

# Especificação do Primeiro Trabalho

## Cálculo do número PI

Considere os seguintes algoritmos para realizar o cálculo aproximado do número PI:

- **Gauss-Legendre**
- **Borwein**
- **Monte Carlo**

Desenvolva um programa para calcular o número PI aproximado utilizando os métodos acima. Utilize o número de iterações  $N = 10^9$ . Imprima o resultado com 6 casas decimais. O arquivo *entrada\_pi.txt* contém a entrada do programa (ele está vazio, portanto, para esse trabalho, não há entrada!) e o arquivo *saida\_pi.txt* contém a saída esperada do programa.

## Aplicação da Simulação de Monte Carlo

### Modelo de Black Scholes

Foi elaborado por dois cientistas chamados Fisher Black e Myron Scholes, que adaptaram uma fórmula física para descrever um fenômeno financeiro que é a precificação de derivativos. Este modelo foi proposto pela primeira vez em 1973, e fez seus criadores ganharem o [prêmio Nobel de 1997](http://pt.wikipedia.org/wiki/Black-Scholes) (<http://pt.wikipedia.org/wiki/Black-Scholes>).

É possível utilizar a Simulação de Monte Carlo para se calcular o preço de opções européias utilizando o modelo de Black Scholes. Este algoritmo será apresentado abaixo, juntamente com a explicação dos parâmetros e variáveis.

Variáveis de entrada:

S: valor da ação

E: preço de exercício da opção

r: taxa de juros livre de risco (SELIC)

$\sigma$ : volatilidade da ação

T : tempo de validade da opção

M : número de iterações

---

#### Pseudo-código do algoritmo de Black Scholes com Monte Carlo

---

```
1: for  $i = 0$  to  $M - 1$  do
2:    $t := S \cdot \exp\left((r - \frac{1}{2}\sigma^2) \cdot T + \sigma\sqrt{T} \cdot \text{randomNumber}()\right)$ 
3:   trials[i] :=  $\exp(-r \cdot T) \cdot \max\{t - E, 0\}$ 
4: end for
5: mean := mean(trials)
6: stddev := stddev(trials, mean)
7: confwidth :=  $1.96 \cdot \text{stddev} / \sqrt{M}$ 
8: confmin := mean - confwidth
9: confmax := mean + confwidth
```

▷  $t$  é uma variável temporária

▷ Cálculo do intervalo de confiança

Adaptado de <http://www.cs.bordeaux.fr/~ydlisk/cs194f07>

---

Saída do programa:

- Variáveis de entrada
- Tempo de execução
- Intervalo de confiança (margem inferior e margem superior)

O arquivo *entrada\_blacksholes.txt* contém a entrada do programa e o arquivo *saida\_blacksholes.txt* contém a saída esperada do programa. Uma maneira de testar a corretude do programa é utilizar a calculadora online ([http://www.soarcorp.com/black\\_scholes\\_calculator.jsp](http://www.soarcorp.com/black_scholes_calculator.jsp)), colocando os parâmetros desejados e verificando o valor resultado no campo "Call Price". Ele deve estar incluído no intervalo de confiança do programa.

**ATENÇÃO:** O gerador de números aleatórios *rand* do C é thread-safe. Isso significa que somente uma thread "acessa" essa região de código por vez. Por que não implementar um gerador próprio? Veja o arquivo *rand.c*!

## Para os Programas

- A entrada deve ser lida da entrada padrão e a saída deve ser escrita na saída padrão. Ou seja, não abra arquivos para leitura nem para escrita dos resultados. Leia os dados necessários utilizando *scanf* e imprima os resultados necessários utilizando o *printf*. O programa deve ser executado da maneira abaixo. Assim, o *scanf* lerá os dados do arquivo **entrada.txt** e o *printf* escreverá os dados no arquivo **saida.txt**:

**`./nomedoprograma < entrada.txt > saida.txt`**

- Fazer uma versão sequencial utilizando C
- Fazer uma versão paralela utilizando Posix Pthreads
- Para calcular o tempo de execução do programa, utilize o comando no Linux:  
**`/usr/bin/time -f "%e" ./nomedoprograma < entrada.txt > saida.txt`**

- Um relatório apresentando uma introdução sobre os algoritmos, os resultados obtidos e as soluções. O relatório deve ter no mínimo 5 páginas e no máximo 10 páginas desconsiderando capas, índices e bibliografia. No máximo 1 página explicando cada algoritmo e seus comentários.
- Discuta as soluções, as dificuldades, os resultados obtidos, o hardware utilizado, a metodologia de execução dos experimentos, etc. (Para obter informações do hardware das máquinas do cluster, utilize o comando abaixo).

### **phoronix-test-suite system-info**

- O relatório deve ser enviado via Moodle conforme combinado no primeiro dia de aula
- O relatório deve apresentar a forma de execução dos códigos (README)
- O relatório deverá ser entregue **SOMENTE** no formato **PDF** com o seguinte nome: **Relatorio-GrupoX**, onde X é o número do grupo.

## **Dicas**

Utilizar a biblioteca GMP para manipulação de big numbers.

<https://gmplib.org/>

## **Avaliação**

- As perguntas para o grupo serão relativas à forma de desenvolvimento, eficiências dos algoritmos, uso correto das ferramentas, legibilidade do código e documentação e participação dos membros no desenvolvimento do trabalho.
- O número de perguntas será decidido no dia da apresentação
- Os horários das apresentações serão definidos em breve

## **GitHub**

- No GitHub, compartilhe o link do projeto com o professor. Padronize o nome do projeto para: [SO2019-GRUPOX \(onde X é o número do grupo\)](#)

## **Prazos**

### **Relatório e Códigos**

**24/06/19 - 23:55h**