Geometria Computacional

Círculos

Prof. Edson Alves

2018

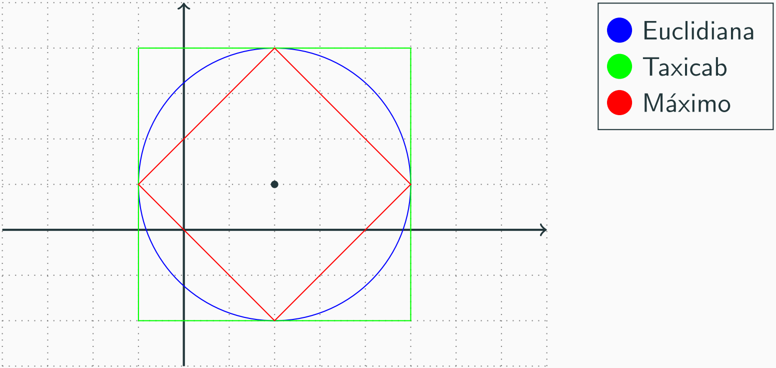
Faculdade UnB Gama

Sumário

Definição de círculo

Definição

- Um círculo é o conjunto de pontos equidistantes de um ponto C
- A distância de um ponto do círculo ao seu centro C é denominada raio r do círculo
- Observe que a visualização do círculo depende da norma utilizada
- As figuras abaixo representação círculos com centro no ponto $(2, 1)$ e com raio $r = 3$



Representação de círculos

- Um círculo pode ser representado através do ponto C e do raio r
- A equação do círculo pode ser deduzida a partir da expressão $d(P, C) = r$, onde $P = (x, y)$ é um ponto do círculo, $C = (x_0, y_0)$ é o centro do círculo e r é o raio
- A equação geral do círculo é dada por

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

- Esta equação é útil para resolver vários problemas envolvendo círculos, como o problema de determinar se um ponto está dentro, fora ou sobre um círculo

```cpp
// Definição da classe Point

template<typename T>
struct Circle {
    Point<T> C;
    T r;
};
```

Características do círculo

A constante π

- Tanto o cálculo do perímetro quanto da área de um círculo envolvem o uso da constante π
- Caso o problema não informe o valor a ser utilizado, há três maneiras de proceder para determinar o valor desta constante
- A primeira é utilizar o valor definido na linguagem Python, que pode ser obtido com o script abaixo.

```python
from math import *
print pi
```

- O valor resultante, 3.141592653589793, está correto nas suas 15 casas decimais
- A segunda forma é utilizar a expressão `acos(-1.0)` em C/C++
- A terceira é usar a macro `M_PI` da biblioteca de matemática padrão do C/C++

Perímetro do círculo

- O perímetro (circunferência) C de um círculo corresponde ao o comprimento do contorno do círculo
- Este valor pode ser computado como o perímetro de um polígono regular de n lados e raio circunscrito r (distância do centro a um dos vértices do polígono), quando n tende a infinito
- O lado L de tal polígono e o raio se relacionam de modo que

$$\sin \frac{\pi}{n} = \frac{L/2}{r}$$

- Assim,

$$C = \lim_{n \to \infty} n \left(2r \sin \frac{\pi}{n} \right) = 2r \left(\lim_{n \to \infty} n \sin \frac{\pi}{n} \right) = 2\pi r$$

Implementação do perímetro em C/C++

```cpp
// Definição do valor de PI

template<typename T>
struct Circle
{
    // Membros e construtores

    double perimeter() const
    {
        return 2.0 * PI * r;
    }
};
```

Área do círculo

- De modo semelhante, a área A de um círculo pode ser aproximada pela área de um polígono regular de n lados e raio circunscrito r quando n tende ao infinito
- A base de cada triângulo interno é igual a L
- A altura é a apótema a, onde

$$\cos \frac{\pi}{n} = \frac{a}{r}$$

- Assim,

$$A = \lim_{n \to \infty} n \left(\frac{La}{2} \right) = \lim_{n \to \infty} n \left(r \sin \frac{\pi}{n} \right) \left(r \cos \frac{\pi}{n} \right),$$

isto é,

$$A = r^2 \left(\lim_{n \to \infty} n \sin \frac{\pi}{n} \cos \frac{\pi}{n} \right) = r^2 \lim_{n \to \infty} n \left(\frac{\sin \frac{2\pi}{n}}{2} \right) = \pi r^2$$

Implementação da área do círculo em C++

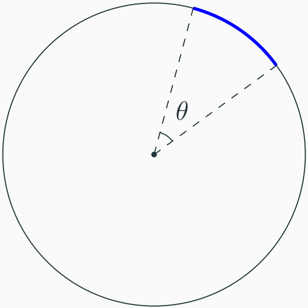
```cpp
// Definição do valor de PI

template<typename T>
struct Circle
{
    // Membros e construtores

    double area() const
    {
        return PI * r * r;
    }
};
```

Arcos

- Um arco de um círculo corresponde a uma seção conectada da circunferência
- O comprimento do arco pode ser determinado através do ângulo central θ, definido pelos vetores gerados pela união de cada um dos pontos extremos do arco com o centro do círculo
- Um ângulo de 2π gera um arco igual a circunferência, isto é, $2\pi r$
- Usando a proporcionalidade e a regra de três, um ângulo θ corresponde a um arco igual a θr



Implementação do arco de um círculo em C/C++

```cpp
template<typename T>
struct Circle
{
    // Membros e construtores

    // A unidade de medida do theta é radianos
    double arc(double theta) const
    {
        return theta * r;
    }
};
```

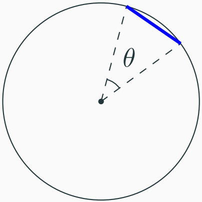
Corda de um círculo

- Uma corda corresponde a qualquer segmento de reta cujos pontos extremos estão sob o círculo
- O diâmetro é a maior dentre todas as cordas possíveis de um círculo, e tem comprimento $d = 2r$
- Conhecidos o raio r e o ângulo central θ do arco definido pela corda, o comprimento L da corda pode ser determinado através da Lei dos Cossenos

$$L = \sqrt{2r^2(1 - \cos\theta)}$$

ou pela Trigonometria

$$L = 2r\sin(\theta/2)$$



Implementação do comprimento de uma corda de um círculo em C/C++

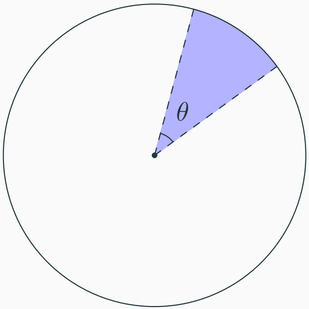
```cpp
template<typename T>
struct Circle
{
    // Membros e construtores

    // A unidade de medida do theta é radianos
    double chord(double theta) const
    {
        return 2 * r * sin(theta/2);
    }
};
```

Setores

- Um setor de um círculo é a área delimitada por um arco cujo ângulo central é θ
- Assim como no caso do arco, a área do setor será a fração da área total correspondente ao ângulo central θ do arco que delimita o setor
- Assim, a área S do setor é dada por

$$S = \frac{\theta r^2}{2}$$



Implementação do setor de um círculo em C++

```cpp
template<typename T>
struct Circle
{
    // Membros e construtores

    // A unidade de medida do theta é radianos
    double sector(double theta) const
    {
        return (theta * r * r)/2;
    }
};
```

Segmento

- Um segmento de um círculo associado a um ângulo central θ corresponde à área resultante da diferença entre o setor delimitado por θ e do triângulo resultante do segmentos de reta que unem os extremos dos arcos ao centro do círculo e os extremos entre si (a corda)
- A área T deste triângulo pode ser determinada pela Fórmula de Heron

$$T = \sqrt{(s-r)(s-r)(s-c)},$$

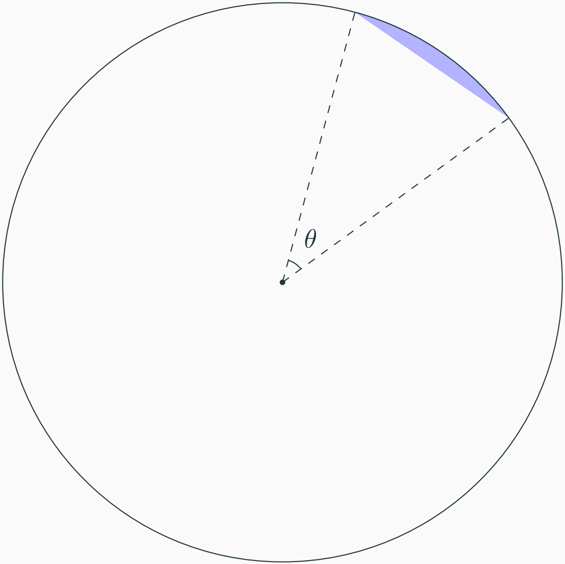
onde s é o semiperímetro, dado por

$$s = \frac{2r+c}{2}$$

e c o comprimento da corda, isto é

$$c = 2r\sin(\theta/2)$$

Visualização do segmento de um círculo



Implementação do segmento de um círculo em C/C++

```cpp
template<typename T>
struct Circle
{
    // Membros e construtores

    // Método sector()

    // A unidade de medida do theta é radianos
    double segment(double a) const
    {
        auto c = chord(a);
        auto s = (r + r + c)/2.0;                      // Semiperímetro
        auto T = sqrt(s*(s - r)*(s - r)*(s - c));    // Área do triângulo

        return sector(a) - T;
    }
};
```

Referências

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.