

# Geometria Computacional

## Retas

---

Prof. Edson Alves

2018

Faculdade UnB Gama

1. Definição de reta
2. Demonstrações\*

## Definição de reta

---

# Definição

- Reta também é um elemento primitivo da Geometria
- No primeiro livro dos elementos, Euclides define reta como “*a line is breadthless length*”, que numa tradução livre diz que “linha é comprimento sem largura”
- As linhas são elementos unidimensionais
- Em C/C++, pontos podem ser representados através ou de sua equação geral, ou de sua equação reduzida
- A equação reduzida de uma reta é a mais conhecida e utilizada nos cursos de ensino médio
  1. tem a vantagem de facilitar comparações entre retas e identificar paralelismo
  2. não é capaz de representar retas verticais
- A equação geral, como o próprio nome diz, pode representar qualquer reta do plano

# Equação reduzida da reta

- A equação reduzida da reta é dada por

$$y = mx + b,$$

onde  $m$  é o coeficiente angular da reta e  $b$  é o coeficiente linear da reta

- O primeiro coeficiente representa a taxa de variação da reta: consiste no número de unidades que  $y$  varia para cada unidade de variação de  $x$  no sentido positivo do eixo horizontal
- O segundo coeficiente é o valor no qual a reta intercepta o eixo  $y$

```
1 template<typename T>
2 struct Line {
3     T m, b;
4
5     Line(T mv, T bv) : m(mv), b(bv) {}
6 };
```

## Equação reduzida a partir de dois pontos dados

- Dados dois pontos  $P = (x_p, y_p)$  e  $Q = (x_q, y_q)$  tais que  $x_p \neq x_q$ , a inclinação da reta é dada por

$$m = \frac{y_q - y_p}{x_q - x_p}$$

- Deste modo, a equação reduzida da reta será dada por

$$y = m(x - x_p) + y_p = mx + (y_p - mx_p)$$

- Se  $x_p = x_q$ , a reta é vertical
- Retas verticais podem ser tratadas por meio de uma variável booleana que indica se a reta é vertical ou não
- Caso seja, o coeficiente  $b$  indica o ponto que a reta intercepta o eixo horizontal

# Implementação da reta através da equação reduzida

```
1 // Definição da função de comparação equals e da classe Point
2
3 template<typename T>
4 struct Line {
5     T m, b;
6     bool vertical;
7
8     Line(const Point& P, const Point& Q) : vertical(false)
9     {
10         if (equals(P.x, Q.x))
11         {
12             vertical = true;
13             b = P.x;
14         } else
15         {
16             m = (Q.y - P.y)/(Q.x - P.x)
17             b = P.y - m * P.x
18         }
19     }
20 };
```

# Equação geral da reta

- A equação geral da reta é dada por

$$ax + by + c = 0$$

- Como dito, a equação geral pode representar retas verticais ( $b = 0$ )
- Nos demais casos, é possível obter a equação reduzida a partir da equação geral

```
1  template<typename T>
2  struct Line {
3      T a, b, c;
4
5      Line(T av, T bv, T cv) : a(av), b(bv), c(cv) {}
6  };
```



# Equação geral da reta a partir de dois pontos

- Dados dois pontos distintos  $P = (x_p, y_p)$  e  $Q = (x_q, y_q)$ , é possível obter os coeficientes da equação geral da seguinte maneira:
  1. Substitua as coordenadas de um dos dois pontos na equação geral
  2. Encontrado o valor de  $c$ , substitua as coordenadas de ambos pontos na equação geral, obtendo um sistema linear
  3. Os valores de  $a$  e  $b$  são a solução deste sistema linear
- Este processo pode ser simplificado através do uso de Álgebra Linear: se três pontos  $P = (x_p, y_p)$ ,  $Q = (x_q, y_q)$  e  $R = (x, y)$  são colineares (isto é, pertencem a uma mesma reta), então

$$\det \begin{bmatrix} x_p & y_p & 1 \\ x_q & y_q & 1 \\ x & y & 1 \end{bmatrix} = 0$$

- A solução da equação acima é

$$a = y_p - y_q, b = x_q - x_p, c = x_p y_q - x_q y_p$$

# Implementação da reta através da equação geral

```
1 // Definição da classe Point
2
3 template<typename T>
4 struct Line {
5     T a, b, c;
6
7     Line(const Point& P, const Point& Q)
8         : a(P.y - Q.y), b (Q.x - P.x), c(P.x * Q.y - Q.x * P.y)
9     {
10    }
11 };
```

## Observações sobre a equação geral da reta

- Um mesma reta pode ter infinitas equações gerais associadas: basta multiplicar toda a equação por uma número real diferente de zero
- Para normalizar a representação, associando uma única equação a cada reta, é necessário dividir toda a equação pelo coeficiente  $a$  (ou por  $b$ , caso  $a$  seja igual a zero)
- Esta estratégia permite a simplificação de algoritmos de comparação entre retas
- Por outro lado, não uniformizar a representação permite manter os coeficientes inteiros, caso as coordenadas dos pontos sejam inteiras
- Importante notar que, em ambas representações, pode acontecer da reta resultante ser degenerada
- Isto ocorre quando os pontos  $P$  e  $Q$  são idênticos: neste caso, a reta se reduz a um único ponto
- O tratamento deste caso especiais nos demais algoritmos aumenta o tamanho e a sofisticação dos códigos
- Porém o não tratamento de casos especiais pode levar ao WA

# Demonstrações\*

---

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. **LAAKSONEN**, Antti. *Competitive Programmer's Handbook*, 2018.
3. **De BERG**, Mark; **CHEONG**, Otfried. *Computational Geometry: Algorithms and Applications*, 2008.
4. David E. Joyce. *Euclid's Elements*. Acesso em 15/02/2019<sup>1</sup>
5. Wikipédia. *Geometria Euclidiana*. Acesso em 15/02/2019<sup>2</sup>.

---

<sup>1</sup><https://mathcs.clarku.edu/~djoyce/elements/bookI/defI2.html>

<sup>2</sup>[https://pt.wikipedia.org/wiki/Geometria\\_euclidiana](https://pt.wikipedia.org/wiki/Geometria_euclidiana)