Metodologia de Monte Carlo

July 4, 2019

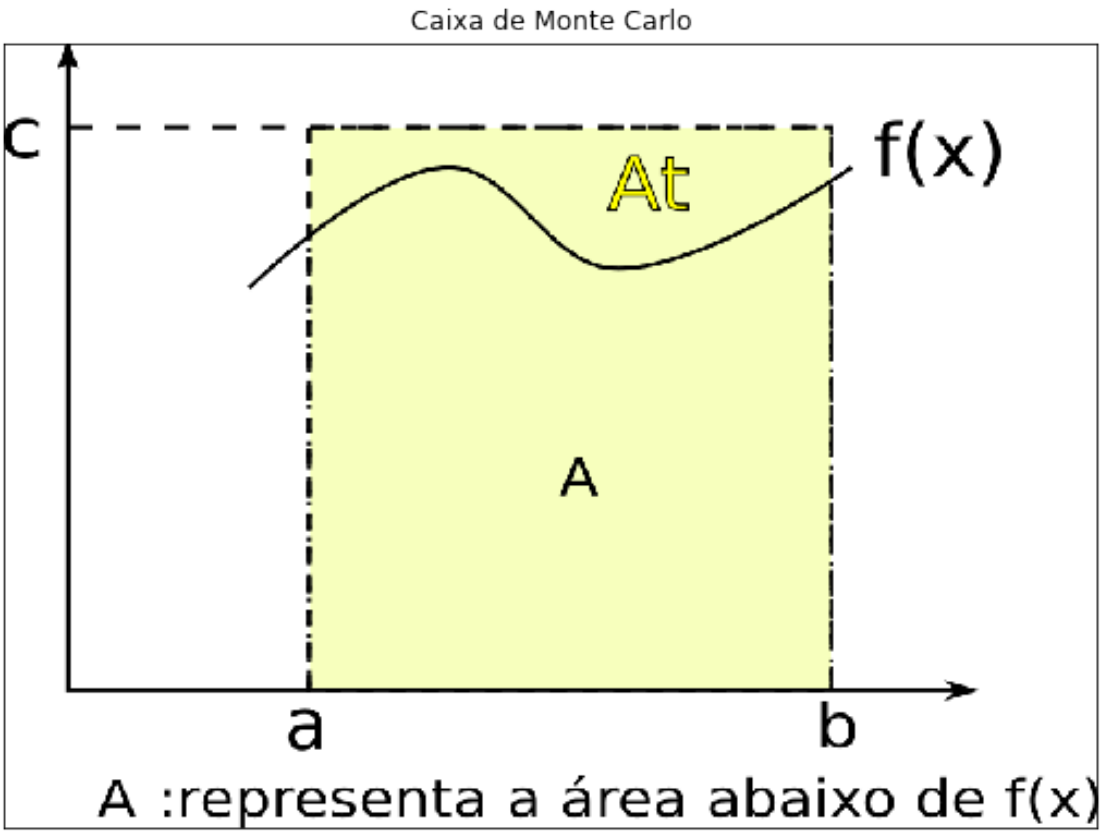
1 Monte Carlo

```
In [3]: import matplotlib.pyplot as plt
        import matplotlib.image as img
        import pylab as py
        import math as ma
        import random
        from random import *
        #novos pacotes de imagem
        import imageio
        from skimage import data, io, filters
        %matplotlib inline
```

O método de monte carlo ou método de ordem zero é um método heurístico de busca baseado na criação de uma caixa e em um sorteio de números aleatórios, cujo espaço de soluções é delimitado por esta caixa.

Foi o primeiro programa computacional criado para resolver uma integral definida sem solucação analítica. Este problema estava relacionado com o projeto de construção da bomba atômica (projeto Manhatam).

```
In [4]: caixa = io.imread( "caixa.png")
        py.rcParams['figure.figsize'] = (9.0, 9.0)
        plt.title("Caixa de Monte Carlo")
        fig = plt.imshow(caixa)
        fig.axes.get_xaxis().set_visible(False)
        fig.axes.get_yaxis().set_visible(False)
        plt.show()
```



A solução do método consiste em sortear n pontos aleatórios dentro do retângulo de área At

$$At = \bar{ab}.c$$

Definimos n_a como sendo o o número total de pontos abaixo da curva f(x).

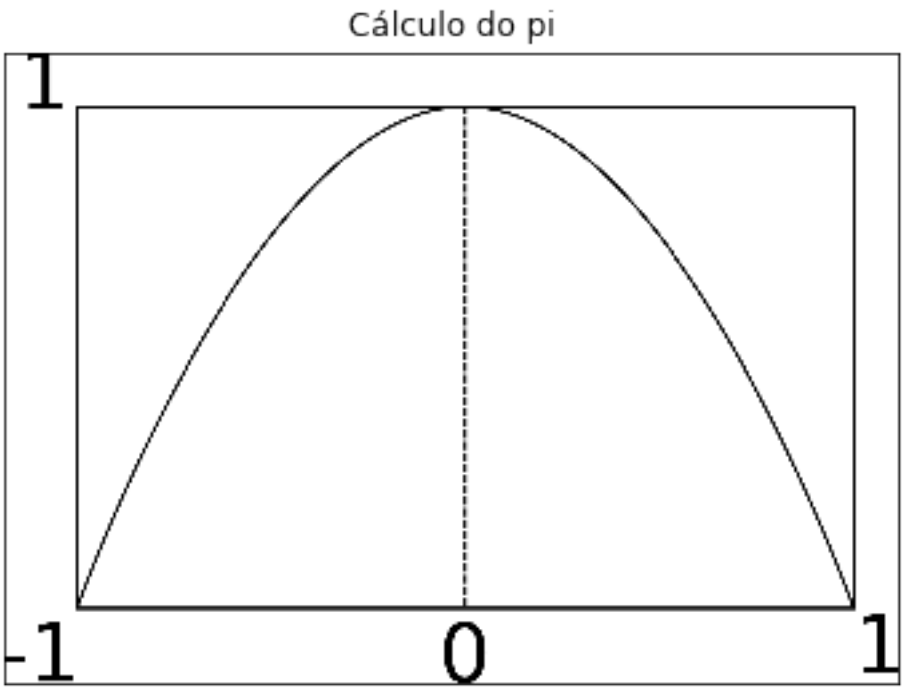
$$n \rightarrow Atna \rightarrow A$$

$$A = \frac{n_a.At}{n}$$

Onde A passa a ser a solução da integral definida.
Em geofísica o principal uso é a minimização de funções.

1.1 Exemplo: o cálculo do valor de pi

```
In [5]: pimage = io.imread( "pi.png")
        py.rcParams['figure.figsize'] = (6.0, 6.0)
        plt.title("Cálculo do pi")
        fig = plt.imshow(pimage)
        fig.axes.get_xaxis().set_visible(False)
        fig.axes.get_yaxis().set_visible(False)
        plt.show()
```



A imagem acima mostra uma metade de uma circunferência $f(x) = \sqrt{1-x^2}$ inserida em uma caixa de monte carlo. Calcule o valor de pi via método de monte carlo

```
In [40]: # Entradas
         pii=2.0*ma.acos(0)
         n=15456456 #número de pontos da caixa

         #Definindo o tamanho da caixa 1
         xmin=-1
         xmax=1

         #Definindo o tamanho da caixa 2

         ymin=0
         ymax=1


         ii=0 # números de pontos abaixo da curva

         for i in range(n):
             x= random()*(xmax-xmin)+xmin
             y= random()*(ymax-ymin)+ymin
             yc=(1-x**2)**0.5 #função objeto
             if y<= yc:
                 ii=ii+1
```

```
        #Cálculo da área abaixo do gráfico
        At=(xmax-xmin)*(ymax-ymin)
        area=At*ii/n


        print("Resultados do valor de pi via monte carlo:")
        print("pontos abaixo da curva",ii)
        print("pi(calculado)=",area*2)
        print("erro=",abs(pii-area*2))
        print("pi(verdadeiro)=",pi)
```

```
Resultados do valor de pi via monte carlo:
pontos abaixo da curva 12140806
pi(calculado)= 3.141937841378386
erro= 0.00034518778859293775
pi(verdadeiro)= 3.141592653589793
```