

Geometria Computacional

Triângulos

Prof. Edson Alves

2018

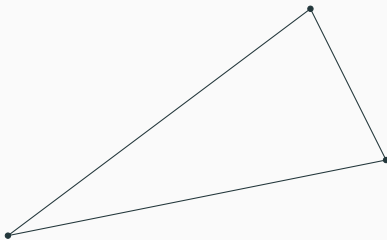
Faculdade UnB Gama

1. Definição de triângulo
2. Classificação de um triângulo
3. Perímetro e área
4. Lugares Geométricos

Definição de triângulo

Definição

- Um triângulo é uma figura geométrica fechada composta por três pontos não-colineares, denominados vértices, e os segmentos de retas formados pelos três pares possíveis entre estes três pontos, denominadas arestas ou lados
- A cada vértice está associado um ângulo, definido pelos dois segmentos de reta dois quais o vértice é um dos extremos
- O triângulo é o mais simples dentre os polígonos, mas possui uma série de características e propriedades notáveis



Representação de um triângulo

- Um triângulo pode ser representado pelas coordenadas de seus vértices
- Outra alternativa é representar o triângulo pelos tamanhos de suas arestas
- É possível deduzir estas valores a partir da primeira representação
- Porém há infinitas possibilidades de coordenadas que satisfaçam as três medidas, uma vez que translações e rotações preservam tais valores
- A representação por vértices pode incluir a representação de um triângulo degenerado (quando os três pontos são colineares)
- A representação por medidas inclui mais casos especiais: pode ser que tais medidas não formem um triângulo
- A Desigualdade Triangular diz que, se a, b, c são números reais, eles serão medidas dos lados de um triângulo se, e somente se,

$$a \leq b + c, \quad b \leq a + c, \quad c \leq a + b$$

Exemplo de representação do triângulo pelos vértices

```
1 // Definição da classe Ponto
2
3 template<typename T>
4 struct Triangle {
5     Point<T> A, B, C;
6 };
```

Classificação de um triângulo

Classificação de acordo com as medidas dos lados

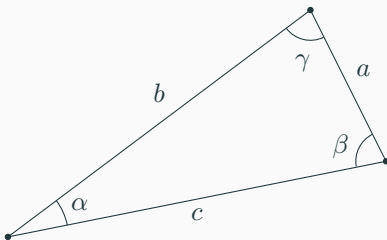
- Sejam a, b, c as medidas dos três lados de um triângulo T
- T é dito equilátero se $a = b = c$
- Se dois lados tem medidas iguais e o terceiro tem medida diferente, T é dito isóceles
- Se $a \neq b \neq c$ o triângulo T é denominado escaleno

Implementação da classificação por medidas dos lados

```
1  template<typename T>
2  struct Triangle {
3      Point<T> A, B, C;
4
5      enum Sides { EQUILATERAL, ISOSCELES, SCALENE };
6
7      Sides classification_by_sides() const
8      {
9          auto a = dist(A, B);
10         auto b = dist(B, C);
11         auto c = dist(C, A);
12
13         if (equals(a, b) and equals(b, c))
14             return EQUILATERAL;
15
16         if (equals(a, b) or equals(a, c) or equals(b, c))
17             return ISOSCELES;
18
19         return SCALENE;
20     }
21 };
```

Classificação de acordo com as medidas dos ângulos internos

- Sejam α, β, γ os ângulos internos de um triângulo T
- Se um destes três ângulos for igual a 90° , T é dito retângulo
- Se um destes três ângulos for maior do que 90° , T é denominado obtusângulo
- Se $\alpha, \beta, \gamma < 90^\circ$, T é chamado acutângulo
- Importante: $\alpha + \beta + \gamma = 180^\circ$



Relação entre medidas dos lados e ângulos

- A Lei dos Cossenos nos diz que

$$a^2 = b^2 + c^2 - 2bc \cos \alpha,$$

- Esta lei permite determinar o ângulo oposto ao um lado escolhido:

$$\alpha = \cos^{-1} \left(\frac{b^2 + c^2 - a^2}{2bc} \right)$$

- Observe que, se $\alpha = 90^\circ$, a Lei dos Cossenos se reduz ao Teorema de Pitágoras
- A Lei dos Senos também relaciona lados e ângulos, com o bônus de permitir determinar o raio R do círculo que circunscreve o triângulo:

$$\frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma} = 2R$$

Implementação da classificação por ângulos internos

```
1 // Definição da classe Point, da função de comparação equals() e
2 // da função de distância entre pontos dist()
3
4 template<typename T>
5 struct Triangle {
6     Point<T> A, B, C;
7
8     enum Angles { RIGHT, ACUTE, OBTUSE };
9
10    Angles classification_by_angles() const
11    {
12        auto a = dist(A, B);
13        auto b = dist(B, C);
14        auto c = dist(C, A);
15
16        auto alpha = acos((a*a - b*b - c*c)/(-2*b*c));
17        auto beta = acos((b*b - a*a - c*c)/(-2*a*c));
18        auto gamma = acos((c*c - a*a - b*b)/(-2*a*b));
19
20        auto right = PI / 2.0;
21
```

Implementação da classificação por ângulos internos

```
22     if (equals(alpha, right) || equals(beta, right)
23         || equals(gamma, right))
24         return RIGHT;
25
26     if (alpha > right || beta > right || gamma > right)
27         return OBTUSE;
28
29     return ACUTE;
30 }
31 };
```

Perímetro e área

Perímetro de um triângulo

- O perímetro de um triângulo é dado pela soma das medidas de seus lados
- Em notação matemática, se o triângulo T tem lados com medidas a, b, c , o perímetro P é dado por

$$P = a + b + c$$

```
1  template<typename T>
2  struct Triangle {
3      Point<T> A, B, C;
4
5      double perimeter() const
6      {
7          auto a = dist(A, B);
8          auto b = dist(B, C);
9          auto c = dist(C, A);
10
11         return a + b + c;
12     }
13 };
```

Área de um triângulo

- Há três formas de se computar a área de um triângulo
- A primeira delas é usar a fórmula ensinada no ensino médio

$$A = \frac{bh}{2}$$

onde b é a medida da base do triângulo (um de seus lados) e h é a altura segmento de reta, perpendicular à base, com um ponto sobre a base e o outro no vértice oposto a esta)

- Contudo, na representação por pontos ou por medidas, esta abordagem é pouco prática, pois envolve o cálculo da altura
- A altura pode ser obtida como a distância do vértice oposto até a base

Cálculo da área por base e altura

```
1 // Definição das estruturas Point e Line
2
3 template<typename T>
4 struct Triangle {
5     Point<T> A, B, C;
6
7     double area() const
8     {
9         Line<T> r(A, B);
10
11         auto b = dist(A, B);
12         auto h = r.distance(C);
13
14         return (b * h)/2;
15     }
16 };
```

Fórmula de Heron

- A segunda maneira de se obter a área de um triângulo é utilizar a fórmula de Heron:

$$A = \sqrt{s(s-a)(s-b)(s-c)},$$

onde A é a área do triângulo de lados a, b, c e s é o semiperímetro, isto é, a metade do perímetro:

$$s = \frac{a + b + c}{2}$$

- Esta abordagem é a mais apropriada na representação do triângulo pela medida de seus lados
- Observe que, caso exista a possibilidade de *overflow* no produto dos quatro termos que estão dentro da raiz, deve-se tirar a raiz de cada fator antes de fazer o produto

Cálculo da área usando a fórmula de Heron

```
1 // Definição da estrutura Point
2
3 template<typename T>
4 struct Triangle {
5     Point<T> A, B, C;
6
7     double area() const
8     {
9         auto a = dist(A, B);
10        auto b = dist(B, C);
11        auto c = dist(C, A);
12
13        auto s = (a + b + c)/2
14
15        return sqrt(s)*sqrt(s - a)*sqrt(s - b)*sqrt(s - c);
16    }
17 };
```

Variantes da fórmula de Heron

- Existem variantes da fórmula de Heron que permitem o cálculo da área do triângulo em termos de outras medidas, como as medianas, as alturas ou os ângulos internos
- Se m_a, m_b, m_c são as medidas das medianas, então

$$A = \frac{4}{3} \sqrt{\sigma(\sigma - m_a)(\sigma - m_b)(\sigma - m_c)}, \quad \sigma = \frac{m_a + m_b + m_c}{2}$$

- Se h_a, h_b, h_c são medidas as alturas, então

$$\frac{1}{A} = 4 \sqrt{H \left(H - \frac{1}{h_a} \right) \left(H - \frac{1}{h_b} \right) \left(H - \frac{1}{h_c} \right)}$$

com

$$H = \frac{1}{2} \left(\frac{1}{h_a} + \frac{1}{h_b} + \frac{1}{h_c} \right)$$

Variantes da fórmula de Heron

- Por fim, sendo usando a notação de lados e ângulos já estabelecida, é possível computar a área, conhecidos os três ângulos e apenas um dos três lados:

$$A = D^2 \sqrt{S(S - \sin \alpha)(S - \sin \beta)(S - \sin \gamma)},$$

onde

$$S = \frac{\sin \alpha + \sin \beta + \sin \gamma}{2}$$

e

$$D = \frac{a}{\sin \alpha} = \frac{b}{\sin \beta} = \frac{c}{\sin \gamma}$$

Cálculo da área por coordenadas dos vértices

- A terceira maneira é computar a área a partir das coordenadas dos vértices
- Se os vértices de um triângulo são $P = (x_1, y_1)$, $Q = (x_2, y_2)$, $R = (x_3, y_3)$, a área A do triângulo é dada por

$$A = \frac{1}{2} \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix}$$

- Esta área é sinalizada: logo deve se considerar o valor absoluto desta expressão
- Observe que não é necessário implementar uma estrutura que represente matrizes e a operação de determinante
- A expansão da expressão acima leva a três termos positivos e a três termos negativos

Cálculo da área por determinante

```
1 // Definição da estrutura Point
2
3 template<typename T>
4 struct Triangle {
5     Point<T> A, B, C;
6
7     double area() const
8     {
9         double det = (A.x*B.y + A.y*C.x + B.x*C.y)
10                    - (C.x*B.y + C.y*A.x + B.x*A.y);
11
12         return 0.5 * fabs(det);
13     }
14 };
```

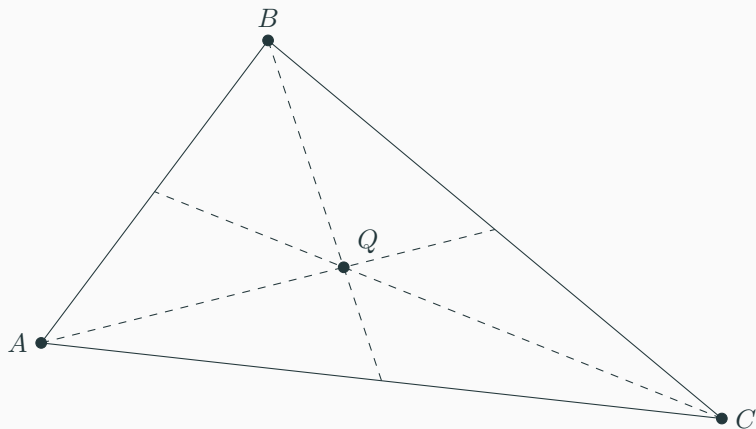
Lugares Geométricos

Baricentro

- Um triângulo possui quatro lugares geométricos notáveis
- O primeiro deles é o baricentro (centróide ou centro de massa), que é o ponto de interseção entre as três medianas (segmentos de reta que unem um vértice ao ponto médio do lado oposto)
- O baricentro divide uma mediana na proporção de 2:1, isto é, ele está a um terço de distância do lado oposto
- As coordenadas do baricentro Q podem ser computadas diretamente a partir das coordenadas dos vértices: serão a média aritmética entre as mesmas

$$Q = \left(\frac{x_A + x_B + x_C}{3}, \frac{y_A + y_B + y_C}{3} \right)$$

Visualização do baricentro



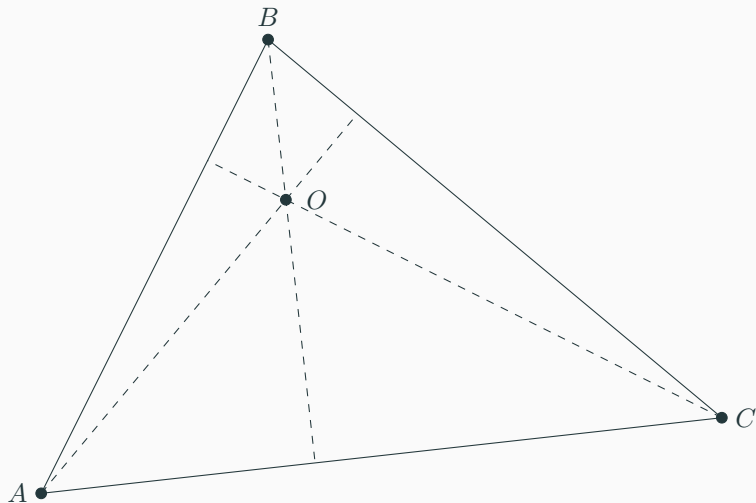
Implementação da identificação do baricentro

```
1 // Definição da estrutura Point
2
3 template<typename T>
4 struct Triangle {
5     Point<T> A, B, C;
6
7     Point<T> barycenter() const
8     {
9         auto x = (A.x + B.x + C.x) / 3.0;
10        auto y = (A.y + B.y + C.y) / 3.0;
11
12        return Point<T> { x, y };
13    }
14 };
```

Ortocentro

- O ortocentro de um triângulo é o ponto de encontro de suas três alturas
- O ortocentro pode mesmo estar fora do triângulo, no caso de um obtusângulo
- No caso de um triângulo retângulo, o ortocentro sempre coincide com o vértice oposto à hipotenusa
- Para obter as coordenadas do ortocentro, é preciso determinar, inicialmente, as retas r e s que contêm os segmentos AB e AC , respectivamente,
- Em seguida, é preciso determinar as retas u e v perpendiculares a r e s que passam por C e B , respectivamente
- O ortocentro O será a interseção entre u e v

Visualização do ortocentro



Implementação da identificação do ortocentro

```
22 template<typename T>
23 struct Triangle {
24     Point<T> A, B, C;
25
26     Point<T> orthocenter() const
27     {
28         Line<T> r(A, B), s(A, C);
29
30         Line<T> u { r.b, -r.a, -(C.x*r.b - C.y*r.a) };
31         Line<T> v { s.b, -s.a, -(B.x*s.b - B.y*s.a) };
32
33         auto det = u.a * v.b - u.b * v.a;
34         auto x = (-u.c * v.b + v.c * u.b) / det;
35         auto y = (-v.c * u.a + u.c * v.a) / det;
36
37         return { x, y };
38     }
39 };
40
```

1. **HALIM**, Felix; **HALIM**, Steve. *Competitive Programming 3*, 2010.
2. Math Open Reference. [Orthocenter of a Triangle](#), acesso em 27/07/2016.
3. Math Open Reference. [Incenter of a Triangle](#), acesso em 27/07/2016.
4. Wikipedia. [Circumscribed Circle](#), acesso em 27/07/2016.
5. Wikipedia. [Heron's Formula](#), acesso em 22/09/2016.