Calculando integrais com métodos estocásticos de Monte Carlo

Laboratório de Computação e Simulação

(MAP2212)

Aluno:	**Milton Leal Neto**
Curso:	Bacharelado de Matemática Aplicada Computacional
NUSP:	8973974
Professor:	Julio Stern

São Paulo, 12 de maio de 2021

Conteúdo

1 Apresentação

Este relatório apresenta uma solução para o segundo Exercício Programa (EP) proposto pelo professor Julio Stern no âmbito da disciplina de Laboratório de Computação e Simulação (MAP 2212) do Bacharelado de Matemática Aplicada Computacional (BMAC) do Instituto de Matemática e Estatística (IME) da Universidade de São Paulo (USP).

O objetivo do EP é utilizar métodos estocásticos de Monte Carlo para calcular o valor de uma integral definida de Riemann.

Os métodos solicitados foram:

1. Crude
2. Hit or Miss
3. Importance Sampling
4. Control Variate

A integral cujo valor será estimado é:

$$\gamma = \int_0^1 e^{-0.326723067x} cos(0.36002998837x)dx$$

O enunciado do EP exige a geração de variáveis aleatórias de distribuições de probabilidade, a escolha dos parâmetros adequados para cada um dos métodos pedidos, além da escolha do tamanho da amostra para que obtenhamos um erro relativo tal que $\frac{|\hat{\gamma} - \gamma|}{\gamma}| \leq 0.0005$, onde γ é o valor exato da integral que, no entanto, desconhecemos.

Desta forma, o EP constitui-se na criação de um programa que estime γ com a acurácia desejada, a apresentação do racional por trás da escolha do número n de pontos que serão aleatoriamente gerados para a simulação de Monte Carlo e, por fim, na comparação da eficiência dos métodos de Monte de Carlo.

2 Definindo n

A estratégia para definição do valor do n_{final} foi a mesma em todos os métodos e deu-se da seguinte forma:

1. Foi escolhido um $n_{inicial}$ de tamanho 50 para rodar uma amostra piloto.

2. Com base na amostra piloto, calculou-se o n_{final} para um nível de confiança de 95% ($\delta = 0.05$) e acurácia $\epsilon = 0.0005$ com a fórmula:

$$n_{final} = \frac{\Phi^{-1}(1 - \frac{\delta}{2})^2 \cdot \hat{\sigma}^2_{50}}{\epsilon^2 \cdot \hat{\gamma}^2_{50}} - n_{inicial},$$

   na qual $\Phi^{-1}(1-\frac{\delta}{2})$ corresponde ao percentil de uma distribuição $Normal(0, 1)$, $\hat{\sigma}^2_{50}$ corresponde à variância da amostra piloto de tamanho 50 e $\hat{\gamma}_{50}$ corresponde a estimativa do valor da integral obtida a partir da amostra piloto.

   Note que o valor do $n_{inicial}$ é subtraído do valor do n_{final}, pois os pontos amostrados na amostra piloto serão incorporados, dentro do programa, ao total de pontos necessários para que atinjamos a acurácia desejada, otimizando dessa forma o tamanho final da amostra.

3 Escolha de parâmetros

Com relação aos métodos Crude e Hit or Miss, não houve margem para discricionaridade na confecção do programa. Ambos os métodos lançam mão de dados gerados aleatoriamente a partir de uma distribuição $Uniforme(0, 1)$.

Em contrapartida, os métodos Importance Sampling e Control Variate exigiam escolhas pertinentes para que obtivéssemos tanto a acurácia desejada quanto um bom nível de eficiência. Abaixo, discorremos sobre as escolhas tomadas em cada um destes dois métodos.

1. **Importance Sampling**

   Neste método, optou-se por escolher uma função densidade de probabilidade Exponencial, truncada no intervalo $[0, 1]$, com taxa $= 0.326723067$, dada por:

   $$g(x) = \frac{0.326723067 \cdot e^{-0.326723067x}}{1 - e^{-0.326723067}}.$$

   Desta forma,

   $$\frac{f(x)}{g(x)} = \frac{e^{-0.326723067x} \cdot \cos(0.36002998837x)}{\frac{0.326723067 \cdot e^{-0.326723067x}}{1 - e^{-0.326723067}}},$$

   resulta em

   $$\frac{f(x)}{g(x)} = \frac{\cos(0.36002998837x) \cdot (1 - e^{-0.326723067})}{0.326723067},$$

   que possui apenas a função cosseno dependendo de x.

   Na próxima página, vemos o plot do comportamento das duas funções e verificamos que a Exponencial truncada escolhida aproxima razoavelmente bem a função que queremos integrar em $[0, 1]$

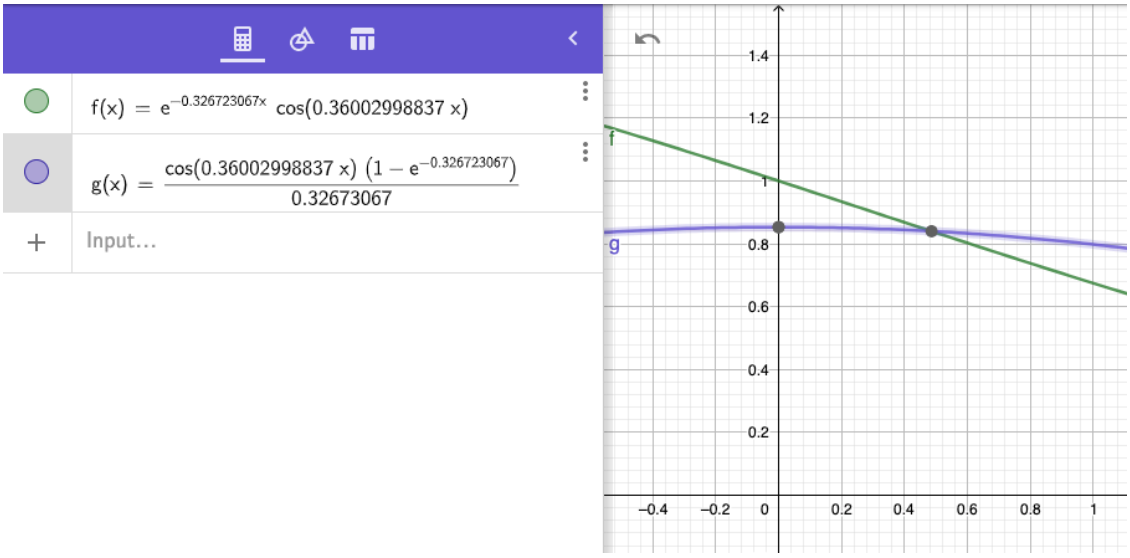

Figura 1: $f(x)$ e $g(x)$ do método Importance Sampling em plot no Geogebra.

Para a geração das variáveis aleatórias no método de Importance Sampling, utilizou-se a função *truncexpon.rvs* da biblioteca *Scipy* da linguagem *Python*.

2. **Control Variate**

Neste método, optou-se por escolher a função polinomial $g(x) = \frac{-x}{4} + 1$ como a variável de controle que melhor se aproxima da função $f(x)$ que queremos integrar.

A escolha foi feita com base o cálculo do coeficiente de correlação ρ entre as funções. No caso, obtivemos:

$$\rho(f(x), g(x)) = 0.99975.$$

Vamos ver agora o comportamento dessas funções no intervalo $[0, 1]$

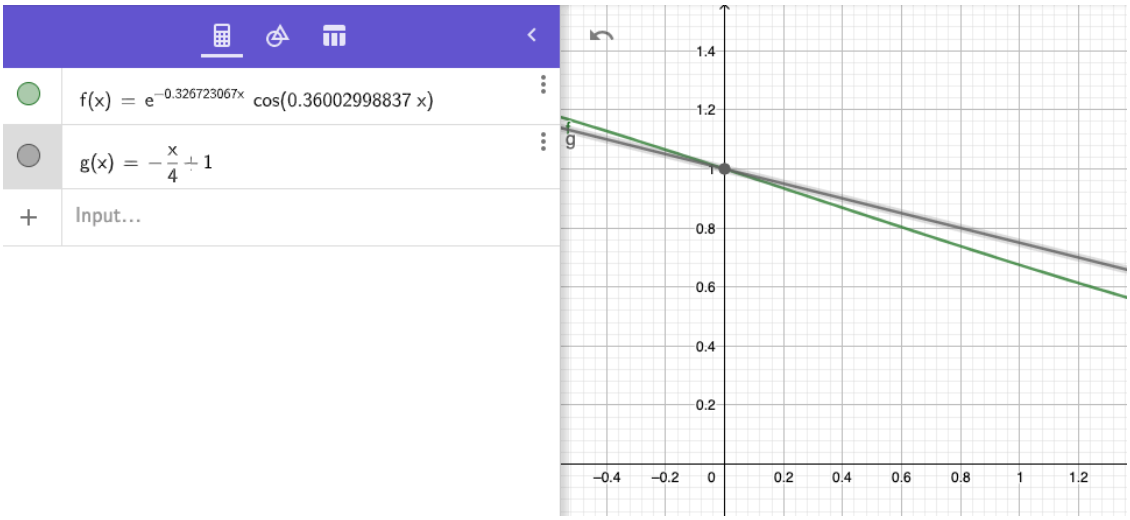

Figura 2: $f(x)$ e $g(x)$ do método Control Variate em plot no Geogebra.

No Control Variate, também utilizamos variáveis aleatórias geradas pela $Uniforme[0,1]$ para avaliar as funções $f(x)$ e $g(x)$ e realizar os cálculos atrelados ao método.

4 Estrutura do programa

O programa escrito em *Python* está estruturado em cinco funções:

1) crude():

Esta função gera inicialmente 50 valores aleatórios x_i entre $[0, 1]$ a partir de uma distribuição $Uniforme$ para construir a amostra piloto e calcular o desvio padrão amostral que será utilizado no cálculo do n_{final}.

Depois, assim como feito feito na amostra piloto, avalia-se a função que queremos integrar com base nos n_{final} valores aleatórios gerados e calcula-se a estimativa do valor da integral com base na média amostral e também o desvio padrão amostral, que será utilizado para comparar os métodos.

2) hitmiss():

Esta função gera inicialmente 50 valores aleatórios de um par (x_i, y_i) entre $[0, 1]$ a partir de uma distribuição $Uniforme$ para construir a amostra piloto e calcular o desvio padrão amostral que será utilizado no cálculo do n_{final}.

Depois, assim como feito feito na amostra piloto, com base nos n_{final} valores aleatórios gerados, verifica-se quantos valores y_i são menores que o valor da função que queremos integrar que foi avaliada com os x_i.

Com isso, obtemos a variável indicadora, que servirá para o cálculo da média amostral, que aproxima o valor real da integral que queremos calcular, e também para o cálculo do desvio padrão amostral, que será utilizado para comparar os métodos.

3) imp_sampling():

Esta função gera inicialmente 50 valores aleatórios x_i entre $[0, 1]$ a partir de uma distribuição $Exponencial$ truncada com taxa = 0.326723067 para construir a amostra piloto e calcular o desvio padrão amostral que será utilizado no cálculo do n_{final}.

Depois, assim como feito feito na amostra piloto, com base nos n_{final} valores aleatórios gerados, avalia-se o quociente $\frac{f(x)}{g(x}$, conforme descrito na

seção anterior, para calcularmos a estimativa do valor da integral com base na média amostral e também o desvio padrão amostral, que será utilizado para comparar os métodos.

4) control_variate():

Esta função gera inicialmente 50 valores aleatórios x_i entre $[0, 1]$ a partir de uma distribuição $Uniforme$ para construir a amostra piloto e calcular o desvio padrão amostral que será utilizado no cálculo do n_{final}.

Depois, assim como feito feito na amostra piloto, com base nos n_{final} valores aleatórios gerados, avalia-se as funções $f(x)$ e $g(x)$, conforme mencionadas na seção anterior, para o cálculo da constante c, que é dada por:

$$c = \frac{Cov(f(x_i), g(x_i)}{Var(g(x_i))},$$

que é utilizada no cálculo da estimativa, dada por:

$$\hat{\gamma} = \sum_{i=0}^{n_{final}} f(x_i) + c(\sum_{i=0}^{n_{final}} g(x_i) - \int_0^1 g(x_i))$$

e do desvio padrão do estimador, dado por:

$$\hat{\sigma\gamma} = \sqrt{\frac{1}{n}(\hat{\sigma^2}(f(x) + \hat{\sigma^2}(g(x)) - 2\rho(f(x), g(x))\hat{\sigma}f(x)\hat{\sigma}g(x))}$$

5) main():

É a função de chamada principal, na qual é estipulada a seed e são impressos os resultados de cada uma das funções anteriores, além de algumas comparações de eficiência dos métodos.

5 Conclusão

Os resultados da simulação via métodos de Monte Carlos mostram que todos os métodos conseguem aproximar o valor da integral desejada com base na acurácia estipulada.

Contudo, como era de se esperar, alguns métodos são melhores que outros. Na simulação realizada, o método Control Variate mostrou-se o mais eficiente, necessitando de uma amostra total de apenas 97 valores para obter a aproximação desejada e desvio padrão amostral de aproximadamente 0.0022.

Na sequência, aparece o método Importance Sampling, que precisou de uma amostra de tamanho 2.919 para retornar um desvio padrão de aproximadamente 0.0160.

Em seguida, vem o método Crude, que necessitou de 99.919 pontos para obter um desvio padrão de aproximadamente 0.09425.

Por fim, o método Hit or Miss precisou de uma amostra total de 957.201 para atingir um desvio padrão de 0.3700.

No que tange aos tempos de execução dos métodos, o Control Variate foi o mais rápido, precisando de apenas 0.002 segundo, seguido pelo Crude (0.122 segundo), Importance Sampling (0.240 segundo) e Hit or Miss (1.475 segundo).

Na próxima página, também destaca-se a razão entre os desvios padrões de cada um dos métodos, que confirma a superioridade do método Control Variate:

```
Comparação entre os métodos pela razão dos Desvios Padrões

Crude X Hit or Miss
DP_Crude / DP_Hit_Miss = 0.2547292341186532

Crude X Importance Sampling
DP_Crude / DP_Importance_Sampling = 5.859507680766991

Crude X Control Variate
DP_Crude / DP_Control_Variate = 42.80645477448895

Hit or Miss X Importance Sampling
DP_Hit_Miss / DP_Importance_Samplig = 23.002886578921775

Hit or Miss X Control Variate
DP_Hit_Miss / DP_Control_Variate = 168.04688681531405

Importance Sampling X Control Variate
DP_Importance_Sampling / DP_Control_Variate = 7.305469521781688
```

Figura 3: O Control Variate é cerca de 170 vezes superior ao Hit or Miss

6 Referências

[1] John Michael Hammersley, D.C. Handscomb - Monte Carlo Methods-Methuen young books (1964)

[2] Morris H DeGroot_ Mark J Schervish - Probability and statistics-Pearson Education (2012)

[3] J.M. Stern. Cognitive Constructivism and the Epistemic Significance of Sharp Statistical Hypotheses in Natural Sciences (2013)