

Modelação Matemática

Adrian Dias

Nº 1, T 12º 3ª

– Disciplina de Física –

2025 - 2026

Resumo

Este relatório apresenta o desenvolvimento de um modelo matemático e a respectiva implementação em `Python/turtle`, com vista à representação de uma curva ou forma encontrada na natureza. São descritos os objectivos, a metodologia seguida e os principais resultados obtidos.

1 Introdução

Neste capítulo apresenta-se o enquadramento teórico do trabalho, incluindo o modelo matemático de base e o modelo específico implementado no código. De acordo com a proposta, seleccionou-se uma forma presente na natureza, a estrutura hexagonal dos favos de mel, a qual serve de inspiração para a construção do script em `Python/turtle`.

1.1 Conjectura do Favo de Mel e Otimização (1.1)

A estrutura dos favos de mel obedece à **Conjectura do Favo de Mel** (Honeycomb Conjecture), demonstrada matematicamente por Thomas Hales em 1999. Esta conjectura estabelece que, entre todos os possíveis padrões de divisão do plano, a configuração hexagonal regular minimiza o perímetro total para uma área fixa. A razão ótima perímetro-área é dada por:

$$\min \frac{P}{A} = \sqrt[4]{12} \approx 1,8612 \quad (1)$$

onde P representa o perímetro e A a área. No caso do hexágono regular com lado s , temos:

$$\frac{P}{A} = \frac{6s}{\frac{3\sqrt{3}}{2}s^2} = \frac{4}{\sqrt{3}s} \quad (2)$$

1.2 Limitações das Tesselações Regulares (1.2)

A geometria euclidiana impõe restrições rigorosas às tesselações regulares do plano. Conforme demonstrado matematicamente, apenas três polígonos regulares preenchem completamente o espaço sem sobreposições ou espaços vazios:

- **Triângulos equiláteros** (notação de Schläfli: $\{3, 6\}$)
- **Quadrados** (notação de Schläfli: $\{4, 4\}$)
- **Hexágonos regulares** (notação de Schläfli: $\{6, 3\}$)

A condição matemática para tesselação regular é dada por:

$$\frac{1}{m} + \frac{1}{n} = \frac{1}{2} \quad (3)$$

onde m e n são inteiros que satisfazem a equação para os casos $\{3, 6\}$, $\{4, 4\}$ e $\{6, 3\}$.

1.3 Impossibilidade de Refinamento Hierárquico (1.3)

Os hexágonos regulares apresentam limitações matemáticas inerentes no que concerne a sistemas de múltiplas escalas. A impossibilidade de subdivisão hierárquica perfeita deve-se a três restrições fundamentais:

- a. **Conservação de área:** $A_{\text{pai}} = \sum A_{\text{filhos}}$
- b. **Hierarquia simples:** Apenas um pai por célula
- c. **Cobertura perfeita:** Sem buracos ou sobreposições

Estas restrições matemáticas explicam por que as abelhas constroem favos de tamanho uniforme em vez de sistemas com múltiplas escalas.

1.4 Modelo de Grade Hexagonal Implementado (1.4)

No script `Python/turtle`, a estrutura hexagonal é implementada através de um sistema de coordenadas baseado em vetores de rede hexagonal. As posições dos hexágonos são calculadas por:

$$\vec{r}_{ij} = i \cdot \vec{a}_1 + j \cdot \vec{a}_2 \quad (4)$$

onde os vetores base da rede hexagonal são definidos como:

$$\vec{a}_1 = (1.5s, 0) \quad (5)$$

$$\vec{a}_2 = \left(0.75s, \frac{\sqrt{3}}{2}s \right) \quad (6)$$

sendo s o comprimento do lado do hexágono.

1.5 Parâmetros e Especificações do Modelo (1.5)

Para a implementação computacional, foram estabelecidos os seguintes parâmetros matemáticos:

- **Tamanho do hexágono:** $s = 25$ unidades turtle
- **Número de camadas:** $n = 7$ camadas concêntricas
- **Total de hexágonos:** $N = 1 + \sum_{k=1}^7 6k = 127$ favos
- **Ângulo de rotação:** $\theta = 60^\circ$ entre lados consecutivos
- **Área individual:** $A_{\text{hex}} = \frac{3\sqrt{3}}{2}s^2$

1.6 Algoritmo de Construção Progressiva (1.6)

O modelo implementado simula a construção natural através de um algoritmo que segue padrão de expansão radial, iniciando em pontos centrais e expandindo concentricamente. A progressão obedece à sequência:

$$N_k = 6k \quad \text{para} \quad k = 1, 2, \dots, n \quad (7)$$

onde N_k representa o número de hexágonos na camada k -ésima, garantindo crescimento simétrico e geometricamente consistente.

2 Parte Experimental

A componente experimental deste trabalho corresponde à elaboração e explicação do código[carvalho2021praticas]. Tal como num procedimento laboratorial, importa detalhar a lógica implementada[martins2015programacao], os algoritmos utilizados e as opções tomadas em cada etapa, de forma a permitir a replicação do processo.

3 Discussão dos Resultados

Apresentam-se e analisam-se, nesta secção, as imagens geradas automaticamente pelo código. Não foram utilizadas capturas de ecrã, mas sim exportações directas produzidas pelo programa. Discutem-se as semelhanças e diferenças entre os resultados e a imagem de referência, identificando as causas dos desvios e avaliando a qualidade da aproximação obtida.

4 Conclusões

As conclusões são redigidas a partir da análise dos resultados. Evitam-se afirmações superficiais ou subjectivas; privilegiam-se observações fundamentadas, como, por exemplo:

- O modelo reproduz com fidelidade parcial a forma natural seleccionada.
- As limitações decorrem de aproximações matemáticas ou restrições do ambiente de programação.
- Futuras melhorias poderão incluir optimizações algorítmicas ou refinamentos gráficos.

5 Bibliografia