MAP2212 - Laboratório de Computação e Simulação

Relatório - EP02

Vítor Garcia Comissoli - 11810411

1 Introdução

O objetivo deste EP é a estimação da integral γ da $f(x)$ estipulada pelo enunciado por meio de quatro métodos de Monte Carlo, sendo eles **Crude**, **Hit or Miss**, **Control Variate** e **Importance Sampling**, de modo que a estimativa para γ obitida tenha um erro de, no máximo, 0.05% de γ, com 95% de confiança, ou seja, que fique dento do Intervalo de Confiança $[\gamma \cdot (1 - 0.0005), \gamma \cdot (1 + 0.0005)]$ com $\alpha = 0.95$.

Vale ressaltar também que, neste EP, deve-se tratar o valor da integral γ como desconhecido, então todas as contas devem ser realizadas com um estimador qualquer para γ ($\hat{\gamma}$).

2 Discussão Teórica

Primeiramente, foi necessário descobrir o valor de **n** (número de loopings realizados pela função "for in range") para que as estimativas de γ se encontrem dentro do intervalo de confiança dado acima com $\alpha = 0.95$

Para isso, foi utilizada a aproximação assintótica para uma Normal(0,1). Disso temos que a fórmula para a obtenção de cada um dos quatro valores de **n** (um para cada método) pode ser dada por:

$$n \geq \frac{z_\alpha 2 \cdot \sigma^2}{\epsilon^2} \Leftrightarrow \frac{z_\alpha \cdot \sigma}{\sqrt{n}} \leq \epsilon$$

Como temos um $\alpha = 0.95$, tomamos pela tabela da normal que $z_\alpha = 1.96$

Temos que $\epsilon = \gamma \cdot 0.0005$. Pode-se dizer que o erro no quadrante se dá por:

$$|\hat{\gamma} - \gamma| \leq 0.0005 \cdot \hat{\gamma}$$

Disso, dizemos que o erro experimental a ser usado na fórmula da aproximação assintótica se dá por $\epsilon = 0.0005 \cdot \hat{\gamma}$, sendo $\hat{\gamma}$ um estimador para γ que será encontrado a seguir.

Sabemos pelo enunciado, que a função $f(x)$ se dá por:

$$f(x) = e^{-Ax} \cdot cos(Bx)$$

Como, pelo enunciado, $A = 0.RG$ e $B = 0.CPF$:

$$f(x) = e^{-0.54018912x} \cdot cos(0.509920698x)$$

Plotando o gráfico de $f(x)$ no intervalo $[0, 1]$ temos a seguinte função:

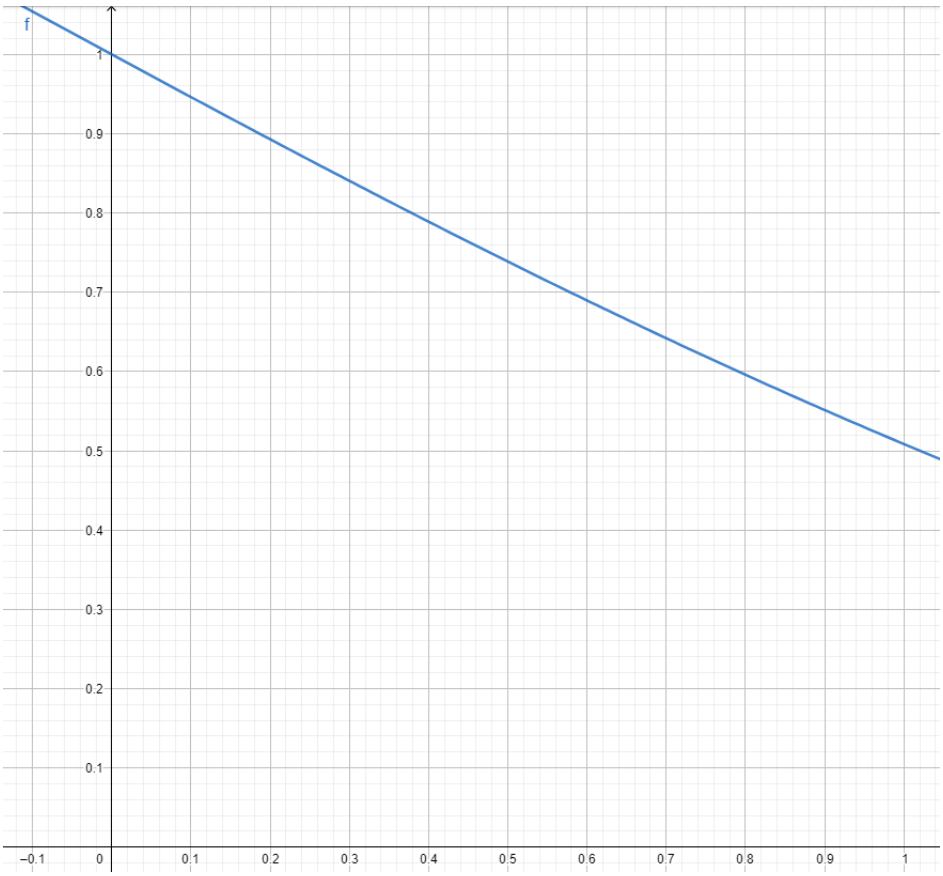

Figura 1: Gráfico da Função $f(x)$ no intervalo $[0, 1]$

Pelo gráfico, e por saber que o exponte da exponencial é negativo, temos que a função é decrescente no intervalo $[0, 1]$, o que implica que:

$$f(1) \leq \gamma \leq f(0) \Leftrightarrow 0.50852 \leq \gamma \leq 1$$

Sabendo disso, podemos assumir $\hat{\gamma} = 0.50852$, já que quanto menor o valor de $\hat{\gamma}$, maior o valor de $\mathbf{n}$, o que implica em um estimador para a integral ainda mais preciso que o enunciado pelo exercício.Tendo isso, podemos agora encontrar o valor de $\mathbf{n}$ nos quatro casos.

Calculemos agora o valor de $\mathbf{n}$ para cada um dos quatro métodos:

2.1 Crude

A variância para esse método, segundo o slide de aula, pode ser dada por:

$$\sigma_c^2 = \frac{1}{n} \int_a^b (f(x) - \gamma)^2 dx$$

Nesse caso específico, a variância se dá por:

$$\sigma^2 = \int_0^1 (f(x) - \hat{\gamma})^2 dx \Leftrightarrow \int_0^1 (e^{-0.54018912x} \cdot cos(0.509920698x) - 0.50852)^2 dx$$

$$\Rightarrow \sigma^2 \approx 0.07575$$

Agora, tendo todos os valores, podemos encontrar o valor de n desejado utilizando a fórmula da aproximação assintótica para uma Normal(0,1) já apresentada acima. Temos então:

$$\frac{z_\alpha \cdot \sigma}{\sqrt{n}} \leq \epsilon \Leftrightarrow \frac{1.96 \cdot \sqrt{0.07575}}{\sqrt{n}} \leq 0.0005 \cdot 0.50852$$

$$\Leftrightarrow \sqrt{n} \geq \frac{1.96 \cdot 0.27522}{0.00025426} \Leftrightarrow \sqrt{n} \geq 2122 \Leftrightarrow n \geq 4502884$$

Temos assim, que o valor de **n** a ser colocado no programa para satisfazer as condições impostas pelo enunciado deve ser igual a 4502884.

2.2 Hit or Miss

A variância para esse método, segundo o slide de aula, pode ser dada por:

$$\sigma_h^2 = \frac{\gamma \cdot (1-\gamma)}{n}$$

Nesse caso específico, a variância se dá por:

$$\sigma^2 = \hat{\gamma} \cdot (1-\hat{\gamma}) \Leftrightarrow 0.50852 \cdot (1-0.50852) \Leftrightarrow 0.50852 \cdot 0.49148$$

$$\Rightarrow \sigma^2 \approx 0.24993$$

Agora, tendo todos os valores, podemos encontrar o valor de n desejado utilizando a fórmula da aproximação assintótica para uma Normal(0,1) já apresentada acima. Temos então:

$$\frac{z_\alpha \cdot \sigma}{\sqrt{n}} \leq \epsilon \Leftrightarrow \frac{1.96 \cdot \sqrt{0.24993}}{\sqrt{n}} \leq 0.0005 \cdot 0.50852$$

$$\Leftrightarrow \sqrt{n} \geq \frac{1.96 \cdot 0.49992}{0.00025426} \Leftrightarrow \sqrt{n} \geq 3854 \Leftrightarrow n \geq 14853316$$

Temos assim, que o valor de **n** a ser colocado no programa para satisfazer as condições impostas pelo enunciado deve ser igual a 14853316.

2.3 Control Variate

A variância para esse método, segundo o slide de aula, pode ser dada por:

$$Var(\hat{\gamma}) = \frac{1}{n} \cdot (\sigma^2(f(x_i)) + \sigma^2(\varphi(x_i)) - 2\rho(f(x_i), \varphi(x_i))\sigma(f(x_i))\sigma(\varphi(x_i))$$

Como não posuímos todos os dados necessários para calcularmos a variância desse método pela equação acima, utilizaremos a variância amostral para calcular o valor de **n**.

Ultilizando um $\hat{n} = 100000$ no código, e repetindo esse processo 100 vezes, obteve-se as estimativas para média e variância de forma amostral, sendo $\mu_a = 0.743915$ e $\sigma_a^2 = 2.57 \cdot 10^{-10}$.

Como a variância amostral obtida é muito baixa, temos que com o próprio $\hat{n} = 100000$ a estimativa para γ já dá dentro do intervalo de acurácia, então tomemos $n = \hat{n} = 100000$

2.4 Importance Sampling

A variância para esse método, segundo o slide de aula, pode ser dada por:

$$\sigma_s^2 = \frac{1}{n} \int (\frac{f(x)}{g(x)} - \gamma)^2 \cdot g(x) dx$$

Nesse caso específico, a variância se dá por:

$$\sigma^2 = \int_0^1 (\frac{f(x)}{g(x)} - \hat{\gamma})^2 \cdot g(x) dx \Leftrightarrow$$

$$\Leftrightarrow \int_0^1 (\frac{e^{-0.54018912x} \cdot cos(0.509920698x)}{g(x)} - 0.50852)^2 \cdot g(x) dx$$

Como g(x) é a função densidade de probabilidade (fdp) de uma beta(1,1.2), temos que:

$$g(x) = \frac{\Gamma(a+b)}{\Gamma(a) \cdot \Gamma(b)} x^{(a-1)} \cdot (1-x)^{(b-1)} \Leftrightarrow 1.2 \cdot (1-x)^{(0.2)}$$

Assim:

$$\sigma^2 = \int_0^1 (\frac{e^{-0.54018912x} \cdot cos(0.509920698x)}{1.2 \cdot (1-x)^{(0.2)}} - 0.50852)^2 \cdot 1.2 \cdot (1-x)^{(0.2)} dx$$

$$\Rightarrow \sigma^2 \approx 0.0596348$$

Agora, tendo todos os valores, podemos encontrar o valor de n desejado utilizando a fórmula da aproximação assintótica para uma Normal(0,1) já apresentada acima. Temos então:

$$\frac{z_\alpha \cdot \sigma}{\sqrt{n}} \leq \epsilon \Leftrightarrow \frac{1.96 \cdot \sqrt{0.0596348}}{\sqrt{n}} \leq 0.0005 \cdot 0.50852$$

$$\Leftrightarrow \sqrt{n} \geq \frac{1.96 \cdot 0.24420}{0.00025426} \Leftrightarrow \sqrt{n} \geq 1882 \Leftrightarrow n \geq 3541924$$

Temos assim, que o valor de **n** a ser colocado no programa para satisfazer as condições impostas pelo enunciado deve ser igual a 3541924.

3 Discussão do Código

Os prints abaixo do código estão comentados linha por linha, explicitando todos os processos tomados.

```python
def f(x):
    """
    Esta funcao deve receber x e devolver f(x), como especifcado no enunciado
    Escreva o seu codigo nas proximas linhas
    """

    func = np.exp(-A*x)*np.cos(B*x) # Calcula o valor de f(x) para o x recebido na função e usando A e B como especificados no enunciado

    return func # Retorna o valor obtido



def crude(Seed = None):
    random.seed(Seed)
    """
    Esta funcao deve retornar a sua estimativa para o valor da integral de f(x)
    usando o metodo crude
    Escreva o seu codigo nas proximas linhas
    """

    soma = 0 # Inicialização da varável que vai obter a soma das n áreas

    n = 4502884 # N obitido pela aproximação assintótica a uma Normal(0,1)

    for i in range(n): # Repete n vezes

        x = random.uniform(0,1) # Coordenada X uniformemente aleatória entre 0 e 1
        valor = f(x) # Valor de F(x) para o X obtido acima
        soma += 1 * valor # Soma acrescenta a área do retângulo de altura f(x) e largura 1
        gamma = soma/n # É a estimativa para o valor da integral de f(x)

    return gamma # Retorna a estimativa
```

Figura 2: Código e comentários das funções f(x) e Crude

```python
def hit_or_miss(Seed = None):
    random.seed(Seed)
    """
    Esta funcao deve retornar a sua estimativa para o valor da integral de f(x)
    usando o metodo hit or miss
    Escreva o seu codigo nas proximas linhas
    """

    dentro = 0 # Inicialização da variável que vai conter o número de pontos dentro da área de gamma

    n = 14853316 # N obitido pela aproximação assintótica a uma Normal(0,1)

    for i in range (n): # Repete n vezes

       x = random.uniform(0,1) # Coordenada X uniformemente aleatória entre 0 e 1
       y = random.uniform(0,1) # Coordenada Y uniformemente aleatória entre 0 e 1

       if y <= f(x): # Testa se a coordenada Y é menor que o valor de f(x), o que implica que Y está dentro da área de gamma
           dentro += 1 # Adiciona 1 ponto à variável "dentro"


    gamma = dentro/n # É a estimativa para o valor da integral de f(x)

    return gamma # Retorna a estimativa
```

Figura 3: Código e comentários da função Hit or Miss

```python
def control_variate(Seed = None):
    random.seed(Seed)
    """
    Esta funcao deve retornar a sua estimativa para o valor da integral de f(x)
    usando o metodo control variate
    Escreva o seu codigo nas proximas linhas
    """

    soma = 0 # Inicialização da varável que vai obter a soma das n equações

    n = 100000 # N obitido pela aproximação assintótica a uma Normal(0,1)

    for i in range(n): # Repete n vezes

        x = random.uniform(0, 1) # Coordenada X uniformemente aleatória entre 0 e 1

        soma += (f(x) - h(x) + (3/4)) # Acrescenta o resultado de f(x) + h(x) + integral definida em [0,1] de h(x) (= 3/4) a variável soma

    gamma = soma/n # É a estimativa para o valor da integral de f(x)

    return gamma # Retorna a estimativa



def h(x): # Função que se assemelha muito ao comportamento de f(x) no intervalo [0,1]

    func = (- 0.5 * x) + 1 # Calcula o valor de g(x) para o x recebido na função

    return func # Retorna o valor obtido
```

Figura 4: Código e comentários da função Control Variate

```python
def importance_sampling(Seed = None):
    random.seed(Seed)
    """
    Esta funcao deve retornar a sua estimativa para o valor da integral de f(x)
    usando o metodo importance sampling
    Escreva o seu codigo nas proximas linhas
    """

    soma = 0 # Inicialização da varável que vai obter a soma das n equações

    n = 3541924 # N obitido pela aproximação assintótica a uma Normal(0,1)

    for i in range(n): # Repete n vezes

        x = random.betavariate (1, 1.2) # Coordenada X aleatória entre 0 e 1 com fdp Beta(1,1.2)
        y = beta.pdf(x,1,1.2) # Uma g(x) que é uma fdp de Beta(1,1.2)

        soma += f(x)/y # Soma o valor de f(x) / g(x)

    gamma = soma/n # É a estimativa para o valor da integral de f(x)

    return gamma # Retorna a estimativa
```

Figura 5: Código e comentários da função Importance Sampling

4 Resultado

O teste dos resultados foi realizado pela criação de uma função main() que chama as funções dos quatro métodos **n** vezes e imprime a razão das estimativas para γ que se encontram dentro do intevalo de confiança sobre o número total de estimativas realizadas.

No método Crude, para $n = 50$, essa razão foi de 1.0, com média $\mu = 0.74392$. O tempo gasto para um loop de $n = 50$ foi de, aproximadamente, 832.59 segundos (13.88 min).

Já no método Hit or Miss, para um $n = 50$, a razão obtida também foi de 1.0, com média $\mu = 0.74390$. O tempo gasto para um loop de $n = 50$ foi de, aproximadamente, 2522.23 segundos (42.04 min).

No método Control Variate, para $n = 50$, essa razão foi de 1.0, com média $\mu = 0.74392$. O tempo gasto para um loop de $n = 50$ foi de, aproximadamente, 17.90 segundos.

Por fim, no método Important Sampling, para $n = 30$, essa razão foi de 1.0, com média $\mu = 0.74392$. O tempo gasto para um loop de $n = 50$ foi de mais de 3 horas.

Todos os valores apresentados acima mostram-se coerentes aos parâmetros de acurácia $\alpha = 0.95$ e $\epsilon = \gamma \cdot 0.0005$, estipulados pelo enunciado do EP.

Com isso podemos concluir que o resultado obtido foi dentro do esperado.

5 Conclusão

Como já dito previamente, todos o métodos se mostraram coerentes aos parâmetros de acurácia de $\alpha = 0.95$ e $\epsilon = \gamma \cdot 0.0005$. Contudo, a outros fatores que podem ser analisados.

Analisando quanto a performance computacional temos que o método de Important Sampling se mostrou muito lento e custoso quando comparado com os outros 3 métodos, seguido pelo método Hit or Miss, o método Crude e, por fim, o método Control Variate, que se mostrou o mais eficiente.

Quanto ao tamanho das variâncias, tivemos Control Variate com a menor variância de todos os métodos, e Hit or Miss com maior variabilidade.

Por fim, quanto ao tamanho do **n** encontrado, temos que o método Hit or Miss necessitou do maior valor de n, enquanto o método Control Variate foi o que utilizou o menor valor de n dos quatro.