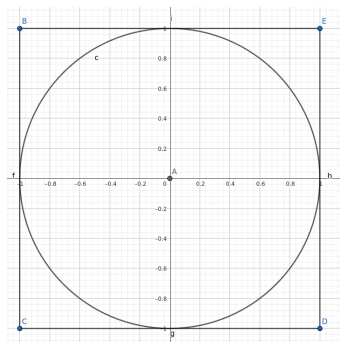


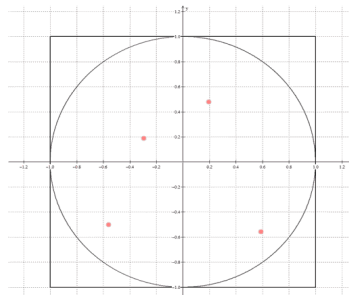
Questão 1 - Estimar a constante π (pi) utilizando o método de monte carlo.

Método de Monte Carlo é um termo utilizado para se referir a qualquer método que resolve um problema gerando números aleatórios e observando se uma dada fração desses números satisfaz uma propriedade previamente estabelecida.

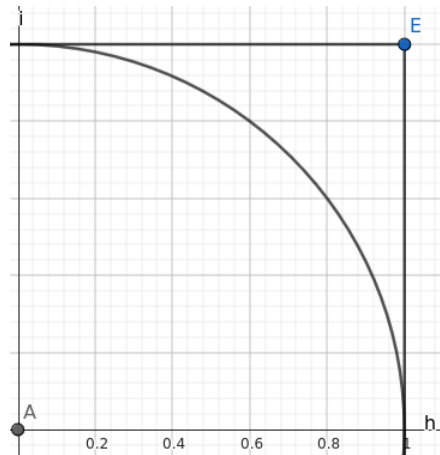
Para estimar a constante, parta do princípio que temos um quadrado de lado 2 ($l = 2$), com uma circunferência de raio 1 ($r = 1$) inscrita nesse quadrado.



Após isso, geramos pontos aleatórios dentro da área do quadrado, dessa maneira:



Sabe-se que a área do círculo é dada por πr^2 e a área do quadrado por l^2 . Para facilitar, utiliza-se apenas um quadrante do círculo:



Por fim, é possível utilizar a relação de proporcionalidade entre número de pontos e as áreas das figuras geométricas. Quanto mais pontos forem gerados, mais o número de pontos se aproxima das áreas das figuras, ou seja

$$\text{Qtd pontos dentro do círculo} / \text{Qtd pontos total} \approx \text{Acírculo} / \text{Aquadrado}$$

Sabendo que a distância entre dois pontos (x,y) se dá por $\sqrt{x^2 + y^2}$, escreva um programa que implemente uma função que recebe o número de pontos a serem gerados pelo usuário, e a partir disso gere os pontos e retorne a constante $\pi(\text{pi})$.

Minha solução em C:

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>

float monte_carlo(int n){
    int points_inside = 0;
    int total_points = 0;

    srand(time(NULL));

    for (size_t i = 0; i < n; ++i) {
        float x = (float)rand() / (float)RAND_MAX;
        float y = (float)rand() / (float)RAND_MAX;

        if (((x * x) + (y * y)) <= 1)
            points_inside += 1;
        total_points++;
    }
    return 4 * (float)points_inside / total_points;
}

int main(int argc, char const *argv[]){
    int n = -1;
    while (n < 0){
        printf("Digite o número de pontos: ");
        scanf("%d", &n);
    }
    float pi = monte_carlo(n);
    printf("Aproximação: %.5f", pi);
    return 0;
}
```

