora: CAPES

Comissão Examinadora

Prof^a. Dr^a. Eliris Cristina Rizziolli
UNESP – Câmpus de Rio Claro - SP
Coorientadora

Prof. Dr. Thiago de Melo
UNESP – Câmpus de Rio Claro - SP

Prof. Dr. Northon Canevari Leme Penteado
UFCA – Câmpus de Brejo Santo - CE

Rio Claro
15 de Fevereiro de 2021

RESUMO

Em Análise Topológica de Dados (TDA) existem diversas ferramentas que combinam conceitos da topologia clássica com a análise de dados e rotinas computacionais. O conceito mais utilizado até então é a homologia de persistência com diversos resultados e aplicações. Desse modo, desejamos encontrar resultados pertinentes a TDA utilizando o grupo fundamental e seus invariantes. Para isso será necessário entendermos conceitos como CW -complexos, retratos zig-zag, funtores e campo de vetores discreto. O presente trabalho é um estudo sobre o cálculo do grupo fundamental de uma nuvem de pontos, no contexto de TDA, dada a relevância que o grupo fundamental já apresenta na Topologia Algébrica. Além disso, utilizamos o software GAP- *Group Algebra Programming* para realizar cálculos pertinentes de programação.

Palavras-chave: Grupo fundamental, GAP, Análise Topológica de Dados, Homologia de Persistência.

ABSTRACT

In Topological Data Analysis (TDA) many tools relate classic topology with data analysis and computational routines. Persistent Homology was the most relevant with many results and applications. Then, we intend to find pertinent results in TDA, implementing the fundamental group and invariants. It will be necessary understanding concepts as CW-complex, zig-zag retracts, functors and discrete vector fields. This work is a study on calculus of fundamental group from point clouds, in TDA context. We used the software GAP- Group Algebra Programming.

Keywords: Fundamental group, GAP, Topological Data Analysis, Persistent Homology.

Ao meu primeiro Amor.

AGRADECIMENTOS

Primeiramente agradeço a Deus, pois sem seu amor e sua benção nada disso seria possível. Agradeço a Deus também pela possibilidade de estudar em uma universidade pública e de qualidade.

À minha família, minha mãe Sônia, meu pai Carlos e meu irmão Rafael, pelo amor, por toda a paciência, motivação, pelo carinho, testemunho de fé e por sempre acreditarem em mim. Agradeço também, pelo apoio nos momentos mais difíceis e por serem uma das maiores bases da minha vida.

À minha noiva, Gabrielle, por todo amor, carinho, por ouvir minhas aflições, desabafos, além das alegrias e risadas juntos. Além disso, agradeço-a por ser minha fiel companheira, e apoiar-me durante todo este trabalho, principalmente no difícil período da pandemia.

Agradeço principalmente minha orientadora, Profa. Dra. Alice Libardi e minha coorientadora Profa. Dra. Eliris Cristina Rizzioli que, mesmo a distância, nunca mediram esforços, tampouco pouparam dedicação para ensinar, pesquisar e me guiar durante esse processo, além de me incentivarem e inspirarem a seguir a área que desejava aprender.

Agradeço a todos os meus amigos de Mogi Guaçu e Rio Claro, por todo apoio, carinho, e todas as conversas, momentos excepcionais e alegrias compartilhadas. Em especial, agradeço aqueles que estiveram mais presentes durante este mestrado, Gabriela, Larissa, Leonardo, Rafaela, Scarlett, Aline, Daniele, Renan, Deborah, Gustavo, Sabrina, Willian, Alissan, Fabio, Francielly e Verusca.

Ao Grupo de Jovens Porta do Céu, ao Grupo de Oração Universitário Fogo do Céu, ao Ministério Universidades Renovadas de Pircacicaba, e todos que conheci neste meio, que foram meu grande apoio espiritual durante toda minha vida universitária.

A todos meus amigos da República Mateca, em especial, Guilherme, Marco, Raul, Rivaldo, que neste período foram como a minha segunda família, e pretendo

levá-los por toda a vida. Agradeço, principalmente, ao Marco Antônio de Freitas Contessoto por todas as horas gastas voluntariamente ajudando-me na pesquisa e a responder minhas dúvidas com relação a conceitos de TDA, e ao Raul Lima que, graças a uma de conversa cotidiana, teve influência direta no tema da minha dissertação.

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

Por fim, agradeço a todos que, direta ou indiretamente, tiveram participação ou influência no desenvolvimento deste trabalho.

Ainda que eu tivesse o dom da profecia, o conhecimento de todos os mistérios e de toda ciência;... Se eu não tivesse o amor, eu não seria nada.

I Co 13, 2

Lista de Figuras

2.1	GAP (Groups, Algorithms, Programming)	28
3.1	Um cubo (esquerda) e um permutaedro (direita).	33
3.2	Construção da célula Dirichlet–Voronoi $D_L(2, 2)$	34
3.3	Permutaedros em mosaico.	35
3.4	Modelos do complexo puro cúbico e permutaedral do nó do trevo. . . .	36
4.1	Amostra de pontos S sobre uma variedade $M \subset \mathbb{R}^2$ desconhecida. . . .	40
4.2	Algumas vizinhanças cúbicas do conjunto S	41
4.3	β_0 barcode dos instantes 0 ao 20 dos espaços da figura 4.2	43
4.4	Imagem digital dos parafusos.	44
4.5	Gráfico de T cartesiano B	45
4.6	β_0 barcode da figura 4.4.	46
5.1	Uma célula e_j^k e seu complexo contato $Cont(e_j^k)$	48
5.2	Retrato zig-zag de um anel.	51
6.1	Conjunto V à esquerda e $nerve(\mathcal{U})$ à direita.	53
6.2	Backbone da proteína 1v2x T. thermophilus representado como um nó.	55
6.3	Backbone da proteína 1v2x T. thermophilus representado como um com- plexo puro cúbico. Visão frontal (esquerda) visão lateral (direita). . . .	56
7.1	Uma estrutura CW regular de um toro munida com um campo de veto- res discreto à esquerda e uma estrutura não regular de um toro à direita.	60
7.2	Os espaços T^0 (à esquerda), V^1 (centro) e T^1 (à direita).	61
8.1	Trajeto dos vértices de e_1^1 e e_2^1	68

8.2	Campo de vetores discreto K à esquerda e a representação do $H(\partial e^2)$ à direita.	69
8.3	Amostra X sobre uma superfície $K \subset \mathbb{R}^3$ (esquerda) e barcode β_1 (direita).	70
8.4	Uma representação cúbica (não variedade) e uma permutaedral (variedade).	72
8.5	Retratos por def. permutaedrais são homeom. diferente dos cúbicos.	73
8.6	Simplificação da estrutura celular de um CW -espaço regular.	75
9.1	Nó borromeano	81

Sumário

1	Introdução	19
2	Preliminares	21
2.1	Elementos básicos	21
2.2	GAP - Groups Algorithms Programming	28
3	CW-espaços	31
4	Homologia de Persistência no HAP	39
4.1	Primeiro exemplo	39
4.2	Segundo exemplo	44
5	Retratos por deformação	47
5.1	Retrato zig-zag	49
6	Complexo de Čech e funtores esquecimento	53
7	Campo de vetores discreto	59
8	Calculando grupos fundamentais	65
8.1	Variedades celulares e um invariante do nó	72
8.2	Calculando subgrupos e invariantes de grupos	76
9	Exemplos do HAP	81
9.1	Investigação	83
10	Conclusão	87
	Referências	89

1 Introdução

A grande motivação para os trabalhos usando resultados sobre o grupo fundamental de um dado espaço surgiu da pergunta: o universo é simplesmente conexo? Gromov considerou que mais do que perguntar se o universo é simplesmente conexo, seria perguntar: **como** o espaço é simplesmente conexo? Facundo et al. considera em [1] o espaço X como sendo uma esfera de dimensão 2 com uma pequena alça grudada, que não é simplesmente conexo. Mas se imaginarmos a alça bem pequena, diríamos que X é quase simplesmente conexo.

A Análise Topológica de Dados (TDA) trabalha em geral com uma quantidade muito grande de dados, porém em número finito, logo seria muito interessante se conseguíssemos calcular o grupo fundamental de uma amostra de uma nuvem X com finitos pontos de um espaço métrico K .

Sabemos que a Análise Topológica de Dados (TDA), possui diversas aplicações referentes a homologia de persistência. Por esta razão, surgem as seguintes perguntas: assim como a homologia, será que conseguimos retirar dados importantes do grupo fundamental? Será que conseguimos informações adicionais do grupo fundamental, em aplicações, além daquelas já conhecidas em homologia de persistência? Essas são as principais perguntas que motivam esta dissertação.

O objetivo principal deste trabalho é calcular o grupo fundamental, $\pi_1(X)$, usando apenas os dados de X . Uma das técnicas usadas em TDA é associar a X um CW -espaço regular filtrado.

Usaremos neste trabalho o programa “GAP - *Groups, Algorithms, Programming* - a System for Computational Discrete Algebra” [2], para realizar o cálculo do grupo fundamental de uma nuvem de pontos. Para tal finalidade é necessário a utilização do pacote “HAP: Homological Algebra Programming” [3], extremamente útil para traba-

lhar com CW -complexos, Homologias e Grupos Fundamentais.

Para entender como é o processo para computar o grupo fundamental no contexto de TDA teremos que relembrar como se dá em Topologia Algébrica. Assim como o grupo fundamental (ou grupos de homotopia, em geral) fornecem mais dados do que as homologias (singular, simplicial, cúbica, etc), usando o grupo fundamental gostaríamos de obter outros resultados além daqueles já obtidos na teoria de homologia de persistência. O estudo de grupo fundamental em TDA é bastante recente, envolvendo o período de 2014 até os dias atuais.

Esta dissertação foi baseada principalmente nos seguintes artigos:

1. Brendel, Piotr; Dłotko, Paweł; Ellis, Graham; Juda, Mateusz; Mrozek, Marian. Computing fundamental groups from point clouds [4].
2. Ellis, Graham; Hegarty, Fintan. Computational homotopy of finite regular CW -spaces [5].
3. Mémoli, Facundo; Zhou, Ling. Persistent Homotopy Groups of Metric Spaces [1].

Com relação a parte computacional, esta dissertação está fundamentada nas referências [2] e [3] já citadas.

As figuras sem fonte neste trabalho foram produzidas utilizando o pacote HAP [3] do GAP, os softwares GeoGebra [6] e Inkscape [7], e capturas de tela do programa GAP [2] e do sistema operacional Linux Ubuntu 5.4.0-58-generic.

2 Preliminares

Neste capítulo apresentaremos conceitos básicos de Topologia, Álgebra e Topologia Algébrica que serão utilizados na dissertação. Além disso, será apresentada uma pequena introdução sobre o programa GAP que é um software que utiliza linguagem de programação escrita em linguagem C que será citado nesta dissertação em diversos momentos.

2.1 Elementos básicos

As definições e resultados expostos neste capítulo são explanadas por Massey [8], Hansen [9], Munkres [10] e Vick [11]. Para este trabalho iremos usar a seguinte terminologia: Para $n \geq 1$,

$$E^n = \{x \in \mathbb{R}^n \mid \|x\| \leq 1\} \text{ (Disco } n\text{-dimensional)};$$

$$U^n = \{x \in \mathbb{R}^n \mid \|x\| < 1\} \text{ (Bola aberta } n\text{-dimensional)};$$

$$S^{n-1} = \{x \in \mathbb{R}^n \mid \|x\| = 1\} \text{ (Esfera } n - 1\text{-dimensional)}.$$

Definição 1. Dada uma relação parcial $\sim'$ (reflexiva, anti-simétrica e transitiva), em um conjunto X , definimos uma relação de equivalência $\sim$ da seguinte forma:

$x \sim y$ se, e somente se, existe uma sequência $x_0, x_1, \dots, x_n$ em X com $x_0 = x$ e $x_n = y$ e para cada i

- $x_{i+1} = x_i$ ou
- $x_{i+1} \sim' x_i$ ou
- $x_i \sim' x_{i+1}$

Dizemos que $\sim$ é uma relação de equivalência gerada por $\sim'$.

Definição 2. Sejam $A \subset X$ e $X \cap Y = \emptyset$ e $f : A \rightarrow Y$ contínua. Considere $X \cup Y$ como um espaço topológico onde X e Y são abertos e fechados em $X \cup Y$. Seja $\sim$ a relação de equivalência em $X \cup Y$ gerada por uma relação parcial, tal que $x \sim f(x)$, para todo $x \in A$. O espaço de identificação $\frac{X \cup Y}{\sim}$ é o **espaço obtido colando X a Y** através de uma aplicação $f : A \rightarrow Y$ e será denotado por $X \cup_f Y$.

Definição 3. Assuma que X^* é um espaço de Hausdorff e que X é um subconjunto fechado de X^* tal que $X^* \setminus X$ é uma união disjunta de abertos e_λ^n , $\lambda \in \Lambda$, Λ um conjunto de índices arbitrário. Cada e_λ^n é homeomorfo a U^n , e são chamados de **n -células** ou **n -células abertas**.

Definição 4. Uma estrutura CW -complexo de um espaço X (que sempre assumimos ser de Hausdorff) é descrita por uma ordenação de uma sequência ascendente de subespaços fechados

$$X^0 \subset X^1 \subset X^2 \subset \dots$$

que satisfazem as seguintes condições:

1. X^0 tem a topologia discreta;
2. Para $n > 0$, X^n é obtido de X^{n-1} pela união de uma coleção de n -células (como explicado na definição 3, visto que temos as condições necessárias);
3. X é a união dos espaços X^i , $i \geq 0$;
4. O espaço X e os subespaços X^q possuem todos a topologia fraca:

Um subconjunto A é fechado em X se, e só se, $A \cap \overline{e^n}$ é fechado para todas as n -células e $n = 0, 1, 2, \dots$

Em outras palavras, podemos dizer que estamos “colando” cada n -célula do CW -complexo e_λ^n a X por uma função característica, isto é, tal que para cada índice $\lambda \in \Lambda$ existe uma aplicação contínua

$$f_\lambda : E^n \rightarrow \overline{e^n}_\lambda,$$

em que f_λ aplica U^n em e_λ^n , de modo homeomorfo, e $f_\lambda(S^{n-1}) \subset X$.

Definição 5. Seja I um conjunto de índices arbitrário. Uma **decomposição celular** de um espaço X é uma família $E = \{e_\alpha : \alpha \in I\}$, de subespaços de X tal que cada e_α é uma célula e

$$X = \bigsqcup_{\alpha \in I} e_\alpha.$$

O **n -esqueleto** de X é o subespaço

$$X^n = \bigsqcup_{\alpha \in J_n} e_\alpha,$$

com $J_n := \{\alpha \in I : \dim(e_\alpha) \leq n\}$. (**Observação:** a notação $\sqcup$ simboliza união disjunta.)

Note que a notação X^n na definição 4 e na definição 5 são coerentes, pois definem o mesmo objeto.

Além destes conceitos topológicos, serão usados os seguintes conceitos provenientes da Álgebra e também explicitadas por Massey [8].

Definição 6. Dizemos que S é um conjunto de geradores do grupo G se todo elemento de G pode ser escrito como um produto de potências de S .

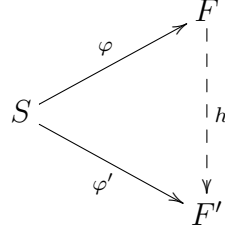
Definição 7. Seja S um conjunto arbitrário. Um grupo livre sobre S (ou gerado por S) é um par constituído por um grupo F e uma função $\varphi : S \rightarrow F$, denotado por (F, φ) tal que as seguintes condições são satisfeitas:

Para todo grupo H e para toda função $\psi : S \rightarrow H$, existe um único homomorfismo $f : F \rightarrow H$ tal que o seguinte diagrama comuta

$$\begin{array}{ccc} & & F \\ & \nearrow \varphi & \vdots \\ S & & \\ & \searrow \psi & \vdots \\ & & H \end{array}$$

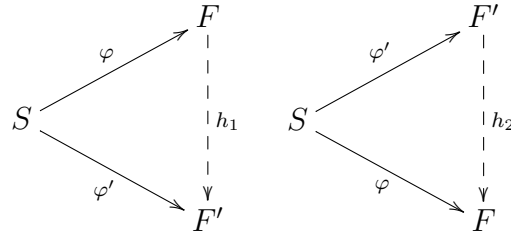
isto é, $f \circ \varphi = \psi$.

Proposição 1. Se (F, φ) e (F', φ') são grupos livres sobre um conjunto S , então, existe um único isomorfismo $h : F \rightarrow F'$ tal que seguinte diagrama comuta



isto é, $h \circ \varphi = \varphi'$.

Demonstração. Pela definição 7, sabemos que um homomorfismo h existe e é único, logo resta provar que h é um isomorfismo. Novamente, pela definição 7 temos os seguintes diagramas



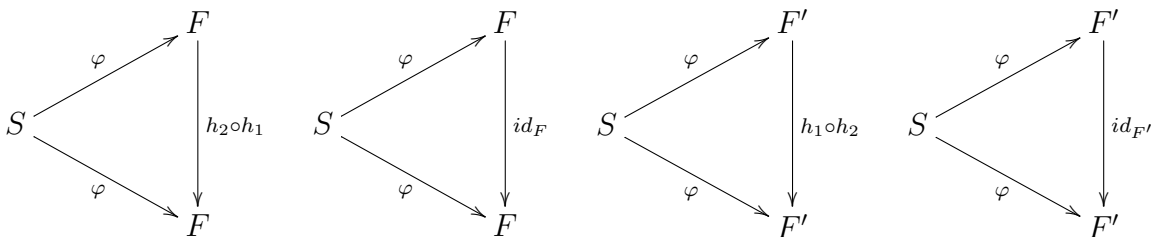
com h_1 e h_2 dois homomorfismos existentes e únicos com a propriedade que

$$h_1 \circ \varphi = \varphi' \quad \text{e} \quad h_2 \circ \varphi' = \varphi \quad (2.1)$$

Pelas equações em (2.1), concluímos que

$$\begin{aligned}
 h_1 \circ (h_2 \circ \varphi') &= \varphi' , \quad h_2 \circ (h_1 \circ \varphi) = \varphi \\
 (h_1 \circ h_2) \circ \varphi' &= \varphi' , \quad (h_2 \circ h_1) \circ \varphi = \varphi
 \end{aligned}$$

por propriedade de homomorfismo, temos que $h_2 \circ h_1 : F \rightarrow F$ e $h_1 \circ h_2 : F' \rightarrow F'$ são homomorfismos, além disso considere $id_F : F \rightarrow F$ e $id_{F'} : F' \rightarrow F'$ os homomorfismos identidade de F e F' respectivamente, pela definição 7, conseguimos construir os seguintes diagramas comutativos.



Por outro lado, pela mesma definição 7 temos que o homomorfismo deve ser único, logo $h_2 \circ h_1 = id_F$ e $h_2 \circ h_1 = id_{F'}$ e com isso, concluímos que h_1 é um isomorfismo. Agora, considerando $h := h_1$ obtemos o resultado desejado. $\square$

Proposição 2. Qualquer grupo é a imagem do homomorfismo de um grupo livre. Mais precisamente, se S é um conjunto qualquer de geradores do grupo G , e F é o grupo livre sobre S , a aplicação inclusão $S \hookrightarrow G$ determina um único epimorfismo de F em G .

Demonstração. Sejam $S \subset G$ um conjunto de geradores de G (pode-se tomar $S = G$), e (F, φ) o grupo livre sobre o conjunto S . Seja $\psi : S \hookrightarrow G$ a aplicação inclusão. Pela definição 7 existe um homomorfismo $f : F \rightarrow G$ tal que o diagrama

$$\begin{array}{ccc} & & F \\ & \nearrow \varphi & \vdots f \\ S & & \\ & \searrow \psi & \downarrow \\ & & G \end{array}$$

comuta. Suponha $S = \{x_i \mid i \in I\}$ em que I é um conjunto de índices qualquer. Como S é um conjunto de geradores de G , dado $g \in G$ temos

$$g = \prod_{i \in I} x_i^{\alpha_i}, \alpha_i \in \mathbb{Z}.$$

Assim, como F é um grupo considere o elemento $y \in F$ tal que

$$y = \prod_{i \in I} \varphi(x_i)^{\alpha_i},$$

desse modo, como f é um homomorfismo,

$$f(y) = f\left(\prod_{i \in I} \varphi(x_i)^{\alpha_i}\right) = \prod_{i \in I} f(\varphi(x_i))^{\alpha_i} = f \circ \varphi(x_i)^{\alpha_i} = \prod_{i \in I} \psi(x_i)^{\alpha_i} = \prod_{i \in I} x_i^{\alpha_i} = g.$$

Logo, temos que f é sobrejetora, então f é um epimorfismo. $\square$

Esta proposição nos dá o significado preciso do termo “relação não trivial entre os geradores”, para este fim, teremos que explorar o fato de que todo subgrupo normal pode ser visto como o núcleo (kernel) de um homomorfismo.

Definição 8. Sejam S o conjunto de geradores do grupo G , (F, φ) o grupo livre sobre conjunto S e $\psi : S \hookrightarrow G$ a aplicação inclusão, e considere f o homomorfismo único tal que $f \circ \varphi = \psi$. Qualquer elemento $r \neq 1$ do núcleo de f é uma **relação** entre os geradores de S do grupo G .

Definição 9. Se $\{r_j\}_j$ é uma coleção de relações e R é uma subcoleção desta relação, então qualquer outra relação r é chamada de **consequência** das relações em R se, e somente se, r está contido no menor subgrupo normal de F que contém as relações R .

O núcleo de f , descrito como na definição 8, é completamente determinado pelo conjunto $\{r_j\}_j$. Note que o núcleo é a intersecção de todos os subgrupos normais de F que contém $\{r_j\}$. Neste caso, o grupo G é completamente determinado por um isomorfismo pelo conjunto de geradores S e o conjunto de relações $\{r_j\}_j$, pois é isomorfo ao quociente de F com o menor subgrupo normal contendo o conjunto $\{r_j\}_j$. Este conjunto de relações, que é o menor subgrupo normal de F contendo $\{r_j\}_j$, é chamado de um conjunto **completo** de relações.

Observação 1. Com as novas informações sobre relação, podemos dar uma nova roupagem a definição 7 da seguinte forma: seja S o conjunto de geradores de G . Dizemos que G é **gerado livremente** por S ou é um **grupo livre** sobre S se não há relações não triviais entre os elementos de S , isto é, as únicas relações possíveis entre os elementos de S são equivalentes aos axiomas de grupo.

Exemplo 1. Se G é um grupo cíclico infinito consistindo de todas as potências do elemento x , então G é um grupo livre sobre o conjunto $S_1 = \{x\}$. Por outro lado, se H é um grupo cíclico finito consistindo nas potências do elemento y , então H não é um grupo livre sobre o conjunto $S_2 = \{y\}$, pois, como H é finito, existe um inteiro positivo n tal que $y^n = e$, em que e representa o elemento identidade do grupo, e esta é uma relação que não é equivalente aos axiomas de grupo.

Definição 10. Uma **apresentação** de um grupo G é um par $(S, \{r_j\}_j)$ consistindo no conjunto de geradores de G e um conjunto de relações entre os geradores que determinam G . A apresentação é **finita** se S e $\{r_j\}_j$ são ambos conjuntos finitos. O grupo G é dito **finitamente apresentado** quando existe pelo menos uma apresentação finita que determina G .

Para mais detalhes sobre apresentação de grupos, consulte [8] e [12].

Agora vejamos algumas definições da Topologia Algébrica explicitadas por Munkres [10] e Vick [11].

Definição 11. Um p -simplexo s em $\mathbb{R}^n$ é a envoltória convexa de uma coleção de $p + 1$ pontos $\{x_0, x_1, x_2, \dots, x_p\}$ no $\mathbb{R}^n$ tal que

$$x_1 - x_0, x_2 - x_0, \dots, x_p - x_0 \text{ é linearmente independente.}$$

A partir de agora consideraremos a definição abaixo como referência para um N -simplexo padrão.

Definição 12. Um N -simplexo padrão Δ^N em $\mathbb{R}^{N+1}$ é uma coleção de pontos definida por

$$\Delta^N := \left\{ (x_0, x_1, \dots, x_N) \mid 0 \leq x_i \leq 1, \sum_{i=0}^N x_i = 1 \right\}.$$

Definição 13. Se f é um caminho em X de x_0 a x_1 e se g é um caminho em X de x_1 a x_2 nós definimos o **produto** $f * g$ de f e g sendo o caminho h dado pelas equações:

$$h(s) = \begin{cases} f(2s); & s \in [0, \frac{1}{2}] \\ g(2s - 1); & s \in [\frac{1}{2}, 1] \end{cases}$$

A função h está bem definida e é contínua, pelo lema da colagem. A função h é um caminho em X de x_0 à x_2 .

A operação produto de caminhos induzidos por f e g é uma operação bem definida nas classes de homotopias de caminhos definida pela equação:

$$[f] * [g] = [f * g]$$

Agora vamos definir a principal estrutura deste trabalho.

Definição 14. Sejam X um espaço e x_0 um ponto de x . O conjunto de classes de homotopia de caminhos de **laços** com base em x_0 , caminhos que começam e terminam em x_0 , com a operação $*$, é chamado de **grupo fundamental** relativo ao ponto base x_0 e é denotado por $\pi_1(X, x_0)$. Quando não há necessidade de explicitar o ponto denotamos por $\pi_1(X)$.

Definição 15. Um espaço X é **simplesmente conexo** se ele é conexo por caminhos e $\pi_1(X, x_0)$ é o grupo unitário formado pelo elemento neutro, para todo $x_0 \in X$.

Por fim, observamos que para calcularmos o grupo fundamental de uma nuvem de pontos é necessário tratar os dados como um **nó**, ou seja, como uma imersão de S^1 no espaço euclidiano $\mathbb{R}^3$ (veja [13]).

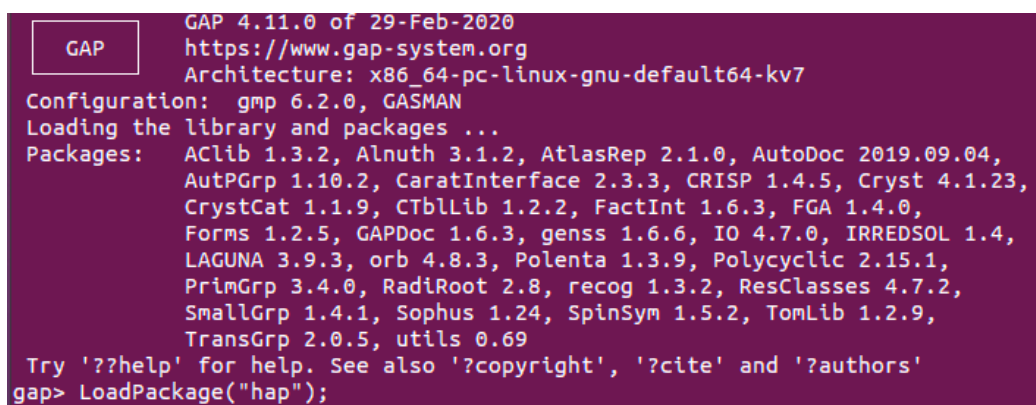
2.2 GAP - Groups Algorithms Programming

Durante todo trabalho usaremos diversas vezes o programa GAP (Groups Algorithms Programming) para a realização de determinados cálculos. Por este motivo, nesta seção queremos dar uma pequena introdução do que é o GAP. Abaixo deixamos um texto sobre “o que é o GAP?” do site oficial referenciado em [2].

GAP é um sistema para álgebra discreta computacional, com ênfase particular na Teoria de Grupo Computacional. GAP fornece uma linguagem de programação, uma biblioteca de milhares de funções que implementam algoritmos algébricos escritos na linguagem GAP, bem como grandes bibliotecas de dados de objetos algébricos. Veja também a visão geral e a descrição das capacidades matemáticas. GAP é usado em pesquisa e ensino de grupos e suas representações, anéis, espaços vetoriais, álgebras, estruturas combinatórias e muito mais. O sistema, incluindo a fonte, é distribuído gratuitamente. Pode-se facilmente modificá-lo ou ampliá-lo para uso particular.

Neste trabalho utilizamos um pacote específico do GAP, a saber, o pacote “HAP - Homological Algebra Programming” para os devidos cálculos.

Figura 2.1: GAP (Groups, Algorithms, Programming)



```
GAP 4.11.0 of 29-Feb-2020
https://www.gap-system.org
Architecture: x86_64-pc-linux-gnu-default64-kv7
Configuration: gmp 6.2.0, GASMAN
Loading the library and packages ...
Packages:  Aclib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04,
AutPGrp 1.10.2, CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23,
CrystCat 1.1.9, CTbllib 1.2.2, FactInt 1.6.3, FGA 1.4.0,
Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0, IRREDSOL 1.4,
LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9, Polycyclic 2.15.1,
PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2,
SmallGrp 1.4.1, Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9,
TransGrp 2.0.5, utils 0.69
Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("hap");
```

Fonte: Gerada pelo software [2].

Na figura 2.1, temos uma captura de tela da interface inicial do GAP, além disso temos o primeiro comando “LoadPackage(“hap”)”, que é justamente a “leitura” do

pacote HAP da biblioteca do GAP.

Para uma maior familiaridade com o programa, vamos expor alguns cálculos extremamente simples no GAP para entendermos como os comandos funcionam.

```
gap> 3+4;  
7  
gap> a:=5*6;  
30  
gap> b:=2^(a/3);a;  
1024  
30
```

Destacamos que:

- ◆ O comando na linha *gap>* representa o “input”, isto é, o comando a ser inserido. O texto que se encontra imediatamente abaixo da linha *gap>* representa o “output”, isto é, o resultado que o programa retorna como resultado ao comando inserido.
- ◆ O ponto e vírgula (;) representa o fim do comando, caso necessário, pode-se realizar um novo comando após o ponto e vírgula.
- ◆ O comando *:=* representa um elemento a ser definido. No exemplo acima definimos *a* como sendo 30 e *b* como sendo 1024 e o programa salva esta informação até o seu encerramento ou até que um novo valor seja definido para *a* ou *b*.

Para uma introdução mais detalhada de como usar o GAP recomendamos a leitura da referência [14].

3 CW -espaços

Neste trabalho tratamos especificamente dos CW -espaços, que são espaços topológicos que possuem uma CW -decomposição. Um CW -espaço na realidade é o espaço subjacente de um CW -complexo. Por esta razão explanamos neste capítulo diversas definições e exemplos desses espaços que serão úteis ao desenvolvimento dessa dissertação. As principais referências deste capítulo são os artigos [4], [5] e o livro [15].

Dada a relevância dos CW -espaços nesta dissertação, apresentaremos conceitos e resultados que serão utilizados, baseados no artigo [5]. Na realidade, estamos interessados em CW -espaços regulares finitos.

Definição 16. Dizemos que um CW -espaço X é **regular** se toda célula é colada por uma aplicação que restrita ao bordo da célula é um homeomorfismo. Além disso, dizemos que X é **finito** se possui uma quantidade finita de células.

Definição 17. Definimos o **fecho** de uma célula e_j^k em X como o menor CW -subespaço fechado $\overline{e_j^k}$ de X contendo e_j^k . De forma mais geral, definimos o fecho de um subconjunto $Y \subset X$ como o menor CW -subespaço $\bar{Y}$ em X contendo Y .

Definição 18. Dizemos que uma célula e_j^n é **maximal** se não pertence ao fecho de nenhuma célula $(n + 1)$ -dimensional.

Definição 19. Um **complexo simplicial** é um par $K = (X, \Phi)$ dado por conjunto finito X e um conjunto Φ de subconjuntos de X , não vazios, tais que:

- $(\forall x \in X), \{x\} \in \Phi,$
- $(\forall \sigma \in \Phi)(\forall \sigma' \subset \sigma, \sigma' \neq \emptyset), \sigma' \in \Phi.$

Os elementos de X são os vértices de K . Os elementos de Φ são os simplexes de K . Se σ é um simplexo de K então todo $\sigma' \subset \sigma$ não vazio é uma face de σ . Um simplexo

σ com $n + 1$ elementos é um n -simplexo e dizemos que é um simplexo de dimensão n . A dimensão de K é a dimensão maximal de seus simplexos.

Definição 20. Um espaço celular n -dimensional é **puro** se qualquer k -célula com $k < n$ está no bordo de alguma $(k + 1)$ -célula.

Há várias maneiras para uma representação de um complexo simplicial puro em um software, mas vamos considerar como um CW -espaço regular devido a sua eficiência no cálculo de grupos fundamentais.

Definição 21. Chamamos um CW -complexo X de **complexo tesselado**, ou **espaço tesselado**, se para algum $n \geq 0$, tem-se que:

1. X é uma união dos fechos de suas n -células;
2. todos os fechos de n -células têm um conjunto parcialmente ordenado de faces isométricas a determinados polítopos fixados.

Nesse caso, dizemos que tal espaço tem **dimensão** n .

Entendemos por **polítopos** ou **n -polítopos** todo objeto geométrico de dimensão n com faces, isto é, com “lados planos”. Polítopos são generalizações do que temos para polígonos e poliedros. Desse modo, polígonos são polítopos 2-dimensionais e poliedros são polítopos 3-dimensionais.

Estes polítopos fixados na definição 21 podem ser, por exemplo, n -cubos, n -permutaedros (Figura 3.1) ou n -simplexos. Falaremos mais sobre eles mais adiante no trabalho.

Definição 22. O fecho de qualquer n -célula é uma **faceta** do complexo tesselado. Uma faceta de um polítopo de dimensão $n + 1$ é uma faceta de dimensão n .

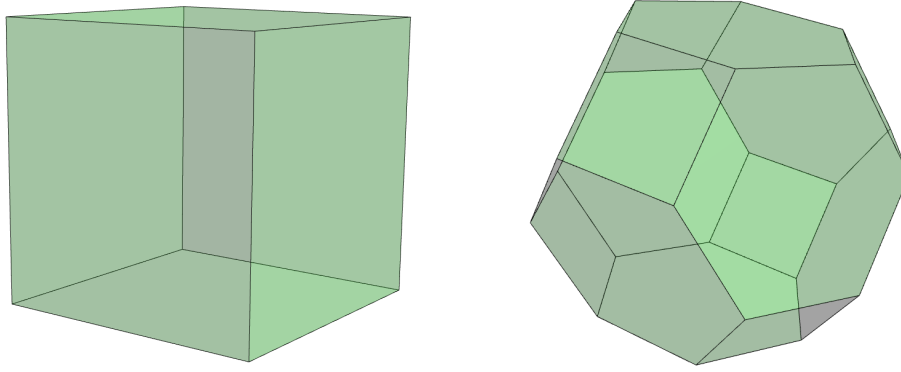
Exemplo 2. Uma faceta de um triângulo é um de seus três lados. Do mesmo modo, dizemos que uma faceta de um cubo 3-dimensional é uma de suas seis faces.

Note que, a grosso modo, a faceta é uma generalização de “face” ou “lado” para um polítopo.

Definição 23.

1. Dizemos que L é um **reticulado** em $\mathbb{R}^n$ se $L \subseteq \mathbb{R}^n$ é um subgrupo aditivo gerado por n vetores linearmente independentes. Denotamos por V_L o conjunto desses geradores.

Figura 3.1: Um cubo (esquerda) e um permutaedro (direita).



Fonte: Gerada pelo software [2].

2. Uma **célula Dirichlet–Voronoi** é um conjunto denotado por $D_L(v)$, em que:

$$D_L(v) := \{x \in \mathbb{R}^n : \|v - x\| \leq \|w - x\| \text{ para todo } w \in L, w \neq v\}$$

Denotamos $D_L(0)$ somente por D_L , em que 0 é o vetor nulo.

O espaço euclidiano $\mathbb{R}^n$ herda a estrutura de um espaço tesselado tendo as células $D_L(v)$ como facetas. Denotamos este espaço tesselado por $\mathbb{R}_L^n$.

Com as definições acima temos alguns reticulados particulares. Um reticulado C gerado por n vetores ortogonais, os polítopos, ou facetas, do espaço $\mathbb{R}_C^n$ em mosaico (em xadrez) são combinatoriamente, equivalentes a um n -**cubo**.

Exemplo 3. Considere $V_L := \{(1, 0); (0, 1)\}$, isto é, $L = \mathbb{Z}^2$. Vejamos como seria $D_L(2, 2)$. Para tanto, veja que este V_L só precisamos comparar o vetor $(2, 2)$ com outros 4 vetores, a saber, $(1, 2)$, $(2, 1)$, $(2, 3)$ e $(3, 2)$. Como a base é ortogonal, a célula Dirichlet–Voronoi seria construída por mediatrizes. Veja a figura abaixo.

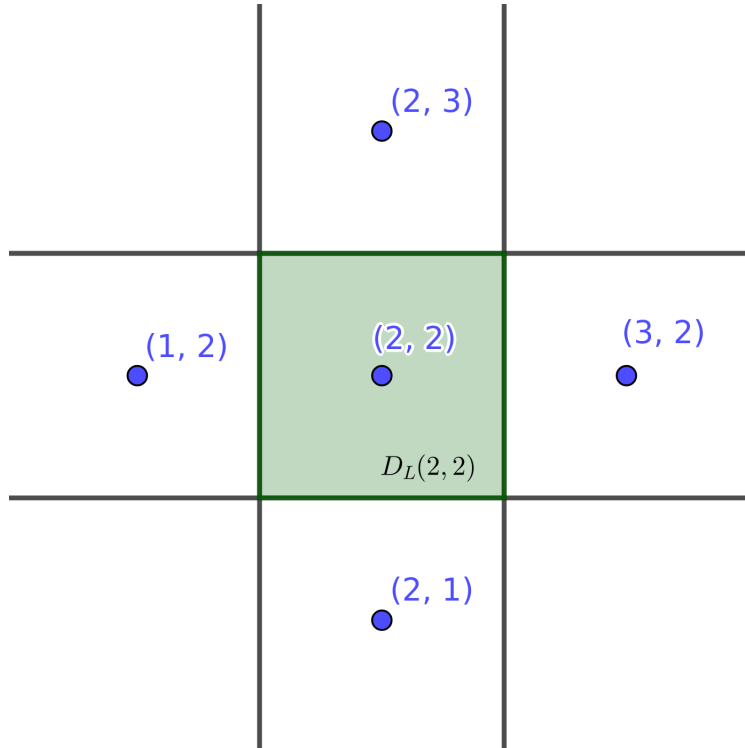
Note que a célula Dirichlet–Voronoi é um quadrado devido ao fato dos vetores de V_L serem ortogonais.

Para descrever um segundo reticulado relevante, identificamos $\mathbb{R}^n$ com o hiperplano em $\mathbb{R}^{n+1}$, composto por todos os vetores cujas coordenadas somadas são zero, isto é:

$$\mathbb{R}^n = \{(x_1, \dots, x_{n+1}) \in \mathbb{R}^{n+1} : x_1 + \dots + x_{n+1} = 0\}.$$

Seja P o grupo abeliano neste hiperplano gerado pelas colunas da matriz $(n+1) \times (n+1)$

Figura 3.2: Construção da célula Dirichlet–Voronoi $D_L(2, 2)$.

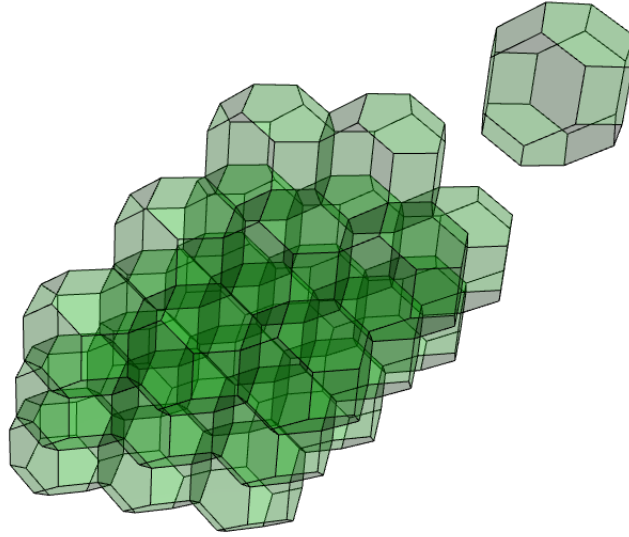


Fonte: Gerada pelo software [2].

$$\begin{pmatrix} -n & 1 & 1 & \cdots & 1 \\ 1 & -n & 1 & \cdots & 1 \\ 1 & 1 & -n & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & -n \end{pmatrix}$$

Os polítopos do espaço $\mathbb{R}_P^n$ em mosaico são combinatorialmente equivalentes ao polítopo convexo n -dimensional conhecido como **permutaedro**. Veja figura.

Figura 3.3: Permutaedros em mosaico.



Fonte: Gerada pelo software [2].

Na figura 3.3 separamos uma célula Dirichlet–Voronoi para melhor visualização do mosaico e para termos uma visão clara que, de fato, é um mosaico de permutaedros.

Outras opções de V_L podem levar a outros polítopos, como prismas. Um cubo tridimensional e um permutaedro tridimensional são ilustrados na figura 3.1.

Definição 24. Seja Λ um subconjunto finito qualquer de L . A união finita de polítopos X definida por,

$$X := \bigcup_{w \in \Lambda} D_L(w),$$

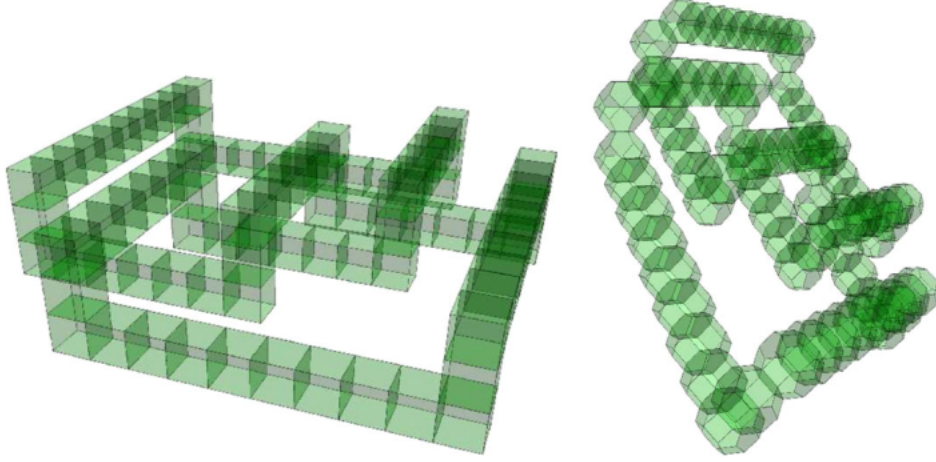
chamamos de **complexo reticulado** ou **L -complexo**.

Um complexo reticulado é chamado de **cúbico** se $L = C$ e **permutaedral** se $L = P$, com C e P definidos anteriormente.

A figura 3.4 mostra um complexo cúbico puro tridimensional e um complexo permutaedral puro tridimensional, ambos no modelo nó do trevo.

Os complexos reticulados admitem uma representação computacional que é mais eficiente que a representação de conjuntos parcialmente ordenados de polítopos para CW -complexos regulares gerais. Dado um reticulado L em $\mathbb{R}^n$, podemos escolher uma base $v_1, \dots, v_n \in L$ e representar qualquer $v \in L$ como uma combinação única $v = \lambda_1 v_1 + \dots + \lambda_n v_n$ com coordenadas inteiras λ_i .

Figura 3.4: Modelos do complexo puro cúbico e permutaedral do nó do trevo.



Fonte: Artigo [4]

Seja X o complexo reticulado em mosaico que surge como uma união finita de diversos polítopos em $\mathbb{R}_L^n$. Aplicando uma translação, se necessário, podemos assumir que X é uma união de polítopos $D_L(v)$ em que v tem apenas coordenadas inteiras positivas λ_i com relação à base escolhida.

Podemos representar X por uma matriz binária n -dimensional $A = (a_{\lambda_1, \lambda_2, \dots, \lambda_n})$ com

$$a_{\lambda_1, \lambda_2, \dots, \lambda_n} = \begin{cases} 1 & \text{se } D_L(\lambda_1 v_1 + \dots + \lambda_n v_n) \text{ pertence a } X, \\ 0 & \text{caso contrário.} \end{cases}$$

Nossa representação computacional de X também precisa armazenar algumas informações sobre o reticulado L . Para nossos algoritmos, as informações extras mais apropriadas podem ser armazenadas como um conjunto finito de vetores inteiros, a saber:

Definição 25. Seja $v \in L$, o conjunto $B_L(v)$ é definido por

$$B_L(v) := \{\lambda = (\lambda_1, \dots, \lambda_n) \in \mathbb{Z}^n : D_L(v) \cap D_L(\lambda_1 v_1 + \dots + \lambda_n v_n) \neq \emptyset\},$$

chamamos $B_L(v)$ de L -**bola** de v . Quando não for necessário especificar o vetor denotaremos simplesmente por B_L . Resumidamente a L -bola $B_L(v)$ é um conjunto de vetores que corresponde as células adjacentes de $D_L(v)$.

A matriz binária $A = (a_{\lambda_1, \lambda_2, \dots, \lambda_n})$ é frequentemente chamada de representação

bitmap. Para $L = C$ a L -bola contém $3^n - 1$ vetores. Para $L = P$ a L -bola contém $2^{n+1} - 2$ vetores.

Exemplo 4. Considere $n = 3$, dessa forma:

1. Quando $L = C$, a L -bola contém 26 vetores, que representa a soma de faces, arestas e vértices de um cubo 3-dimensional.
2. Quando $L = P$, a L -bola contém 14 vetores que representa o número de faces de um permutaedro 3-dimensional.

4 Homologia de Persistência no HAP

Reservamos este capítulo para apresentar exemplos de como utilizar o pacote HAP do GAP para calcular homologias de persistência. Os exemplos em questão, podem ser encontrados no artigo [5]. Para tanto, precisamos da seguinte definição:

Definição 26. Seja X um complexo tesselado. Dizemos que uma faceta e de X é **vizinha** de uma faceta f de X se $e \cap f \neq \emptyset$. Em particular, definimos a **vizinhança** de f , $N_X(f)$, pela união de todos os vizinhos de f , em outras palavras,

$$N_X(f) := \bigcup_{i \in I} e_i ; \quad e_i \cap f \neq \emptyset, '$$

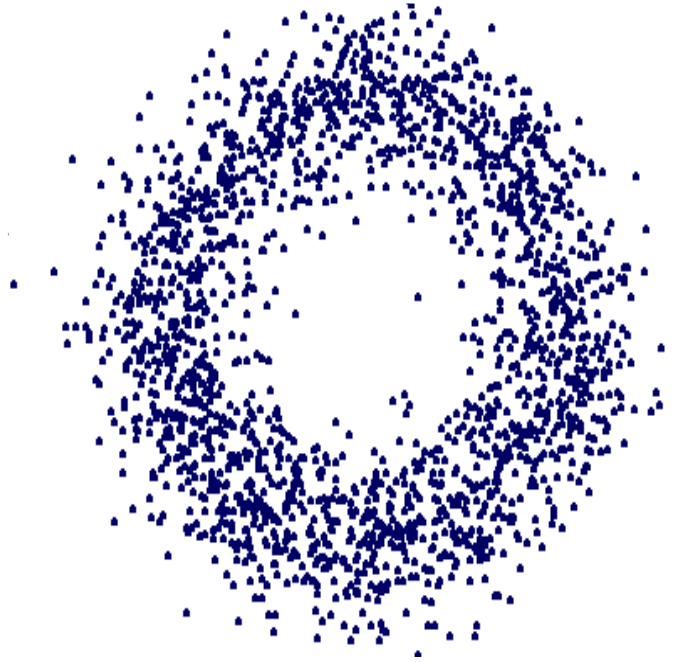
com I um conjunto de índices.

Os espaços reticulados surgem naturalmente na topologia aplicada. Como uma primeira ilustração, consideramos dois exemplos que chamaremos de exemplo brinquedo (Toy example) e exemplo dos parafusos. Para estes exemplos, estamos interessados em saber que informação conseguimos através das ferramentas de Homologia de Persistência.

4.1 Primeiro exemplo

Para o exemplo brinquedo, suponhamos que $M \subset \mathbb{R}^2$ e consideramos a amostra aleatória S de 43935 pontos distintos $(x, y) \in \mathbb{N}^2$ representado na figura 4.1.

Figura 4.1: Amostra de pontos S sobre uma variedade $M \subset \mathbb{R}^2$ desconhecida.



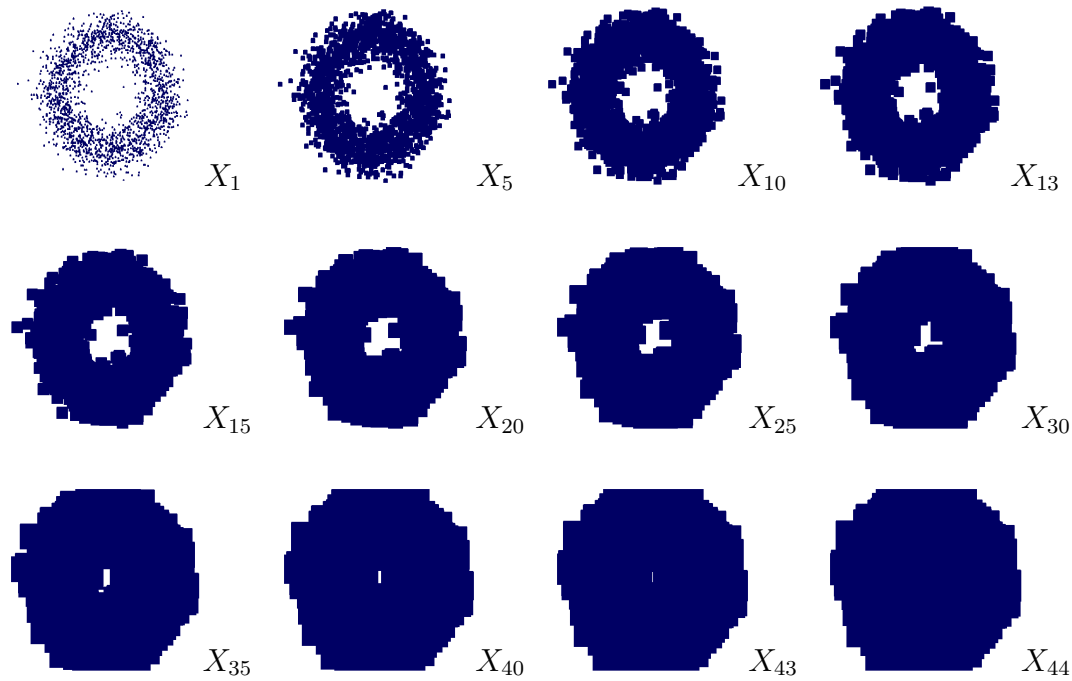
Fonte: referências [5, 16].

No que segue, o objetivo é respondermos a seguinte pergunta: o que podemos inferir sobre a topologia de M ? O número de pontos é dado pelo GAP através dos seguintes comandos:

```
gap> S:=ReadImageAsPureCubicalComplex("datacloud.eps",100000);
Pure cubical complex of dimension 2.

gap> Size(S);
43935
```

Seja C o reticulado gerado por $(1, 0)$ e $(0, 1)$. Associamos a S o espaço reticulado X_1 consistindo dessas facetas $D_C(v)$ de $\mathbb{R}_C^2$ com $v \in S$. Uma abordagem natural para o entendimento da variedade M é gerar uma sequência $X_1 \subset X_2 \subset X_3 \subset \dots$ de sucessivas vizinhanças $X_{i+1} = N_{\mathbb{R}_C^2}(X_i)$ e procurar por propriedades topológicas “persistentes” na sequência. Muitos desses X_i são ilustrados na figura 4.2.

Figura 4.2: Algumas vizinhanças cúbicas do conjunto S .

Fonte: Gerada pelo software [2].

Para realizar a filtração em questão no GAP utilizamos os seguintes comandos:

```
gap> T:=ThickeningFiltration(S,45);
Filtered pure cubical complex of dimension 2.

gap> I:=[1..45];
[ 1 .. 45 ]
gap> Apply(I,i->FiltrationTerm( T, i ));

gap> X1:=ViewPureCubicalComplex(I[1]);

gap> X5:=ViewPureCubicalComplex(I[5]);

gap> X44:=ViewPureCubicalComplex(I[44]);
```

O comando “ViewPureCubicalComplex(I[.])” permite a visualização das imagens na figura 4.2.

A expectativa é que as propriedades persistentes reflitam as propriedades da varie-

dade M desconhecida. O i -ésimo **número de Betti** $\beta_i(X)$ é o posto do subgrupo livre de torção da homologia $H_i(X, \mathbb{Z})$ e estão registrados nos comandos do GAP a seguir para esta sequência de espaços.

```
gap> Apply(I,x->BettiNumber( x, 0 ));
gap> I;
[ 478, 206, 107, 57, 28, 24, 15, 12, 10, 7, 5, 4, 4, 2, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1 ]

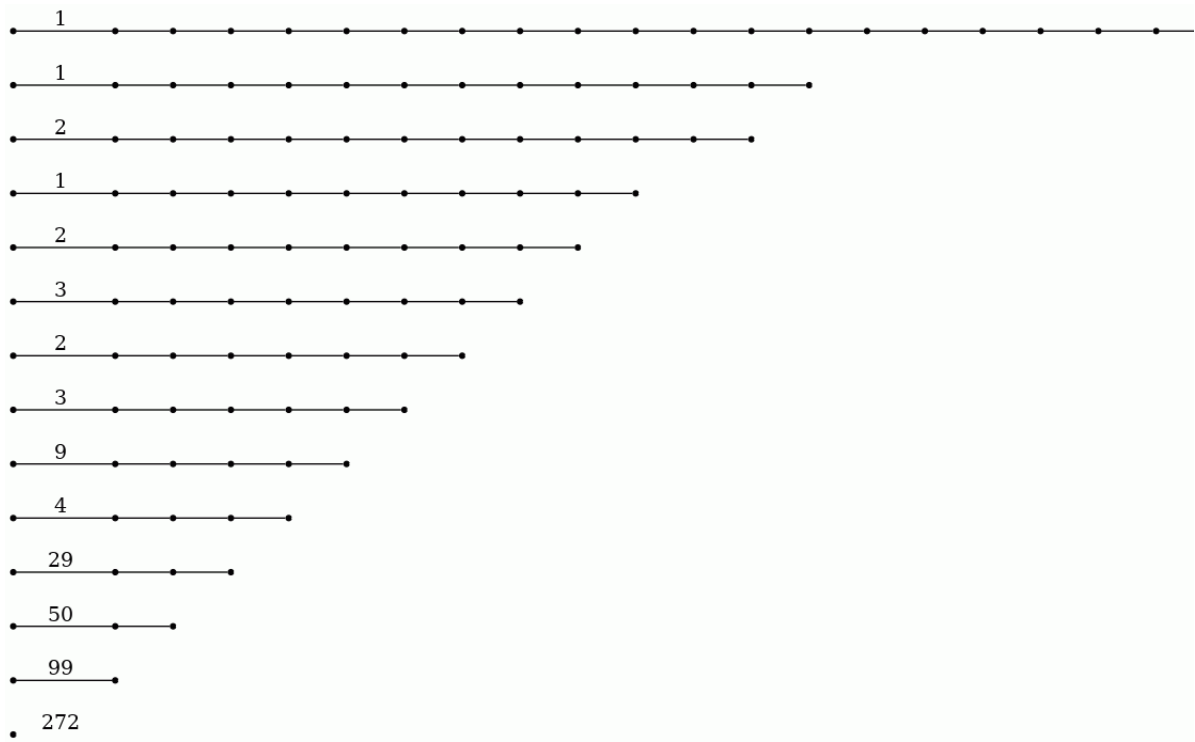
gap> J:=[1..45];;
gap> Apply(J,i->FiltrationTerm( T, i ));;
gap> Apply(J,x->BettiNumber( x, 1 ));;
gap> J;
[ 80, 153, 222, 177, 112, 66, 42, 29, 22, 17, 10, 5, 3, 4, 4,
  4, 2, 2, 1, 1, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1,
  1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 1, 0, 0 ]
```

Os números acima fornecem apenas uma indicação parcial da persistência homológica. Eles indicam, por exemplo, que $H_1(X_{10}, \mathbb{Z})$ é gerado por 17 ciclos independentes e $H_1(X_{15}, \mathbb{Z})$ é gerado por 4 ciclos. Mas eles não indicam quantos dos 17 ciclos estão no núcleo do homomorfismo induzido $H_1(X_{10}, \mathbb{Z}) \rightarrow H_1(X_{15}, \mathbb{Z})$.

Para melhor capturar os ciclos da homologia de persistência, definimos a **persistência dos números de Betti** β_n^{ij} . Para $i \leq j$, definimos β_n^{ij} igual ao posto de um subgrupo livre de torção da imagem do homomorfismo induzido $H_n(X_i; \mathbb{Z}) \rightarrow H_n(X_j; \mathbb{Z})$. Para $i > j$, define-se $\beta_n^{ij} = 0$.

Representamos a matriz (β_n^{ij}) por um gráfico, chamado **barcode** com linhas horizontais e vértices dispostos em coluna. A i -ésima coluna tem $\beta_n^{ii} = \beta_n(X_i)$ vértices. Existem β_n^{ij} linhas da i -ésima coluna à j -ésima coluna.

Quando um barcode tem muitas linhas é conveniente representar uma coleção de n linhas horizontais com a coluna inicial e final comuns como uma única linha rotulada pelo inteiro n . Dizemos que tal barcode está na **forma compacta**. Veja um exemplo de barcode na forma compacta na figura 4.3 referente ao exemplo brinquedo.

Figura 4.3: β_0 barcode dos instantes 0 ao 20 dos espaços da figura 4.2

Fonte: Gerada pelo software [2].

Note que se o barcode não estivesse na forma compacta ele deveria ter um total de 478 linhas. Além disso a primeira linha persiste até o instante 45.

O barcode β_0 nos mostra que pontos tendem a se aglutinar em um único ciclo a partir do instante 15, isto é, os pontos não estão aglomerados em duas regiões distantes. Além disso, o barcode nos mostra que os pontos em geral estão próximos mas não estão colados, caso contrário teríamos uma aglutinação em um instante menor do que 15.

Perceba também que β_1 torna-se 0 no instante 44 que é justamente quando o conjunto se torna um espaço contrátil.

Segundo [5], existe uma única linha horizontal longa no β_1 barcode dos instantes 5 ao 43 que indicam que há um único gerador de homologia 1-dimensional que persiste de X_5 a X_{43} . O palpite é, portanto, que o conjunto S foi amostrado aleatoriamente de uma variedade $M \subset \mathbb{R}^2$ com $H_1(M, \mathbb{Z}) = \mathbb{Z}$ o que é coerente, pois coincide com a homologia da S^1 .

4.2 Segundo exemplo

Como uma segunda ilustração do uso de espaços reticulados na topologia aplicada, vamos considerar imagens digitais. A imagem digital da figura 4.4 pode ser representada como um espaço reticulado cúbico 5-dimensional cujas facetas estão centradas nos vetores inteiros (x, y, r, g, b) com x, y as coordenadas euclidianas de um pixel, e com r, g, b inteiros no intervalo de 0 a 255 que representam as cores vermelha, verde e azul, respectivamente do pixel.

Figura 4.4: Imagem digital dos parafusos.



Fonte: imagem editada de [5].

Um espaço reticulado cúbico bidimensional Y_t pode ser obtido escolhendo um número limitante t e incluindo uma faceta 2-dimensional centrada em (x, y) para cada pixel com $r + g + b \leq t$ (a cor branca tem $r = 255, g = 255, b = 255$). Para calcular o número $\beta_0(Y_t)$ das componentes conexas por caminho de Y_t para vários limitantes t utilizaremos o algoritmo 1 descrito abaixo.

Algoritmo 1.

Entrada: Um espaço tesselado finito X representado de forma que as vizinhanças das facetas podem ser facilmente computadas.

Saída: Retorna uma lista $X_1, \dots, X_c$ de componentes conexas por caminho de X .

Procedimento: Inicie com $c := 0$;

desse modo, todas as facetas de X serão incolor e nenhuma faceta é colorida;

enquanto existir facetas de X incolores, faça

defina um novo c por $c := c + 1$;

escolha alguma faceta descolorida e atribua a ela a cor c ;

enquanto existir uma faceta incolor f na vizinhança de alguma faceta de cor c

atribua a cor c a f ;

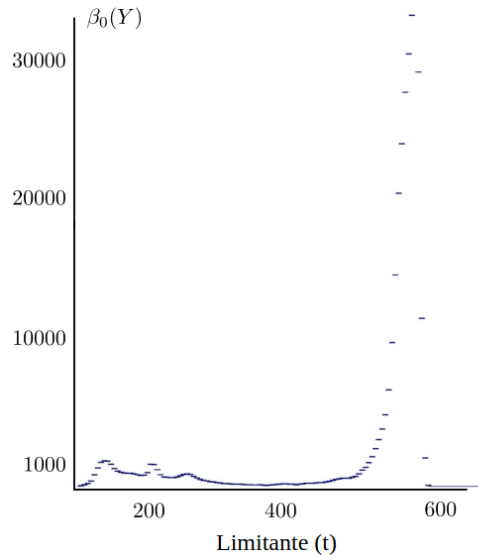
para $0 \leq i \leq c$ seja X_i a união de todas as facetas de cor i ;

A figura 4.5 mostra um gráfico do conjunto

dos limitantes $T := \{t \in \mathbb{Z} : 0 \leq t \leq 620\}$ Figura 4.5: Gráfico de T cartesiano B cartesiano $B := \{\beta_0(Y_t) : t \in T\}$.

Para limitantes próximos de $t = 300$, o valor de $\beta_0(Y_t)$ é praticamente estável. Conforme o limitante t aumenta, novas componentes conexas por caminho nascem e, ao mesmo tempo, grupos de componentes conexas por caminho existentes distintos se aglutinam para formar uma única componente conexa por caminho. O gráfico de $\beta_0(Y_t)$, portanto, aumenta e depois diminui. O número $\beta_0^{st}(Y)$ de componentes conexas por caminho que persistem de Y_s a Y_t pode ser obtido usando o Algoritmo 1 para determinar o número de componentes de Y_t que se intersectam de forma não trivial com o espaço Y_s .

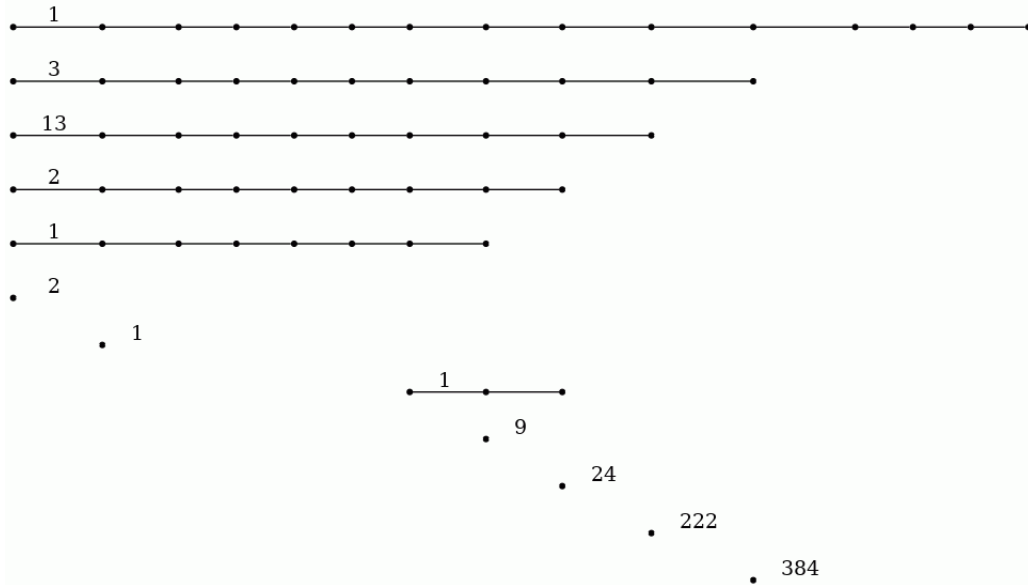
Para $|t - s|$ grande esta deve ser uma estimativa robusta do número de objetos na imagem que seriam detectados pelo olho humano. Para exemplificar, vejamos um



Fonte: artigo [5]

cálculo computacional que determina barcode da figura 4.6. Note que as 20 linhas mais longas do barcode representam os 20 objetos que são vistos na imagem.

Figura 4.6: β_0 barcode da figura 4.4.



Fonte: Gerada pelo software [2].

Os exemplos acima nos mostraram as informações que conseguimos utilizando homologia de persistência, mas o sentimento que fica é que poderíamos obter resultados ainda mais sofisticados e detalhados sobre as variedades estudadas. Por esta razão, estamos procurando encontrar uma aplicação do grupo fundamental na topologia aplicada.

5 Retratos por deformação

Neste capítulo abordamos um tipo específico de ‘retrato por deformação’, a saber, o retrato por zig-zag, a partir das referências [4] e [5]. Inicialmente é necessário relembrarmos algumas definições de topologia.

Definição 27. Um subconjunto A de um espaço topológico X é chamado de um **retrato** de X se existe uma aplicação contínua $r : X \rightarrow A$, chamada de **retração**, tal que $r(a) = a$ para todo $a \in A$.

Definição 28. Um subconjunto A de X é um **retrato por deformação** de X se existe uma retração $r : X \rightarrow A$ e uma homotopia $f : X \times I \rightarrow X$ tal que, para quaisquer $x \in X$, $a \in A$ e $t \in I$, temos:

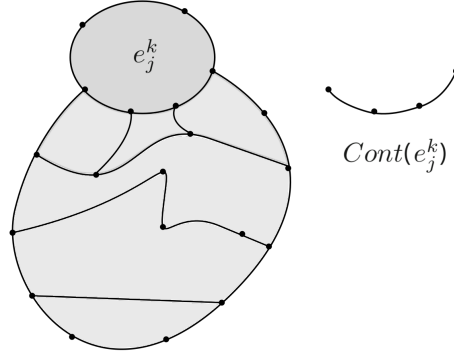
$$\begin{aligned}f(x, 0) &= x \\f(x, 1) &= r(x) \\f(a, t) &= a.\end{aligned}$$

Definição 29.

1. Definimos o **complexo contato** de uma célula maximal e_j^k em X pelo CW -espaço $Cont(e_j^k) = \overline{e_j^k} \cap \overline{X \setminus e_j^k}$ que é a interseção entre o fecho da célula e_j^k e o fecho do complementar (veja figura 5.1).
2. Uma célula maximal e_j^k em X é **redundante** se o complexo contato $Cont(e_j^k)$ é um CW -espaço contrátil.

Proposição 3. Se e_j^k é uma célula redundante maximal no CW -espaço regular finito X então a aplicação inclusão $\overline{X \setminus e_j^k} \hookrightarrow X$ é uma equivalência de homotopia, e $\overline{X \setminus e_j^k}$ é um retrato por deformação de X .

Figura 5.1: Uma célula e_j^k e seu complexo contato $Cont(e_j^k)$



Fonte: Gerada pelo software [7].

Demonstração. Suponha que $A = \overline{X \setminus \overline{e_j^k}}$ e $\varphi : A \hookrightarrow X$ a aplicação inclusão. Como e_j^k é uma célula maximal redundante, temos que $Cont(e_j^k)$ é contrátil. Além disso, $\overline{e_j^k}$ também é contrátil, logo como $Cont(e_j^k) \subset \overline{e_j^k}$ existe uma aplicação contínua $h_1 : \overline{e_j^k} \rightarrow Cont(e_j^k)$ que é uma equivalência de homotopia. Desse modo, sabemos que existem uma aplicação inclusão $i_1 : Cont(e_j^k) \hookrightarrow \overline{e_j^k}$ e uma homotopia $H : \overline{e_j^k} \times I \rightarrow \overline{e_j^k}$ que satisfaz as seguintes condições

$$H(x, 0) = Id_{\overline{e_j^k}}(x)$$

$$H(x, 1) = i_1 \circ h_1(x) = h_1(x).$$

Considere $h_2 : A \rightarrow A$ a aplicação identidade. Pelo lema da colagem existe uma aplicação contínua $\psi : X \rightarrow A$ definida por

$$\psi(x) = \begin{cases} h_1(x), & \text{se } x \in \overline{e_j^k}, \\ h_2(x), & \text{se } x \in A. \end{cases}$$

Dessa forma,

$$\psi \circ \varphi(x) = \psi(\varphi(x)) = \psi(x) = h_2(x) = x, \forall x \in \overline{X \setminus \overline{e_j^k}}.$$

Logo $\psi \circ \varphi = Id_A \sim Id_A$.

Resta provar que existe uma homotopia $G : X \times I \rightarrow X$ tal que

$$G(x, 0) = Id_X(x)$$

$$G(x, 1) = \psi(x)$$

Note que a aplicação $G_1 : A \times I \rightarrow X$ definida por $G_1(x, t) = x$ é uma homotopia. Resta provar que existe uma homotopia $G_2 : \overline{e_j^k} \times I \rightarrow X$. Assim, queremos definir G_2 de modo que esta seja contínua e satisfaça as seguintes condições

$$\begin{aligned} G_2(x, 0) &= Id_{\overline{e_j^k}}(x) \\ G_2(x, 1) &= \psi(x) = h_1(x) = i_1 \circ h_1(x) \end{aligned}$$

Assim defina $G_2(x, t) = H(x, t)$. Logo, G_2 é uma homotopia e pelo lema da colagem conseguimos construir uma $G : X \times I \rightarrow I$ tal que G é contínua, $G(x, 0) = Id_X$ e $G(x, 1) = \psi(x)$, ou seja, G é uma homotopia.

Portanto φ é uma equivalência de homotopia.

Além disso, note que pela homotopia G e pela retração ψ , A é um retrato por deformação de X . $\square$

5.1 Retrato zig-zag

A seguir apresentamos dois algoritmos úteis que determinam um retrato por deformação e que são imprescindíveis para a compreensão da definição 30 de retrato zig-zag.

Algoritmo 2.

Entrada: Um CW -espaço regular finito X , um CW subespaço $A \subset X$.

Objetivo: Encontrar um subespaço CW (mínimo) $X' \subset X$ que seja um retrato por deformação de X e que contém A .

Procedimento: Inicie com $X' := X$;

encontre uma célula maximal redundante e_j^k em X' que não pertence a A ;

defina um novo X' por $X' := \overline{X' \setminus e_j^k}$.

Algoritmo 3.

Entrada: Um CW -espaço finito X , um CW subespaço $A \subset X$ não vazio.

Objetivo: Encontrar um subespaço CW (máximo) $X' \subset X$ que contenha A como um retrato por deformação.

Procedimento: Inicie com $X' := A$;

encontre uma célula e_j^k em X que seja maximal redundante em $X' \cup \overline{e_j^k}$;

defina um novo X' por $X' := X' \cup \overline{e_j^k}$.

Os algoritmos acima serão de extrema importância para o entendimento da definição a seguir.

Definição 30. Dizemos que um espaço tesselado X' é um **retrato zig-zag**, ou **retrato por deformação em zig-zag**, de um espaço tesselado X se existe uma sequência de retratos de deformação

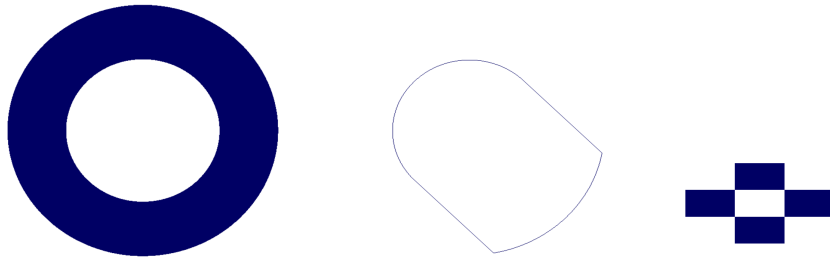
$$X = B_0 \leftarrow A_1 \hookrightarrow B_1 \leftarrow A_2 \hookrightarrow B_2 \leftarrow \cdots \hookrightarrow B_{k-1} \leftarrow A_k = X'$$

Os dois Algoritmos 2 e 3 podem ser usados para encontrar os retratos por deformação que são potencialmente menores que os atuais retratos. Para produzir um retrato por zig-zag de um espaço tesselado $X = B_0$ podemos usar de modo iterado os algoritmos 2 e 3. Primeiramente podemos usar o Algoritmo 2 sobre B_0 para produzir um retrato A_1 . Desse modo, construímos um pequeno espaço delimitador $A_1 \subset W_1$. Então, usando o Algoritmo 3, podemos calcular um subespaço $B_1 \subset W_1$ que é máximo em relação a conter A_1 como retrato por deformação. Podemos então usar o Algoritmo 2 para encontrar um retrato $A_2 \subset B_1$. Embora A_{n+1} tenha menos facetas do que A_n , na maioria das vezes, esse processo pode ser repetido até alcançar um X' desejado, ou até alcançarmos um índice m tal que se $n > m$ temos que $A_n = A_m$.

A razão por trás deste método é justamente simplificar os cálculos computacionais, assim não necessitamos de uma super máquina para realizá-los. Muitas das vezes os espaços tesselados X' são muito menores que X e tem o mesmo tipo de homotopia, mas não estão contidos em X . O exemplo 5 a seguir, nos mostra esta condição e nos revela como o retrato por zig-zag é uma ferramenta extremamente eficiente.

Exemplo 5. Um exemplo bem simples e conhecido de retrato zig-zag é o caso do anel com um raio interno largo, qualquer retrato por deformação nos fornecerá uma imagem do mesmo tipo de homotopia que um círculo (S^1) com “raio” relativamente grande, por outro lado o anel possui o mesmo tipo de homotopia que um círculo de raio muito pequeno. A seguir podemos ver um exemplo de um anel, do seu retrato por deformação e do seu retrato zig-zag (imagem ampliada) respectivamente.

Figura 5.2: Retrato zig-zag de um anel.



Fonte: Gerada pelo software [2].

Utilizando o GAP podemos ver claramente a diferença da quantidade de facetas que temos entre cada uma das imagens.

```
gap> L:=ReadImageAsPureCubicalComplex("annulus.bmp", 3000);
Pure cubical complex of dimension 2.

gap> Size(L);
162833

gap> M:=ContractedComplex(L);
Pure cubical complex of dimension 2.

gap> Size(M);
1057

gap> ZC1:=ZigZagContractedComplex(L);
Pure cubical complex of dimension 2.

gap> Size(ZC1);
4
```

O anel tem uma quantidade inicial de 162833 facetas. Realizando um retrato por deformação simplificamos este anel por um espaço com um total de 1057 facetas, por outro lado, realizando um retrato por zig-zag obtemos um espaço com apenas 4 facetas. A última imagem da figura 5.2 é uma imagem extremamente ampliada para visualizarmos as 4 facetas.

O comando “ZigZagContractedComplex” pode ser aplicado para complexos cúbicos

e permutaedrais puros. Por esta razão, os complexos foram apresentados no capítulo 3.

Para obter as imagens obtidas na figura 5.2 realizamos os seguintes comandos,

```
gap> ViewPureCubicalComplex(L);  
gap> ViewPureCubicalComplex(M);  
gap> ViewPureCubicalComplex(ZC1);
```

com L , M e $ZC1$ definidos como anteriormente.

Para mais detalhes sobre equivalência de homotopia e retratos por deformação, consulte as referências [17] e [18].

6 Complexo de Čech e funtores esquecimento

A topologia algébrica envolve uma gama de funtores para converter um tipo de complexo celular em outro tipo. Para calcular o grupo fundamental, a nível computacional, necessitamos utilizar determinados funtores.

Este capítulo é destinado a apresentação de dois funtores, a saber, o complexo de Čech e os funtores esquecimento. As principal referência deste capítulo é o artigo [4].

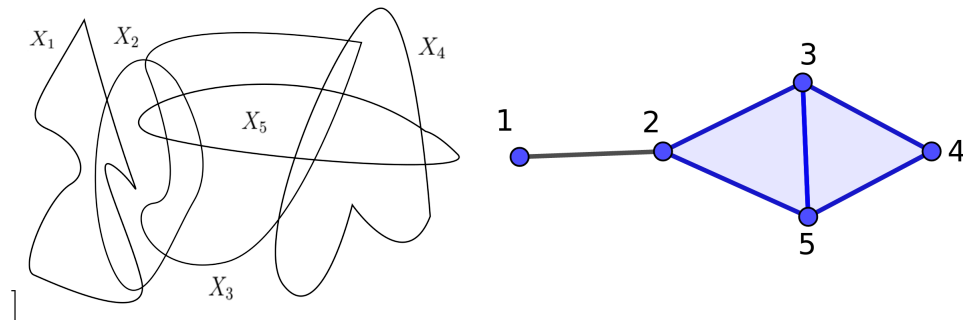
Primeiramente, a fim de tratarmos do complexo de Čech, definimos o que é um nervo de um complexo.

Definição 31. Seja $\mathcal{U} = \{U_v\}_{v \in V}$ uma cobertura aberta de um complexo C . O **nervo** de $\mathcal{U}$ é um complexo simplicial finito $nerve(\mathcal{U})$ definido por

$$nerve(\mathcal{U}) = \left\{ \sigma \subset V : \bigcap_{v \in \sigma} U_v \neq \emptyset \right\}.$$

Vejamos um exemplo de nervo.

Figura 6.1: Conjunto V à esquerda e $nerve(\mathcal{U})$ à direita.



Fonte: Gerada pelos softwares [7] e [6].

Exemplo 6. Considere $V := \{X_1, X_2, X_3, X_4, X_5\}$ mostrados na figura 6.1 à esquerda, e considere U_i o menor aberto que contém o espaço X_i . Se $\mathcal{U} = \{U_i\}$ então o $\text{nerve}(\mathcal{U})$ é dado pela figura 6.1 à direita.

A construção do **complexo Čech** pode ser usada para associar um complexo simplicial $\check{\text{Cech}}(P)$ a qualquer complexo reticulado n -dimensional P . O complexo simplicial $\check{\text{Cech}}(P)$ possui um vértice para cada n -célula em P e um simplexo para cada coleção de n -células de P cujos fechos têm interseção comum não trivial. Essa construção é facilmente implementada em um computador usando as L -bolas B_L para determinar interseções não triviais. Para cada n -célula e^n em P , podemos construir $U(e)$ uma pequena vizinhança aberta do fecho de e^n de modo que os conjuntos abertos $U(e)$ formem uma cobertura aberta de P . Então $\check{\text{Cech}}(P)$ pode ser visto como um nervo do recobrimento $\{U(e)\}$, isto é, o $\check{\text{Cech}}(P)$ é um nervo em que os abertos $\{U(e)\}$ recobrem o conjunto P . Além disso, o $\check{\text{Cech}}(P)$ é bem conhecido por ter o mesmo tipo de homotopia que P .

As referências [19, 20, 21] contêm detalhes sobre nervo e complexo de Čech.

Uma segunda classe de functor útil e de fácil implementação são os chamados **funtores esquecimento** que convertem um complexo simplicial K , cúbico ou permutaedral puro em um CW -complexo regular simplesmente esquecendo parte da estrutura de K . Esses funtores esquecimento são, mais uma vez, facilmente implementados no GAP e são realizados pelos comandos:

```
gap>SimplicialComplexToRegularCWComplex(K);
```

em que K é um complexo simplicial.

```
gap>CubicalComplexToRegularCWComplex(K);
```

onde K é um complexo cúbico puro.

```
gap>PermutahedralComplexToRegularCWComplex(K);
```

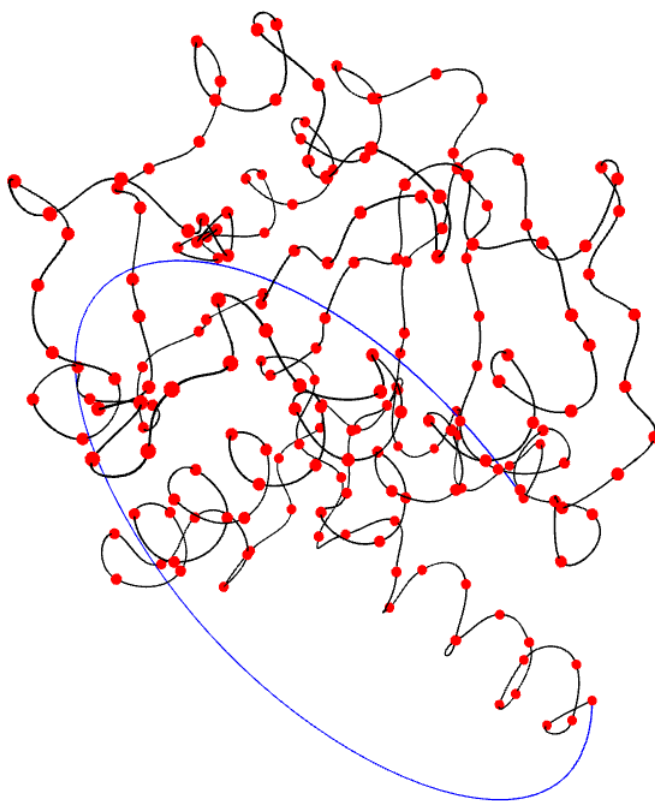
com K um complexo permutaedral puro.

Nos três comandos acima, o GAP retorna o CW -complexo regular correspondente. Veja no próximo exemplo, como o complexo de Čech e alguns desses comandos são utilizados. O exemplo foi intercalado entre a parte teórica e a computacional.

Exemplo 7. A Figura 6.2 mostra o backbone da proteína da bactéria *Thermus Thermophilus*. As coordenadas euclidianas de todos os átomos da proteína estão disponíveis no Protein Data Bank [22]. Essas coordenadas são dados experimentais. O backbone é obtida tomando as coordenadas do átomo de carbono alfa em cada um dos 191 aminoácidos na proteína e ajustando a uma curva para essa sequência de átomos de carbono alfa. Os átomos de carbono alfa inicial e final, que ficam próximos à ‘superfície’ da estrutura da proteína, foram artificialmente unidos por uma curva, de modo a formar uma imersão contínua $S^1 \rightarrow \mathbb{R}^3$, isto é, de modo a formar um nó (veja figura 6.2).

A figura 6.3 mostra uma representação do nó como um complexo cúbico puro tri-dimensional K . O complexo K foi formado de um retrato por deformação de um complexo cúbico puro K' , tendo o complexo K' sido construído de modo a conter a imagem do nó $S^1 \rightarrow \mathbb{R}^3$ como um retrato por deformação.

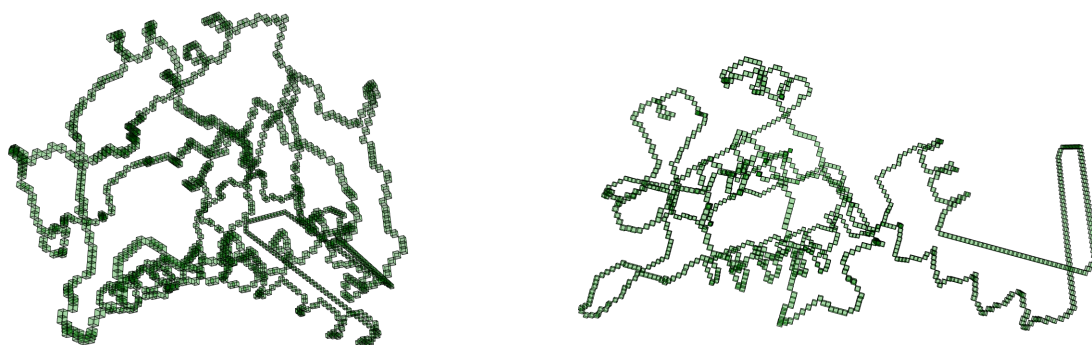
Figura 6.2: Backbone da proteína 1v2x T. thermophilus representado como um nó.



Fonte: Protein Data Bank [22].

Pode ser visto na figura 6.3 que como o complexo cúbico puro K não é uma variedade topológica, discutiremos sobre isso no capítulo 8.

Figura 6.3: Backbone da proteína 1v2x T. thermophilus representado como um complexo puro cúbico. Visão frontal (esquerda) visão lateral (direita).



Fonte: Gerada pelo software [2].

No entanto, K pode ser imerso em um variedade M da seguinte maneira. Seja $A = (a_\lambda)$ a matriz binária que representa K . Seja $A' = (a'_\lambda)$ uma nova matriz binária definida por

$$(a'_\lambda) := \begin{cases} a'_\lambda = 1 & \text{se } a_{\lambda+\mu} = 1 \text{ para algum } \mu \in B_L \\ a'_\lambda = 0 & \text{caso contrário} \end{cases}$$

com $B_L = \{-1, 0, 1\}^3$, isto é, $B_L = \{(x, y, z) \mid x, y, z \in \{-1, 0, 1\}\}$. Seja L o complexo cúbico puro representado por A' . Então L é uma variedade topológica contendo K . Neste exemplo, acontece que K é um retrato por deformação de L (em outros exemplos, pode ser necessário redimensionar e usar cubos menores para garantir que K seja um retrato por deformação de L). Os complexos cúbicos puros K e L contêm 1071 e 16408 cubos tridimensionais, respectivamente.

```
gap> L:=ReadPDBfileAsPureCubicalComplex("1v2x.pdb");
Reading chain containing 191 atoms.
Pure cubical complex of dimension 3.

gap> K:=ContractedComplex(L);
Pure cubical complex of dimension 3.

gap> Size(L);
16408
```

```
gap> Size(K);
1071
```

Como estamos interessados no complementar $\mathbb{R}^3 \setminus L$, formamos um complexo cúbico puro C com a matriz binária $A'' = (a''_\lambda)$ definida por

$$\begin{cases} a''_\lambda = 1 & \text{se } a'_\lambda = 0 \\ a''_\lambda = 0 & \text{se } a'_\lambda = 1. \end{cases}$$

Para garantir que C seja um complexo finito, restringimos o índice λ a um intervalo finito que inclui todos os casos em que $a'_\lambda = 1$. O complexo C possui 691684 cubos tridimensionais e é homeomorfo ao complementar $I^3 \setminus L$ com I^3 algum cubo tridimensional sólido e fechado cujo interior contém a variedade.

```
gap> C:=ComplementOfPureComplex(L);
Pure cubical complex of dimension 3.

gap> Size(C);
691684
```

Ao aplicar o retrato zig-zag no GAP, obtemos um complexo cúbico puro C' tendo 4892 cubos tridimensionais que tem o mesmo tipo de homotopia que C . O CW -complexo C' visto como um CW -complexo regular tem 77077 células.

```
gap> C1:=ZigZagContractedPureComplex(C);
Pure cubical complex of dimension 3.

gap> Size(C1);
4892

gap> C1R:=CubicalComplexToRegularCWComplex(C1);
Regular CW-complex of dimension 3

gap> Size(C1R);
77077
```

Um CW -complexo regular menor, mas do mesmo tipo de homotopia, pode ser obtido por $C'' = \check{C}ech(C')$. O CW -complexo C'' tem um total de 51607 células e é 5-dimensional. Em geral, o complexo Čech de um complexo cúbico puro será de dimensão maior que o complexo original.

```
gap> C2:=CechComplexOfPureCubicalComplex(C1);
Simplicial complex of dimension 5.

gap> Size(C2);
51607
```

Agora transformamos C'' em um complexo regular e um retrato por deformação $C''' \subset C''$ pode ser calculado usando o comando “ContractedComplex”. O CW -complexo regular C''' tem um total de 30743 células simpliciais, em comparação com um total de 5674743 células cúbicas no CW -complexo C de mesmo tipo homotópico. Além disso, o CW -complexo C''' é de dimensão 2.

```
gap> C2R:=SimplicialComplexToRegularCWComplex(C2);
Regular CW-complex of dimension 5

gap> C3:=ContractedComplex(C2R);
Regular CW-complex of dimension 2

gap> Size(C3);
30743

CR:=CubicalComplexToRegularCWComplex(C);
Regular CW-complex of dimension 3

gap> Size(CR);
5674743
```

7 Campo de vetores discreto

Este capítulo trata da última ferramenta necessária para o desenvolvimento do cálculo computacional do grupo fundamental. O papel principal de um campo de vetores discreto é captar informações estruturais dos complexos que sejam de fato pertinentes à implementação da rotina computacional. Para tanto, observamos inicialmente que para um CW -espaço ser regular, geralmente precisamos incluir mais células do que seria necessário em uma CW -decomposição do espaço não regular. A dificuldade com CW -complexos não regulares é que não está claro como melhor representá-los de forma combinatória em um computador sem perder informações importantes da homotopia.

Optamos por uma representação envolvendo uma tripla (X, Y, h) em que X é um CW -complexo regular e Y é um (possivelmente) CW -espaço não regular, tais que X e Y têm o mesmo tipo de homotopia e $h : X \rightarrow Y$ uma equivalência de homotopia entre eles. Dessa forma, todas as informações são armazenadas em X , enquanto Y pode ser usado em algoritmos.

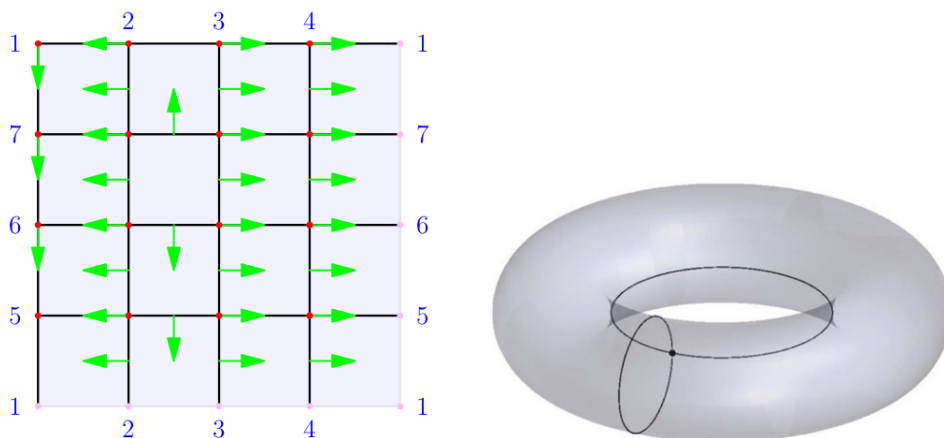
Definição 32. Um **campo de vetores discreto** em um CW -complexo X é uma coleção formal de setas $s \rightarrow t$ onde

1. s, t são células de X com $\dim(t) = \dim(s) + 1$ e com s pertencendo a fronteira de t . Dizemos que s e t estão **envolvidas** na seta, que s é a **fonte** da seta e que t é a **meta** da seta.
2. qualquer célula está envolvida em no máximo uma seta.

Um exemplo de um campo de vetores discreto em uma CW -decomposição regular de um toro é ilustrado na figura 7.1. O campo de vetores discreto é devido a Forman [23].

Agora vejamos algumas definições.

Figura 7.1: Uma estrutura CW regular de um toro munida com um campo de vetores discreto à esquerda e uma estrutura não regular de um toro à direita.



Fonte: artigo [4].

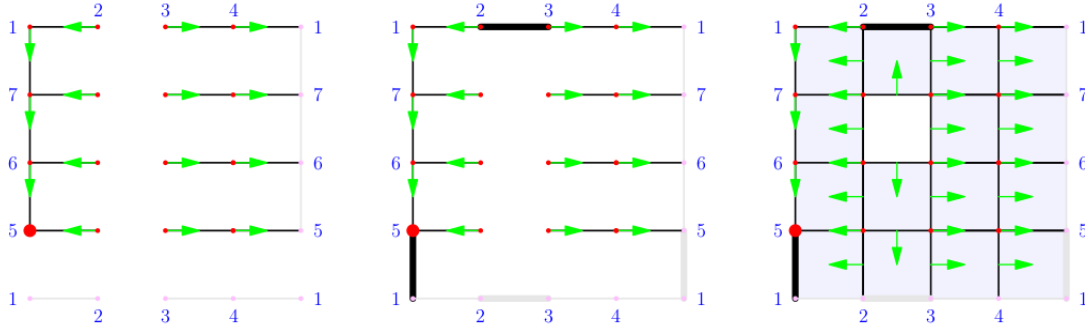
Definição 33.

1. Uma **cadeia** em um campo de vetores discreto é uma sequência de setas

$$\dots, s_1 \rightarrow t_1, s_2 \rightarrow t_2, s_3 \rightarrow t_3, \dots$$

onde s_{i+1} está no bordo de t_i para cada i .

2. Diz-se que uma cadeia é um **circuito** se tiver um comprimento finito, com a fonte s_1 da seta inicial $s_1 \rightarrow t_1$ pertencendo ao bordo da meta t_n da seta final $s_n \rightarrow t_n$.
3. Um campo de vetores discreto é considerado **admissível** se não contiver circuitos, nem cadeias que se estendam infinitamente para a direita. Estamos interessados apenas com CW -espaços finitos; então, um campo de vetores discreto é **admissível**, ou **acíclico** se não contiver circuitos.
4. Dizemos que um campo de vetores discreto admissível é **máximo** se não for possível adicionar mais setas preservando a admissibilidade.
5. Diz-se que uma célula em X é **crítica**, no campo de vetores discreto, se não estiver envolvida em alguma seta.

Figura 7.2: Os espaços T^0 (à esquerda), V^1 (centro) e T^1 (à direita).


Fonte: artigo [4].

Teorema 1. ([23], [24], [18]) Se X é um CW -complexo regular com campo de vetores discreto admissível, então há uma equivalência de homotopia

$$h : X \rightarrow Y$$

onde Y é um CW -espaço cujas células estão em correspondência biunívoca com as células críticas de X .

O CW -complexo Y neste teorema é determinado por X e pelo campo de vetores discreto em X . Para termos uma ideia da demonstração do teorema 1 vejamos o exemplo abaixo referente a figura 7.1.

Exemplo 8. Considere o campo de vetores discreto acíclico na CW -decomposição regular do toro X mostrado na figura 7.1 (à esquerda). O complexo possui uma 0-célula crítica, duas 1-células críticas e uma 2-célula crítica. Devemos encontrar uma equivalência de homotopia $h : X \rightarrow Y$, onde Y é um CW -espaço com uma 0-célula, duas 1-células e uma 2-célula, conforme ilustrado na figura 7.1 (à direita).

No que segue, sejam X^n o n -esqueleto de X ; T^n a união de X^n com todas as $(n+1)$ -células que são metas de alguma seta em X ; V^{n+1} a união de T^n com todas as $(n+1)$ -células críticas de X . Consideramos V^0 como a coleção de 0-células críticas. Consequentemente temos as inclusões:

$$V^0 \subseteq X^0 \subseteq T^0 \subseteq V^1 \subseteq X^1 \subseteq T^1 \subseteq V^2 \subseteq X^2$$

Os espaços T^0 , V^1 e T^1 estão ilustrados na figura 7.2.

Construímos o CW -complexo Y recursivamente, definindo seu 0-esqueleto Y^0 consistindo nas 0-células críticas, então $V^0 = Y^0$. Suponha agora, que conseguimos construir uma equivalência de homotopia $\nu : V^n \rightarrow Y^n$ em que as células de Y^n estão em uma correspondência biunívoca com as k -células críticas de X para $k \leq n$.

Note que podemos estabelecer que V^n é um retrato por deformação de T^n . Para fazer isso, observe que, como o campo de vetores discreto é admissível, deve haver uma n -célula e^n em T^n que é a fonte da seta inicial $e^n \rightarrow e^{n+1}$ de alguma cadeia no campo de vetores em T^n . Seja $T_1^n \subset T^n$ o CW -espaço regular obtido de T^n removendo e^n e e^{n+1} . Isso constitui um colapso elementar $T^n \searrow T^{n+1}$ e expansão elementar $T_1^n \nearrow T^n$ no sentido da teoria simples de homotopia [17], visto que $T^n = T_1^n \cup e_1^n \cup e_1^{n+1}$ e $e_1^n \in e_1^{n+1}$. Em particular, T_1^n é um retrato por deformação de T^n . Aplicando o mesmo procedimento a T_1^n , obtemos um retrato por deformação $T_2^n \hookrightarrow T_1^n$ com $V^n \subseteq T_2^n$. Repetindo o procedimento, obtemos uma sequência de retratos por deformação

$$V^n \nearrow T_k^n \nearrow T_{k-1}^n \nearrow \dots \nearrow T_1^n \nearrow T^n.$$

Veja, por exemplo, que $V^1 = T_{15}^1$, pois 15 é a quantidade de setas que possuem uma 2-célula como meta. Assim existe uma aplicação $\tau : T^n \rightarrow V^n$ que é uma equivalência de homotopia.

Para cada $(n+1)$ -célula e^{n+1} crítica em X temos uma aplicação característica de E^{n+1} a X tal que, restrita a S^n , define a aplicação $\phi : S^n \rightarrow X^n$. A aplicação composta

$$\phi' : S^n \xrightarrow{\phi} X^n \subseteq T^n \xrightarrow{\tau} V^n \xrightarrow{\nu} Y^n$$

pode ser usada para formar a união $Y^n \cup e^{n+1}$ com estrutura CW na qual ϕ' se estende a uma aplicação característica para e^{n+1} . Consideramos Y^{n+1} o espaço obtido anexando a Y^n todas as $(n+1)$ -células críticas de X dessa maneira. A construção de Y^{n+1} é tal que a equivalência de homotopia $\nu : V^n \rightarrow Y^n$ se estende a uma equivalência de homotopia $v : V^{n+1} \rightarrow Y^{n+1}$.

Sejam V, Y denotados pelos CW -complexos com V^n, Y^n seus respectivos n -esqueletos. Para $n = \dim(X)$, temos a equivalência de homotopia $h : X = V^n \rightarrow Y$ desejada.

8 Calculando grupos fundamentais

Este é o principal capítulo deste trabalho pois, depois de uma sequência de capítulos, com diversos exemplos e definições, finalmente podemos apresentar o cálculo computacional do grupo fundamental, via o programa GAP, de maneira coerente e bem fundamentada.

Especificamente, abordamos os seguintes objetivos:

1. O algoritmo por trás do cálculo do grupo fundamental;
2. O cálculo do grupo fundamental, propriamente dito, e de uma aplicação de grupos fundamentais.
3. O cálculo envolvendo os subgrupos de um grupo fundamental, a fim de testar a igualdade, a nível computacional, de grupos finitamente apresentados.

Para tanto, primeiramente vejamos as seguintes definições,

Definição 34. Um CW -espaço é chamado de **reduzido** se possui apenas uma 0-célula.

Definição 35.

1. Um complexo celular 1-dimensional $X = X^1$ é chamado de **grafo** que consiste em vértices (as 0-células) às quais as arestas (as 1-células) estão conectadas. As duas extremidades de uma aresta podem ser coladas ao mesmo vértice.
2. Um **subgrafo** de um grafo X é um subespaço $Y \subset X$ que é uma união de vértices e arestas de X , de modo que $e_\alpha \subset Y$ implica que $\bar{e}_\alpha \subset Y$. A última condição diz apenas que Y é um subespaço fechado de X .
3. Uma **árvore** é um grafo contrátil. Por uma árvore em um grafo X , queremos dizer um subgrafo que é uma árvore.

4. Chamamos uma árvore em X **maximal** se ela contiver todos os vértices de X .

Sabemos que a estrutura celular do 2-esqueleto de um CW -espaço Y reduzido é codificada por uma apresentação correspondente do seu grupo fundamental em termos de um conjunto de geradores $\underline{x}$ de um grupo livre $F = F(\underline{x})$ e um conjunto $\underline{r} \subset F$ de relações. Os geradores estão em bijeção com as 1-células de Y e as relações estão em bijeção com as 2-células. Para mais detalhes, consulte [25, 26]. Sabemos que, se Y é um CW -complexo conexo por caminhos então, contraindo qualquer árvore máxima em seu 1-esqueleto, a estrutura CW no espaço Y pode ser modificada para a de um CW -complexo reduzido.

Agora, o intuito é, no contexto do teorema 1, determinar uma apresentação para o grupo fundamental $\pi_1(Y)$ do CW -espaço Y o qual corresponde a um CW -complexo regular K com campo de vetores discreto admissível.

Definição 36. Dizemos que uma célula em K é **terminal** se é crítica ou é meta de uma seta. Caso contrário, isto é, se é fonte de uma seta, dizemos que a célula é **inicial**. Observe que uma 0-célula é terminal se, e somente se, for crítica.

Qualquer 0-célula e^0 em K pode ser associada a uma única 0-célula $H(e^0)$ terminal definindo recursivamente:

$$H(e^0) = \begin{cases} e^0, & \text{se } e^0 \text{ é terminal,} \\ H(e'^0), & \text{se existe uma seta } e^0 \rightarrow e^1, \text{ com } e'^0 \neq e^0, \text{ um bordo da 1-célula } e^1. \end{cases}$$

De um modo mais simples, se uma 0-célula não é crítica, a associamos à outra 0-célula ‘adjacente’.

Definição 37. Dizemos que uma 1-célula e_i^1 é orientada ou possui uma **orientação** quando seus vértices são ordenados, denotados por $\partial^- e_i^1$ e $\partial^+ e_i^1$. Dizemos que $\partial^- e_i^1$ é o **primeiro vértice** e que $\partial^+ e_i^1$ é o **segundo vértice**.

Definição 38. Dizemos que $\mathcal{E} = (e_1^1, e_2^1, \dots, e_n^1)$ é um **trajeto**, ou **caminho**, de 1-células terminais em K , se $\partial^+ e_i^1 = \partial^- e_{i+1}^1$

Em $H(e^0)$, é fácil definirmos a recursão visto que em cada 1-célula temos no máximo duas 0-células em seu fecho. Mas para definirmos $H(e^1)$ isto será um pouco mais delicado, visto que uma 2-célula pode possuir diversas 1-células em seu fecho.

Para alguma 1-célula orientada inicial e^1 temos uma 2-célula associada $e^1 \rightarrow e^2$. O bordo de e^2 especifica um trajeto $f_1^1, f_2^1, \dots, f_m^1$ em que o primeiro vértice de f_1^1 é o primeiro vértice de e^1 , e o segundo vértice de f_m^1 é o segundo vértice de e^1 . Denotamos este trajeto por $\leftarrow e^1 \rightarrow$. O trajeto envolve todas as 1-células do bordo de e^2 exceto a 1-célula e^1 .

Definimos $H(e^1)$ recursivamente por

$$H(e^1) = \begin{cases} e^1, & \text{se } e^1 \text{ é terminal,} \\ H(\leftarrow e^1 \rightarrow), & \text{caso contrário.} \end{cases}$$

Se definirmos $H(\leftarrow e^1 \rightarrow)$ de forma coerente a definição recursiva de $H(e^1)$ estará completa. Se $\mathcal{E} = (e_1^1, e_2^1, \dots, e_n^1)$ é um trajeto de 1-células qualquer, definimos o $H(\mathcal{E})$ por

$$H(\mathcal{E}) := (H(e_1^1), H(e_2^1), \dots, H(e_n^1))$$

desse modo, visto que $\leftarrow e^1 \rightarrow := (f_1^1, f_2^1, \dots, f_m^1)$,

$$H(\leftarrow e^1 \rightarrow) = (H(f_1^1), H(f_2^1), \dots, H(f_m^1)),$$

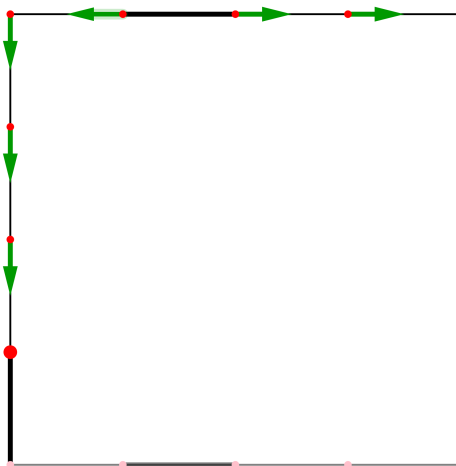
isto é, definimos o trajeto $H(\leftarrow e^1 \rightarrow)$ como sendo a concatenação da sequência ordenada $H(f_1^1), H(f_2^1), \dots, H(f_m^1)$.

Observe que nas definições recursivas de $H(e^0)$ e $H(e^1)$ o processo será encerrado devido à condição de admissibilidade do campo de vetores discreto.

Podemos agora descrever o CW -complexo Y do teorema 1. O 0-esqueleto Y^0 consiste em uma 0-célula para cada 0-célula crítica e^0 em K , denotamos por ρe^0 a 0-célula de Y correspondente a e^0 crítica de K . O 1-esqueleto Y^1 consiste em Y^0 unido com uma 1-célula ρe^1 para cada 1-célula e^1 crítica de K . A 1-célula ρe^1 é colada às 0-células $\rho H(e^0), \rho H(e'^0)$ onde e^0 e e'^0 são as 0-células no bordo de e^1 . O 2-esqueleto Y^2 consiste em Y^1 unido com uma 2-célula crítica ρe^2 para cada 2-célula crítica em K . A 2-célula ρe^2 é colada pelo trajeto $(\rho(e_1^1), \rho(e_2^1), \dots, \rho(e_n^1))$ onde a sequência $(e_1^1, e_2^1, \dots, e_n^1)$ é obtida aplicando H a todo o bordo ∂e^2 de e^2 e excluindo qualquer 1-célula não crítica do caminho resultante $H(\partial e^2)$. O bordo ∂e^2 é um trajeto fechado (0-célula inicial e final se coincidem) de 1-células e não importa em qual 1-célula o trajeto fechado é iniciado.

Exemplo 9. No campo de vetores discreto da figura 7.1, o CW -complexo Y tem uma 0-célula, duas 1- células e uma 2-célula. Se denotarmos as 1-células orientadas arbitrariamente de Y por x e y , e suas orientações opostas por x^{-1} e y^{-1} , então, pela explicação anterior, a 2-célula será colada pelo trajeto $xyx^{-1}y^{-1}$. Com efeito, para Y^0 só temos uma célula crítica, logo não há o que ser feito. Para Y^1 temos que $Y^1 = Y^0 \cup \rho e_1^1 \cup \rho e_2^1$, veja na imagem a seguir o trajeto dos vértices de e_1^1 e e_2^1 .

Figura 8.1: Trajeto dos vértices de e_1^1 e e_2^1 .



Fonte: Gerada pelo software [6].

Note que como só temos uma única 0-célula crítica, os vértices de ρe_1^1 e ρe_2^1 em Y serão iguais ao ρe^0 . Por fim, em K , possuímos uma única 2-célula crítica e^2 , assim temos na figura 8.2 à direita o $H(\partial e^2)$.

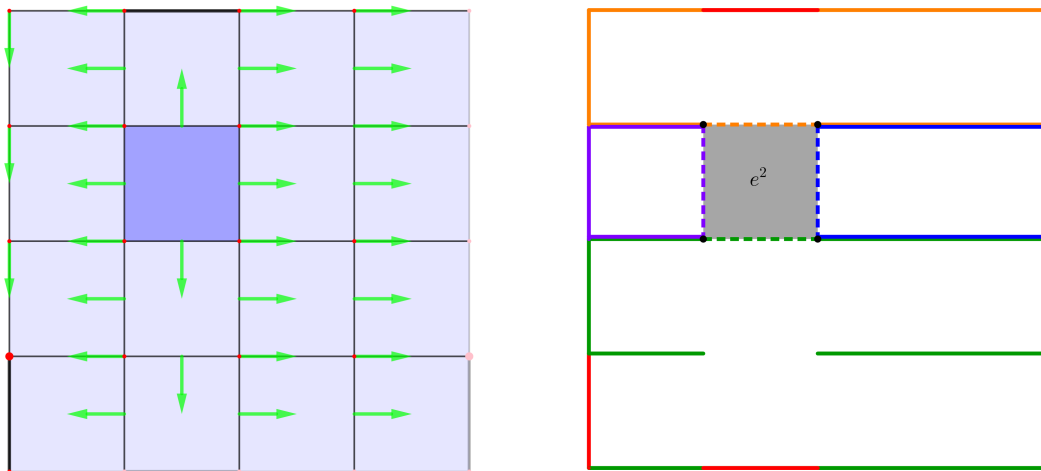
Agora considere a seguinte tabela:

Gerador	Direção	Orientação
x	horizontal	esquerda para direita
x^{-1}	horizontal	direita para esquerda
y	vertical	baixo para cima
y^{-1}	vertical	cima para baixo

Fonte: Gerada pelo autor.

Assim, basta seguirmos todo trajeto de $H(\partial e^2)$ a partir de uma 0-célula no fecho de e^2 . Porém, como explicamos anteriormente, consideramos apenas as 1-células críticas

Figura 8.2: Campo de vetores discreto K à esquerda e a representação do $H(\partial e^2)$ à direita.



Fonte: Gerada pelo software [6].

que estão em vermelho na figura 8.2 à direita, desse modo obtemos o trajeto $xyx^{-1}y^{-1}$.

Usando a correspondência entre CW -complexos 2-dimensionais reduzidos e as apresentações de grupo, obtemos a apresentação $\pi_1(Y) \cong \langle x, y \mid xyx^{-1}y^{-1} \rangle$.

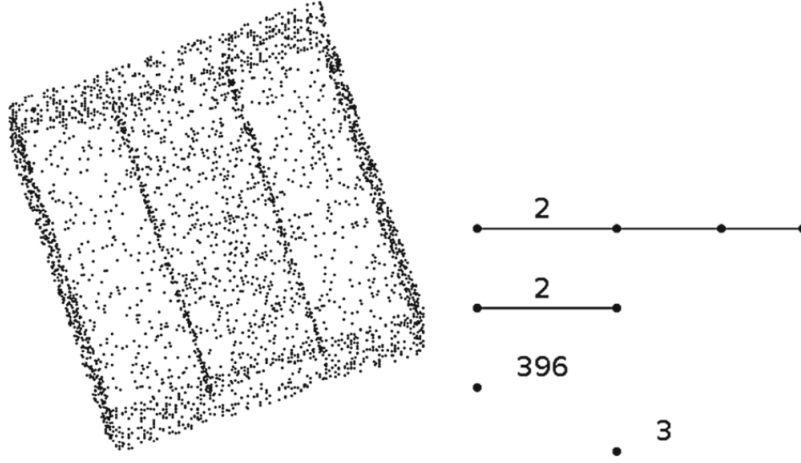
Veja que, no contexto do teorema 1, o CW -espaço Y pode não estar reduzido, se este for o caso, é possível construir uma árvore máxima a partir de seu 1-esqueleto e então realizar uma contração deste árvore a um ponto e desta forma obtemos uma estrutura CW reduzida para Y . Obtemos uma apresentação para o grupo fundamental de um CW -complexo Y reduzido diretamente de seu 2-esqueleto.

O exemplo 10 a seguir, pode ser encontrado na íntegra no artigo [4], e todos os dados e resultados também são provenientes de lá, visto que a amostra de pontos não estava disponível para o teste dos cálculos a seguir. Mesmo assim, o exemplo será explanado pois é extremamente útil para o entendimento da construção dos grupos fundamentais para análise topológica de dados.

Exemplo 10. A figura 8.3 (esquerda) mostra um conjunto X de 3397 pontos amostrados em uma superfície desconhecida $K \subset \mathbb{R}^3$. A superfície K está contida em um cubo de lado 100 e os pontos em X foram escolhidos para terem coordenadas inteiras. Seja K_0^X o complexo cúbico puro que consiste em uma unidade de 3-cubo centrada no vetor inteiro x , para cada $x \in X$. Definimos K_i^X recursivamente como a união das

unidades 3-cubos centralizadas em vetores inteiros que se intersectam de forma não trivial com K_{i-1}^X , isto é, estamos desconsiderando as interseções vazias. Como meio de inferir informações sobre K , podemos seguir a abordagem padrão de homologia persistente para análise de dados e investigar as aplicações induzidas em homologia $\alpha_n^{ij} : H_n(K_i^X, \mathbb{Q}) \rightarrow H_n(K_j^X, \mathbb{Q})$ para $i \leq j$.

Figura 8.3: Amostra X sobre uma superfície $K \subset \mathbb{R}^3$ (esquerda) e barcode β_1 (direita).



Fonte: artigo [4].

Estas são aplicações de espaços vetoriais e portanto determinadas pelo seu posto $\beta_m^{i,j} = \text{rank}(\alpha_n^{ij})$.

A sequência de aplicações em homologia de grau 1

$$H_1(K_2^X, \mathbb{Q}) \rightarrow H_1(K_6^X, \mathbb{Q}) \rightarrow H_1(K_{10}^X, \mathbb{Q}) \rightarrow H_1(K_{14}^X, \mathbb{Q}) \quad (8.1)$$

são descritas pelo barcode da figura 8.3 que é produzido usando a implementação da homologia de persistência, de grau 1 em [3]. Veja que este barcode em específico é composto por quatro colunas, onde cada coluna representa um índice das aplicações em (8.1), assim, as colunas que possuem os vértices são representados pelos índices 2, 6, 10 e 14, exatamente nesta ordem. O número de vértices de cada coluna do barcode, contado pela multiplicidade, é igual ao número de Betti $\beta_1^{i,i} = \beta_1(K_i^X) = \text{rank}(H_1(K_i^X, \mathbb{Q}))$. Então $\beta_1^{2,2} = 2 + 2 + 396 + 0 = 400$, $\beta_1^{6,6} = 2 + 2 + 0 + 3 = 7$, $\beta_1^{10,10} = 2$ e $\beta_1^{14,14} = 2$. Para os demais números de Betti, analisamos quantas barras, ou linhas, existem entre uma coluna e outra, em outras palavras, o número de linhas da i -ésima coluna para a j -ésima coluna é igual a número de Betti $\beta_1^{i,j}$. Assim $\beta_1^{2,6} = 2 + 2 = 4$, $\beta_1^{2,10} = 2$,

$\beta_1^{2,14} = 2$, $\beta_1^{6,10} = 2$, $\beta_1^{6,14} = 2$, $\beta_1^{10,14} = 2$. A linha longa corresponde ao $\beta_1^{2,14} = 2$ implica que duas classes de homologia 1-dimensional em K_2^X persistem como classes de homologia distintas em K_{14}^X .

O barcode é consistente com X sendo mostrado na variedade K com $\beta_1(K) = 2$. Os barcodes β_0 e β_2 correspondentes (que não são mostrados) sugerem que $\beta_0(K) = 1$ e $\beta_2(K) = 1$. Como esses números de Betti são consistentes com K sendo um toro $K = S^1 \times S^1$ ou uma esfera com duas circunferências pinçadas em um ponto $K = S^2 \vee S^1 \vee S^1$.

Após aplicarmos as técnicas de homologia de persistência, o próximo passo é calcular o grupo fundamental de K_{14}^X . Para tanto, veja que este complexo cúbico puro possui 1321192 3-cubos. Um retrato zig-zag $L \simeq K_{14}^X$ pode ser construído com apenas 664 3-cubos. O espaço L e a apresentação do grupo fundamental $\pi_1(L) \cong \langle x, y \mid xyx^{-1}y^{-1} \rangle$ foram calculados usando [3]. O cálculo envolveu a construção de um campo de vetores discreto em L com uma 0-célula crítica, duas 1-células críticas, uma 2-célula crítica e nenhuma outra célula crítica. A apresentação sugere que X foi amostrado em um toro (note que a apresentação coincide com a vista no exemplo 9).

Para dados de dimensões mais altas, uma variante do exemplo 10 seria escolher uma sequência crescente de números reais $\epsilon_1, \epsilon_2, \dots$ e considerar K_i^X como o complexo simplicial Vietoris-Rips (também chamado de “clique complex”), tendo um vértice para cada ponto em X e um k -simplexo para cada subconjunto de vértices $\{x_1, \dots, x_{k+1}\} \subset X$ onde a distância entre cada par de pontos no subconjunto satisfaz $d(x_i, x_j) \leq \epsilon_k$. A métrica d pode ser a distância euclidiana ou qualquer outra distância.

Uma característica notável do exemplo 10 é a natureza relativamente simples da apresentação do grupo fundamental que foi produzido usando o GAP. Quando uma apresentação é muito “complexa”, isto é, possui uma grande quantidade de geradores, relações e as relações são extensas, trabalhar com estes grupos com o programa GAP torna-se inviável.

Em geral, podemos tentar calcular $\pi_1^{ij} : \pi_1(K_i^X) \rightarrow \pi_1(K_j^X)$ através um campo de vetores discreto maximal em K_i^X e um campo de vetores discreto maximal auxiliar, possivelmente não relacionado com K_j^X .

Para simplificar a discussão, vamos supor que K_i^X, K_j^X são espaços conexos por caminho e que escolhemos construir um campo de vetores em K_i^X com uma única 0-

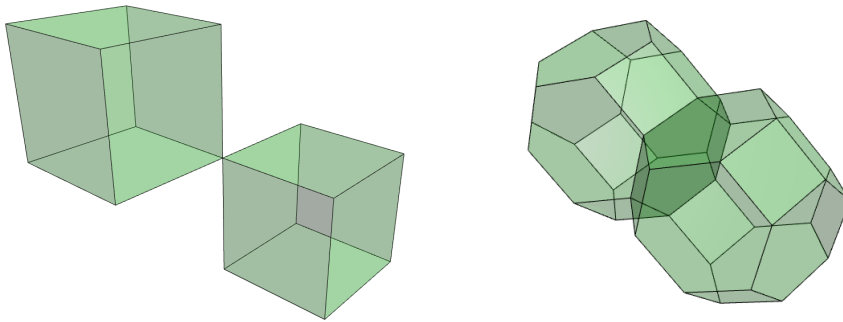
célula crítica em K_i^X . Este campo de vetores determina uma árvore $T_i \subset K_i^X$ em que todas as 1-células são metas de setas. Cada gerador $x \in \pi_1(K_i^X)$ corresponde a alguma 1-célula crítica e_x^1 em K_i^X . Esta 1-célula crítica e_x^1 determina um laço γ_x , visto que só temos uma 0-célula crítica, de 1-células representando x , com todas estas 1-células pertencendo a T_i com 1-célula crítica. Cada 1-célula no laço γ_x é também uma 1-célula em K_j^X . Para cada 1-célula f no laço γ_x , denote $H(f)$ pela sequência de 1-células em K_j^X definidas anteriormente com respeito ao campo de vetores discreto em K_j^X . Desse modo, $\pi_1^{ij}(x)$ é representado pela concatenação das sequências $H(f)$ assim como f percorre as arestas do caminho γ .

Este cálculo de homomorfismos de grupos finitamente apresentados aplica-se ao homomorfismo $\pi_1(K) \rightarrow \pi_1(L)$ induzido por qualquer inclusão $K \subseteq L$ de CW -complexos regulares como veremos no exemplo 11 mais adiante.

8.1 Variedades celulares e um invariante do nó

O complexo permutaedral puro n -dimensional K tem a propriedade de que quaisquer duas n -células distintas e^n, e'^n com interseção não trivial dos bordos compartilham exatamente uma célula bordo de dimensão $n - 1$. Isso implica que K é uma variedade topológica, isto é, localmente homeomorfo a $\mathbb{R}^n$, veja figura 8.4.

Figura 8.4: Uma representação cúbica (não variedade) e uma permutaedral (variedade).



Fonte: Gerada pelo software [2].

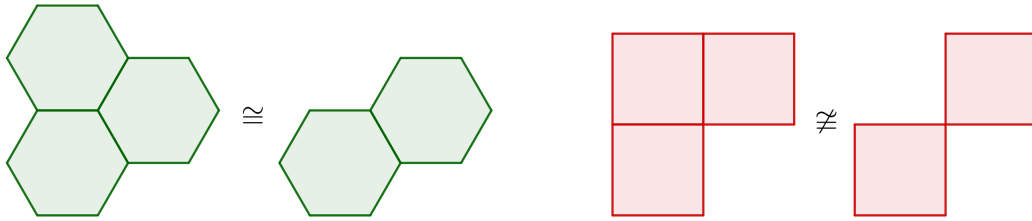
Além disso, implica que, para qualquer retrato por deformação $\overline{K \setminus e^n} \subset K$ existe um homeomorfismo $\overline{K \setminus e^n} \cong K$ (Veja fig. 8.5). Como qualquer retrato de deformação em zig-zag é apenas uma sequência dessas deformações elementares, segue que qualquer

retrato de deformação em zig-zag de um complexo permutaedral puro K é, na verdade, homeomorfo a K .

Uma segunda característica importante desses retratos por deformação no complexo permutaedral puro e no complexo cúbico puro é que estes retratos preservam o tipo de isotopia ambiente. Ou seja, se K' é um retrato zig-zag de K , existe uma aplicação contínua $F : \mathbb{R}^n \times [0, 1] \rightarrow \mathbb{R}^n$ de modo que F_0 é a identidade, F_t é um homeomorfismo e F_1 leva K em K' .

Essas propriedades tornam os complexos permutaedrais tridimensionais puros um bom cenário para a teoria computacional dos nós.

Figura 8.5: Retratos por def. permutaedrais são homeom. diferente dos cúbicos.



Fonte: Gerada pelo software [6].

Exemplo 11. Vejamos novamente o exemplo 7 do backbone da proteína 1v2x T. thermophilus. Utilizando o GAP, construímos um complexo permutaedral puro 3-dimensional K de ambiente isotópico a uma pequena vizinhança fechada do nó na figura 6.2. O complexo K contém 1190 permutaedros 3-dimensional.

```
gap> K:=ReadPDBfileAsPurePermutahedralComplex("1v2x.pdb");
Reading chain containing 191 atoms.
Pure permutahedral complex of dimension 3.

gap> Size(K);
1190
```

Uma variedade compacta e contrátil $R \subset \mathbb{R}^3$ foi construída de forma que K pertence ao interior de R . Para calcular o grupo fundamental, calculamos o fecho do complementar $\overline{R \setminus K}$ obtendo um complexo permutaedral puro 3-dimensional C envolvendo 3438 células permutaedrais 3-dimensional.

```

gap> C2:=ComplementOfPureComplex(K);
Pure permutahedral complex of dimension 3.

gap> C:=ZigZagContractedPureComplex(C2);
Pure permutahedral complex of dimension 3.

gap> Size(C);
3438

```

A variedade C está então representada como um CW -complexo regular. O bordo da variedade C , isto é, $T = \partial C$, foi calculado como um CW -espaço puro 2-dimensional. O número total de células em C e T é 147751 e 120854, respectivamente.

```

gap> Y:=PermutahedralComplexToRegularCWComplex(C);
Regular CW-complex of dimension 3

gap> Size(Y);
147751

gap> T:=BoundaryOfPureComplex(Y);
Regular CW-complex of dimension 2

gap> Size(T);
120854

```

Como a teoria prevê, o complexo T possui duas componentes conexas, uma homeomorfa ao toro e outra homeomorfa a 2-esfera correspondente ao bordo de R . Escolhendo um ponto base comum para T e C na componente toro de T , usamos o grupo fundamental descrito acima, para calcular o homomorfismo induzido $\pi_1(T) \rightarrow \pi_1(C)$ como o homomorfismo de grupos apresentados finitamente.

$$\begin{aligned}
\langle u, v \mid uvu^{-1}v^{-1} \rangle &\rightarrow \langle x, y \mid y^2x^{-2}xyx^{-1}, yx^{-2}y^{-1}x(xy^{-1})^2 \rangle \\
u &\mapsto yx^{-3}y^2x^{-2}xyx^{-1} \\
v &\mapsto yx^{-1}y^{-1}x^2y^{-1}.
\end{aligned}$$

O seguinte código GAP foi usado para construir $\pi_1(T) \rightarrow \pi_1(C)$.

```

i:=BoundaryPairOfPureRegularCWComplex(Y);
Map of regular CW-complexes

gap> CriticalCellsOfRegularCWComplex(Source(i));
[ [ 2, 1 ], [ 2, 261 ], [ 1, 1043 ], [ 1, 1626 ], [ 0, 2892 ],
[ 0, 24715 ] ]

gap> phi:=FundamentalGroup(i,2892);
[ f1, f2 ] -> [ f2*f1^-3*f2^2*f1^-2*f2*f1*f2^-1,
f2*f1^-1*f2^-1*f1^2*f2^-1 ]

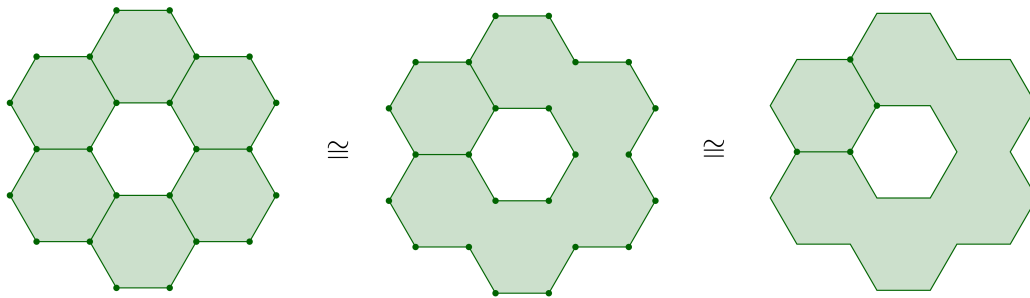
gap> R1:=RelatorsOfFpGroup(Source(phi));
[ <identity ...>, f1*f2^-1*f1^-1*f2 ]

gap> R2:=RelatorsOfFpGroup(Target(phi));
[ f2^2*f1^-2*f2*f1*f2^-1, f2*f1^-2*f2^-1*f1*(f1*f2^-1)^2 ]

```

Seja K uma variedade arbitrária dotada com a estrutura celular de um espaço CW . Mencionamos anteriormente que qualquer k -célula e^k pertencente ao bordo de exatamente uma célula e^{k+1} de dimensão $k+1$ gera um retrato por deformação $K \setminus (e^{k+1} \cup e^k) \hookrightarrow K$. Em geral este retrato por deformação não será uma variedade.

Figura 8.6: Simplificação da estrutura celular de um CW -espaço regular.

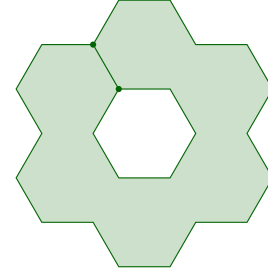


Fonte: Gerada pelo software [6].

Existe uma operação básica alternativa que pode ser aplicada a estrutura CW regular de uma variedade K para gerar uma variedade CW regular homeomorfa com duas células a menos, a saber:

Suponha que K contém uma k -célula e^k pertencendo ao bordo de precisamente duas $(k+1)$ -células e_1^{k+1} e e_2^{k+1} com cobordos idênticos (isto é, e_1^{k+1} pertence ao bordo de uma célula e^{k+2} se, e só se, e_2^{k+1} pertence ao bordo de e^{k+2}). Então as três células e^k , e_1^{k+1} e e_2^{k+1} podem ser removidas e trocadas por uma única célula de dimensão $k+1$. O espaço topológico K permanece imutável; somente sua estrutura CW muda.

A estrutura CW resultante não é, em geral, regular. Porém, será regular se os conjuntos de vértices V_0, V_1, V_2 pertencentes aos bordos de $e^k, e_1^{k+1}, e_2^{k+1}$ respectivamente são tais que $V_0 = V_1 \cap V_2$. Diversas dessas aplicações são mostradas no exemplo da ‘flor hexagonal’ na figura 8.6. Veja que na figura ao lado, a estrutura CW não é regular. Uma simplificação para variedades CW regular, baseado nesta operação, está implementada em [3].



Fonte: Gerada pelo software [6].

8.2 Calculando subgrupos e invariantes de grupos

Para determinadas apresentações de grupos, é impossível encontrar um algoritmo para decidir a igualdade entre os elementos do grupo dados pelos produtos das potências dos geradores. No entanto, existem algoritmos para determinar alguns isomorfismos invariantes básicos de grupos finitamente apresentados. O sistema GAP, também, contém certos comandos, que parecem bastante práticos para grupos G com poucos geradores e poucas relações de comprimento curto, para saber um pouco mais sobre estes comandos consulte o manual em [2].

Ademais, muitas das vezes os grupos fundamentais encontrados são grupos infinitos e incomuns, isto é, não são grupos que estamos habituados a trabalhar, como o $\mathbb{Z}_n$ por exemplo, o que torna difícil a implementação de determinados cálculos.

Para entendermos estes algoritmos que estamos interessados é necessário alguns conhecimentos básicos que podem ser encontrados em [27].

Definição 39. Uma **série subnormal** de um grupo G é uma cadeia de subgrupos $G = G_0 > G_1 > \dots > G_n$ tal que G_{i+1} é normal a G_i , $0 \leq i < n$. Uma serie subnormal tal que G_i é normal em G para todo i é chamada de **série normal**.

Definição 40. Os subgrupos $\gamma_n G$ de uma **série central decrescente** ou **cadeia central decrescente** de G , é dada por

$$G = \gamma_1(G) \geq \gamma_2(G) \geq \cdots \geq \gamma_n(G) \geq \gamma_{n+1}(G) \geq \cdots$$

são definidos recursivamente por $\gamma_1 G = G$ e $\gamma_{n+1} G = [\gamma_n G, G]$, $n = 1, 2, \dots$, em que $[G, G]$ denota o subgrupo dos comutadores. Para simplificar a notação definimos $\gamma_i G := G_i$. Dado um grupo G e um inteiro positivo N , se $G_N = 1$ então G é chamado de **nilpotente**.

Proposição 4. A cadeia central decrescente de G é uma série normal.

Demonstração. Basta provar que cada G_i é normal a G . Sabemos que G_1 é normal a G . Suponha por hipótese de indução que G_k é normal a G . Desse modo, queremos provar que G_{k+1} é normal a G .

Como $G_{k+1} = [G_k, G]$, tome $xyx^{-1}y^{-1} \in G_{k+1}$ qualquer, onde $x \in G_k$ e $y \in G$. Para todo $g \in G$, temos:

$$\begin{aligned} g(xyx^{-1}y^{-1})g^{-1} &= gxyx^{-1}y^{-1}g^{-1} \\ &= gx(g^{-1}g)y(g^{-1}g)x^{-1}(g^{-1}g)y^{-1}g^{-1} \\ &= (gxg^{-1})(gyg^{-1})(gx^{-1}g^{-1})(gy^{-1}g^{-1}) \\ &= (gxg^{-1})(gyg^{-1})(gxg^{-1})^{-1}(gyg^{-1})^{-1}. \end{aligned}$$

Como G_k é normal, $gxg^{-1} \in G_k$. Assim $g(xyx^{-1}y^{-1})g^{-1} \in G_{k+1}$, portanto G_{k+1} é normal. $\square$

Dados n, p números inteiros positivos, com p primo, definimos $\gamma_{n+1}^p G$ recursivamente por

$$\gamma_1^p G := G \text{ e } \gamma_{n+1}^p G := [\gamma_n^p G, G](\gamma_n^p G)^p.$$

Agora, observamos que temos definições suficientes para calcular os seguintes invariantes de um grupo G .

- Os invariantes do quociente abeliano $G/[G, G]$.
- Dado um inteiro n , uma lista de apresentações finitas para os subgrupos $S < G$ de índices até no máximo n .

- Dado um inteiro c , uma apresentação finita para o grupo nilpotente $G/\gamma_{c+1}G$. O procedimento provê um algoritmo para determinar a igualdade entre elementos do grupo.
- Dados inteiros c e p , uma apresentação finita para o p -grupo finito $G/\gamma_{c+1}^p G$.
- Dados inteiros c, p e m , os invariantes dos grupos de homologia integral $H_m(G/\gamma_{c+1}G, \mathbb{Z})$ e $H_m(G/\gamma_{c+1}^p G, \mathbb{Z})$.

Analisando a viabilidade dos algoritmos acima consideramos os seguintes invariantes do grupo G e o homomorfismo de grupos $\varphi : G \rightarrow G'$.

Definição 41. Dado um grupo G e inteiros $m, n, c \geq 1$ definimos o conjunto de grupos de homologia por

$$I^{[n,c,m]}(G) = \{H_m(S/\gamma_{c+1}S, \mathbb{Z}) : S \leq G, |G : S| \leq n\}.$$

Sejam $\varphi : G \rightarrow G'$ um homomorfismo entre grupos, $p > 1$ um inteiro primo e inteiros $m, c \geq 1$. Definimos o homomorfismo canônico entre grupos abelianos por

$$J^{[p,c,m]}(\varphi) : H_m(S/\gamma_c S, \mathbb{Z}) \rightarrow H_m(S'/\gamma_{c+1} S', \mathbb{Z})$$

para $S = [G, G]G^p$, $S' = [G', G']G'^p$.

Ambos os invariantes na Definição 41 podem, a princípio, ser calculados no GAP para qualquer grupo finitamente apresentado e qualquer homomorfismo de grupos finitamente apresentados. Na prática, o cálculo não é viável para grupos com grande quantidade de geradores ou grande quantidade de relações, ou pequenas quantidades de relações extensas, isto é, em que o comprimento de uma relação é extenso, devido ao tempo necessário para fazer esses cálculos.

É difícil quantificar o que é ‘grande’ ou o que é ‘pequeno’, pois depende, em certo aspecto, da natureza das apresentações envolvidas. No que segue exploramos exemplos sobre cálculos de invariantes.

Exemplo 12. Os códigos GAP mostrados a seguir calculam o invariante $I^{[5,2,3]}(G)$ para G o complementar do nó da proteína mostrado na figura 6.2. O conjunto $I^{[5,2,3]}(G)$ é mostrado (‘print’) e G possui nove subgrupos ($S < G$) de índice $|G : S| \leq 5$.


```

gap> K:=ReadPDBfileAsPurePermutahedralComplex("1v2x.pdb");
Reading chain containing 191 atoms.
Pure permutahedral complex of dimension 3.

gap> G:=KnotGroup(K);
#I  there are 2 generators and 1 relator of total length 6
<fp group of size infinity on the generators [ f1, f2 ]>
gap> R:= RelatorsOfFpGroup(G);
[ f2^-1*f1*f2*f1*f2^-1*f1^-1 ]
gap> L:=LowIndexSubgroupsFpGroup(G,5);
gap> Apply(L,F->Range(IsomorphismFpGroup(F)));
gap> Apply(L,S->NilpotentQuotient(S,2));
gap> Apply(L,Q->GroupHomology(Q,3));
gap> Print(SortedList(L));
[ [ ], [ ], [ 2, 2, 2, 4, 4, 4, 4 ], [ 3, 3 ], [ 3, 3 ], [ 3, 3 ],
[ 4 ], [ 4 ], [ 9 ] ]

```

Observe que neste exemplo utilizamos o comando ‘KnotGroup’ para encontrar o grupo fundamental e a relação deste grupo é diferente da que encontramos no exemplo 11. Por esta razão vamos realizar novamente o cálculo com o algoritmo feito anteriormente.

```

gap> K:=ReadPDBfileAsPurePermutahedralComplex("1v2x.pdb");;
gap> C2:=ComplementOfPureComplex(K);;
gap> C:=ZigZagContractedPureComplex(C2);;
gap> Y:=PermutahedralComplexToRegularCWComplex(C);;
gap> F:=FundamentalGroup(Y,2892);
#I  there are 2 generators and 1 relator of total length 5
<fp group of size infinity on the generators [ f1, f2 ]>
gap> R:= RelatorsOfFpGroup(F);
[ f2*f1~-2*f2*f1 ]
gap> L2:=LowIndexSubgroupsFpGroup(F,5);;
gap> Apply(L2,F->Range(IsomorphismFpGroup(F)));
gap> Apply(L2,S->NilpotentQuotient(S,2));
gap> Apply(L2,Q->GroupHomology(Q,3));
gap> Print(SortedList(L2));
[ [ ], [ ], [ 2, 2, 2, 4, 4, 4, 4 ], [ 3, 3 ], [ 3, 3 ], [ 3, 3 ],
[ 4 ], [ 4 ], [ 9 ] ]

```

Em ambos os casos obtivemos 9 subgrupos de índice 5 e obtivemos os mesmos grupos de homologia, a saber

$$\mathbb{Z}, \mathbb{Z}, \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_2 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4 \oplus \mathbb{Z}_4, \mathbb{Z}_3 \oplus \mathbb{Z}_3, \mathbb{Z}_3 \oplus \mathbb{Z}_3, \mathbb{Z}_3 \oplus \mathbb{Z}_3, \mathbb{Z}_4, \mathbb{Z}_4, \mathbb{Z}_9;$$

isto é,

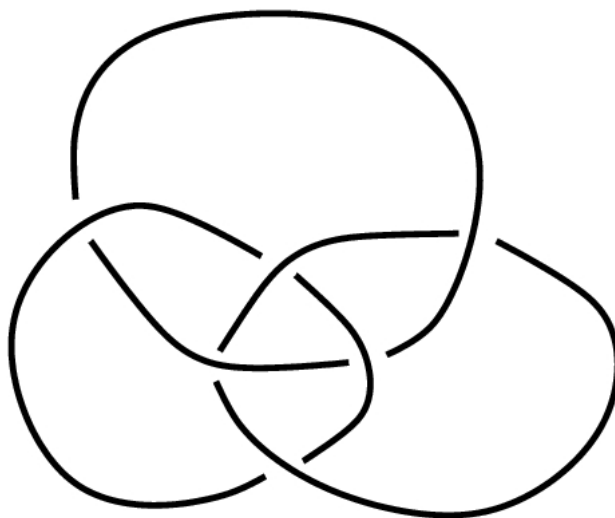
$$\mathbb{Z}, \mathbb{Z}, \mathbb{Z}_2^3 \oplus \mathbb{Z}_4^4, \mathbb{Z}_3^2, \mathbb{Z}_3^2, \mathbb{Z}_3^2, \mathbb{Z}_4, \mathbb{Z}_4, \mathbb{Z}_9.$$

9 Exemplos do HAP

Além dos cálculos realizados nos capítulos anteriores, durante esta pesquisa, realizamos outros cálculos de grupo fundamental. Neste capítulo, explicitamos dois desses exemplos, o exemplo do nó borromeano e a leitura da proteína ‘1XD3’ do corpo humano. Exemplos que podem ser encontrados nas referências [5, 16, 28].

Exemplo 13. Vejamos inicialmente o nó borromeano mostrado na figura 9.1.

Figura 9.1: Nó borromeano



Fonte: Referência [29].

Inicialmente, observe que o nó borromeano não é uma imersão da S^1 no $\mathbb{R}^3$. Veja que utilizando os comandos a seguir confirmamos que “ L ” tem a homologia correta.

```
gap> L:=ReadLinkImageAsPureCubicalComplex("borromean.jpg");
Pure cubical complex of dimension 3.

gap> Homology(L,0);
[ 0, 0, 0 ]

gap> Homology(L,1);
[ 0, 0, 0 ]

gap> Homology(L,2);
[ ]
```

Agora vamos calcular não só o seu grupo fundamental, mas também o respectivo grupo de homologia.

```
gap> F:=KnotGroup(L);
#I there are 3 generators and 2 relators of total length 20
<fp group of size infinity on the generators [ f1, f2, f3 ]>
gap> R:= RelatorsOfFpGroup(F);
[ f2*f1^-1*f3*f1*f2^-1*f1^-1*f2*f3^-1*f2^-1*f1,
  f3^-1*f1^-1*f3*f2*f3^-1*f2^-1*f1*f2*f3*f2^-1 ]
gap> G:=NilpotentQuotient(F,3);
Pcp-group with orders [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
gap> GroupHomology(G,3);
[ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]
```

Lembrando que, o comando “ $\text{NilpotentQuotient}(F, c)$ ” vai gerar o grupo $\gamma_{c+1}F$, e o comando “ $\text{GroupHomology}(G, i)$ ” gera o grupo de homologia $H_i(G, \mathbb{Z})$. Neste exemplo, $H_3(G, \mathbb{Z}) = \mathbb{Z}^{40}$.

Exemplo 14. No que seque, estudaremos a proteína ‘1XD3’, com uma estrutura mais complexa que a proteína ‘1v2x’ da *T. Thermophilus*.

```

gap> K:=ReadPDBfileAsPurePermutahedralComplex("1XD3.pdb");
Reading chain containing 243 atoms.
Pure permutahedral complex of dimension 3.

gap> F:=KnotGroup(K);
<fp group of size infinity on the generators [ f1, f2, f3, f5, f6 ]>
gap> RelatorsOfFpGroup(F);
[ ]
gap> G:=NilpotentQuotient(F,3);
Pcp-group with orders [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ]

```

Diferente da outra proteína essa possui 243 átomos e seu grupo fundamental tem um total de 5 geradores, além disso, só temos as relações triviais de grupo. Por outro lado, calcular o grupo de homologia desta proteína já é algo pouco viável, visto que o cálculo do Grupo de Homologia seria demorado demais ou impossível em certas máquinas. Neste exemplo temos que $\gamma_4 F = \mathbb{Z}^{56}$.

9.1 Investigação

Podemos perceber, pelos exemplos da seção anterior, que calcular o grupo fundamental ou o grupo de homologia isoladamente pode não nos dar tantas informações quanto desejado. Por esta razão uma das investigações feitas foi tentar calcular grupos fundamentais de uma filtração, e tentar observar se existe alguma ‘característica persistente’.

Para tentar calcular o grupo fundamental de uma filtração usaremos novamente o exemplo 7 por ser um exemplo simples, no sentido de viabilidade dos cálculos em um computador, e ao mesmo tempo não é um exemplo trivial como o grupo identidade.

Exemplo 15. A investigação é tentar calcular o grupo fundamental direto da filtração, como mostrado nos comandos a seguir:

```

gap> L:=ReadPDBfileAsPureCubicalComplex("1v2x.pdb");
Reading chain containing 191 atoms.
Pure cubical complex of dimension 3.

gap> T:=ThickeningFiltration( L, 30 );
Filtered pure cubical complex of dimension 3.

gap> N:=[1..10];
[ 1 .. 10 ]
gap> Apply(N,i->FiltrationTerm( T, i ));
gap> Apply(N,F->KnotGroup(F));
#I there are 2 generators and 1 relator of total length 6
#I there are 4 generators and 1 relator of total length 6

```

Realizando estes comandos temos o conjunto N como sendo o conjunto de todos 10 primeiros grupos fundamentais dados pela filtração. Considere X_i o i -ésimo termo da filtração. Vejamos a lista desse grupo e de seus geradores, na lista abaixo.

```

gap> N;
[ <fp group of size infinity on the generators [ f1, f2 ]>,
  <fp group of size infinity on the generators [ f1, f2, f3, f4 ]>,
  <fp group of size infinity on the generators
    [ f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14,
      f15, f16, f17 ]>,
  <fp group of size infinity on the generators [ f1, f2, f3, f4,
    f5, f6, f7, f8, f9, f10, f11, f12, f13, f14, f15, f16, f17,
    f18, f19, f20, f21, f22, f23 ]>,
  <fp group of size infinity on the generators
    [ f1, f2, f3, f4, f5, f6, f7, f8, f9, f10, f11, f12, f13, f14,
      f15, f16, f17 ]>,

```

```

<fp group of size infinity on the generators [ f1, f2, f3, f4,
      f5, f6, f7, f8, f9 ]>,
<fp group of size infinity on the generators [ f1 ]>,
<free group on the generators [ ]>,
<free group on the generators [ ]>,
<free group on the generators [ ]>,

```

A partir do instante 8, todos os demais grupos fundamentais da filtração são isomorfos ao identidade.

Pelos geradores, podemos inferir que a partir do instante três é bem provável que o termo da filtração comece a perder sua estrutura de nó, fazendo com que diversos outros geradores apareçam.

A partir do instante 8 é provável que os termos da filtração X_i sejam similar a um disco 3-dimensional, isto é, similar ao espaço E^3 , em que nenhum laço seja capaz de ‘perfurar’ o espaço X_i .

Além disso, os grupos possuem as seguintes relações:

```

Apply(N,F-> RelatorsOfFpGroup(F));
gap> N;
[ [ f2*f1*f2^-1*f1^-1*f2^-1*f1 ], [ f3^-1*f2*f3*f2*f3^-1*f2^-1 ],
  [ ], [ ], [ ], [ ], [ ], [ ], [ ], [ ] ]

```

Apenas os grupos fundamentais referentes a X_1 e X_2 possuem relações não triviais.

10 Conclusão

Muitos exemplos e cálculos computacionais foram realizados para chegarmos a este ponto da pesquisa. Muitos cálculos computacionais foram na base da tentativa e erro, pois trabalhar com GAP, e consequentemente com o pacote HAP, foi algo completamente novo para todos envolvidos na pesquisa.

Quando começamos o trabalho, desejávamos encontrar uma aplicação concreta para o grupo fundamental, porém percebemos que o uso do grupo fundamental na análise topológica de dados (TDA) ainda está engatinhando. Por esta razão ainda é muito cedo dizer se realmente conseguimos informações úteis do grupo fundamental, ou até mesmo se conseguimos informações adicionais do grupo fundamental, além daquelas já conhecidas em homologia de persistência. Porém, esperamos que estas dúvidas sejam respondidas em breve com novas publicações sobre o tema.

Através de todos os exemplos, podemos verificar que é possível calcular o grupo fundamental de uma nuvem de pontos, isto é, existem ferramentas que são capazes de calcular o grupo fundamental contruindo um nó que passa por uma nuvem de pontos, principalmente considerando uma quantidade de pontos em que seria impossível, ou inviável para os mais persistentes, de calcular utilizando apenas papel e caneta.

Ainda há muito estudo a ser feito sobre o assunto e durante o fatídico ano de 2020 novos artigos sobre uma nova teoria intitulada ‘Persistent Homotopy’ (homotopia de persistência) vem sendo publicados, em que o grupo fundamental, também conhecido como grupo de homotopia de ordem 1, é um dos protagonistas.

Para mim, espero que a pesquisa seja o começo de algo ainda maior e que sirva de orientação para outras pessoas que desejam encontrar grandes aplicações no TDA.

Referências

- [1] MÉMOLI, F.; ZHOU, L. *Persistent Homotopy Groups of Metric Spaces*. 2020. ArXiv:1912.12399 [math.AT].
- [2] THE GAP GROUP. *GAP – Groups, Algorithms, and Programming, Version 4.11.0*. [S.l.], (2020). <https://www.gap-system.org>.
- [3] ELLIS, G. *HAP, Homological Algebra Programming, Version 1.26*. 2020. <https://gap-packages.github.io/hap>. Refereed GAP package.
- [4] BRENDDEL, P. et al. Computing fundamental groups from point clouds. *Applicable Algebra in Engineering, Communication and Computing* 26, 27–48, (2015). <https://doi.org/10.1007/s00200-014-0244-1>.
- [5] ELLIS, G.; HEGARTY, F. Computational homotopy of finite regular *cw*-spaces. *J. Homotopy Relat. Struct.* 9, 25–54, (2014). <https://doi.org/10.1007/s40062-013-0029-4>.
- [6] HOHENWARTER, M. et al. *GeoGebra 6.0*. 2020. <http://www.geogebra.org>.
- [7] Inkscape Project. *Inkscape*. 2020. <https://inkscape.org>.
- [8] MASSEY, W. et al. *A Basic Course in Algebraic Topology*. [S.l.]: Springer New York, 1991. (Graduate Texts in Mathematics). ISBN 9780387974309.
- [9] HANSEN, S. *CW complexes*. <https://www.math.ksu.edu/~hansen/CWcomplexes.pdf>.
- [10] MUNKRES, J. *Topology*. 2nd. ed. [S.l.]: Prentice Hall, Incorporated, 2000. ISBN 9780131816299.

-
- [11] VICK, J. W. Book. *Homology theory; an introduction to algebraic topology*. [S.l.]: Academic Press New York, 1973. ISBN 0127212507.
- [12] MAGNUS, W.; KARRASS, A.; SOLITAR, D. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*. [S.l.]: Dover Publications, 2004. (Dover books on mathematics). ISBN 9780486438306.
- [13] CROWELL, R.; FOX, R. *Introduction to Knot Theory*. [S.l.]: Dover Publications, 2008. (Dover books on mathematics). ISBN 9780486468945.
- [14] THE GAP GROUP. *GAP – Groups, Algorithms, and Programming, Version 4.11.0*. [S.l.], (2020). GAP Tutorial. <https://www.gap-system.org/Manuals/doc/tut/chap0.html>.
- [15] HATCHER, A. *Algebraic topology*. Cambridge: Cambridge Univ. Press, 2000.
- [16] ELLIS, G. *HAP, Homological Algebra Programming, Version 1.26*. 2020. More comprehensive older tutorial. <https://www.gap-system.org/Manuals/pkg/hap-1.25/www/SideLinks/About/aboutContents.html>.
- [17] COHEN, M. M. *A course in simple-homotopy theory*. New York: Springer, 1973. (Graduate Texts in Mathematics). <https://doi.org/10.1007/978-1-4684-9372-6>. Disponível em: <<https://cds.cern.ch/record/1614791>>.
- [18] WHITEHEAD, J. H. C. Simple homotopy types. *American Journal of Mathematics*, Johns Hopkins University Press, v. 72, n. 1, 1950. ISSN 00029327, 10806377. Disponível em: <<http://www.jstor.org/stable/2372133>>.
- [19] SPANIER, E. *Algebraic Topology*. [S.l.]: Springer New York, 2012. ISBN 9781468493221.
- [20] ITSKOV, V.; ROSEN, Z. *Applied Algebraic Topology Notes*. 2015. <http://zvihrosen.com/notes/AppliedAT.pdf>.
- [21] GHRIST, R. W. *Elementary applied topology*. 1st. ed. [S.l.]: Createspace, 2014.
- [22] NUREKI, D. et al. *Deep Knot Structure for Construction of Active Site and Co-factor Binding Site of tRNA Modification Enzyme*. 2004. <https://www.rcsb.org/structure/1V2X>.

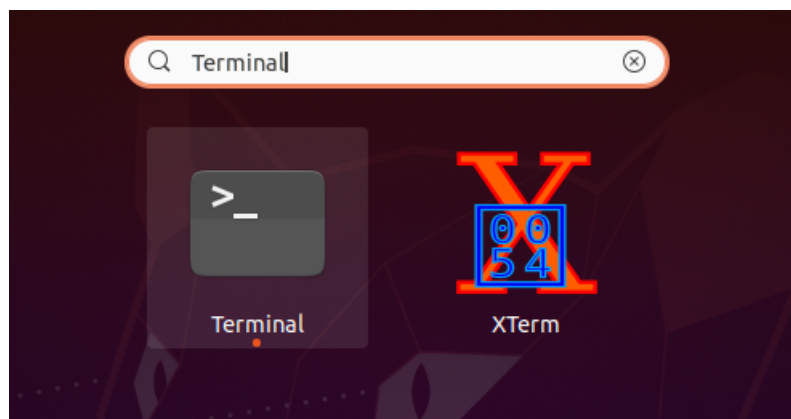
-
- [23] FORMAN, R. Morse theory for cell complexes. *Advances in Mathematics*, v. 134, p. 90–145, 1998.
- [24] FORMAN, R. A user’s guide to discrete morse theory. *Séminaire Lotharingien de Combinatoire*, v. 48, 2001.
- [25] WHITEHEAD, J. H. C. Combinatorial homotopy. I. *Bulletin of the American Mathematical Society*, American Mathematical Society, v. 55, n. 3.P1, p. 213 – 245, 1949. Disponível em: <<https://doi.org/>>.
- [26] WHITEHEAD, J. H. C. Combinatorial homotopy. II. *Bulletin of the American Mathematical Society*, American Mathematical Society, v. 55, p. 453 – 496, 1949. Disponível em: <<https://doi.org/>>.
- [27] KURDACHENKO, L.; SUBBOTIN, I. On some properties of the upper and lower central series. *Southeast Asian Bulletin of Mathematics*, v. 37, 01 2013.
- [28] MISAGHI, S. et al. *Structure of the Ubiquitin Hydrolase UCH-L3 Complexed with a Suicide Substrate*. 2004. <https://www.rcsb.org/structure/1XD3>.
- [29] ELLIS, G. *HAP, Homological Algebra Programming, Version 1.26*. [S.l.], 2020. Manual. https://www.gap-system.org/Manuals/pkg/hap/doc/chap0_mj.html.
- [30] THE GAP GROUP. *GAP – Groups, Algorithms, and Programming, Version 4.11.0*. [S.l.], (2020). GAP Installing. <https://www.gap-system.org/Download/>.
- [31] GAWRILOW, E.; JOSWIG, M. *polymake*: a framework for analyzing convex polytopes. In: *Polytopes—combinatorics and computation (Oberwolfach, 1997)*. [S.l.]: Birkhäuser, Basel, 2000, (DMV Sem., v. 29). p. 43–73. <https://polymake.org/doku.php>.
- [32] HAMMERLINDL, A.; BOWMAN, J.; PRINCE, T. *Asymptote: The Vector Graphics Language*. <https://asymptote.sourceforge.io>.

Apêndice A - Instalação GAP

Durante a pesquisa, diversas barreiras foram encontradas causando um sentimento misto de frustração e curiosidade. Mas de todos os empecílios, instalar o GAP foi, com certeza, um dos mais problemáticos. A primeira tentativa foi instalar o GAP no sistema operacional Windows, mas não obtive sucesso, pois para realizar alguns comandos era necessário pacotes externos, isto é, que não pertenciam ao GAP, os quais eu não consegui instalar no sistema. Então no meio do estudo, troquei o sistema operacional para o Linux, no qual depois de várias tentativas, obtive sucesso. Como não encontrei em lugar algum um passo a passo eficiente de como instalar a versão mais atualizada do GAP no Linux, decidi criar este apêndice para explicar como realizar a instalação. Lembrando que dependendo da versão Linux, a instalação pode ter algumas pequenas diferenças.

Lembrando que o tutorial a seguir é um exemplo de tutorial em que fiz e funcionou, pode ser que existam outros modos de instalar ou até que alguns passos realizados tenham sido desnecessários. Todas as imagens a seguir foram geradas pelo autor no sistema ubuntu Linux.

1. Abra o terminal (na aba de aplicativos, por exemplo);



2. Entre como administrador(root) utilizando o comando “sudo su” e em seguida coloque a senha de administrador;

```
rodrigo@rodrigo-Aspire-V5-471:~$ sudo su  
[sudo] senha para rodrigo:  
root@rodrigo-Aspire-V5-471:/home/rodrigo#
```

3. Use o comando “cd” para entrar no diretório inicial;

```
root@rodrigo-Aspire-V5-471:/home/rodrigo# cd  
root@rodrigo-Aspire-V5-471:~#
```

4. Instale o pacote “g++” para conseguir compilar o GAP utilizando o comando “apt install g++” (talvez ele já esteja instalado), no final responda “s” para instalação;

```
root@rodrigo-Aspire-V5-471:~# apt install g++
```

5. Baixe o GAP no site oficial [30] e extraia a pasta com os arquivos em um diretório (não escolha um diretório muito complicado);
6. Abra a pasta extraída no terminal e execute o arquivo configure usando o comando “./configure”;

```
rodrigo@rodrigo-Aspire-V5-471:~$ cd gap-4.11.0  
rodrigo@rodrigo-Aspire-V5-471:~/gap-4.11.0$ ./configure
```

7. Se não houver erro execute o comando “make” ou “gmake” dependendo da versão do Linux;

```
rodrigo@rodrigo-Aspire-V5-471:~/gap-4.11.0$ make
```

8. Execute o arquivo BuildPackages.sh com o comando “./bin/BuildPackages.sh”

```
rodrigo@rodrigo-Aspire-V5-471:~/gap-4.11.0$ ./bin/BuildPackages.sh
```

9. Execute o GAP usando o comando “./bin/gap.sh” e em seguida utilize o comando “gap> LoadPackage ("hap");”;


```

rodrigo@rodrigo-Aspire-V5-471:~/gap-4.11.0$ ./bin/gap.sh
GAP 4.11.0 of 29-Feb-2020
https://www.gap-system.org
Architecture: x86_64-pc-linux-gnu-default64-kv7
Configuration: gmp 6.2.0, GASMAN
Loading the library and packages ...
Packages: AClib 1.3.2, Alnuth 3.1.2, AtlasRep 2.1.0, AutoDoc 2019.09.04, AutPGrp 1.10.2,
CaratInterface 2.3.3, CRISP 1.4.5, Cryst 4.1.23, CrystCat 1.1.9, CTbLLib 1.2.2,
FactInt 1.6.3, FGA 1.4.0, Forms 1.2.5, GAPDoc 1.6.3, genss 1.6.6, IO 4.7.0,
IRREDSOL 1.4, LAGUNA 3.9.3, orb 4.8.3, Polenta 1.3.9, Polycyclic 2.15.1,
PrimGrp 3.4.0, RadiRoot 2.8, recog 1.3.2, ResClasses 4.7.2, SmallGrp 1.4.1,
Sophus 1.24, SpinSym 1.5.2, TomLib 1.2.9, TransGrp 2.0.5, utils 0.69
Try '??help' for help. See also '?copyright', '?cite' and '?authors'
gap> LoadPackage("hap");

```

10. Observe atentamente as linhas com a frase “#I HAP warning” como na foto;

```

-----
#I HAP warning: Set POLYMAKE_PATH manually if needed.
#I HAP warning: Set NEATO_PATH manually if needed.
#I HAP warning: Set DOT_PATH manually if needed.
#I HAP warning: Set DISPLAY_PATH manually if needed.
#I HAP warning: Set ASY_PATH manually if needed.
Loading HAP 1.25 ...

```

11. Saia do GAP usando o comando “gap>quit;” e novamente entre no terminal como administrador com o comando “sudo su” e em seguida utilize o comando “cd”;

```

gap> quit;
rodrigo@rodrigo-Aspire-V5-471:~/gap-4.11.0$ sudo su
[sudo] senha para rodrigo:
root@rodrigo-Aspire-V5-471:/home/rodrigo/gap-4.11.0# cd
root@rodrigo-Aspire-V5-471:~#

```

12. Instale os pacotes restantes necessários, no nosso caso são o polymake [31] e o asymptote [32], então utilizamos os comandos “apt install polymake” e “apt install asymptote” e respondemos “s” para ambas as instalações;

```

root@rodrigo-Aspire-V5-471:~# apt install asymptote

```

```

root@rodrigo-Aspire-V5-471:~# apt install polymake

```

13. Execute o GAP novamente e mais uma vez use o comando “gap> LoadPackage("hap");” e verifique se as frases com o aviso “#I HAP warning” desapareceram;
14. Utilize o GAP e o pacote HAP sem problemas.

Caso tenha dificuldade na instalação ou queira instalar o GAP em outros sistemas operacionais veja a referência [30].