

Exercícios



Aproximações de pi

As aproximações do valor de pi costumam serem calculados através de soma de séries infinitas, a precisão é relacionada com a velocidade de convergência da série.

Série de Ramanujan

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

cada termo aproxima oito dígitos de pi. Base para diversos algoritmos para cálculo de pi.

Série de Ramanujan

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{k=0}^{\infty} \frac{(4k)!(1103 + 26390k)}{(k!)^4 396^{4k}}$$

Quantas funções você visualiza para calcular pi com essa fórmula? Esboce um programa modularizado para isso!

Chudnovsky

$$\frac{1}{\pi} = 12 \sum_{k=0}^{\infty} \frac{(-1)^k (6k)! (13591409 + 545140134k)}{(3k)! (k!)^3 640320^{3k+3/2}}.$$

Baseado na série de Ramanujan. Dado o programa anterior, você pode reaproveitar alguma das funções criadas?

Soma da área da Circunferência

Sabemos que a área da circunferência é:

$$A = \pi \cdot r^2$$

Soma da área da Circunferência

Sabemos que a área da circunferência é:

$$A = \pi \cdot r^2$$

Se fizermos uma circunferência com raio ($\frac{1}{2}$) temos:

$$A = \pi / 4$$

Soma da área da Circunferência

Essa circunferência pode ser inscrita em um quadrado de lado = 1, sendo o centro $(0.5, 0.5)$.

Com isso podemos fazer uma integração de Monte Carlo com o seguinte procedimento.

Soma da área da Circunferência

Sorteie dois números aleatórios entre 0 e 1, representando as coordenadas (x, y) de um ponto.

Se esse ponto estiver dentro da circunferência incrementamos +1 em `ptos_circ`.

Além disso, incrementamos +1 em `total`.

Soma da área da Circunferência

Após vários sorteios, temos que $\text{ptos_circ} / \text{total}$ aproxima a área da circunferência, que aproxima $\pi / 4$.

Implemente a função que atualiza utilizando variável do tipo `static` para `ptos_circ` e `total`.

rand()

Na biblioteca **stdlib.h**, você alimenta o seed com **srand()** e obtém um número aleatório com **rand()**:

```
#include <stdlib.h>
#include <time.h>
int main ( )
{
    srand(time(NULL));
    printf("%d\n", rand()); /* inteiro [0, RAND_MAX] */
    return 0;
}
```

Alterando o range

`rand()` de $[0, n[$:

`rand() % n;` /* não uniforme para n grande */

`floor(rand() * n / (RAND_MAX + 1));`