

# Reconhecimento de Propriedades em Polígonos

## Trabalho de Geometria Computacional

Richard Fernando Heise Ferreira  
Universidade Federal do Paraná  
Programa de Pós-Graduação em Informática

14/04/2025

### Resumo

Este projeto tem como objetivo a implementação de estruturas de dados e algoritmos eficientes para análise de propriedades geométricas de polígonos em um plano bidimensional. São abordadas as verificações de simplicidade, convexidade e inclusão de pontos em polígonos simples.

## 1 Introdução

O reconhecimento de propriedades geométricas em polígonos é uma tarefa fundamental em diversas áreas, como computação gráfica, geoprocessamento e modelagem computacional. Neste trabalho, propomos a implementação de algoritmos para verificação de três propriedades importantes:

- Simplicidade de um polígono: ausência de auto-intersecções.
- Convexidade: todos os ângulos internos menores que  $180^\circ$ .
- Inclusão de pontos: determinar se um ponto está dentro, fora ou na borda de um polígono.

Todas as soluções foram desenvolvidas em linguagem C++ e testadas com conjuntos de dados diversos.

## 2 Algoritmos e Corretude

### 2.1 Verificação de Simplicidade

Para verificar se um polígono é simples, utilizamos um algoritmo de força bruta que verifica todas as combinações de pares de arestas não adjacentes. A corretude se baseia no fato de que um polígono simples não possui intersecções internas — portanto, qualquer intersecção detectada entre segmentos não consecutivos invalida a simplicidade.

## 2.2 Verificação de Convexidade

Assumindo que os vértices do polígono são fornecidos em ordem anti-horária, é possível percorrer os vértices e calcular o sinal do produto vetorial entre os vetores consecutivos. Se o sinal se mantiver constante, o polígono é convexo. A verificação de simplicidade é pré-requisito, pois um polígono com interseções pode apresentar falsa convexidade.

## 2.3 Classificação de Pontos

Foi utilizado o algoritmo de *Ray Casting*, no qual uma semirreta é projetada a partir do ponto em questão. Conta-se o número de interseções dessa reta com as arestas do polígono:

- Se o número de interseções é ímpar, o ponto está dentro.
- Se for par, está fora.

Casos especiais como pontos sobre arestas ou vértices são tratados com checagens específicas para garantir precisão.

## 2.4 Interseções e Produtos Vetoriais

Todas as verificações geométricas utilizam operações com vetores, especialmente produtos vetoriais para determinar orientação entre pontos e segmentos. Essas operações são fundamentais para a robustez dos algoritmos.

# 3 Estrutura do Código

O projeto está dividido em arquivos conforme a seguir:

- `main.cpp`: Leitura da entrada e impressão da saída.
- `polygons.cpp`: Implementações das funções de verificação.
- `polygons.hpp`: Declaração das estruturas de dados (`Dot` e `Polygon`) e interfaces.
- `Makefile`: Compilação do projeto.

A pasta `tests/` contém:

- `in/`: Arquivos de entrada.
- `out/`: Saídas geradas pelo programa.
- `answers/`: Saídas esperadas.
- `generator.cpp`: Geração de testes aleatórios.
- `printer.py`: Geração de imagens a partir dos arquivos de entrada.
- `run_tests.py`: Execução automatizada dos testes.