As bases da Dinâmica Molecular - 2

Alexandre Diehl

Departamento de Física - UFPel

O algoritmo de velocity-Verlet

J. Chem. Phys. 76, 637 (1982)

A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters

William C. Swope and Hans C. Andersen

Department of Chemistry, Stanford University, Stanford, California 94305

Peter H. Berens and Kent R. Wilson

Department of Chemistry, University of California—San Diego, La Jolla, California 92903
(Received 16 July 1981; accepted 17 September 1981)

We present a molecular dynamics computer simulation method for calculating equilibrium constants for the formation of physical clusters of molecules. The method is based on Hill's formal theory of physical clusters. In the method, a molecular dynamics calculation is used to calculate the average potential energy of a cluster of molecules as a function of temperature, and the equilibrium constants are calculated from the integral of the energy with respect to reciprocal temperature. The method is illustrated by calculations of the equilibrium constants for the formation of clusters of two to five water molecules that interact with each other by an intermolecular potential devised by Watts. The method is compared with other procedures for calculating the thermodynamic properties of clusters.

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2 \qquad (1)$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \qquad (2)$$

Da equação (2) temos $x(t-\Delta t) = x(t+\Delta t) - 2\dot{x}(t)\Delta t$

Que substituída na equação (1) resulta em

$$x(t+\Delta t) = 2x(t) - [x(t+\Delta t) - 2\dot{x}\Delta t] + \ddot{x}(t)\Delta t^2$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2 \qquad (1)$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \qquad (2)$$

Da equação (2) temos $x(t-\Delta t) = x(t+\Delta t) - 2\dot{x}(t)\Delta t$

Que substituída na equação (1) resulta em

$$2x(t+\Delta t) = 2x(t) + 2\dot{x}\Delta t + \ddot{x}(t)\Delta t^2$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2 \qquad (1)$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \qquad (2)$$

Da equação (2) temos $x(t-\Delta t) = x(t+\Delta t) - 2\dot{x}(t)\Delta t$

Que substituída na equação (1) resulta na **equação de evolução da posição**

$$\boxed{x(t+\Delta t) = x(t) + \dot{x}\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2} \qquad (3)$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \qquad (2)$$

$$t \rightarrow t + \Delta t$$

Para obter a equação de evolução para a velocidades, usamos a equação (2):

$$\dot{x}(t+\Delta t) = \frac{x(t+\Delta t+\Delta t) - x(t+\cancel{\Delta t}-\cancel{\Delta t})}{2\Delta t}$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t} \qquad (2)$$

Para obter a equação de evolução para a velocidades, usamos a equação (2):

$$\dot{x}(t+\Delta t) = \frac{x(t+2\Delta t) - x(t)}{2\Delta t} \qquad (4)$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2 \qquad (1)$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Para obter $x(t+2\Delta t)$ usamos (1), com $t \rightarrow t+\Delta t$

$$x(t+\Delta t+\Delta t) = 2x(t+\Delta t) - x(t+\cancel{\Delta} t-\cancel{\Delta} t) + \ddot{x}(t+\Delta t)\Delta t^2$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2 \qquad (1)$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Para obter $x(t+2\Delta t)$ usamos (1), com $t \to t+\Delta t$

$$x(t+2\Delta t) = 2x(t+\Delta t) - x(t) + \ddot{x}(t+\Delta t)\Delta t^2 \qquad (5)$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Substituindo (5) em (4) teremos

$$\dot{x}(t+\Delta t) = \frac{2x(t+\Delta t) - \overset{1}{x(t)} + \ddot{x}(t+\Delta t)\Delta t^2 - \overset{1}{x(t)}}{2\Delta t}$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Substituindo (5) em (4) teremos

$$\dot{x}(t+\Delta t) = \frac{2x(t+\Delta t) - 2x(t) + \ddot{x}(t+\Delta t)\Delta t^2}{2\Delta t}$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Substituindo (5) em (4) teremos

$$\dot{x}(t+\Delta t) = \frac{x(t+\Delta t) - x(t)}{\Delta t} + \frac{1}{2}\ddot{x}(t+\Delta t)\Delta t \qquad (6)$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Substituindo (3) em (6) teremos

$$\dot{x}(t+\Delta t) = \frac{\cancel{x(t)} + \dot{x}(t)\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2 - \cancel{x(t)}}{\Delta t} + \frac{1}{2}\ddot{x}(t+\Delta t)\Delta t$$

O algoritmo de velocity-Verlet

Derivação, usando as equações de Verlet

$$x(t+\Delta t) = 2x(t) - x(t-\Delta t) + \ddot{x}(t)\Delta t^2$$

$$\dot{x}(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$$

Substituindo (3) em (6) teremos e **equação de evolução da velocidade**

$$\boxed{\dot{x}(t+\Delta t) = \dot{x}(t) + \frac{1}{2}\left[\ddot{x}(t+\Delta t) + \ddot{x}(t)\right]\Delta t} \quad (7)$$

O algoritmo de velocity-Verlet

Equações de evolução para velocity-Verlet

$$\boxed{x(t+\Delta t) = x(t) + \dot{x}\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2}$$

$$\boxed{\dot{x}(t+\Delta t) = \dot{x}(t) + \frac{1}{2}\left[\ddot{x}(t+\Delta t) + \ddot{x}(t)\right]\Delta t}$$

Vantagens do método de velocity-Verlet

- Pode ser usado a partir de tempo igual a zero.
- Velocidade não é obtida a partir da razão entre números pequenos.
- Posição e velocidade avaliados no mesmo tempo.

O algoritmo de velocity-Verlet

Equações de evolução para velocity-Verlet

$$\boxed{x(t+\Delta t) = x(t) + \dot{x}\Delta t + \frac{1}{2}\ddot{x}(t)\Delta t^2}$$

$$\boxed{\dot{x}(t+\Delta t) = \dot{x}(t) + \frac{1}{2}\left[\ddot{x}(t+\Delta t) + \ddot{x}(t)\right]\Delta t}$$

Desvantagens do método de velocity-Verlet

- Exige o cálculo das acelerações duas vezes num mesmo ciclo de integração das EDOs.

t e $t+\Delta t$

O algoritmo de velocity-Verlet

A cada ciclo de integração das EDOs

1. Avançamos as velocidades até metade do incremento de tempo, usando:

$$\dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right)=\dot{\vec{r}}(t)+\ddot{\vec{r}}(t)\frac{\Delta t}{2}$$

$$t \rightarrow t+\frac{\Delta t}{2}$$

2. Com estas velocidades, avançamos as posições por um incremento de tempo, usando:

$$\vec{r}(t+\Delta t)=\vec{r}(t)+\dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right)\Delta t$$

pois isto será equivalente à equação de evolução das posições de $t \rightarrow t+\Delta t$

$$\vec{r}(t+\Delta t)=\vec{r}(t)+\dot{\vec{r}}\Delta t+\frac{1}{2}\ddot{\vec{r}}(t)\Delta t^2$$

O algoritmo de velocity-Verlet

A cada ciclo de integração das EDOs

$\ddot{\vec{r}}(t+\Delta t)$

3. Com as novas posições, calculamos as novas acelerações,
usando a rotina derivada a partir do potencial de interação proposto.

4. Com as novas acelerações, atualizamos as velocidades até $t+\Delta t$

$$\dot{\vec{r}}(t+\Delta t) = \dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right) + \ddot{\vec{r}}(t+\Delta t)\frac{\Delta t}{2}$$

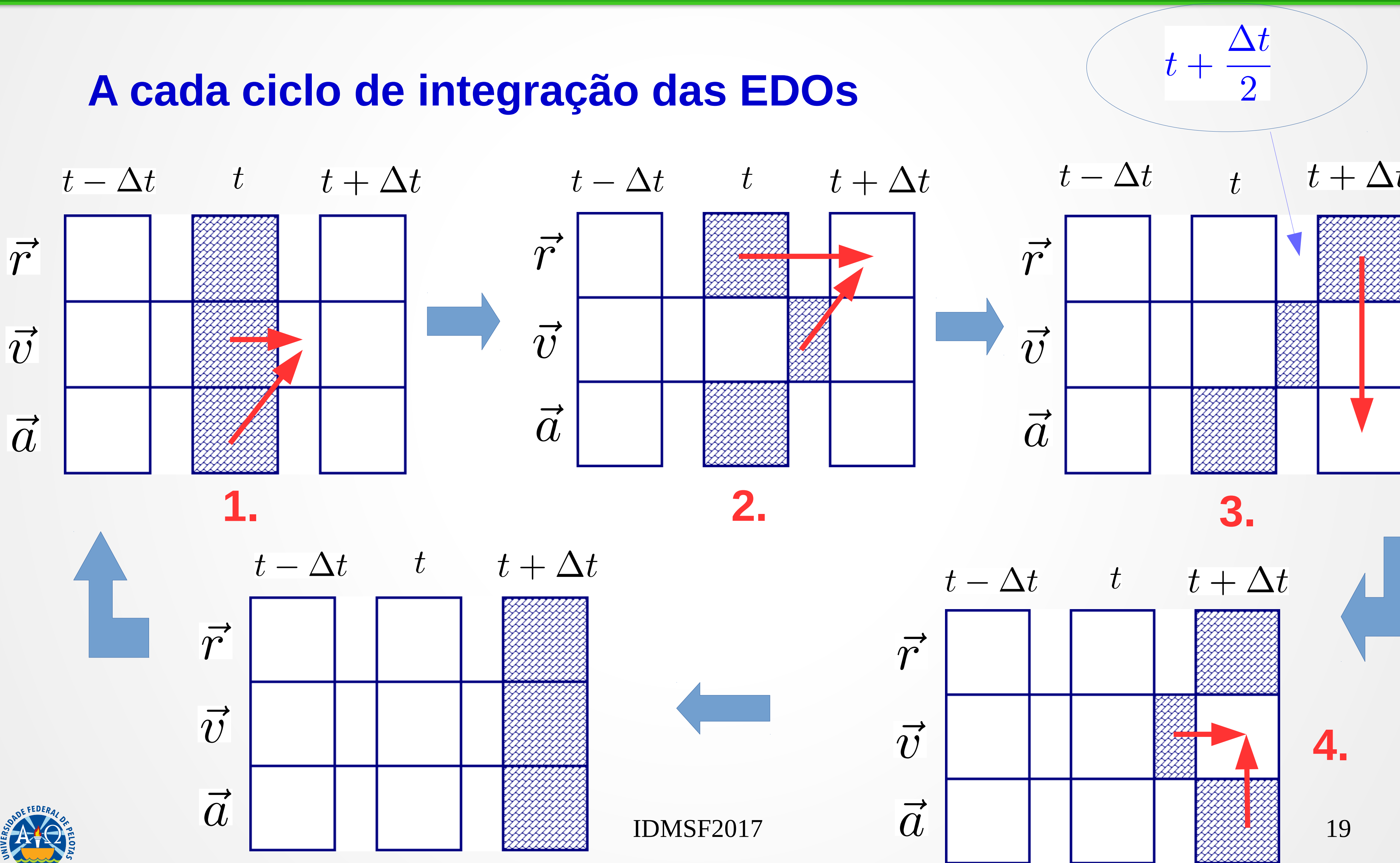
pois isto será equivalente à equação de evolução das velocidades de $t \to t+\Delta t$

$$\dot{\vec{r}}(t+\Delta t) = \dot{\vec{r}}(t) + \frac{1}{2}\left[\ddot{\vec{r}}(t+\Delta t) + \ddot{\vec{r}}(t)\right]\Delta t$$

Repetimos o ciclo, avançando as posições e velocidades para cada novo incremento de tempo.

O algoritmo de velocity-Verlet

A cada ciclo de integração das EDOs



O algoritmo de velocity-Verlet

Rotina de integração das EDOs

Etapas 1 e 2

$$\dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right)=\dot{\vec{r}}(t)+\ddot{\vec{r}}(t)\frac{\Delta t}{2}$$

Δt

```
do i = 1, N
   do k = 1,3
      vxyz(k,i) = vxyz(k,i) + 0.5 * axyz(k,i) * dt

      xyz(k,i) = xyz(k,i) + vxyz(k,i) * dt

   end do
end do
```

$$\vec{r}(t+\Delta t)=\vec{r}(t)+\dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right)\Delta t$$

Etapa 3

Com $\vec{r}\,(t+\Delta t)$ calculamos as novas acelerações $\ddot{\vec{r}}\,(t+\Delta t)$

O algoritmo de velocity-Verlet

Rotina de integração das EDOs

Etapa 4

$$\dot{\vec{r}}(t + \Delta t) = \dot{\vec{r}}\left(t + \frac{\Delta t}{2}\right) + \ddot{\vec{r}}(t + \Delta t)\frac{\Delta t}{2}$$

```
do i = 1, N
   do k = 1,3
      vxyz(k,i) = vxyz(k,i) + 0.5 * axyz(k,i) * dt
   end do
end do
```

O cálculo das velocidades deve ser feito em dois momentos:

- Um com os valores de aceleração a tempo t
- Outro com os valores de aceleração a tempo $t + \Delta t$

O algoritmo de velocity-Verlet

Proposta de estrutura em MD, usando velocity-Verlet

```
program velocity_verlet
  call init ()
  call force ()
  t = 0.0
  do while (t < tmax)
     call integrate (1)
     call force ()
     call integrate (2)
     t = t + dt
     call sample ()
  end do
end program velocity_verlet
```

Inicializa posições e velocidades

Cálculo de acelerações $\ddot{\vec{r}}(t)$

$\dot{\vec{r}}\left(t+\frac{\Delta t}{2}\right)$ e $\vec{r}(t+\Delta t)$

$\ddot{\vec{r}}(t+\Delta t)$

$\dot{\vec{r}}(t+\Delta t)$

Cálculo de propriedades de interesse

O algoritmo de velocity-Verlet

Proposta de estrutura em MD, usando velocity-Verlet

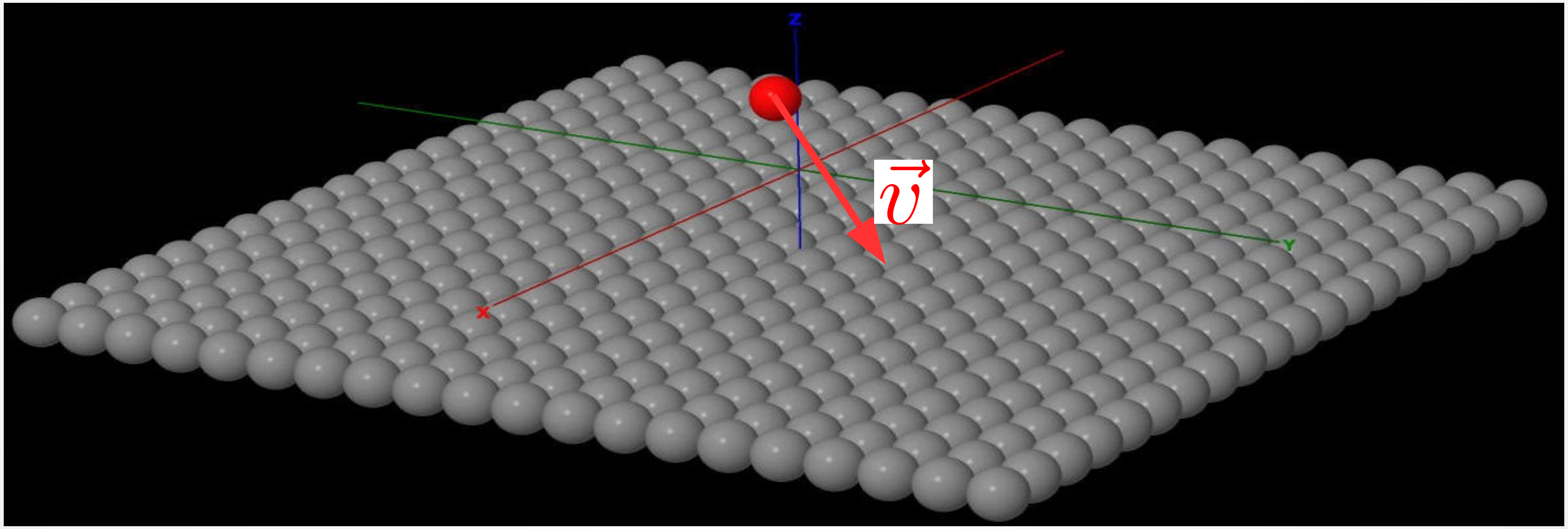
```
subroutine integrate (switch)
  if (switch == 1) then
     do i = 1, N
        do k = 1,dim
           vxyz(k,i) = vxyz(k,i) + 0.5 * axyz(k,i) * dt
           xyz(k,i) = xyz(k,i) + vxyz(k,i) * dt
        end do
     end do
  else
     do i = 1, N
        do k = 1,dim
           vxyz(k,i) = vxyz(k,i) + 0.5 * axyz(k,i) * dt
        end do
     end do
  end if
end subroutine integrate
```

Dimensão do espaço

O algoritmo de velocity-Verlet

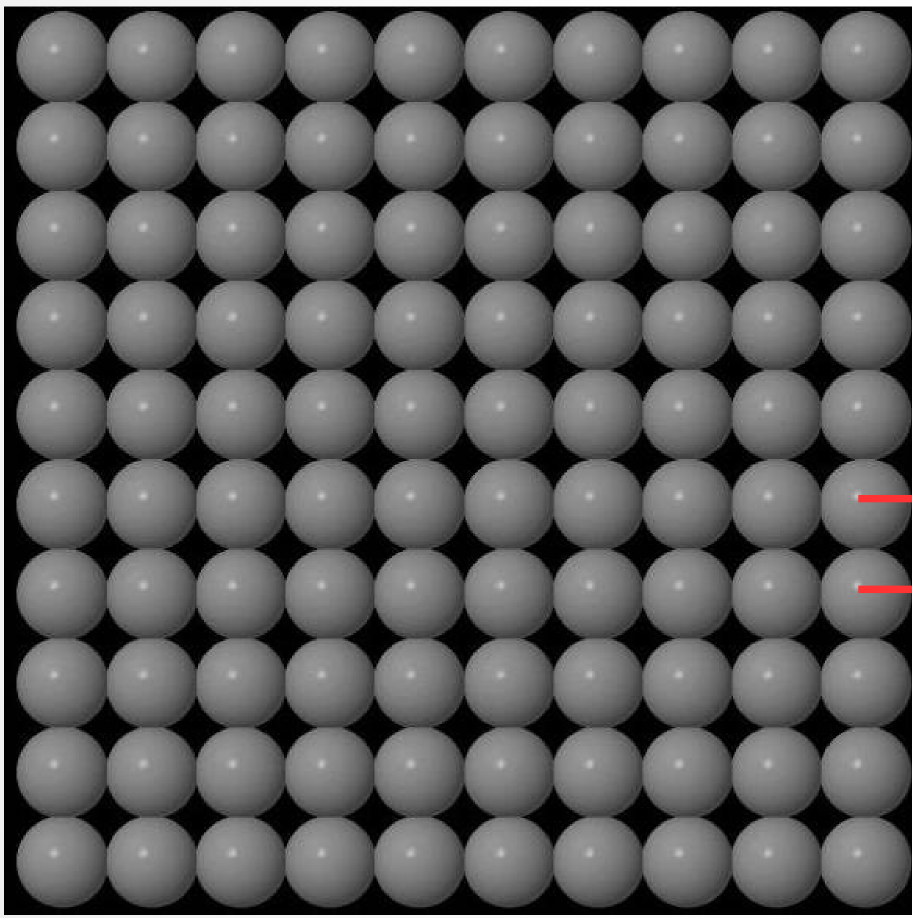
Exercício: partícula colidindo com uma parede

1. Usar o potencial WCA para descrever a interação entre a partícula e a parede.
2. A parede é formada por N_{wall} partículas fixas e de mesmo tamanho.
3. Colocar a parede em $z = 0$.
4. Use um ciclo temporal de integração, que permita uma colisão com a parede, dada uma velocidade inicial para a partícula.



O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede



Construção da parede

A parede é quadrada: $L_x = L_y = L$

Diâmetro das partículas na parede: σ_{wall}

Número de partículas na parede: N_{wall}

Número de linhas (e colunas) na parede:

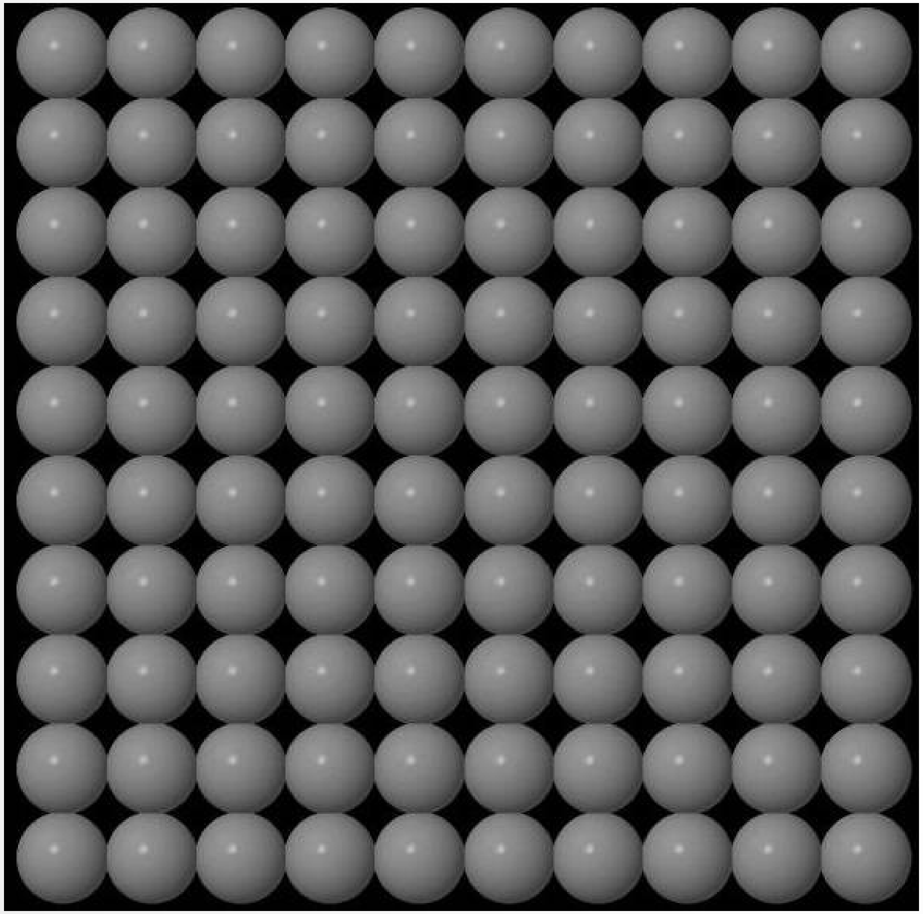
$$n_{\text{row}} = \sqrt{N_{\text{wall}}}$$

Dimensão linear da parede:

$$L = n_{\text{row}} \sigma_{\text{wall}}$$

O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede



Propriedades para calcular

- Faça um gráfico das componentes ***x***, ***y*** e ***z*** do vetor velocidade da partícula em função do tempo de simulação.
- Faça um gráfico da projeção ***z*** do vetor de posição da partícula em função do tempo.

Sugestão:

Adapte as rotinas de força e integração, levando em conta que apenas uma partícula deve ser movimentada.

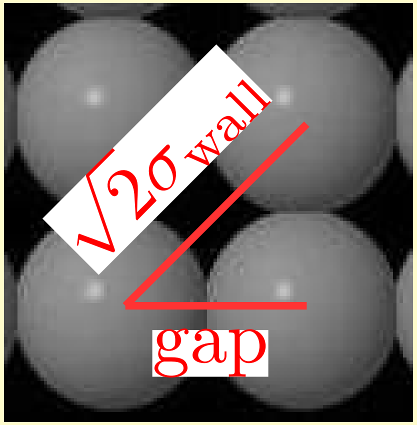
O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

```
subroutine init ()
  implicit none
  integer :: i, j, row, wsite
  real :: dx, dy, gap, L, theta
  row = int(Nwalls**(1.0/2.0))
  L = row*sigma;  gap = L/real(row);  wsite = 0;  dx = -gap/2.0
  do i = 1, row
     dx = dx + gap
     dy = -gap/2.0
     do j = 1,row
        dy = dy + gap
        if (wsite < Nwalls) then
           wsite = wsite + 1
           xyz_wall(1,wsite) = dx
           xyz_wall(2,wsite) = dy
           xyz_wall(3,wsite) = 0.0
        end if
     end do
  end do
  xyz(1,1) = 0.5;  xyz(2,1) = 0.5;  xyz(3,1) = 3.5
  vxyz(1,1) = 0.87;  vxyz(2,1) = 0.0;  vxyz(3,1) = -0.5

end subroutine init
```

$\text{gap} = \sigma_{\text{wall}}$



Condições iniciais para a partícula que colide

O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

Visualização do movimento da partícula incidente

Use um programa de visualização de moléculas, como por exemplo Jmol e VMD.

O **Jmol** pode ser instalado dos repositórios do Ubuntu.

Nos dois casos, um arquivo com a extensão **.xyz** deve ser fornecido, contendo as coordenadas das partículas envolvidas na visualização.

Formato do arquivo .xyz

- **1ª linha**: número N de átomos que serão visualizados.
- **2ª linha**: reservada para algum comentário.
- **3ª linha em diante**: as coordenadas das N partículas, x, y e z, e uma letra indicando o elemento químico representado.

O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

Visualização do movimento da partícula incidente

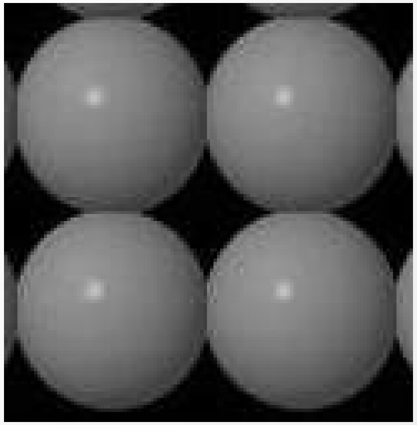
Use um programa de visualização de moléculas, como por exemplo Jmol e VMD.

O **Jmol** pode ser instalado dos repositórios do Ubuntu.

Nos dois casos, um arquivo com a extensão **.xyz** deve ser fornecido, contendo as coordenadas das partículas envolvidas na visualização.

Formato do arquivo .xyz

```
4
Parede com 4 partículas
'C'   0.5  0.5  0.0
'C'   1.0  0.5  0.0
'C'   0.5  1.0  0.0
'C'   1.0  1.0  0.0
```



C de Carbono

O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

Visualização do movimento da partícula incidente

Para fazer um "filme" da dinâmica da partícula, basta criar um arquivo **.xyz** com a repetição da estrutura básica, uma para cada instante de tempo da dinâmica.

```
4
Parede com 4 partículas
'C'   0.5  0.5  0.0
'C'   1.0  0.5  0.0
'C'   0.5  1.0  0.0
'C'   1.0  1.0  0.0
4
Parede com 4 partículas
'C'   0.5  0.5  0.0
'C'   1.0  0.5  0.0
'C'   0.5  1.0  0.0
'C'   1.0  1.0  0.0
.
.
.
```

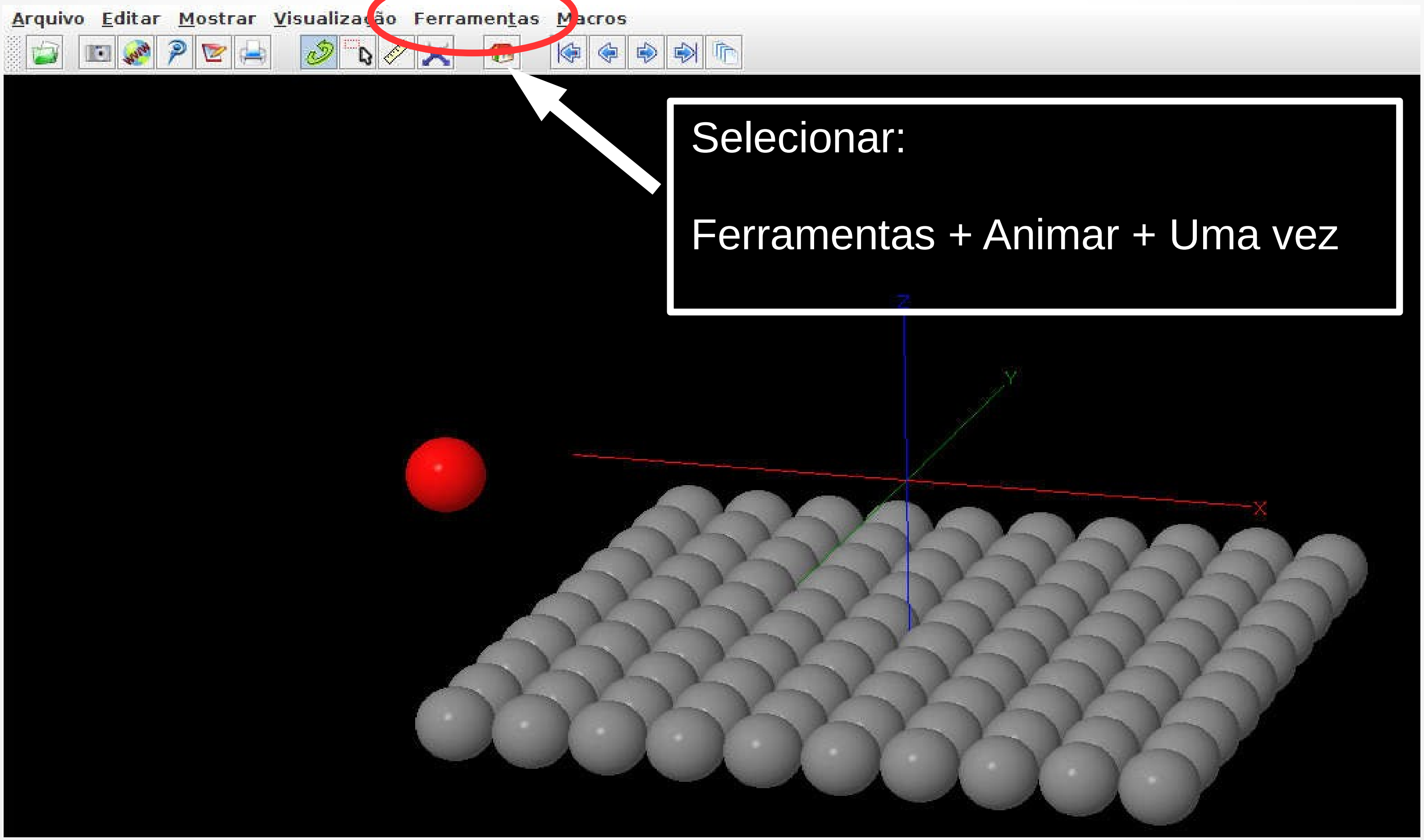
→ $\text{Instante } t$

→ $\text{Instante } t + \Delta t$

O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

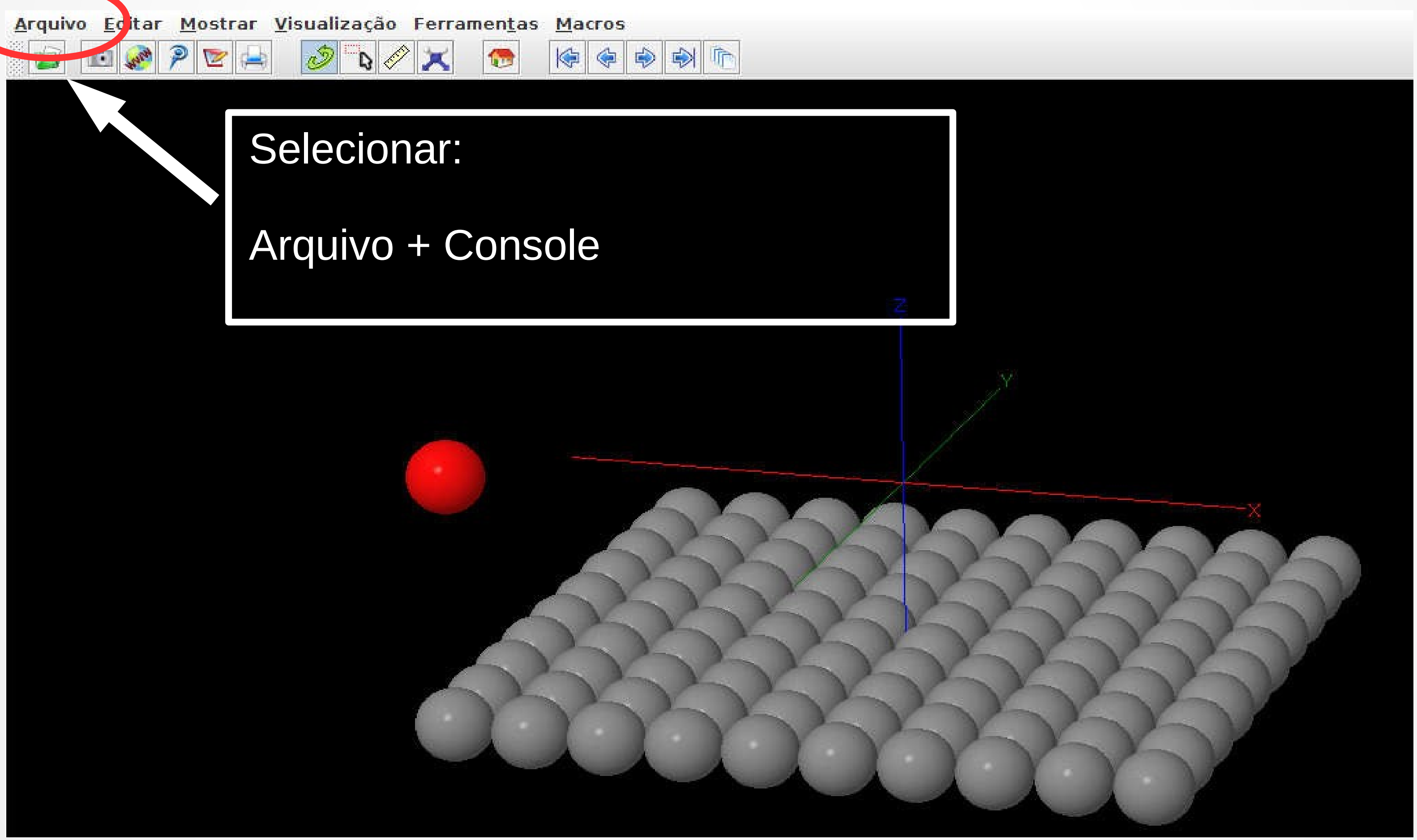
Visualização do movimento da partícula incidente



O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

Filme do movimento da partícula incidente



O algoritmo de velocity-Verlet

Exercício: partícula colidindo com uma parede

Filme do movimento da partícula incidente

- No Console, digite:

write frames 800 600 “filme.jpg”

- Feche o Jmol.

- Os diversos arquivos **.jpg** que são criados com o comando **write** devem ser agora reunidos num único arquivo **.gif**

Criação de um GIF animado ← **convert -coalesce -delay 9 *.jpg filme.gif**