

# 1. Rasterização de Linhas - Algoritmos

A rasterização de linhas é um processo fundamental em computação gráfica para desenhar segmentos de reta em uma grade de pixels. Três algoritmos clássicos são usados para isso:

1. **Analítico** – Usa a equação da reta para calcular os pontos intermediários.
2. **DDA (Digital Differential Analyzer)** – Aproxima a reta incrementando valores ao longo de um eixo.
3. **Bresenham** – Algoritmo eficiente que evita operações de ponto flutuante e usa apenas adições e subtrações.

A implementação de cada algoritmo em Python é descrita a seguir.

---

## 2. Implementação dos Algoritmos

### 2.1 Algoritmo Analítico

O método analítico utiliza a equação da reta na forma:

$$y=mx+b$$

Onde m (coeficiente angular) é dado por:

$$m=\frac{y_2-y_1}{x_2-x_1}$$

#### **Vantagens:**

- Simples de implementar.
- Funciona bem para retas não muito inclinadas.

#### **Desvantagens:**

- Pode causar imprecisão devido ao arredondamento.
  - Ineficiente devido ao uso de operações de ponto flutuante.
- 

### 2.2 Algoritmo DDA (Digital Differential Analyzer)

O DDA é um método incremental que evita o cálculo direto da equação da reta, usando apenas adições.

#### **Vantagens:**

- Evita multiplicações repetidas.
- Mais preciso do que o método analítico.

#### **Desvantagens:**

- Ainda usa operações de ponto flutuante, o que pode afetar a performance em hardware antigo.

## 2.3 Algoritmo de Bresenham

O algoritmo de Bresenham é um método eficiente que utiliza apenas operações inteiras para determinar os pixels da linha.

### Vantagens:

- Usa apenas adições e subtrações, sendo muito eficiente.
- Evita erros acumulados de ponto flutuante.

### Desvantagens:

- Mais complexo de entender e implementar.

## 3. Comparação dos Algoritmos

Algoritmo	Precisão	Eficiência Computacional	Uso de Números Reais	Facilidade de Implementação
<b>Analítico</b>	Média	Baixa	Sim	Fácil
<b>DDA</b>	Alta	Média	Sim	Média
<b>Bresenham</b>	Muito Alta	Alta	Não	Complexa

- **Eficiência:** Bresenham é o mais eficiente porque usa apenas adições e subtrações inteiras.
  - **Precisão:** O DDA e Bresenham são mais precisos do que o método analítico.
  - **Facilidade:** O método analítico é o mais simples de implementar.
- 

## 4. Conclusão

O método a ser escolhido depende do contexto:

- Para aprendizado e simplicidade, o método **analítico** é suficiente.
- Para um equilíbrio entre precisão e facilidade, o **DDA** é uma boa escolha.
- Para máximo desempenho e precisão, o **Bresenham** é a melhor opção.

Cada algoritmo tem suas vantagens e desvantagens, mas o algoritmo de Bresenham continua sendo amplamente utilizado em sistemas gráficos devido à sua eficiência.

### 4.1

#### Comparativo

O algoritmo analítico é preciso, mas pode ser menos eficiente em sistemas sem suporte a ponto flutuante. O DDA é um pouco mais eficiente, especialmente em sistemas com recursos limitados, uma vez que evita cálculos de ponto flutuante. No entanto, o algoritmo de Bresenham é altamente eficiente e mantém a precisão, tornando-se a escolha preferida em muitos casos, especialmente em sistemas embarcados ou com recursos limitados.