Instituto Federal Catarinense - Campus Videira

Turma - Ciência da computação - 1º semestre 2019

Aluno: Pedro Henrique Fagundes

Trabalho Final.

**Problema 2: Calcular PI pelo método de monte Carlo**

Oque é o método de Monte Carlo? Essa técnica foi, e é usado para várias áreas de estudo, como por exemplo no mundo financeiro, calculando e analisando o mercado para investimentos, além disso esse método foi muito importante na Segunda Guerra Mundial, sendo parte da construção das primeiras bombas atômicas, ele consiste em fazer simulações através de contas matemáticas e algoritmos criando vários cenários diferentes com valores aleatórios, assim, procurando o que chega mais perto do seu resultado desejado.

Nesse experimento, vamos usar o método de Monte Carlo para descobrir o valor de PI, será aplicado em algoritmos mais especificamente em linguagem c. Como já explicado o método faz simulações com valores aleatórios, então como cenário base usaremos esse gráfico:

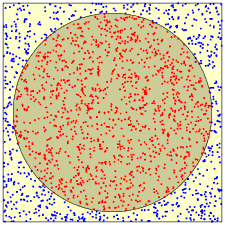


Figura 1

Como esse gráfico se encaixa no método de Monte Carlo? Como se pode ver, a figura 1 se compõe de um quadrado e dentro dele um círculo, seguidos de vários pontos azuis e vermelhos. Como dito anteriormente o método de Monte Carlo gera e lê valores aleatórios, pois bem, cada ponto dentro desta figura é um valor gerado aleatoriamente dentro do cenário, com isso podemos fazer um algoritmo que calcule o valor de PI.

Para começar precisamos entender as operações matemáticas e relações geométricas que vamos aplicar.

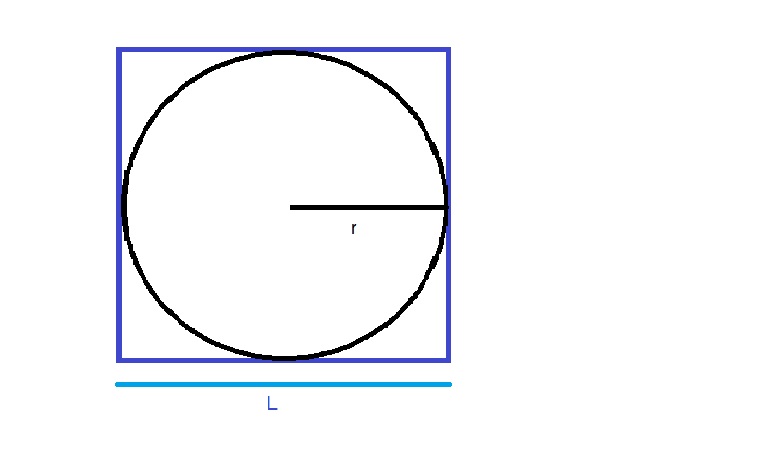
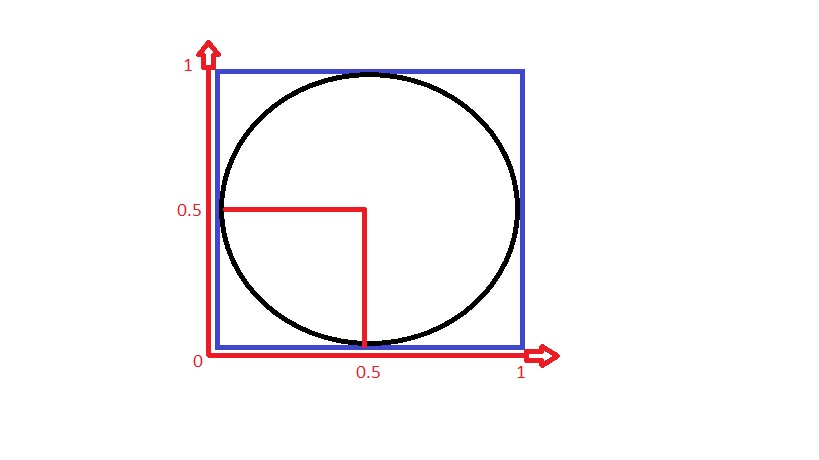


Figura 2

Tendo como base a figura 2, já temos por conhecimento que a **área da circunferência** = **π** \* **r²**, em que **r** é o raio da circunferência, e também a **área do quadrado** = **L²**, onde L é o lado do quadrado. Fazendo uma relação entre o **r** do círculo e o **L** quadrado podemos transformar de **L²** para **4r²**, pois **L** é a mesma coisa que **(**2**r) ²**. Sendo a formula de PI: **π = área do circulo / área do quadrado (π r² / 4r²),** podemos cancelar os elementos em comum tendo por final **π = 4 \* área do círculo / área do quadrado.**

Definindo a formula do **π**, vamos compreender quais operações matemáticas iremos usar e entende-las:

Figura 3

Para melhor compreensão vamos imaginar que o cenário do problema é um plano cartesiano, tendo coordenadas (x, y) com o tamanho máximo 1. Pois então, como dito anteriormente no método de Monte Carlo é gerado pontos aleatórios dentro do quadrado e do círculo, automaticamente cada ponto irá ficar em uma coordenada do plano cartesiano (x, y), sendo assim, alguns pontos irão ficar fora da área do círculo, enquanto outros irão ficar dentro. Mas como vamos saber isso por meio de um algoritmo? Pois bem, nós vamos determinar se os pontos estam dentro ou fora da circunferência, por meio do cálculo da distância euclidiana, que calcula a distância entre dois pontos, sendo o ponto de origem no meio da circunferência (x = 0.5, y = 0.5), como mostra na figura 3.

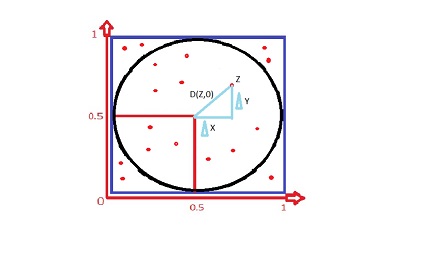
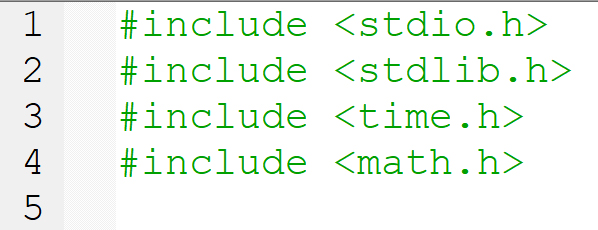


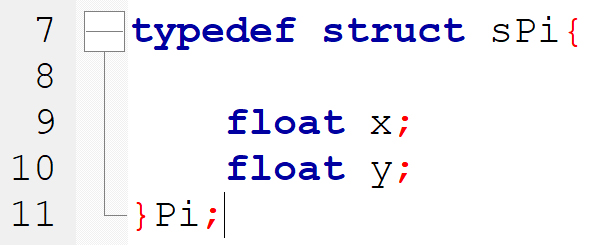
Figura 4

Com base na figura 4, é escolhido um ponto Z que foi aleatoriamente gerado, calculamos a distância entre ele e o ponto de origem. Se traçar uma linha entre o ponto Z e a origem percebesse que se forma um triangulo retângulo, com sua hipotenusa D(Z,0), e seus catetos que são dados como Δx =( xZ – 0.5), e Δy =( yZ – 0.5), nisso podemos aplicar o teorema de Pitágoras onde a formula seria : D(Z, 0) = √ ( Δx)² + (Δy)² , com a formula de Pitágoras conseguiremos calcular a distância entre os dois pontos, onde se o ponto tiver um valor menor ou igual a 0.5, ele vai estar dentro do círculo, se tiver um valor maior que 0.5, estará fora.

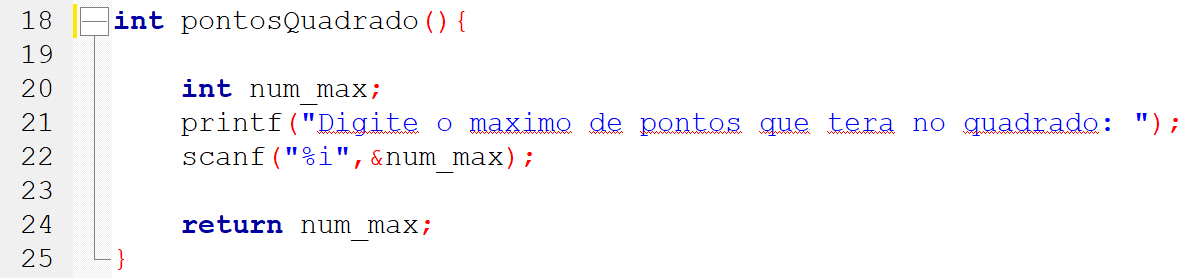
Com todos os valores e operações definidos podemos iniciar assim o nosso algoritmo. Primeiramente vamos definir as bibliotecas necessarias.



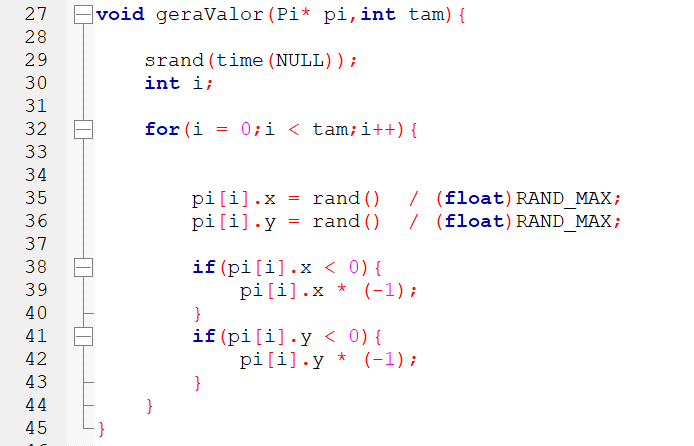
Após isso, abrimos um registro ***< typedef struct >***, com as variáveis *float x* e *float y* que guardará os valores das coordenadas.



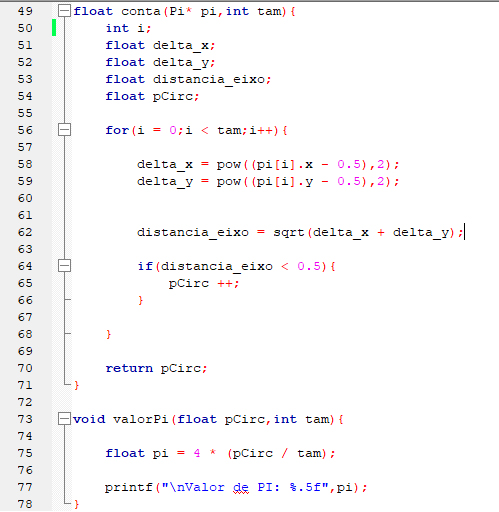
Então começamos a definir os valores, abrimos uma função *int pontosQuadrados,* onde o usuário irá definir o máximo de pontos que irá ter no cenário.



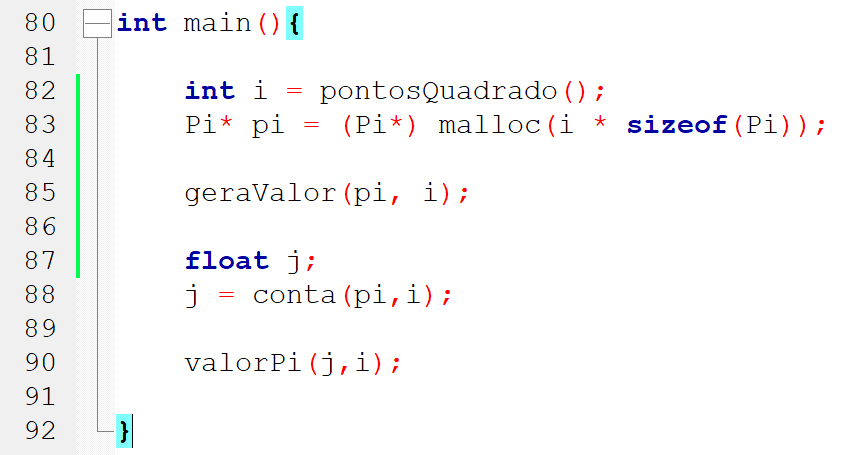
Com o cenário já delimitado, geramos as coordenadas aleatórias *x , y* de 0 até 1.



Com os valores gerados e guardados em sua respectivas variáveis, vazemos as operações matemáticas para saber se o pontos esta dentro ou fora do círculo, e após isso calcular o valor aproximado de **π** com a formula já definida **π = 4 \* área do círculo / área do quadrado.**



Por final para que cada função seja executada, abrimos a função principal *int main.*



.

Referencias:

Camylla Cantanheide, Canal Youtube, vídeo <https://www.youtube.com/watch?v=VIFj-f6fSdM>.

Site GeeksforGeeks, Estimating the value of Pi using method Monte Carlo, <https://www.geeksforgeeks.org/estimating-value-pi-using-monte-carlo/>.