

ALGORITMO VERLET

O algoritmo de *verlet* é usado para integrar as equações de movimento de *Car-Parrinello* no programa *Quantum Espresso*. O objetivo deste algoritmo de integração é determinar a posição da partícula na posição $t + \delta t$. O algoritmo é derivado usando a expansão de *Taylor*, ou seja,

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}(t)(\delta t)^2 + \frac{1}{3!}\dddot{\mathbf{r}}(t)(\delta t)^3 + O((\delta t)^4)$$

$$\mathbf{r}(t - \delta t) = \mathbf{r}(t) - \dot{\mathbf{r}}(t)\delta t + \frac{1}{2}\ddot{\mathbf{r}}(t)(\delta t)^2 - \frac{1}{3!}\dddot{\mathbf{r}}(t)(\delta t)^3 + O((\delta t)^4)$$

Todos os símbolos usados aqui apresentam significado usual, ou seja, \mathbf{r} representa a posição do átomo considerado, t representa o tempo e δt é o passo de integração. Somando estas duas equações, obtemos

$$\mathbf{r}(t + \delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \delta t) + \ddot{\mathbf{r}}(t)(\delta t)^2 + O((\delta t)^4).$$

Substituindo a aceleração $\ddot{\mathbf{r}}(t)$ nesta equação por $\ddot{\mathbf{r}}(t) = \mathbf{F}/m$, obtemos o algoritmo de *Verlet* para as posições dos átomos no instante $t + \delta t$:

$$\mathbf{r}(t + \delta t) \approx 2\mathbf{r}(t) - \mathbf{r}(t - \delta t) + \frac{\mathbf{F}(t)}{m}(\delta t)^2.$$

(Error! No text of specified style in document..1)

Para calcularmos a posição dos átomos na posição $\mathbf{r}(t + \delta t)$, precisamos conhecer a posição dos átomos nas posições $\mathbf{r}(t)$ e $\mathbf{r}(t - \delta t)$ e a força $\mathbf{F}(t)$ sobre os átomos no instante t . A força pode ser calculada usando a Equação

$$\mathbf{F} = -\nabla V(\mathbf{r}) = -\left(\frac{\partial V(x, y, z)}{\partial x}, \frac{\partial V(x, y, z)}{\partial y}, \frac{\partial V(x, y, z)}{\partial z}\right),$$

onde $\nabla V(\mathbf{r})$ é o gradiente da energia potencial na posição \mathbf{r} . O erro deste algoritmo é de 4ª ordem, $O((\delta t)^4)$, e tem a vantagem de ser simples, acurado, estável e é bastante popular entre os simuladores. No entanto, tem a desvantagem de não calcular as velocidades diretamente a partir das forças, embora não precisemos das velocidades para encontrar as novas posições.

As velocidades $\mathbf{v}(t)$ podem ser calculadas usando o método das diferenças finitas centradas, isto é,

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{1}{2}\mathbf{a}(t)(\delta t)^2 + \frac{1}{3!}\dot{\mathbf{a}}(t)(\delta t)^3 + O((\delta t)^4)$$

$$\mathbf{r}(t - \delta t) = \mathbf{r}(t) - \mathbf{v}(t)\delta t + \frac{1}{2}\mathbf{a}(t)(\delta t)^2 - \frac{1}{3!}\dot{\mathbf{a}}(t)(\delta t)^3 + O((\delta t)^4)$$

(Error! No text of specified style in document..2)
(Error! No text of specified style in document..3)

Subtraindo a segunda da primeira, obtemos:

$$\mathbf{r}(t + \delta t) - \mathbf{r}(t - \delta t) = 2\mathbf{v}(t)\delta t + O((\delta t)^3)$$

$$\mathbf{v}(t) = \frac{\mathbf{r}(t + \delta t) - \mathbf{r}(t - \delta t)}{2\delta t}.$$

Observe que o erro no cálculo das velocidades $\mathbf{v}(t)$ está na terceira ordem. As velocidades são usadas no algoritmo de Verlet para calcular as energias cinéticas dos átomos do sistema. Com as energias cinéticas, podemos calcular a temperatura instantânea do sistema.

ALGORITMO VELOCITY VERLET

Como já mencionamos anteriormente, o principal problema do algoritmo de Verlet é que as velocidades no instante t só são calculadas após obter as posições no instante $t + \delta t$. Isto dificulta a implementação do algoritmo e torna mais difícil os cálculos com pressões constantes em que as forças dependem das velocidades no instante t . Vale a pena mencionar que estamos particularmente interessados em algoritmos que levem a conservação da energia e que sejam reversíveis, pelo menos em intervalos de tempo pequenos. Uma modificação do algoritmo de Verlet, conhecido como *Velocity Verlet*, procura sanar estas dificuldades. As posições no instante $t + \delta t$, no algoritmo de *Velocity Verlet*, são derivadas usando a expansão de Taylor:

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \mathbf{v}(t)\delta t + \frac{1}{2}\mathbf{a}(t)(\delta t)^2,$$

(Error!
No text of
specified
style in
document..4)

Derivando (Error! **No text of specified style in document..4**) em relação ao tempo, obtemos

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \mathbf{a}(t)\delta t + \frac{1}{2!}\dot{\mathbf{a}}(t)(\delta t)^2.$$

(Error!
No text of
specified
style in
document..5)

Usando o método das diferenças finitas progressivas, podemos escrever $\dot{\mathbf{a}}(t)$ como

$$\dot{\mathbf{a}}(t) = \frac{\mathbf{a}(t + \delta t) - \mathbf{a}(t)}{\delta t}.$$

Usando este resultado em (Error! **No text of specified style in document..5**), obtemos

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \mathbf{a}(t)\delta t + \frac{1}{2!} \frac{\mathbf{a}(t + \delta t) - \mathbf{a}(t)}{\delta t} (\delta t)^2$$

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \mathbf{a}(t)\delta t + \frac{\mathbf{a}(t + \delta t) - \mathbf{a}(t)}{2} \delta t$$

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \frac{\mathbf{a}(t + \delta t) + \mathbf{a}(t)}{2} \delta t$$

(Error!
No text of
specified
style in
document..6)

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \frac{\mathbf{F}(t+\delta t)+\mathbf{F}(t)}{2m} \delta t$$

A Equação (Error! **No text of specified style in document..6**) mostra que as velocidades no algoritmo de Velocity Verlet são calculadas diretamente das forças e no mesmo tempo em que se calculam as posições. As equações

$$\mathbf{r}(t + \delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \delta t) + \frac{\mathbf{F}(t)}{m}(\delta t)^2.$$

$$\mathbf{v}(t + \delta t) = \mathbf{v}(t) + \frac{\mathbf{F}(t + \delta t) + \mathbf{F}(t)}{2m} \delta t$$

(Error! **No text of specified style in document..7**)

(Error! **No text of specified style in document..8**)

são as equações de integração do algoritmo de Velocity Verlet. Os algoritmos de Verlet e Velocity Verlet apresentam as vantagens de conservarem a energia por um longo período e de serem, por um curto período, reversível temporalmente.

ALGORITMO *LEAP-FROG*

O algoritmo de integração *leap-frog* é usado no programa de química quântica HyperChem. Este algoritmo pode ser derivado fazendo a expansão em série de Taylor nos tempos $t + \delta t/2$ e $t - \delta t/2$ e depois fazendo a subtração entre as duas expansões e deslocando o tempo nos argumentos em $t + \delta t/2$:

$$\mathbf{r}\left(t + \frac{\delta t}{2}\right) = \mathbf{r}(t) + \dot{\mathbf{r}}(t) \frac{\delta t}{2}$$

$$\mathbf{r}\left(t - \frac{\delta t}{2}\right) = \mathbf{r}(t) - \dot{\mathbf{r}}(t) \frac{\delta t}{2}$$

Subtraindo a segunda equação da primeira, obtemos

$$\mathbf{r}\left(t + \frac{\delta t}{2}\right) - \mathbf{r}\left(t - \frac{\delta t}{2}\right) = 2\dot{\mathbf{r}}(t) \frac{\delta t}{2}.$$

Agora, fazendo $t = t + \delta t/2$, obtemos

$$\mathbf{r}\left(t + \frac{\delta t}{2} + \frac{\delta t}{2}\right) = \mathbf{r}\left(t + \frac{\delta t}{2} - \frac{\delta t}{2}\right) + \dot{\mathbf{r}}\left(t + \frac{\delta t}{2}\right) \delta t$$

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \dot{\mathbf{r}}\left(t + \frac{\delta t}{2}\right) \delta t$$

(Error!

No text of

specified

style in

document..9)

$$\mathbf{r}(t + \delta t) = \mathbf{r}(t) + \mathbf{v}\left(t + \frac{\delta t}{2}\right) \delta t.$$

Esta é a Equação de leap-frog para o cálculo das posições. Esta equação depende das velocidades, cuja equação pode ser derivada fazendo a expansão da série de Taylor ou usando diferenças finitas centrais:

$$\mathbf{v}\left(t + \frac{\delta t}{2}\right) = \mathbf{v}(t) + \dot{\mathbf{v}}(t) \frac{\delta t}{2} + \frac{1}{2!} \ddot{\mathbf{v}}(t) \left(\frac{\delta t}{2}\right)^2 + O\left(\left(\frac{\delta t}{2}\right)^3\right)$$

$$\mathbf{v}\left(t - \frac{\delta t}{2}\right) = \mathbf{v}(t) - \dot{\mathbf{v}}(t) \frac{\delta t}{2} + \frac{1}{2!} \ddot{\mathbf{v}}(t) \left(\frac{\delta t}{2}\right)^2 - O\left(\left(\frac{\delta t}{2}\right)^3\right)$$

Subtraindo a segunda da primeira, obtemos:

$$\mathbf{v}\left(t + \frac{\delta t}{2}\right) = \mathbf{v}\left(t - \frac{\delta t}{2}\right) + \mathbf{a}(t) \delta t$$

ou

(Error!

No text of

specified

style in

document..10)

$$\mathbf{v}\left(t + \frac{\delta t}{2}\right) = \mathbf{v}\left(t - \frac{\delta t}{2}\right) + \frac{\mathbf{F}(t)}{m} \delta t$$

As equações (Error! **No text of specified style in document..9**) e (Error! **No text of specified style in document..10**) são as equações de integração do algoritmo de Leap-Frog. Este algoritmo calcula as velocidades no instante $t + \delta t/2$ e em seguida calcula as posições no instante $t + \delta t$. Portanto, a posição pula o tempo da velocidade e depois a velocidade pula o tempo da posição. Daí o nome *leap-frog*.

A acuracidade para um passo de integração maior é importante na simulação, pois quanto maior for o passo de integração menor será o número de cálculo das forças por unidade de tempo de simulação. Portanto, é vantagem usar um algoritmo sofisticado que permite o uso de um passo de integração grande. Algoritmos mais precisos usam

derivadas de mais alta ordem, o que aumenta a quantidade de memória necessária para armazenamento dos dados. Isso não chega a ser um problema, pois a maioria dos computadores usados em simulações possuem memórias suficientes, a não ser que o sistema seja realmente grande. A conservação de energia é algo importante na simulação. Algoritmos de mais alta ordens tendem ser bons para conservar a energia por períodos curtos (poucos passos), mas, em geral, apresentam desvios sistemáticos por longos períodos de simulação. Os algoritmos no estilo de Verlet apresentam conservação de energia moderada em períodos curtos de simulação. No entanto, os desvios em períodos longos de simulação são moderados.

Duas simulações com as mesmas condições iniciais apresentaram trajetórias que serão bem diferentes devidos a propagação dos erros dos algoritmos de integração. Contudo, não devemos nos desesperar com isso, pois estamos interessados em previsões estatísticas. As trajetórias assim obtidas são representativas do espaço de fase, mesmo que não sejam as verdadeiras trajetórias. Em resumo, as principais características dos algoritmos no estilo Verlet são:

- 1) São bastante rápidos;
- 2) Não são muito acurados para simulações longas;
- 3) Requer pouca memória para armazenamento de dados;
- 4) A conservação da energia para simulações pequenas é razoável;
- 5) Os desvios sistemáticos da energia nas simulações longas são pequenos.