

Alteração dos geradores do EP3 para geradores de baixa discrepância e comparações

MAP 2212 EP3

Luiz Guilherme De Padua Sanches

Maio 2023, NUSP = 13686431

1 Requisitos e considerações

Foi solicitado a alteração dos geradores dos métodos do EP2 para geradores Quasi-Aletórios e que seja feita a comparação entre a eficiência utilizando esse geradores Quasi-Random e os geradores Pseudoaleatórios nas funções.

Rodaremos o programa com uma SEED(k), com k variando de 0 a 199 chegamos ao número N de eventos médios de cada item para serem realizadas as comparações.

As imagens de gráficos foram geradas no [Geogebra](#).

O cálculo da integral e do polinômio aproximador foi feito no [Symbolab](#).

A gerações de números aleatórios Quasi-Random e o cálculo da discrepância foi utilizando a biblioteca [scipy.stats.qmc](#).

Esses geradores impossibilitam o uso de variância como método de precisão, assim utilizaremos o conceito de discrepância. Podemos encontrar mais informações em [Quasi-Monte Carlo method](#).

2 Mudança na aplicação dos métodos de Monte Carlo

Os métodos já foram descritos no relatório anterior, nesse falaremos as diferenças na implementação causadas pela mudança de gerador e critério de parada.

2.1 Método de Monte Carlo Cru e Monte Carlo Control Variables

Nesses métodos o gerador aleatório foi substituído pelo gerador Halton sem maiores problemas. O critério de parada utilizado foi:

$$|E| \leq VD^*$$

Esse critério nos diz que o módulo do erro é menor que a variação da função multiplicado pela discrepância da sequência X gerada (no caso, a sequência de Halton). Pode-se encontrar mais informações no seguinte texto [William J. Morokoff and Russel E. Caflisch, Quasi-Monte Carlo integration](#).

Para ambos o critério convergiu em um bom tempo, com um erro dentro do esperado em poucas repetições.

2.2 Método de Monte Carlo Hit or Miss e Monte Carlo Important Sampling

Nesses métodos o gerador aleatório foi substituído pelo gerador Halton sem maiores problemas. No método Important Sampling foi utilizado uma transformação de uniforme para beta com finalidade de gerar a amostra com aquela distribuição, baseado no seguinte artigo [inversion sampling](#), e para isso foi utilizado a função `scipy.stats.beta.pdf(x, a, b)`.

Entretanto, o cálculo dos erros, através da mesma forma que os métodos anteriores, não conseguiram ser realizados. Seja devido ao custoso tempo para o cálculo da divergência ou por problemas na convergência, não foi possível em minha parte aplicar o mesmo método.

Assim, foram pensadas duas opções para resolver esse problema:

- Randomizar os números Quasi-Aleatórios, o que permitiria o uso do parâmetro de variância para o cálculo de N . O problema nesse método é existir uma perda de eficiência.
- Utilizar os métodos anteriores para estabelecer uma estimativa confiável, criar um intervalo aceitável do erro sobre esse parâmetro e estimar N dessa forma.

Em um problema sem a finalidade de observar as vantagens do método, a primeira opção teria sido escolhida. Entretanto, como temos essa finalidade, a segunda opção foi escolhida.

2.2.1 Parâmetros de parada para esses métodos

Utilizando o método Control Variables para estimar o valor da integral e repetindo o processo 10000 vezes chegamos ao seguinte valor em média 0.7430356447175345.

Multiplicando por 0.9995 e 1.0005 para encontrar os limites chegamos ao intervalo de: [0.74266412689 ; 0.74340716254]

Rodando cada interação diversas vezes (testes empíricos), com SEED K variando de 0 a 1000, computando os valores de N que causam a estimativa a caírem dentro do intervalo e tirando a moda desses valores N definimos um valor fixo para cada um dos métodos.

2.3 Comparações

Comparemos os seguintes requisitos:

- Média dos número de interações.
- Precisão média das integrais.
- Relação entre os números de interações.
- Taxa de acerto.

Método	Precisão	Média N	Taxa de acerto
Hit or Miss Random	0.9999955698704618	5533445.49	0.955
Crude Random	0.999996831832542	579654.135	0.945
Importance Random	0.9999970221602653	182344.75	0.955
Control Random	1.000020887922342	12034.07	0.95
Hit or Miss Quasi*	0.9999911067803328	34999	0.957
Crude Quasi	1.0000005679513833	419.32	1.0
Importance Quasi*	0.999997620786358	6152	0.956
Control Quasi	0.9999976131768897	105.475	1.0

OBS: amostras * foram calculadas da maneira explicada em 2.2.1, possivelmente causou a diferença em N e a diferença de precisão.

Relação entre o número de interações do método de Monte Carlo utilizando Quasi com o número de interações utilizando Random:

Método	Relação a N Quasi
Hit or Miss Random	158.1 vezes mais N
Crude Random	1382.367 vezes mais N
Importance Random	29.639 vezes mais N
Control Random	114.094 vezes mais N

Não houve comparações de tempo oficiais, visto que algumas aplicações diferiram bastante e haveriam resultados injustos ao comparar uma função que precisa calcular N no programa com outra já calculada externamente. Entretanto, será falado de modo geral e empírico o que foi sentindo durante os testes.

2.4 Vantagens e desvantagens observadas:

2.4.1 Gerador Quasi:

Vantagens:

- As interações que usaram o cálculo do erro através da discrepâncias e variação tiveram acertos 100% das vezes testadas.
- Para todos os métodos houveram um baixo número de interações para encontrarmos a precisão necessária.
- O método Crude (o mais simples) já é muito eficiente, não precisando de um tratamento prévio.
- Para baixas interações sua velocidade é rápida.

Desvantagens:

- O cálculo do erro para muitas interações se revelou impraticável por causa de sua demora.
- Uma mudança de uma distribuição uniforme para outra distribuição atrapalha no funcionamento do método.

2.4.2 Gerador Random:

Vantagens:

- A variância pode ser aplicada de forma mais geral que o erro utilizado nos métodos Quasi.
- Não há problema de mudança de distribuição.
- Dependendo do método, possui velocidade rápida.

Desvantagens:

- A variância nos gera um intervalo de confiança e pode ser custosa a ser calculada (mas, aparentemente, menos que a discrepância).
- É necessário um tratamento prévio da função para ter alta eficiência.

2.5 Conclusão

Os métodos utilizando o gerador Quasi se revelam boas alternativas, mas é necessário certo cuidado em sua aplicação.

O método Crude utilizando esse gerador se mostra altamente efetivo, dispensando um tratamento anterior e sendo fácil de aplicar.

Talvez utilizar a randomização seja uma opção válida para a solução de problemas mais complexos, onde o cálculo do erro seja muito problemático.

Em casos que o número de interações seja muito grande, acho preferível o uso de um gerador Random, pois aparenta ser mais rápidos com grandes interações do que o gerador Halton.

Assim, em caso de rápida convergência com o gerador Quasi esse método é o preferível, mas caso exista muitas interações a alteração para um gerador Random se revela uma opção provavelmente mais confortável do que a randomização do Quasi.