

# EP2 - Método de Monte Carlo

George Othon  
NUSP 103xxxxx

March 2020

## 1 Introdução

Este relatório tem como objetivo analisar os resultados obtidos pelo Método de Monte Carlo nas suas 4 variações ao integrar  $f(x)$ .

$$f(x) = e^{-ax} \cos bx$$

Com  $a = 0.RG$  e  $b = 0.NUSP$ , no intervalo  $[0,1]$ . (Com  $RG = 399xxxxxx$  e  $NUSP = 103xxxxx$ , logo,  $a = 0.399xxxxxx$  e  $b = 103xxxxx$  ).

## 2 Bibliotecas

Para nos auxiliar no manuseio das listas, gerar os números aleatórios e para utilizar funções matemáticas e estatísticas que não temos nas built-in do Python, foi necessário importar as seguintes bibliotecas com as seguintes funções:

### 1. Numpy

- (a) `exp()` - Função exponencial
- (b) `cos()` - Cosseno
- (c) `std()` - Desvio padrão
- (d) `sqrt()` - Raíz quadrada
- (e) `mean()` - Média
- (f) `random.beta()` - Gera números aleatórios com distribuição beta
- (g) `cov()` - Covariância
- (h) `var()` - Variância

### 2. Random

- (a) `random()` Gera números aleatórios entre 0 e 1

### 3. Scipy

- (a) `stats.beta.pdf()` - Função de densidade de probabilidade

### 3 Critério de parada

Como critério de parada utilizamos o erro padrão, onde a cada iteração verificamos se o erro é menor que 1%, e assim que atendesse o critério, ele calcula a média das iterações e retorna como resultado final.

### 4 Método Crud

Este foi o primeiro método que testamos e o mais simples onde apenas calculamos  $\frac{1}{n} \sum_{i=1}^n f(x)$ , que rendeu bons resultados, e teve desvio padrão de  $\sigma = 0.0203$

### 5 Método Hit or miss

Ao tentar aproximar a integral de  $f(x)$  no intervalo  $[0,1]$  geramos diversos pares ordenados  $(x,y)$  e verificamos se  $y \leq f(x)$ , ou seja, se o par ordenado tem imagem acima ou abaixo da função  $f$ , e usamos essa proporção para aproximar  $\int_0^1 f(x)dx$ . Apresentou desvio padrão  $\sigma = 0.0108$

### 6 Método Importance Sampling

No método em questão, utilizamos um gerador de números pseudo-aleatórios com distribuição beta. Após diversos testes tivemos os melhores resultados com os parâmetros  $\alpha = 0.9$  e  $\beta = 1.0$ . Nessas condições tivemos boas aproximações, e com diversos testes o desvio padrão foi de  $\sigma = 0.0048$

### 7 Método Control Variate

No Control Variate, após algumas análises escolhemos como uma aproximação para  $f(x)$ , a função  $g(x) = e^{-ax}$ , com  $a = 0.399104525$ . Como  $g(x) > f(x)$ ,  $\forall x \in [0, 1]$ , e  $g(x)$  é extremamente próxima à  $f(x)$ .

Dividimos este método em duas partes. Na primeira calculamos o resultado da integral pelo método Crud para  $f(x)$  e para  $g(x)$ . Na segunda parte, calculamos o fator  $c$  dado por

$$c = \frac{-Cov[f(x), g(x)]}{Var[g(x)]}$$

e em seguida calculamos

$$Crud(f(x)) + c * (Crud(g(x)) - \int_0^1 g(x)dx)$$

onde  $\int_0^1 g(x)dx = 0.824$ , para obter a aproximação final, que teve excelentes resultados apresentando desvio padrão  $\sigma = 0.0006$

## 8 Comparando os métodos

Rodamos cada método 1000 vezes e fizemos o seguinte resumo estatístico com o auxílio da biblioteca pandas.

Utilizamos o jupyter notebook para gerar a tabela a seguir, e incluímos como uma imagem no artigo. Para a criação da tabela foram usadas as funções `DataFrame()` e `describe()`.

	Crud	Hit or miss	Importance Sampling	Control Variate
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.822487	0.824923	0.823170	0.822733
std	0.020304	0.010854	0.004848	0.000667
min	0.680146	0.791621	0.807193	0.821175
25%	0.815592	0.817249	0.819918	0.822233
50%	0.823014	0.824620	0.823348	0.822722
75%	0.830016	0.831905	0.826457	0.823220
max	0.984073	0.866772	0.840504	0.824516

Figure 1: Análise Descritiva ( n = 1000 )

Ao analisar a tabela acima percebemos que para todos os métodos a média e a mediana ( 50% ) fica próximo à 0.82. O método Crud foi o que teve a maior variação e o Control Variate teve a menor. Mas escolhemos o desvio padrão para analisar o desempenho de cada método, e com base nesse parametro fizemos a seguinte classificação dos métodos.

	Método	Desvio padrão
1	Control Variate	0.0006
2	Importance Sampling	0.0048
3	Hit or miss	0.0108
4	Crud	0.0203

Table 1: Classificação dos métodos

Portanto, dentre os quatro métodos que implementamos, o que retornou os melhores resultados foi o Método Control Variate utilizando a função g como aproximação para f.

## References

- [1] <https://numpy.org/doc/>
- [2] <https://docs.python.org/3/library/random.html>
- [3] <https://docs.scipy.org/doc/scipy/reference/stats.html>
- [4] Cognitive Constructivism and the Epistemic Significance of Sharp Statistical Hypotheses in Natural Sciences - Julio Stern