

INSTITUTO FEDERAL  
CATARINENSE  
Câmpus Videira

Turma: 1 fase - Ciência da computação

Aluno: Bruno Pergher

Disciplina: Algoritmos

Professor: Manasses Ribeiro

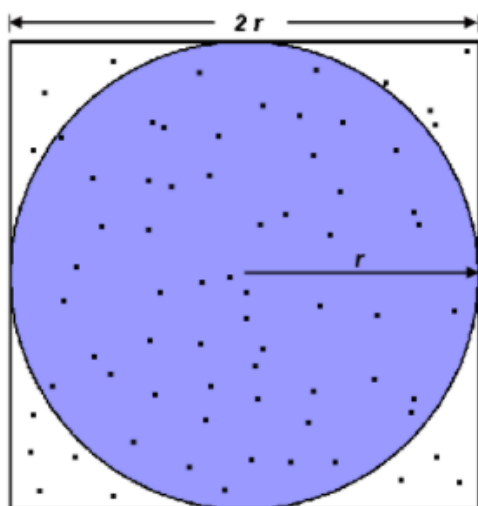
---

## Cálculo do valor de PI pelo método de Monte Carlo implementado em C

### 1. Problema

O presente trabalho tem como objetivo entender a lógica mostrada por Monte Carlo e aplicá-la para o cálculo do valor aproximado de PI, implementado a um sistema criado com a linguagem de programação C.

O método surgiu no tempo da 2 guerra mundial, onde se baseava principalmente na geração de pontos aleatórios em um área quadrada e assim seguindo uma linha tênue de raciocínio aplicado às fórmulas referentes a circunferência era possível saber se o ponto aleatório fazia parte ou não da circunferência utilizando o valor desse ponto, seguindo a base que a área de um quadrado e de um círculo possuindo o mesmo valor do raio pode se criar uma relação onde a área do quadrado é equivalente a  $4r^2$  e a da circunferência  $\pi r^2$  podendo formular a seguinte situação:



$$\text{área Quadrado} / \text{área Círculo} = \pi / 4 \Rightarrow$$

$$\pi = 4 * \text{área Quadrado} / \text{área Círculo},$$

Por fim é proposto que a realização de **muitos** pontos aleatórios dentro dessa figura ao serem verificados se fazem parte ou não do círculo constroem uma última relação necessária para o cálculo: área do quadrado = total de pontos aleatórios criados.

área do círculo = pontos gerados dentro do círculo.

### 2. Desenvolvimento do algoritmo

Após analisar o problema e a lógica presente nele, o desafio mais importante é realizar uma validação para sabermos quando o número está dentro ou não do círculo, após algumas pesquisas e “quebrar a cabeça” cheguei a conclusão que usufruindo da fórmula da circunferência no plano cartesiano que é  $x^2 + y^2 = r^2$  temos a possibilidade de calcular a borda do círculo que vai ser igual a  $r^2$ , assim qualquer valor igual ou menor o  $r^2$  está dentro do círculo, assim podendo aplicar a validação da quantidade de pontos presentes no

círculo, baseando se em uma figura que possui uma dimensão de 2x2 conseguimos concluir que o raio seguindo o conceito de que o raio é igual a área lateral dividido por 2, resultando em 1 e como  $1^2$  é 1 a validação foi feita utilizando raio igual a 1, como mostrado a seguir:

```
if (pp -> x * pp -> x + pp -> y * pp -> y <= 1)
{
    insideCircle++;
}
```

Podendo validar os pontos dentro do círculo tempos mais algumas etapas importantes a geração de números aleatórios sem padrões, selecionar a quantidade de números aleatórios e o cálculo de pi, primeiro vamos decidir a quantidade de pontos a serem criados, essa informação irá ser dada pelo usuário mas tendo em vista que quanto mais pontos maior é a chance da aproximação ficar melhor, após isso vamos gerar números aleatórios dentro de uma figura 2x2, onde decidi gerar números aleatórios de -1 até 1 utilizando a função rand com uma pequena configuração, como podemos ver na imagem abaixo:

```
srand(time(NULL));

for (i = 0; i < n; i++)
{
    n1 = (double)rand() / RAND_MAX * 2.0 - 1.0;
    n2 = (double)rand() / RAND_MAX * 2.0 - 1.0;
    pp = gerarPonto(n1, n2);

    if (pp -> x * pp -> x + pp -> y * pp -> y <= 1)
    {
        insideCircle++;
    }
    free(pp);
}

return (double)4 * insideCircle / n;
}
```

Nesse momento já possuindo a informação de quantos pontos serão gerados é criando as coordenadas aleatórias já ocorre a validação, após todo esse processo possuímos todas as informações necessárias para calcular o PI, sendo  $\pi = 4 * \text{pontos no círculo} / \text{pontos totais}$ .

### 3. Conclusão

Ao testarmos o código e analisarmos que tudo é fruto de números aleatórios, é perceptível que quanto maior o número de pontos, maior será a chance de a aproximação

ter um melhor resultado, dizendo isso temos em base resultados feitos pelo algoritmo criado onde o menor número que consegui alcançar as primeiras 2 casas de pi foi de 200 pontos,

```
C:\Windows\system32\cmd.exe
=====
Descubra o valor de PI segundo o m|t|do de Monte Carlo
=====

Escolha a quantidade de pontos aleatorios a serem gerados(Quanto mais, mais aproximado vai ser o resultado)
200
3.140000
Press any key to continue . . .
```

porém ainda são poucos dados, ao trabalharmos com grandes quantidade de pontos se torna mais comum alcançarmos o mesmo resultado, chegando até mesmo a alcançar os 4 primeiros decimais de PI como ocorre no teste com 600000 pontos gerados:

```
C:\Windows\system32\cmd.exe
=====
Descubra o valor de PI segundo o m|t|do de Monte Carlo
=====

Escolha a quantidade de pontos aleatorios a serem gerados(Quanto mais, mais aproximado vai ser o resultado)
600000
3.141553
Press any key to continue . . .
```

Assim concluindo que apesar de não ser o método mais preciso de se calcular o valor de PI é sim possível chegarmos a proximidade do mesmo apenas partindo de valores aleatórios.