Disciplina INF2604 – Geometria Computacional

RELATÓRIO DE EXERCÍCIO:
Triangulação de polígonos

Aluno: Gustavo Coelho

## 1. Objetivo

O objetivo deste relatório é expor uma abordagem para o problema de triangulação de polígonos através do método de remoção de orelhas. O conteúdo do relatório se resume na explicação teórica dos métodos utilizados ao longo do algoritmo proposto e uma breve demonstração dos resultados de sua implementação, incluindo a demonstração da complexidade temporal assintótica.

## 2. Método

Como mencionado, o método escolhido para a triangulação de polígonos é o método de remoção de orelhas. Este método parte do princípio de que todo polígono com mais de 3 vértices possui ao menos duas orelhas. Portanto, sua implementação se resume a percorrer todos os vértices, buscando por orelhas em potencial, e removendo-as, até que o polígono resultante tenha somente 3 vértices.

Para isso, precisamos primeiramente definir o método para certificar-se de que um determinado vértice forma uma orelha em potencial. Consideramos o polígono abaixo:

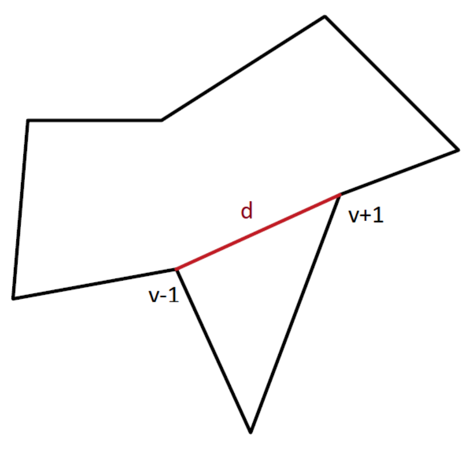


Figura 1: Orelha de um polígono.

O vértice v do polígono forma uma orelha devido às seguintes condições serem satisfeitas:

- Os vértices v-1 e v+1 formam uma diagonal d que está contida na região interna delimitada pela fronteira do polígono.
- A intersecção da diagonal d com a fronteira do polígono se dá apenas em v-1 e v+1.

Portanto, o problema de identificação das orelhas se resume na verificação destas duas condições para cada vértice do polígono. Vejamos inicialmente a primeira condição.

Para determinar se a diagonal d está contida no polígono, consideraremos os vetores $v_1$ , $v_2$ e $v_3$, conforme ilustrado abaixo.

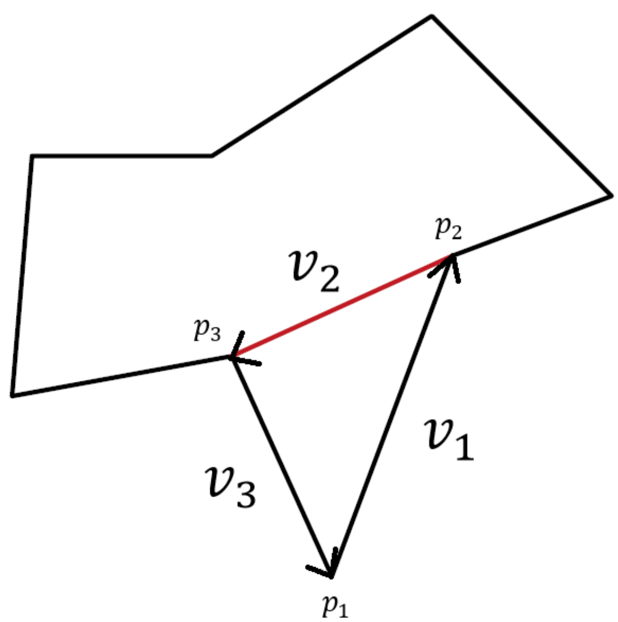


Figura 2: Vetores formatos por um vértices e seus adjacentes.

Onde:

$$v_1 = p_2 - p_1\,, \quad v_2 = p_3 - p_2 \quad e \quad v_3 = p_1 - p_3$$

Dessa forma, podemos afirmar que caso a diagonal d esteja contida no polígono, então a orientação dos vetores será positiva. Ou seja:

$$\begin{vmatrix} p_{1_x} & p_{1_y} & 1 \\ p_{2_x} & p_{2_y} & 1 \\ p_{3_x} & p_{3_y} & 1 \end{vmatrix} > 0$$

Desenvolvendo o determinante, temos:

$$p_{1_x}p_{2_y} + p_{3_x}p_{1_y} + p_{2_x}p_{3_y} - p_{3_x}p_{2_y} - p_{1_x}p_{3_y} - p_{2_x}p_{4_y} > 0$$

Caso esta condição não seja satisfeita, então podemos descartar este vértice como uma orelha em potencial e seguir para o seguinte.

Caso a condição seja satisfeita, precisamos nos certificar de que a diagonal d intersecta somente os pontos $p_2$ e $p_3$. Para isso, assumiremos que: caso haja alguma outra intersecção, então ao menos um vértice não adjacente do polígono está situado dentro do triângulo formado por $p_1, p_2$ e $p_3$. Considerando um ponto $p_i$ representado pela coordenada baricêntrica formada por $p_1, p_2$ e $p_3$, temos:

$$p_i = \lambda_1 p_1 + \lambda_2 p_2 + \lambda_3 p_3 \text{ , onde } \lambda_1 + \lambda_2 + \lambda_3 = 1$$

Tomando uma das propriedades de coordenadas baricêntrica, dizemos que caso o ponto $p_i$ esteja contido no triângulo, então:

$$\lambda_1 > 0,\ \lambda_2 > 0 \text{ e } \lambda_3 > 0$$

Portanto, o problema se resume em encontrar os valores de $\lambda_1$, $\lambda_2$ e $\lambda_3$. Encontramos estes valores através das seguintes equações:

$$\lambda_1 p_{1_x} + \lambda_2 p_{2_x} + \lambda_3 p_{3_x} = p_{i_x}$$
$$\lambda_1 p_{1_y} + \lambda_2 p_{2_y} + \lambda_3 p_{3_y} = p_{i_y}$$
$$\lambda_1 + \lambda_2 + \lambda_3 = 1$$

Usando a regra de Cramer, temos:

$$\lambda_1 = \frac{D_{\lambda_1}}{D}, \lambda_2 = \frac{D_{\lambda_2}}{D} \text{ e } \lambda_3 = \frac{D_{\lambda_3}}{D}$$

Onde:

$$D = \begin{vmatrix} p_{1_x} & p_{2_x} & p_{3_x} \\ p_{1_y} & p_{2_y} & p_{3_y} \\ 1 & 1 & 1 \end{vmatrix}$$

$$D_{\lambda_1} = \begin{vmatrix} p_{i_x} & p_{2_x} & p_{3_x} \\ p_{i_y} & p_{2_y} & p_{3_y} \\ 1 & 1 & 1 \end{vmatrix}$$

$$D_{\lambda_2} = \begin{vmatrix} p_{1_x} & p_{i_x} & p_{3_x} \\ p_{1_y} & p_{i_y} & p_{3_y} \\ 1 & 1 & 1 \end{vmatrix}$$

$$D_{\lambda_3} = \begin{vmatrix} p_{1_x} & p_{2_x} & p_{i_x} \\ p_{1_y} & p_{2_y} & p_{i_y} \\ 1 & 1 & 1 \end{vmatrix}$$

Decompondo os determinantes, temos:

$$\lambda_1 = \frac{p_{i_x}p_{2_y} + p_{2_x}p_{3_y} + p_{3_x}p_{i_y} - p_{3_x}p_{2_y} - p_{i_x}p_{3_y} - p_{2_x}p_{i_y}}{p_{1_x}p_{2_y} + p_{2_x}p_{3_y} + p_{3_x}p_{1_y} - p_{3_x}p_{2_y} - p_{1_x}p_{3_y} - p_{2_x}p_{1_y}}$$

$$\lambda_2 = \frac{p_{1_x}p_{i_y} + p_{i_x}p_{3_y} + p_{3_x}p_{1_y} - p_{3_x}p_{1_y} - p_{1_x}p_{3_y} - p_{i_x}p_{1_y}}{p_{1_x}p_{2_y} + p_{2_x}p_{3_y} + p_{3_x}p_{1_y} - p_{3_x}p_{2_y} - p_{1_x}p_{3_y} - p_{2_x}p_{1_y}}$$

$$\lambda_3 = \frac{p_{1_x}p_{2_y} + p_{2_x}p_{i_y} + p_{i_x}p_{1_y} - p_{i_x}p_{2_y} - p_{1_x}p_{i_y} - p_{2_x}p_{1_y}}{p_{1_x}p_{2_y} + p_{2_x}p_{3_y} + p_{3_x}p_{1_y} - p_{3_x}p_{2_y} - p_{1_x}p_{3_y} - p_{2_x}p_{1_y}}$$

Caso esta condição também seja satisfeita, então podemos assumir que o vértice em questão forma uma orelha do polígono. O algoritmo então remove este vértice e segue para o próximo.

O algoritmo percorre o polígono por inteiro, e para cada vértice, a comparação é feita para cada um dos outros vértices. Portanto, a complexidade esperada do algoritmo é $O_{(n^2)}$.

## 3. Implementação

Após a implementação do algoritmo proposto em linguagem Python, podemos apresentar os exemplos abaixo:

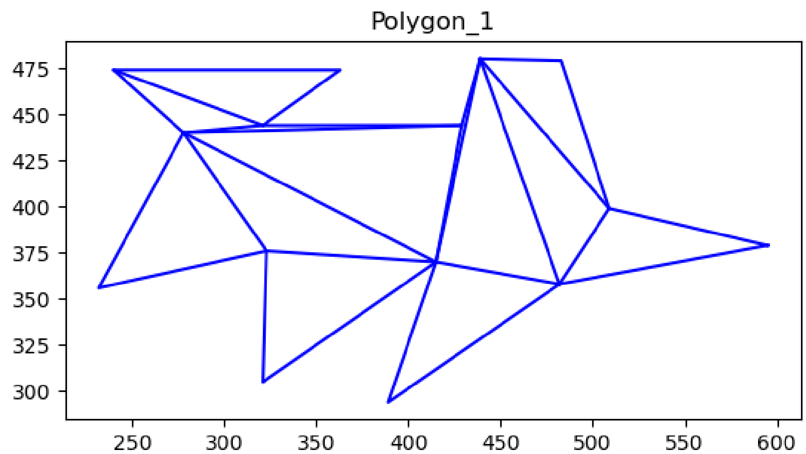


Figura 3: Polígono 1 triangulado

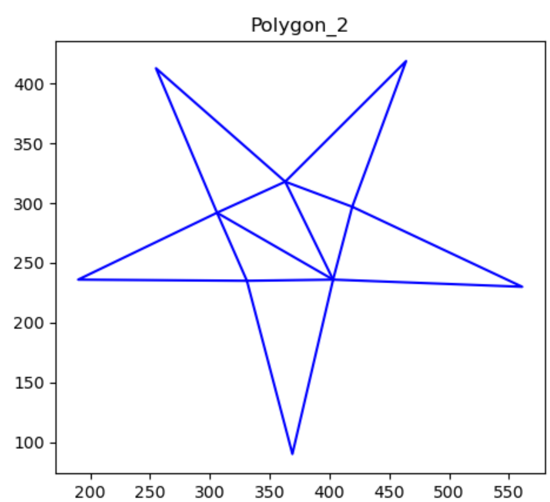


Figura 4: Polígono 2 triangulado

Como esperado, a complexidade do algoritmo se aproxima de um polinômio de segundo grau:

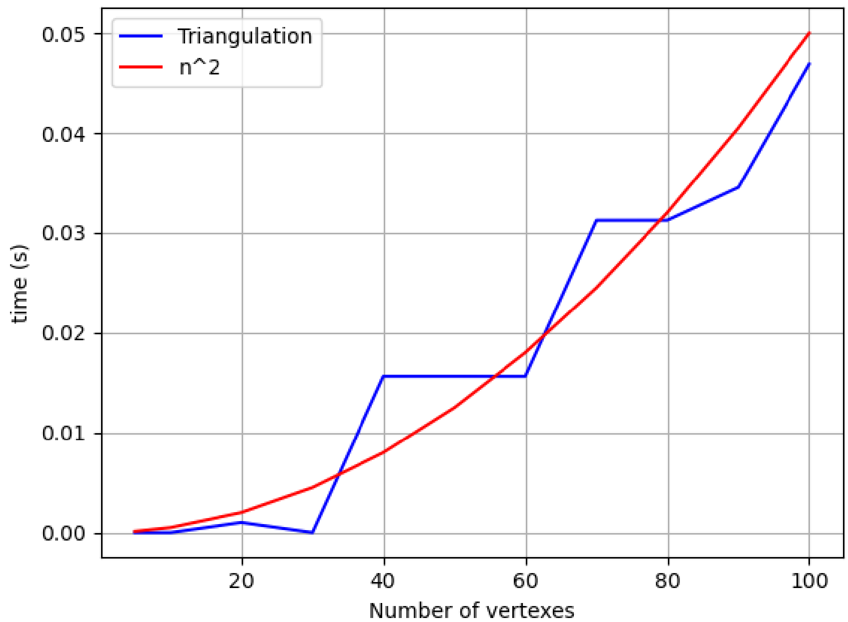


Figura 5: Tempo de execução de acordo com número de vértices

## 4. Conclusão

O algoritmo de triangulação de polígonos através de remoção de orelhas foi implementado em linguagem Python através dos métodos demonstrados neste relatório. A complexidade assintótica apresentada representa uma considerável melhoria de performance comparada à solução “força bruta”, onde a complexidade é proporcional a $O_{(n^4)}$. Há ainda outros métodos com maiores ganhos de performance, porém com maior complexidade teórica. Portanto o método proposto apresenta uma solução intermediária que pode ser útil em determinadas aplicações.