

---

Universidade Estadual de Campinas  
Instituto de Física Gleb Wataghin

Projeto referente à disciplina F-809 –  
Instrumentação para o ensino

# Simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D



Aluno: Marcelo Augusto dos Reis RA: 009258  
Orientador: Prof. Dr. Silvio A. S. Vitiello

Campinas - SP  
Novembro de 2004

---

# Resumo

---

Esse projeto tem por objetivo ilustrar um poderoso método numérico que pode ser implementado na resolução de problemas físicos, em especial, um problema de muita importância em física: o oscilador harmônico quântico unidimensional. Para tal concretização, uma simulação em computador foi desenvolvida permitindo a interatividade do usuário com o programa.

Inicialmente, discutimos os aspectos teóricos do oscilador harmônico e em seguida apresentamos o método de Monte Carlo e o algoritmo de Metropolis. A partir dessas considerações é apresentada a implementação do método de Monte Carlo ao problema proposto.

Na segunda parte desse projeto estão contidos os desenvolvimentos computacionais, tanto o algoritmo que calcula a energia do sistema como o da sua implementação gráfica permitindo que o programa como um todo rode praticamente em qualquer plataforma computacional da atualidade.

Finalmente, na terceira parte os resultados são abordados. Uma análise dos parâmetros de entrada e suas respectivas saídas são discutidos tendo como objetivo o entendimento do método utilizado.

# Conteúdo

---

1. Introdução	4
2. O oscilador harmônico unidimensional	5
3. Cálculo variacional	7
3.1 O método variacional.....	7
3.2 O método variacional de Monte Carlo.....	8
3.3 O algoritmo de Metropolis.....	10
4. Implementação	12
4.1 O programa	12
4.2 A construção da Applet-Java	13
4.3 Funcionamento da Applet-Java	14
5. Resultados	18
5.1 O parâmetro delta.....	18
5.2 O número de blocos.....	19
5.3 O parâmetro alfa.....	21
5.4 O número de passos.....	22
6. Conclusão	23
7. Referências	24

# 1. Introdução

---

Atualmente, cada vez mais os diversos sistemas computacionais são usados para o ensino de ciências. Esses sistemas são muitas vezes simulações em computador de problemas muito conhecidos nas diversas disciplinas científicas tal como a física.

O oscilador harmônico quântico unidimensional foi escolhido como sistema físico para o qual foi desenvolvida uma simulação para estudar aspectos do método empregado para a obtenção da energia do sistema. Essa simulação foi desenvolvida tendo em vista sua utilização nos diferentes sistemas operacionais, mais especificamente foi construído um Applet-Java que pode rodar em praticamente todos os navegadores atuais inclusive naqueles pertencentes à classe "software livre", isso significa uma grande portabilidade ao programa e também fácil acesso aos estudantes, já que se trata de um programa Web.

Muitos problemas em mecânica quântica não possuem soluções conhecidas. Uma maneira de contornar esse problema é encontrar funções que satisfaçam algumas características já conhecidas dos sistemas e obter uma solução aproximada para as quantidades de interesse. Entretanto, muitas vezes estas soluções aproximadas não permitem um tratamento analítico. Esta dificuldade, porém, pode ser superada através de métodos numéricos. O método de Monte Carlo (MC) utilizando o algoritmo de Metropolis é uma maneira nestes casos de obtermos uma solução sujeitas apenas a incertezas de natureza estatística.

Embora o problema do oscilador possua solução analítica conhecida e desse modo não se justifica qualquer tratamento numérico na busca de uma solução, o método será empregado para esse sistema e seus resultados confrontados com os valores esperados na tentativa de ilustrar uma aplicação do método de Monte Carlo.

O usuário do programa poderá interagir com parâmetros do método e obter valores da energia e seu erro padrão assim como visualizar a "função de onda" tentativa para cada conjunto desses parâmetros.

## 2. O oscilador harmônico quântico unidimensional

---

Neste capítulo faremos uma breve descrição teórica do sistema oscilador harmônico quântico unidimensional. Os resultados dessa teoria servirão de parâmetros de análise para nossa simulação numérica.

Em mecânica quântica, o operador Hamiltoniana desse sistema é dado por:

$$H = \frac{P^2}{2m} + \frac{1}{2}m\omega^2 X^2 \quad (1)$$

Onde  $P$  representa o momento e  $X$  a posição da partícula de massa  $m$  oscilando com frequência angular  $\omega$ .

Desde que  $H$  seja independente do tempo, para obter os níveis de energia desse sistema basta resolver a equação de autovalor:

$$H|\varphi\rangle = E|\varphi\rangle \quad (2)$$

Podemos escrever essa equação na representação das coordenadas  $\{|x\rangle\}$  e obtermos, portanto, a equação de Schrödinger independente do tempo:

$$\left[ -\frac{\hbar^2}{2m} \frac{d^2}{dx^2} + \frac{1}{2}m\omega^2 x^2 \right] \varphi_n(x) = E_n \varphi_n(x) \quad (3)$$

Onde  $E_n$  é a energia para um dado número quântico  $n$  e  $\varphi_n(x)$  é o autoestado correspondente para esse energia.

Resolvendo essa equação, obtém-se:

$$\boxed{E_n = \left( n + \frac{1}{2} \right) \hbar \omega} \quad (4)$$

$$\varphi_n(x) = \left[ \frac{1}{2^n n!} \left( \frac{\hbar}{m\omega} \right)^n \right]^{1/2} \left( \frac{m\omega}{\pi\hbar} \right)^{1/4} \left[ \frac{m\omega}{\hbar} x - \frac{d}{dx} \right]^n e^{-\frac{1}{2} \frac{m\omega}{\hbar} x^2} \quad (5)$$

Para o estado fundamental  $n=0$  temos:

$$E_0 = \frac{\hbar\omega}{2} \quad e \quad \varphi_0(x) = \left( \frac{m\omega}{\pi\hbar} \right)^{1/4} e^{-\frac{1}{2} \frac{m\omega}{\hbar} x^2} \quad (6)$$

## 3. Cálculo Variacional

---

A seguir, apresentamos a idéia do método variacional e mostramos como é possível implementá-lo numericamente através do método variacional de Monte Carlo.

### 3.1 O Método Variacional

Em situações onde não é possível determinarmos a solução exata da equação de Schrödinger, o método variacional é de grande utilidade. Ao construirmos uma função de onda tentativa  $\Psi_T$ , é importante que ela incorpore da melhor maneira possível os conhecimentos que temos do sistema.

A energia variacional do sistema pode ser escrita analogamente a (2), na representação das coordenadas:

$$H\Psi_T(x) = E_T\Psi_T(x) \Rightarrow \int \Psi_T^*(x)H\Psi_T(x)dx = \int \Psi_T^*(x)E_T\Psi_T(x)dx$$

Logo,

$$E_T = \frac{\int \Psi_T^*(x)H\Psi_T(x)dx}{\int |\Psi_T(x)|^2 dx} \quad (7)$$

É interessante notar que a função tentativa  $\Psi_T(x)$  não precisa ser normalizada para esse propósito pois essa possível constante de normalização seria cancelada algebricamente em (7).

Para o nosso problema em particular, vamos usar uma função tentativa normalizada, isto porque não só calcularemos a energia do sistema para o estado fundamental como também visualizaremos essa função em contraste com a função exata  $\varphi_0(x)$  do estado fundamental.

Assim, nossa função tentativa para o estado fundamental em um sistema onde  $m = \hbar = \omega = 1$  é simplesmente:

$$\Psi_T^\alpha(x) = \left(\frac{2\alpha}{\pi}\right)^{1/4} e^{-\alpha x^2} \quad (8)$$

Portanto, se substituirmos a função tentativa dada em (8) na expressão da energia variacional em (7), obtemos:

$$E_T^{(\alpha)} = \frac{\alpha}{2} + \frac{1}{8\alpha} \quad (9)$$

Como a expressão depende do parâmetro  $\alpha$ , minimizamos a expressão (8) em relação à  $\alpha$ .

Dessa forma obtemos o mínimo em  $\alpha = \frac{1}{2}$ , nos fornecendo:

$$E_T^{(1/2)} = \frac{1}{2} \quad (10)$$

Não há nada de surpreendente nesse resultado já que, verificando (6) para o mesmo sistema  $m = \hbar = \omega = 1$  obtemos a função tentativa igual a função original do problema consecutivamente as mesmas energias.

Mas como dissemos anteriormente, o método variacional não pressupõe que conheçamos as funções analíticas dos problemas no caso, as funções (5), em particular as da forma (6), já que estamos tratando o estado fundamental do sistema.

Tudo que foi feito até agora foi apenas supor que o estado fundamental para esse sistema seja uma gaussiana não necessariamente normalizada e calcular analiticamente a energia para esse estado, a normalização surge aqui apenas para visualizar essa função de onda graficamente no programa.

### 3.2 O Método Variacional de Monte Carlo

Uma vez construída a função tentativa  $\Psi_T(x)$ , surge o problema de calcular a integral expressa em (7). É claro que em nosso problema não precisamos de nenhum método numérico, pois se trata de um problema simples e ainda mais, já conhecemos sua solução. Não é demais lembrar que em um problema de investigação científica não conhecemos as



autofunções do sistema e podemos chegar em integrais do tipo (7) que podem ser extremamente difíceis de calcular analiticamente. Um exemplo de aplicação dessas idéias e que ilustra tal dificuldade seria um sistema de muitas partículas quânticas tais como o Hélio.

Logo, o nosso problema do oscilador harmônico é tratado aqui como se não conhecêssemos sua solução, apenas analisaremos os resultados conhecidos para estudar o método de Monte Carlo.

Para resolver a integral dada em (7) reescrevemos de outra forma:

$$E_T^{(\alpha)} = \frac{\int \Psi_T^\alpha(x)^2 \left( \frac{H\Psi_T^\alpha(x)}{\Psi_T^\alpha(x)} \right) dx}{\int \Psi_T^\alpha(x)^2 dx}$$

Definindo respectivamente a densidade de probabilidade e a energia local:

$$p(x) = \frac{\Psi_T^\alpha(x)^2}{\int \Psi_T^\alpha(x)^2 dx} \quad e \quad E_L^{(\alpha)}(x) = \frac{H\Psi_T^\alpha(x)}{\Psi_T^\alpha(x)} \quad (11)$$

Temos:

$$E_T^{(\alpha)} = \int p(x) E_L^{(\alpha)}(x) dx \quad (12)$$

A energia local pode ser calculada usando a Hamiltoniana (1) na representação das coordenadas  $x$  e no sistema onde  $m = \hbar = \omega = 1$  que fica:

$$H = -\frac{1}{2} \frac{d^2}{dx^2} + \frac{1}{2} x^2 \quad (13)$$

Logo, usando (8) e (13), obtemos:

$$E_L^{(\alpha)}(x) = x^2 \left( \frac{1}{2} - 2\alpha^2 \right) + \alpha \quad (14)$$

Resta-nos amostrar a densidade de probabilidade  $p(x)$  de modo a calcular numericamente (12) da seguinte forma:

$$E_T^{(\alpha)} = \frac{1}{M} \sum_{i=1}^M E_L^{(\alpha)}(x_i) \quad (15)$$

Ou seja  $E_T^{(\alpha)}$  é a média das possíveis M configurações  $x_i$  amostradas a partir da densidade de probabilidade  $p(x)$ .

Os resultados estão sujeitos a incertezas estatísticas, o que torna essencial à estimativa da variância:

$$\sigma^2 = \frac{1}{M-1} \sum_{i=1}^M \left( E_L^{(\alpha)}(x_i) - E_T^{(\alpha)}(x_i) \right)^2 \quad (16)$$

O erro padrão fica:

$$\Delta E_T^{(\alpha)} = \frac{\sigma}{\sqrt{M}} \quad (17)$$

### 3.3 O Algoritmo de Metropolis

Para amostrar as M configurações do problema a partir da densidade de probabilidade  $p(x)$  usamos o algoritmo de Metropolis. Uma observação aqui é notar que  $p(x)$  pode ser qualquer.

Um simples fluxograma desse algoritmo pode ser analisado a seguir.

Na figura  $\zeta_1, \zeta_2$  e  $\zeta_3$  são números aleatórios no intervalo  $[0,1]$ ,  $\Delta$  é um parâmetro que determina a amplitude do deslocamento da partícula e do qual os resultados são independentes. O número de elementos da amostra é M e  $p(x)$  é a mesma de (11).

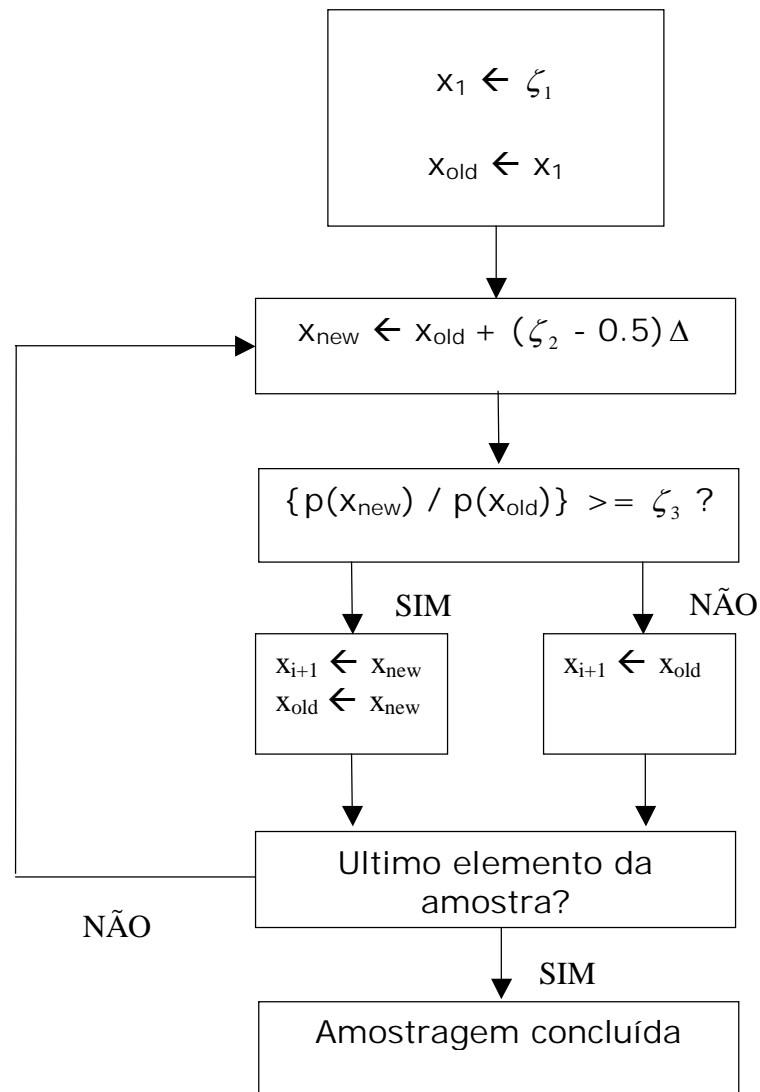


Figura 1 - Algoritmo de Metropolis

Fica claro, portanto, que o próprio algoritmo traz correlações entre os elementos amostrados a partir de um primeiro elemento arbitrário, devemos considerar esse fato e faremos isso desprezando os primeiros elementos amostrados.

## 4. Implementação

---

### 4.1 O Programa

O programa foi inicialmente escrito em Fortran 90 e posteriormente, adaptado ao Java. Neste capítulo mostraremos os aspectos fundamentais do programa.

O usuário pode interagir com quatro parâmetros de entrada os quais são necessários para o cálculo da energia.

Tabela 1 – Parâmetros de entrada do programa fornecidos

Parâmetro	Descrição
Passos (M)	Dimensão da amostra $\{ x_1, x_2, x_3, \dots, x_M \}$
Alfa ( $\alpha$ )	Parâmetro da função tentativa $\Psi_T^\alpha(x)$
Delta ( $\Delta$ )	Amplitude de deslocamento de um elemento da amostra (Fig. 1)
Blocos	A partir dos M elementos da amostra podemos dividi-la em Blocos para calcular a energia

De posse desses parâmetros o programa produz a amostragem com M elementos conforme descrito na seção 3.3.

Em seguida é chamada uma função que calcula a energia a partir do número de blocos especificados.

A idéia aqui é a seguinte:

Se o #Blocos for igual a um, o programa simplesmente utiliza a equação (15) no cálculo da energia e nos dá um erro padrão calculado a partir da equação (17).

Agora, para valores de Blocos maiores que um, o programa faz o cálculo da energia para cada bloco  $E_{B_i}^{(\alpha)}$ , ou seja, ele utiliza a equação (15) para cada bloco onde M agora é substituído por M/#Blocos.

A energia calculada desse modo será a médias das energias dos blocos e a variância será calculada normalmente onde a energia local será substituída por  $E_{B_i}^{(\alpha)}$  e  $E_T^{(\alpha)}$  será a média dessas energias e dessa forma obtemos o erro padrão para a energia.

Uma melhor explicação da influência desse parâmetro no método será comentada nos resultados, mas podemos imaginar que esse parâmetro afetará a variância da energia média total já que haverá, para cada # Blocos fornecidos, um respectivo número de elementos em cada bloco os quais produzirão uma nova energia local ou  $E_{B_i}^{(\alpha)}$ .

Essa idéia da média por blocos tenta minimizar as correlações produzidas pelo próprio algoritmo de Metropolis, mas nem sempre é fácil encontrar um número de Blocos adequado para cada situação.

A saída produzida pelo programa é evidentemente a energia e seu respectivo erro padrão. Além dessas duas variáveis, a fração de aceitos dos elementos da amostra também é mostrada (ver Fig.1), esse parâmetro é importante pois permite ajustar o  $\Delta$ . Outro parâmetro de saída é o desvio que a energia calculada está da prevista que no caso é 0.5. Além da função de onda visualizada, esse parâmetro de desvio ajuda o usuário a perceber se os parâmetros fornecidos ao programa estão adequados.

## 4.2 A construção da Applet-Java

O objetivo da utilização de uma Applet nesse projeto é tornar o aprendizado dos conceitos do cálculo variacional de Monte Carlo mais agradável e para isso uma interface gráfica se faz necessária. Nosso foco, portanto, é o método em si, porém vale descrever mesmo que básica e sucintamente as características do programa em Java.

Fortran é uma linguagem por procedimentos, isto é, o compilador vai seguindo a seqüência de instruções linha a linha acionando funções e sub-rotinas até alcançar a instrução "parar". Essa metodologia é muitas vezes conhecida como programação estruturada.

Java é uma linguagem orientada a objetos e difere de Fortran pela metodologia empregada na programação.

A versão utilizada do Java foi a j2sdk 1.4.1 e o IDE (o editor) o Eclipse.

A primeira coisa a se ter em mente na implementação de um programa estruturado para um orientado a objetos é justamente definir

as classes que farão parte do projeto, definir objetos e seus métodos e atributos.

As principais classes que permitem o funcionamento do programa são: `Oscilador.class` e `OsciladorApplet.class`. A primeira é a adaptação à linguagem Java do programa escrito em Fortran. A segunda é responsável pela interface gráfica que roda no navegador.

Basicamente, a classe `OsciladorApplet.class` utiliza um objeto da classe `Oscilador.class` e seus métodos são as mesmas funções programadas em Fortran.

### 4.3 Funcionamento da Applet-Java

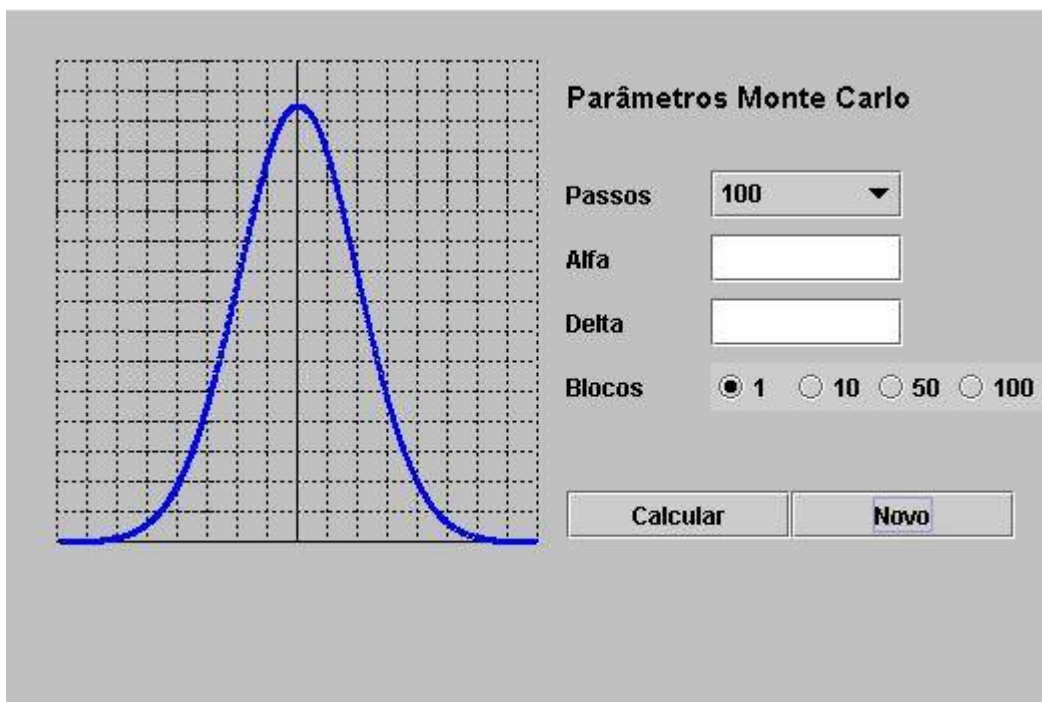


Figura 2 – Applet: Simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D

Na Fig. 2 podemos ver a Applet logo após ser iniciada. À esquerda temos o plano  $xy$  e a função de onda para o estado fundamental do oscilador no sistema onde  $m = \hbar = \omega = 1$  tendo como energia o valor  $E_0 = 0.5$ .

À direita estão os campos de entrada e os controles de comando.

Podemos fazer nossa primeira simulação ilustrativa com parâmetros arbitrários.

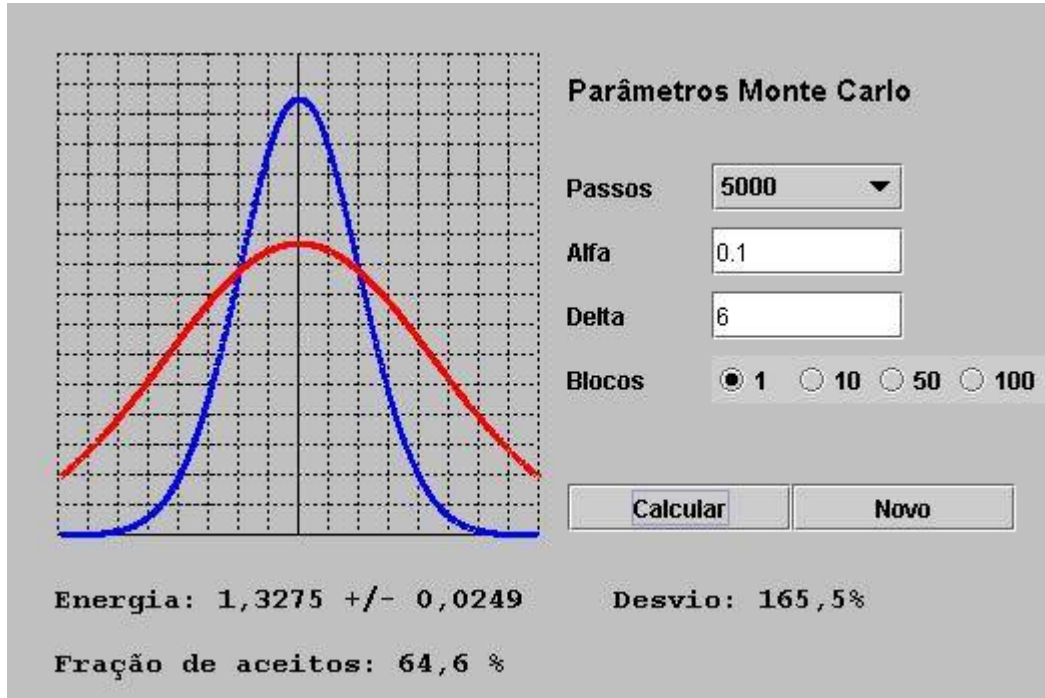


Figura 3 – Ilustração da simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D

Nessa simulação podemos ver em vermelho a função de onda tentativa com parâmetro  $\alpha = 0.1$  em contraste com a função de onda inicial com  $\alpha = 0.5$ .

A energia obtida com esses parâmetros foi  $E = 1,3275 \pm 0,0249$

Vemos que a energia difere 165,5% do valor esperado  $E_0 = 0.5$ .

A amostragem foi realizada com 64,6% de aceitação para delta com valor de 6,0.

A escala do gráfico é fixa e é a seguinte: os picos das curvas estão em  $x = 0$  e os limites horizontais em  $x = \pm 4,0$ , a escala vertical vai de 0 até aproximadamente 9,0.

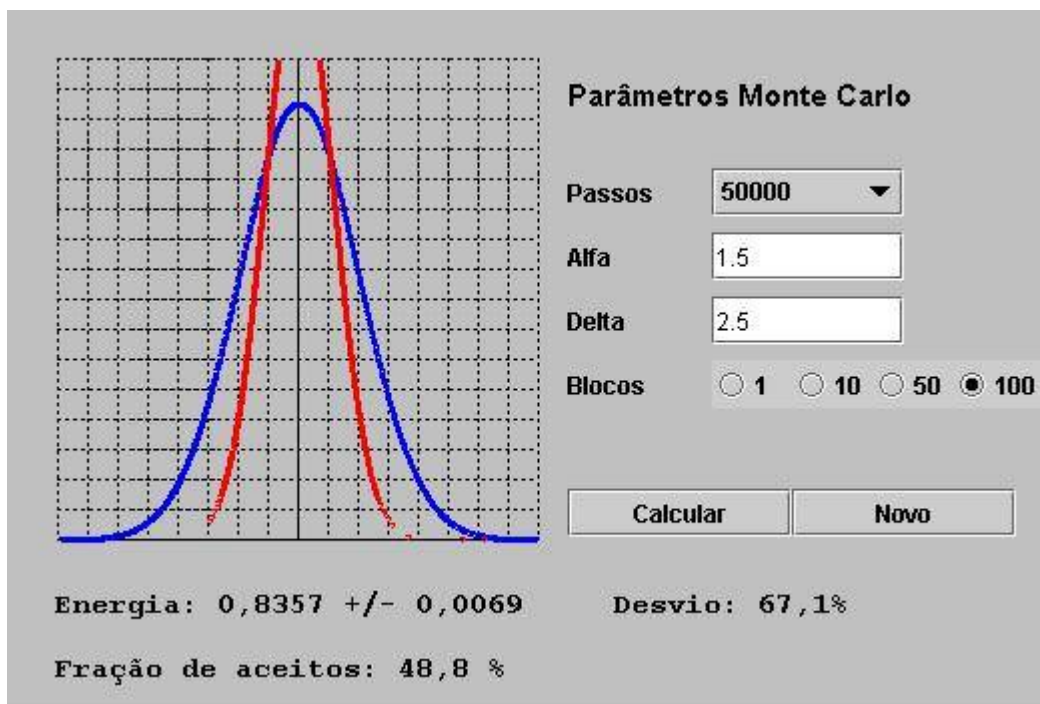


Figura 4 – Ilustração da simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D

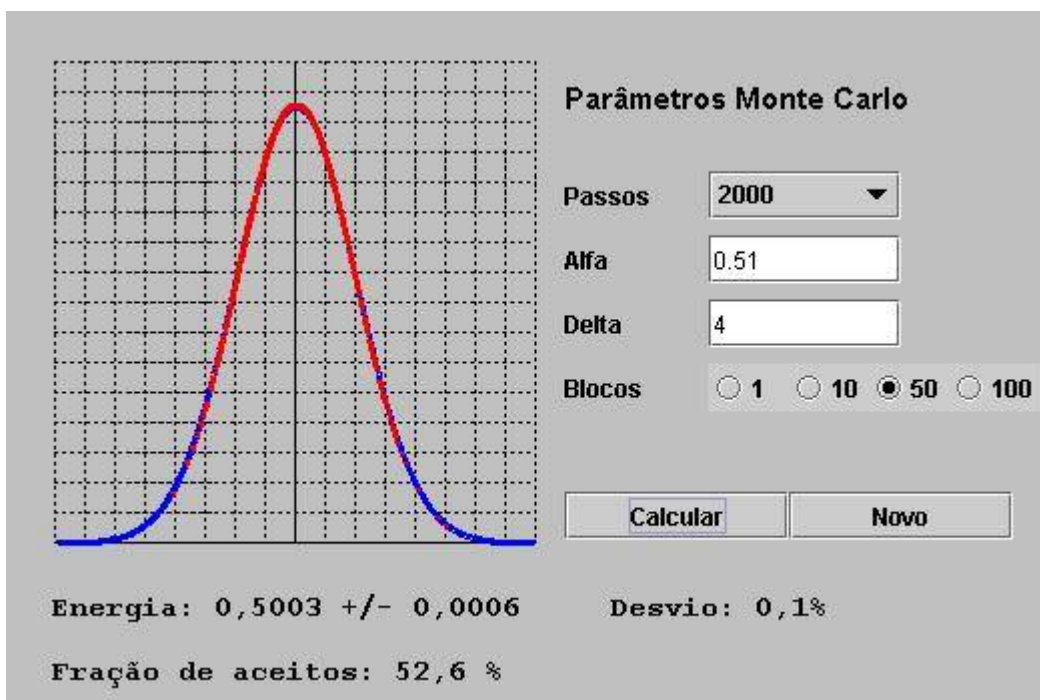


Figura 5 – Ilustração da simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D



Temos agora, a oportunidade de ver a influência do delta em relação ao número de passos conservando ou outros parâmetros da simulação anterior (Fig. 5).

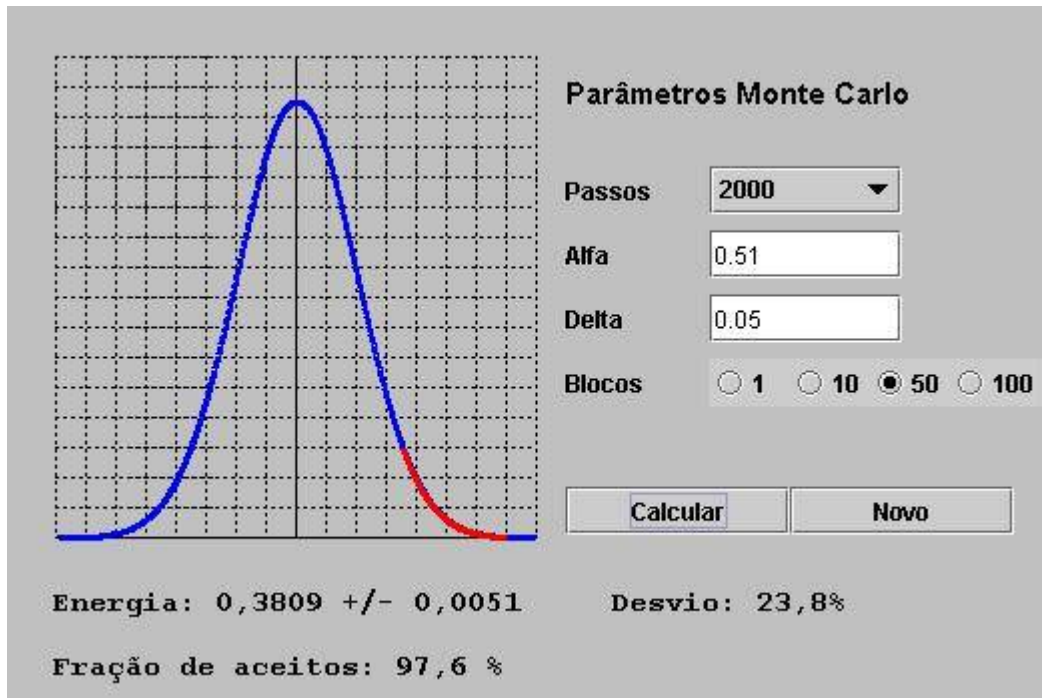


Figura 6 – Ilustração da simulação Monte Carlo aplicado ao oscilador harmônico quântico 1D

Com o valor do delta muito menor agora, não conseguimos produzir uma amostragem satisfatória que preencha a região de maior probabilidade. Isso acontece porque o primeiro elemento da amostra é arbitrário e com um delta pequeno temos consecutivamente descolamentos pequenos, os pontos da amostra não conseguem alcançar a região de maior probabilidade com apenas 2000 passos. É interessante notar que a fração de aceitação aumentou muito em relação à simulação da fig. 5, isso porque os pontos da amostra tem probabilidades “parecidas” e a aceitação está condicionada a esse fato mais um número aleatório de aceite ou rejeite ver Fig. 1.

## 5. Resultados

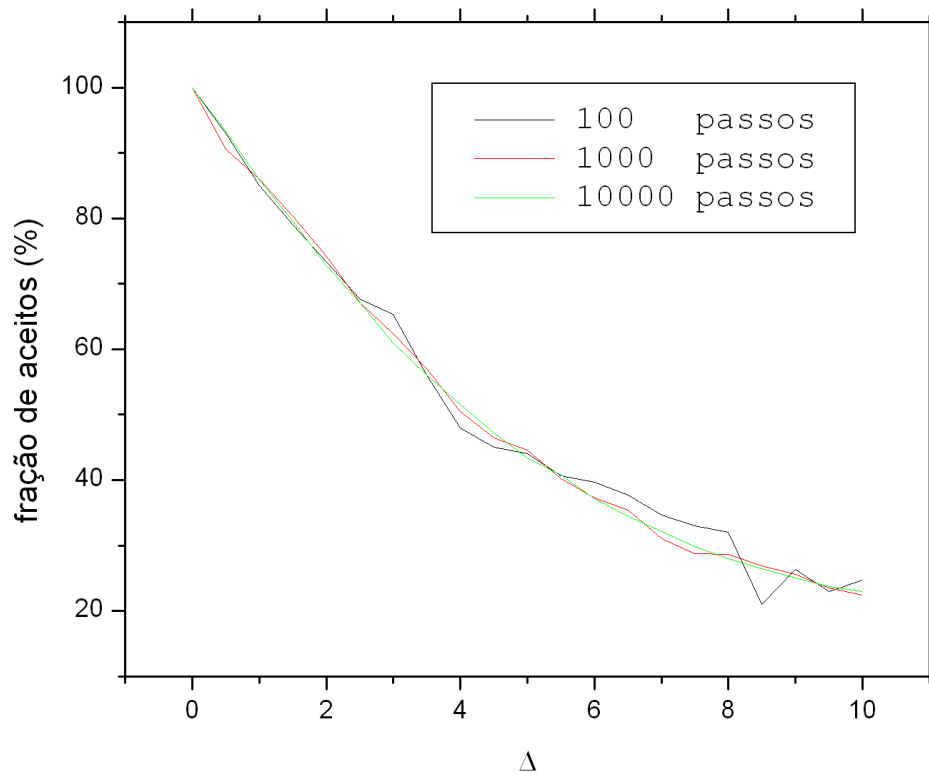
---

A seguir, o algoritmo é analisado em função dos parâmetros.

### 5.1 O parâmetro Delta ( $\Delta$ )

Para analisar o comportamento do  $\Delta$  em função a fração de aceitos, vamos atribuir  $\alpha = 0.5$  de modo que para todos os passos tenhamos a mesma função de onda tentativa e mesma energia.

Gráfico I - Análise do parâmetro  $\Delta$



Como era de se esperar, quanto menor o valor do delta maior aceitação se tem. Isso ocorre porque quando permitimos somente deslocamentos pequenos a probabilidade da candidata à nova configuração tem um valor próximo ao da probabilidade do elemento anterior que já pertence à amostra, como a condição de aceitação verifica se a razão das probabilidades do candidato à amostra e o seu elemento anterior é maior que um número aleatório em um intervalo de  $[0,1]$  é muito provável que haja aceitação dessa nova configuração já que a razão é aproximadamente a unidade.

O contrário ocorre para deltas muito grandes, a razão das probabilidades pode ser muito diferente uma da outra, pois com um delta grande permitimos grandes deslocamentos e assim comparando com o número aleatório no intervalo  $[0,1]$  pode ocorrer muita rejeição como de fato ocorre.

Uma regra empírica mostra que um bom valor para o delta é manter uma faixa de aceitação em torno de 50%, o que corresponde a um  $\Delta = 4$ .

Podemos ver também no gráfico que para um maior número de passos, mais suave é a variação da fração de aceitação em função do delta. Isso porque temos mais elementos na amostra permitindo abranger uma faixa maior de probabilidades com pequenos espaçamentos entre os elementos da amostra.

## 5.2 O número de blocos

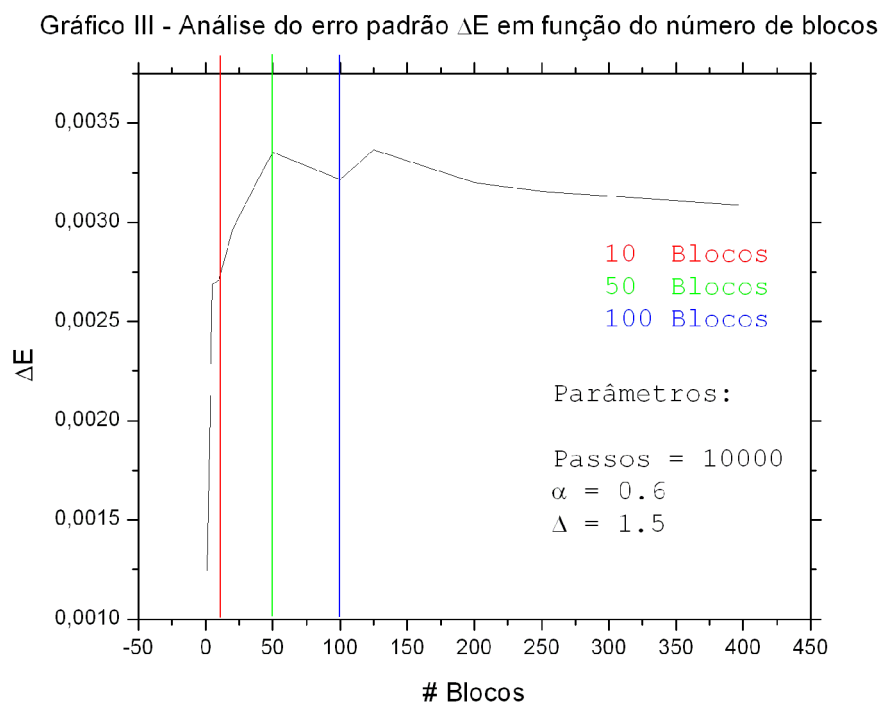
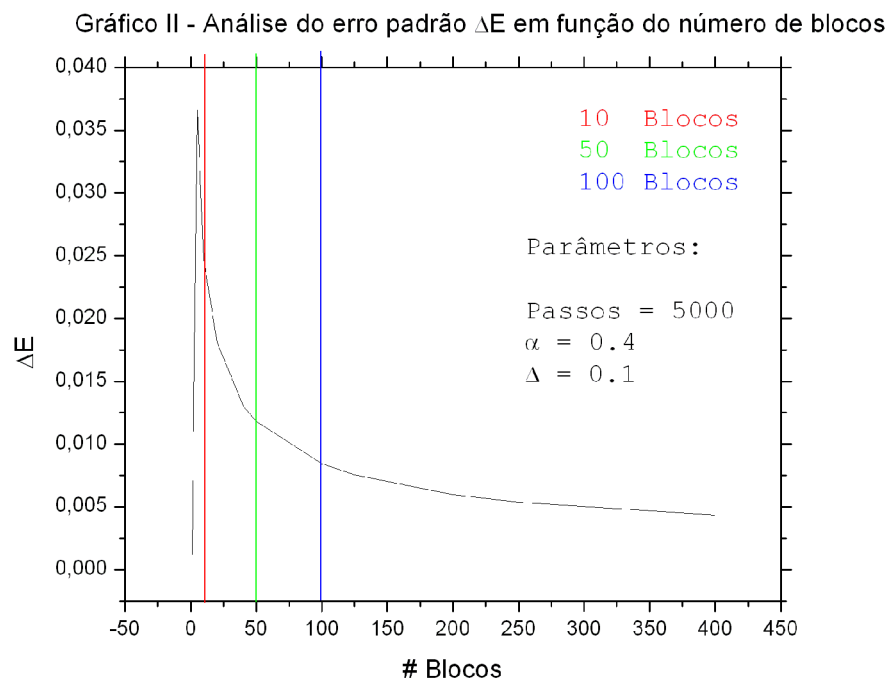
Fizemos um teste para saber qual o número de blocos adequado para estimarmos um erro padrão confiável. Após essa análise disponibilizamos somente alguns valores especiais de blocos para o usuário interagir. Esses valores darão uma estimativa mais correta da variância.

A análise, portanto, foi feita através do programa escrito em Fortran.

A metodologia para essa análise foi a seguinte: primeiro estipulamos um valor para  $\alpha$  próximo de 0.5 e um determinado número de passos. O delta foi escolhido tendo como referencia o gráfico I. Escolhemos um valor inferior a 4 para haver um número maior de aceitação para por em evidência o comportamento do erro padrão. Isso é justificado para simulações em que envolvam muitos corpos.

A partir de uma mesma amostragem as energias são calculadas com diferentes números de Blocos, neste caso, a energia será a mesma

para todas as configurações. Portanto, podemos estudar o comportamento do erro padrão em função do número de blocos.



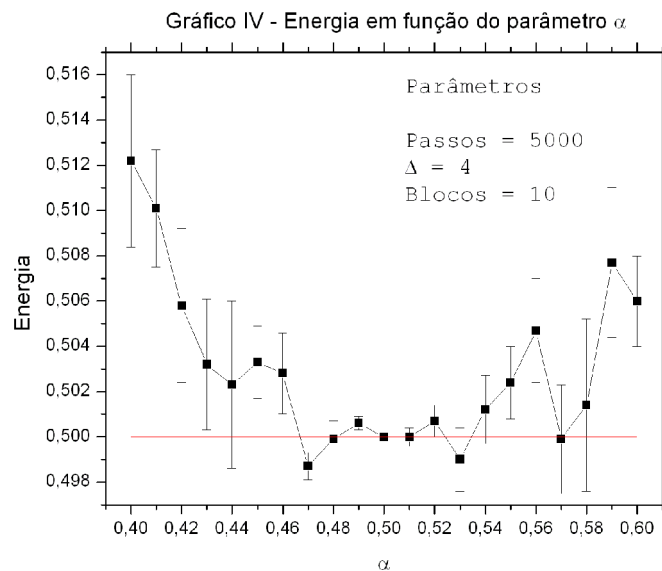
Os comportamentos de  $\Delta E$  para diferentes número de blocos são bem diferentes para parâmetros iniciais diferentes. Os gráficos II e III ilustram tal situação. Porém, nota-se uma região onde há um número de bloco específico para cada conjunto de simulação em que o erro padrão é máximo. Se para cada conjunto de parâmetros de entrada fizessemos o mesmo procedimento, ao final teríamos uma idéia do número de blocos que poderíamos usar.

Nos conjuntos de parâmetros ilustrados nos gráficos II e III, uma boa estimativa do erro seria usar um número de blocos entre 10 e 50, portanto, disponibilizamos para o usuário as quantidades 1, 10, 50 e 100 número de blocos.

### 5.3 O parâmetro alfa ( $\alpha$ )

Utilizando o programa e já sabendo qual valor de energia teremos que obter (10) podemos, a propósito de ilustração, podemos verificar qual o intervalo em que  $\alpha$  pode estar inserido levando em conto o erro padrão associado da energia utilizando um certo número de blocos.

Para estimarmos um erro padrão, vamos usar 10 blocos baseados nas discussões apresentadas na seção anterior.



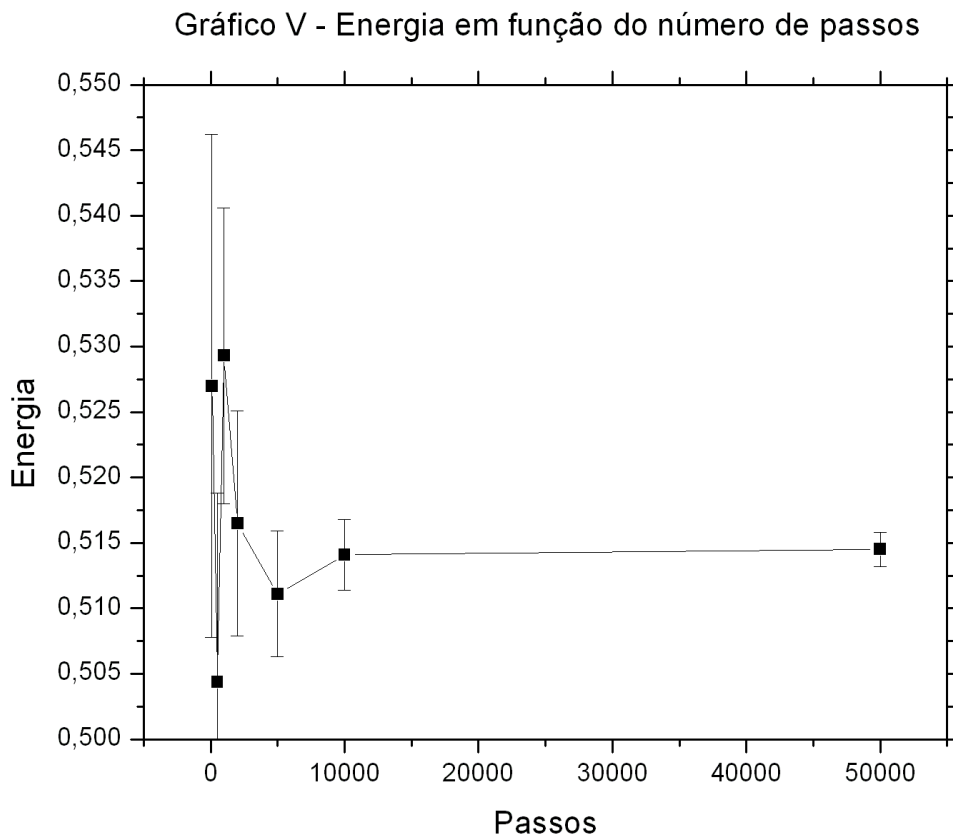
## 5.4 O número de passos

O objetivo aqui é ver o comportamento do cálculo da energia para diferentes valores de passos mantendo fixos os parâmetros alfa, delta e blocos, a saber:

$$\alpha = 0.4$$

$$\Delta = 4$$

$$\text{Blocos} = 10$$



Verificamos que quanto maior o número de passos mais estável é o valor da energia, pois o número de passos é justamente o número de medidas de energia local, isto é, a variância diminui com o aumento de elementos da amostra.

## 6. Conclusão

---

Na pesquisa científica onde os métodos exemplificados neste trabalho são extensamente empregados, as funções tentativas para o sistema muitas vezes não permitem soluções analíticas e, portanto, métodos numéricos devem ser empregados nos cálculos das quantidades de interesse.

Esse trabalho teve por objetivo elucidar o método de Monte Carlo utilizando uma aplicação computacional gráfica de fácil interatividade.

Podemos através dessa aplicação, conhecer os principais parâmetros do método e verificar sua influência no algoritmo e ter uma idéia da poderosa ferramenta que é o cálculo variacional.

Usamos um sistema físico bastante simples que dispensa a aplicação de métodos numéricos, mas esse sistema serviu de comparação aos nossos resultados.

O algoritmo de Metropolis é somente um dos métodos que fazem a amostragem de configurações.

Quando usamos um  $\alpha = 0.5$  para a função tentativa essa se torna o próprio autoestado do sistema do oscilador para o estado fundamental e obviamente o valor calculado para a energia concorda com o valor previsto e a variância do resultado numérico é nula.

## 7. Referências

---

- [1] Cohen-Tannoudji, C.; Bernard, D; Laloë, F. – “*Quantum Mechanics*”- Wiley-Interscience Publication, 1977
- [2] Gould, H.; Tobochnik, J. – “*An introduction to computer simulation methods: applications to physical systems*” – vol. 2 – Addison-Welwy, 1998
- [3] Canabrava, R; Vitiello, S. A. – Revista Brasileira de Ensino de Física, vol. 25 n° 1, São Paulo, 2003
- [4] Lawrence, N.; “*Compaq Visual Fortran*”, Digital Press, 2001.
- [5] Deitel, H. M.; Deitel, P. J. – “*Java, como programar*” trad. Carlo Arthur Lang Lisboa – Porto Alegre – Bookman, 2003
- [6] <http://java.sun.com/applets>
- [7] <http://java.sun.com/j2se/1.4.2/download.html>
- [8] <http://www.fortran.com>