

# As bases da Dinâmica Molecular - 3

Alexandre Diehl

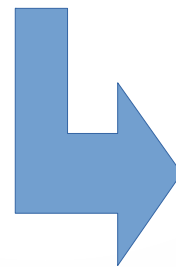
Departamento de Física - UFPel

# Sistemas finitos x sistemas macroscópicos

## Sistemas finitos em MD

### Sistemas finitos

- O número de partículas é pequeno.
- O sistema **não tem fronteiras** (contornos abertos) ou **usa confinamento** na forma de paredes.
- Não se pretende simular o limite termodinâmico.



$$\begin{cases} N \rightarrow \infty \text{ e } V \rightarrow \infty \\ \frac{N}{V} \rightarrow \text{constante} \end{cases}$$

# Sistemas finitos

## Fronteiras abertas (contornos abertos)

- Preparamos o sistema numa **condição inicial** (posições e velocidades).
- **Evoluímos o sistema no tempo**, resolvendo as equações de movimento durante o tempo pretendido.
- A **configuração final** do sistema será determinada pelo potencial de interação entre as partículas do sistema.

**Exemplo:**  $N$  partículas numa caixa cúbica  $L \times L \times L$

Condições iniciais:

1. **Posições aleatórias** ou **posições numa rede**.
2. Evitar superposições ou distâncias muito pequenas entre as partículas (problema de acelerações grandes no tempo  $t = 0$ ).
3. **Velocidades aleatórias** (sem se preocupar com o módulo).
4. **Velocidade do Centro de Massa** (CM) igual à **zero** (se queremos que o sistema como um todo não tenha translação do CM).

# Sistemas finitos

## Fronteiras abertas (contornos abertos)

Como fixar a velocidade do CM igual à zero?

- Defina as velocidades iniciais das partículas  $\vec{v}_i$
- Calcule a velocidade do CM

$$\vec{V}_{CM} = \frac{1}{M} \sum_{i=1}^N m_i \vec{v}_i \longrightarrow M = \sum_{i=1}^N m_i$$

Massa total  
↙

- Como em geral esta velocidade é diferente de zero (as velocidades iniciais são quaisquer), redefina as velocidades  $\vec{v}_i^*$  como

$$\vec{v}_i^* = \vec{v}_i + \vec{a} \longrightarrow \vec{a} \text{ é um vetor constante a ser definido, tal que}$$

$$\vec{V}_{CM}^* = \frac{1}{M} \sum_{i=1}^N m_i \vec{v}_i^* = \vec{0}$$

# Sistemas finitos

## Fronteiras abertas (contornos abertos)

Como fixar a velocidade do CM igual à zero?

$$\vec{v}_i^* = \vec{v}_i + \vec{a}$$

$$\begin{aligned}\vec{V}_{CM}^* &= \frac{1}{M} \sum_{i=1}^N m_i \vec{v}_i^* = \vec{0} \\ &= \frac{1}{M} \sum_{i=1}^N m_i (\vec{v}_i + \vec{a}) = \vec{0} \\ &= \frac{1}{M} \sum_{i=1}^N m_i \vec{v}_i + \frac{1}{M} \sum_{i=1}^N m_i \vec{a} = \vec{0}\end{aligned}$$

$$\equiv \vec{V}_{CM}$$

SCEF

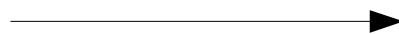
# Sistemas finitos

## Fronteiras abertas (contornos abertos)

Como fixar a velocidade do CM igual à zero?

- como  $\vec{a} = -\vec{V}_{CM}$
- Uma vez definidas as velocidades iniciais de forma aleatória, propomos a seguinte mudança de variável

$$\vec{v}_i^* = \vec{v}_i + \vec{a}$$



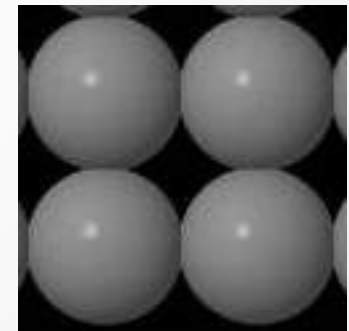
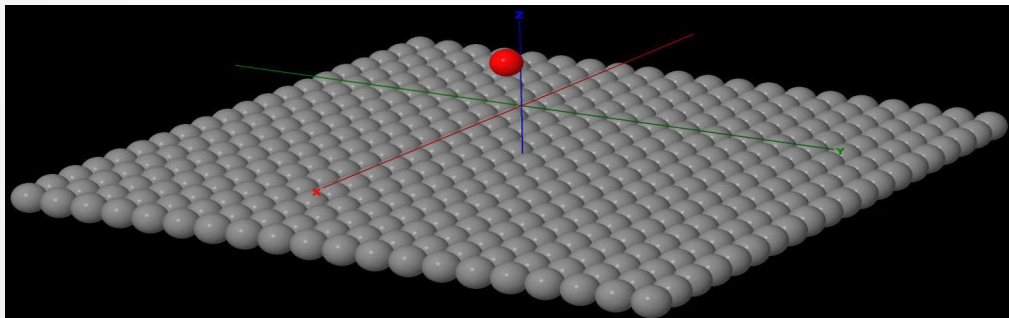
$$\vec{v}_i^* = \vec{v}_i - \vec{V}_{CM}$$

- Esta mudança deve ser feita apenas no tempo  $t = 0$ , antes do ciclo de MD.
- Na ausência de formas externas ao sistema, a velocidade do CM deve se manter próxima de zero (erros numéricos advindos da técnica de integração).

# Sistemas finitos

## Fronteiras fechadas (confinamento por paredes)

- Podemos confinar **usando partículas** para definir as paredes
  - Demanda o **uso de muitas partículas** para este fim, o que pode trazer problemas de tempo de computação.
  - Exige que se defina um **potencial de interação entre as partículas da parede e àquelas confinadas**.
  - Usado nos casos em que as **paredes têm “estrutura”**.



# Sistemas finitos

## Fronteiras fechadas (confinamento por paredes)

- Podemos confinar **usando um potencial de interação** para definir as paredes
  - As **paredes** não são formadas por partículas.
  - Exige que se defina um **potencial de interação** entre as paredes e as partículas confinadas.
    - A **força de interação** é calculada usando a distância entre as partículas confinadas e a posição das paredes.
  - Usado nos casos em que as **paredes não têm “estrutura”**.
  - Uma escolha óbvia é o **potencial de WCA**.



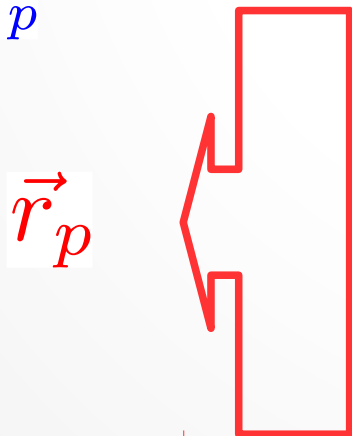
# Sistemas finitos

## Fronteiras fechadas (confinamento por paredes)

- Podemos confinar usando um potencial de interação

$$\vec{f}_{ip}^{WCA} = \begin{cases} \left( \frac{48\varepsilon_i}{\sigma_i^2} \right) \left[ \left( \frac{\sigma_i}{r_{ip}} \right)^{14} - \frac{1}{2} \left( \frac{\sigma_i}{r_{ip}} \right)^8 \right] \vec{r}_{ip}, & r_{ip} < 2^{1/6} \sigma_i \\ 0, & r_{ip} \geq 2^{1/6} \sigma_i \end{cases}$$

$$\vec{r}_{ip} = \vec{r}_i - \vec{r}_p$$



→ É o vetor que localiza as posições das paredes.

→ Para uma caixa cúbica  $L \times L \times L$  temos 6 paredes:

$$\begin{aligned} x_p &= 0 \quad \text{e} \quad x_p = L \\ y_p &= 0 \quad \text{e} \quad y_p = L \\ z_p &= 0 \quad \text{e} \quad z_p = L \end{aligned}$$

# Sistemas finitos

## Fronteiras fechadas (confinamento por paredes)

- Podemos confinar usando um **potencial de interação**

$$\vec{f}_{ip}^{WCA} = \begin{cases} \left( \frac{48\varepsilon_i}{\sigma_i^2} \right) \left[ \left( \frac{\sigma_i}{r_{ip}} \right)^{14} - \frac{1}{2} \left( \frac{\sigma_i}{r_{ip}} \right)^8 \right] \vec{r}_{ip}, & r_{ip} < 2^{1/6} \sigma_i \\ 0, & r_{ip} \geq 2^{1/6} \sigma_i \end{cases}$$

- Assim, cada partícula  $i$  sofrerá a ação das **forças internas**, produzidas pela interação de  $i$  com as outras  $j$  partículas, mais as **forças externas**, produzidas por cada uma das paredes sobre a partícula.

$$\vec{f}_i = \sum_j \vec{f}_{ij} + \sum_p \vec{f}_{ip}^{WCA}$$

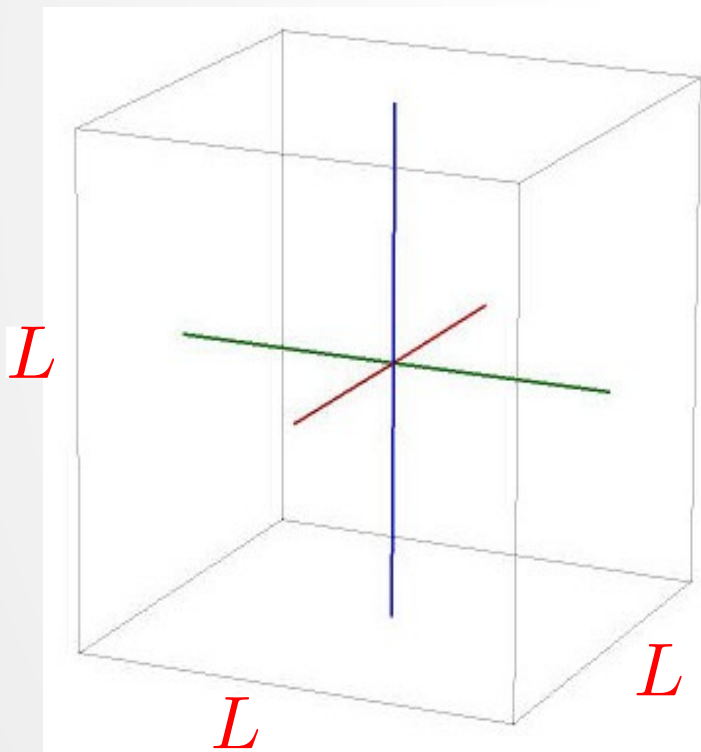
SCEF

Soma sobre as paredes

# Sistemas finitos

## Fronteiras fechadas (confinamento por paredes)

- Podemos confinar usando um potencial de interação



Força exercida pela parede em  $x = L$  sobre  $i$

```
dr = xyz(1,i) - L      ! parede em x = L
if (abs(dr) < rcut) then
  r2i = 1.0d0/dr/dr
  r6i = r2i*r2i*r2i
  factor = 48.0*r6i*(r6i-0.5)*r2i
  xyz(1,i) = xyz(1,i) + factor*dr
endif
```

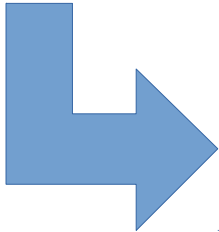
$rcut = 2^{1/6}$

Deve ser incluída uma força deste tipo para cada uma das paredes, sobre cada uma das  $N$  partículas do sistema.

# Sistemas macroscópicos

## Sistemas macroscópicos em MD

- O número de partículas é grande ( $\sim 10^{23}$ ), muito além da capacidade de MD.
- Se pretende simular o limite termodinâmico


$$\left\{ \begin{array}{l} N \rightarrow \infty \text{ e } V \rightarrow \infty \\ \frac{N}{V} \rightarrow \text{constante} \end{array} \right.$$

- **Como simular tal condição num computador?**

# Sistemas macroscópicos

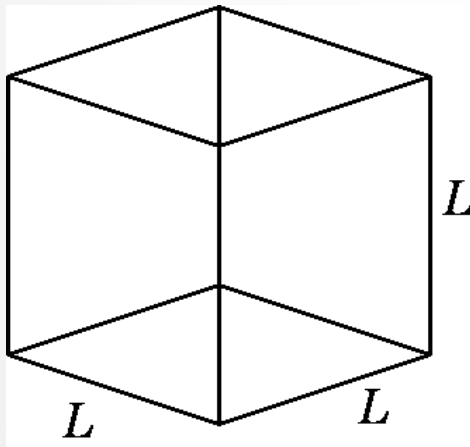
## Sistemas macroscópicos

- Mantemos o número de partículas pequeno  $100 \leq N \leq 10000$ 
  - Buscamos **economia** (ou limitação) **de memória** computacional.
  - Queremos **velocidade de execução** do código de simulação.
- Pretendemos simular o **limite termodinâmico** (condições de **bulk**)
  - O sistema **não tem fronteiras** físicas.
  - As interações entre as partículas não dependem das posições destas dentro da região de simulação.
  - Não são observadas estruturas (**layering**) nas bordas da região de simulação.
  - Queremos usar **regiões de simulação pequenas** (menor  $N$ ).

# Sistemas macroscópicos

## Sistemas macroscópicos

- Qual o **formato** e **quão pequena** deve ser a região de simulação?
  - O **formato** deve ser o **mais simples possível** (**CUBO**, por exemplo), que **não impacte na Física** pretendida pela simulação.
  - **Pequena o suficiente** tal que:

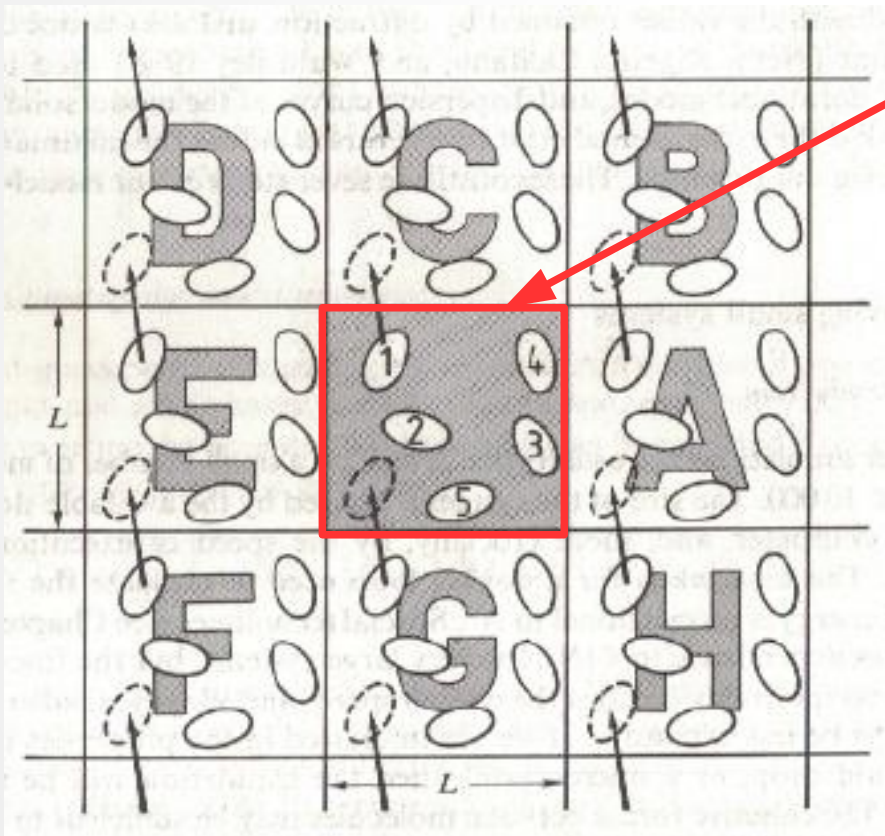


$$\frac{L}{2} >$$

- ★ **Alcance das interações intermoleculares** (OK para potenciais de curto alcance).
- ★ **Alcance** das **correlações físicas** no sistema.
- ★ **Escala característica dominante** do sistema (**comensurabilidade**): formação de camadas, aglomerados, estruturas sólidas, etc).

# Sistemas macroscópicos

## Condições de contorno periódicas (PBC) (Born & von Karman 1912)



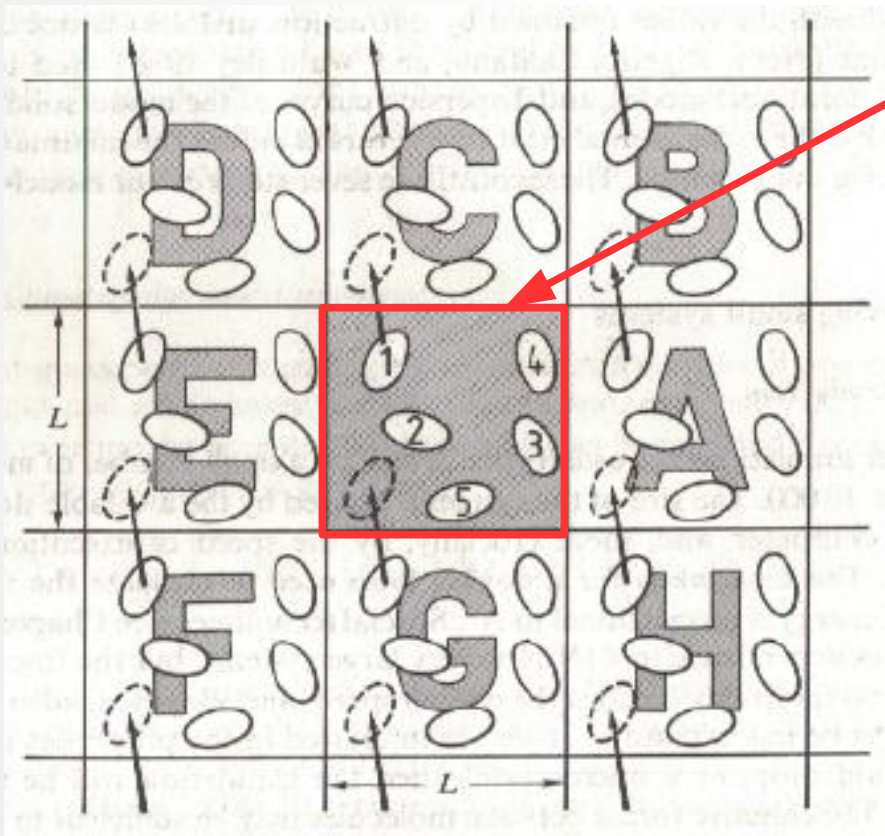
- **Caixa central cúbica de simulação** é replicada em todas as direções, formando uma **rede cúbica infinita**.
- Cada partícula na região de simulação (**1, 2, 3, ...**) tem **infinitas cópias**, localizadas nas **imagens A, B, C, ...**
  - **Partícula 1** tem imagens  **$1_A, 1_B, 1_C, \dots$**
  - O **movimento** da **imagem** é **igual** ao da **partícula** na **região de simulação**.

Allen & Tildesley, pg. 24



# Sistemas macroscópicos

## Condições de contorno periódicas (PBC) (Born & von Karman 1912)



Allen & Tildesley, pg. 24

- **Caixa central cúbica de simulação** é replicada em todas as direções, formando uma **rede cúbica infinita**.
- Cada partícula na região de simulação (**1, 2, 3, ...**) tem **infinitas cópias**, localizadas nas **imagens A, B, C, ...**

→ A **posição de cada imagem** é conhecida:

$$\vec{r}_{\text{im}} = \vec{r} + \vec{n}L$$

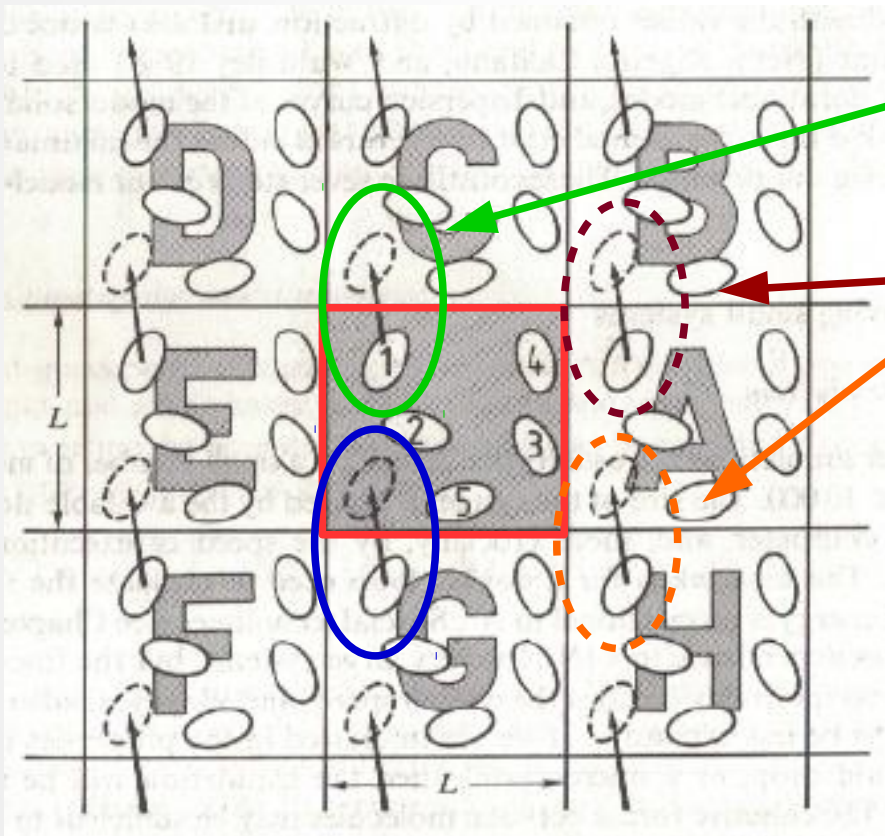
com  $\vec{n} = (n_x, n_y, n_z)$

onde  $n_x, n_y, n_z$  são inteiros  $0, \pm 1, \pm 2, \dots$



# Sistemas macroscópicos

## Condições de contorno periódicas (PBC) (Born & von Karman 1912)

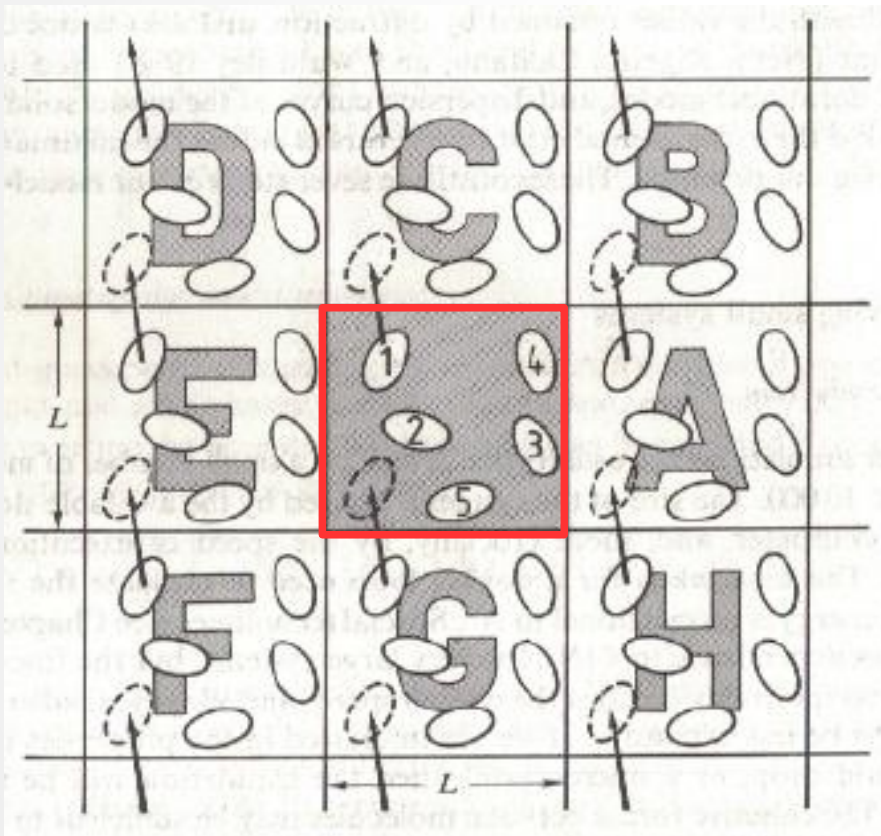


- Se uma **partícula** deixa a **região de simulação**, uma de suas **imagens** entrará **nesta região** pela face oposta.
- O mesmo acontece nas infinitas imagens.
- O **número** (**densidade**) de **partículas** é **conservado**.
- **Não existem paredes** na caixa principal de simulação.
- **Não existem partículas na superfície** do sistema (caixa central+réplicas).

Allen & Tildesley, pg. 24

# Sistemas macroscópicos

## Condições de contorno periódicas (PBC) (Born & von Karman 1912)

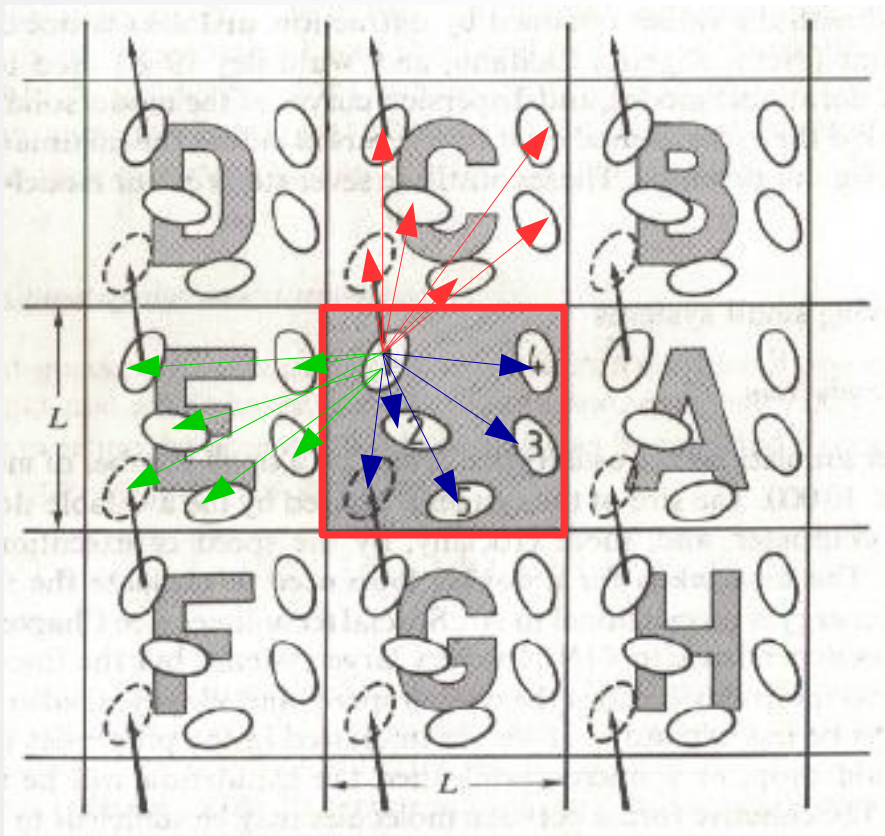


Allen & Tildesley, pg. 24

- Se uma **partícula** **deixa** a **região de simulação**, uma de suas **imagens** **entrará** **nesta região** pela face oposta.
  - O mesmo acontece nas infinitas imagens.
  - O **número** (**densidade**) de **partículas** é **conservado**.
- Sistema contém um **número “ilimitado” de partículas** (caixa+imagens).
  - **Computacionalmente impossível** de **calcular as interações** entre as partículas.

# Sistemas macroscópicos

## Condições de contorno periódicas (PBC) (Born & von Karman 1912)



Allen & Tildesley, pg. 24

- Força sobre uma partícula  $i$  na caixa central de simulação (supondo um potencial de pares aditivo  $u_{ij}$ )

$$\vec{f}_i = -\vec{\nabla} U_i$$

onde

$$U_i = \sum_j \sum_{\vec{n}}' u_{ij}(|\vec{r}_i - \vec{r}_j + \vec{n}L|)$$

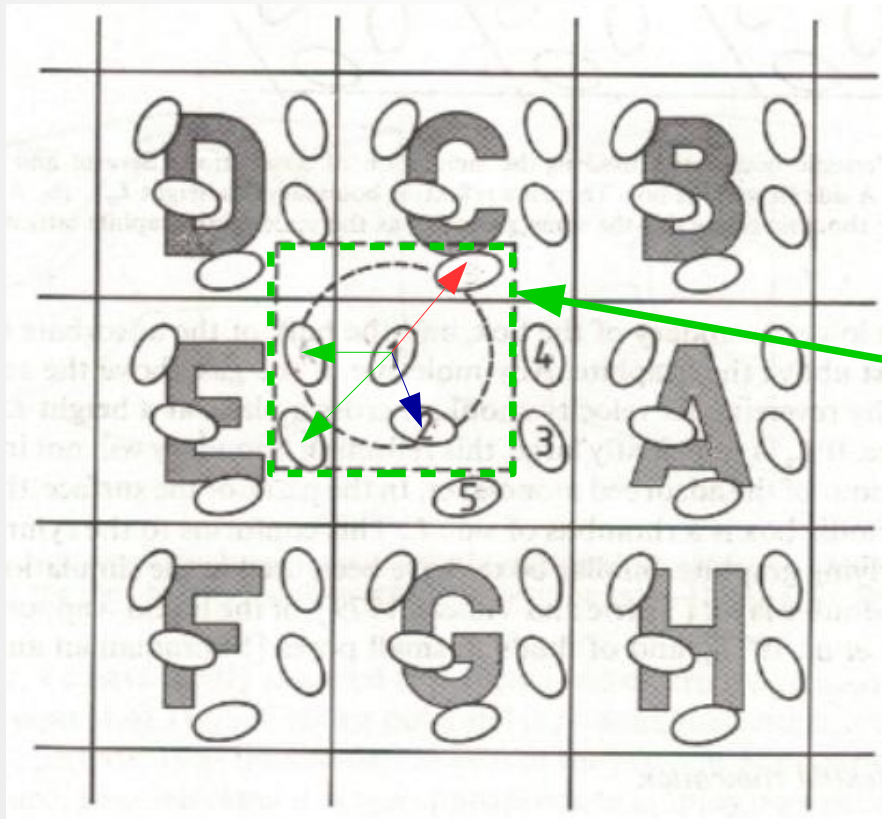
$$\sum_{\vec{n}}' \longrightarrow \underbrace{\vec{n} \neq (0, 0, 0)}$$

Soma apenas sobre as imagens (todas)



# Sistemas macroscópicos

## Convenção de Mínima Imagem (MI) (Metropolis et al 1953)

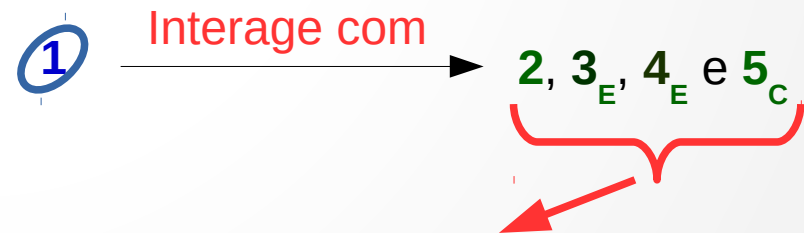


Allen & Tildesley, pg. 28

### Potenciais de curto alcance

- **Força** sobre a **partícula 1** na **caixa central de simulação**:

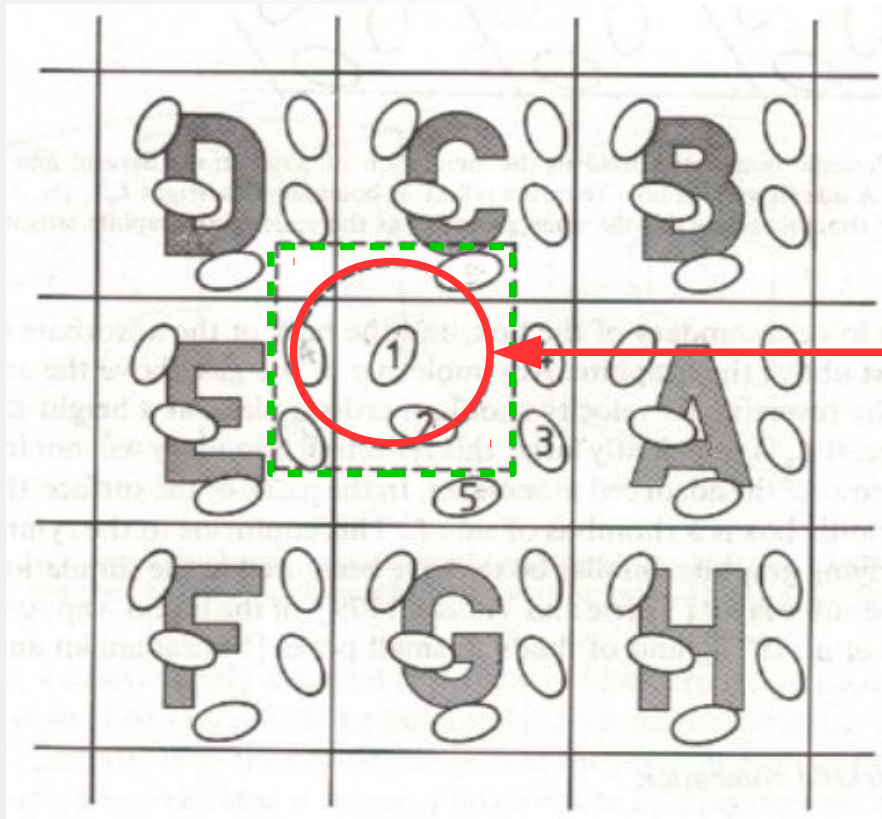
→ Apenas as **partículas localizadas dentro de uma região em torno da partícula** (**idêntica à caixa de simulação**) contribuem para o cálculo da força sobre a partícula:



**Mínima Imagem (MI)**

# Sistemas macroscópicos

## Convenção de Mínima Imagem (MI) (Metropolis et al 1953)



Allen & Tildesley, pg. 28

## Potenciais de curto alcance

- **Força** sobre a **partícula 1** na **caixa central de simulação**:

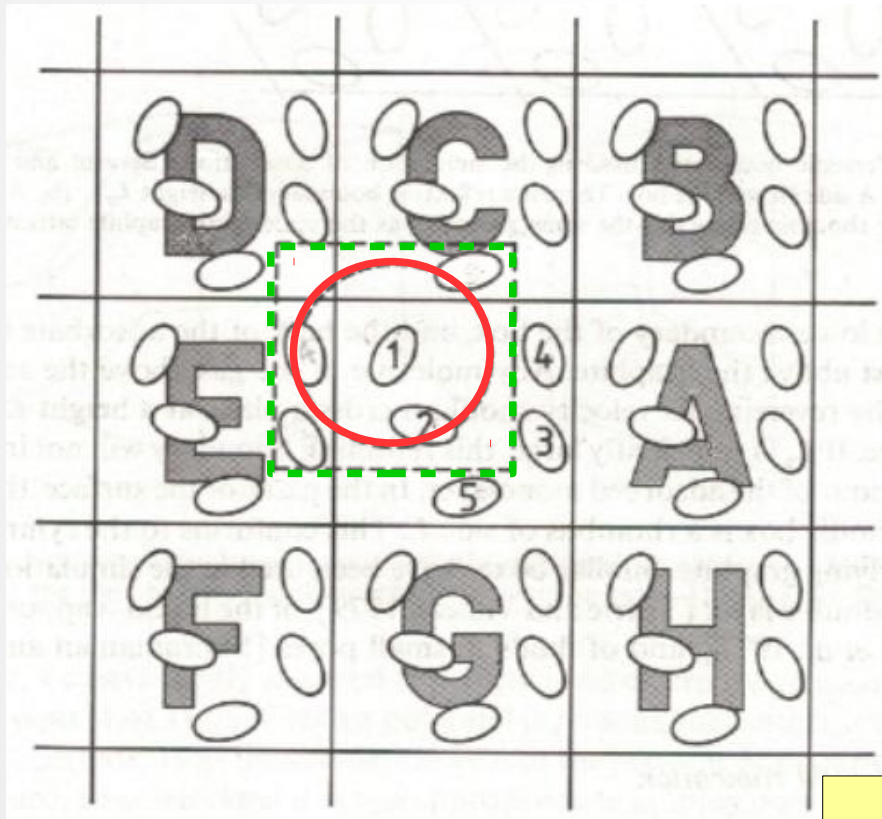
→ **Reduzimos** ainda mais o **número de termos**, usando um **cutoff esférico** em torno da partícula:



**Mínima Imagem (MI)**  
**com cutoff**

# Sistemas macroscópicos

## Convenção de Mínima Imagem (MI) (Metropolis et al 1953)



Allen & Tildesley, pg. 28

### Potenciais de curto alcance

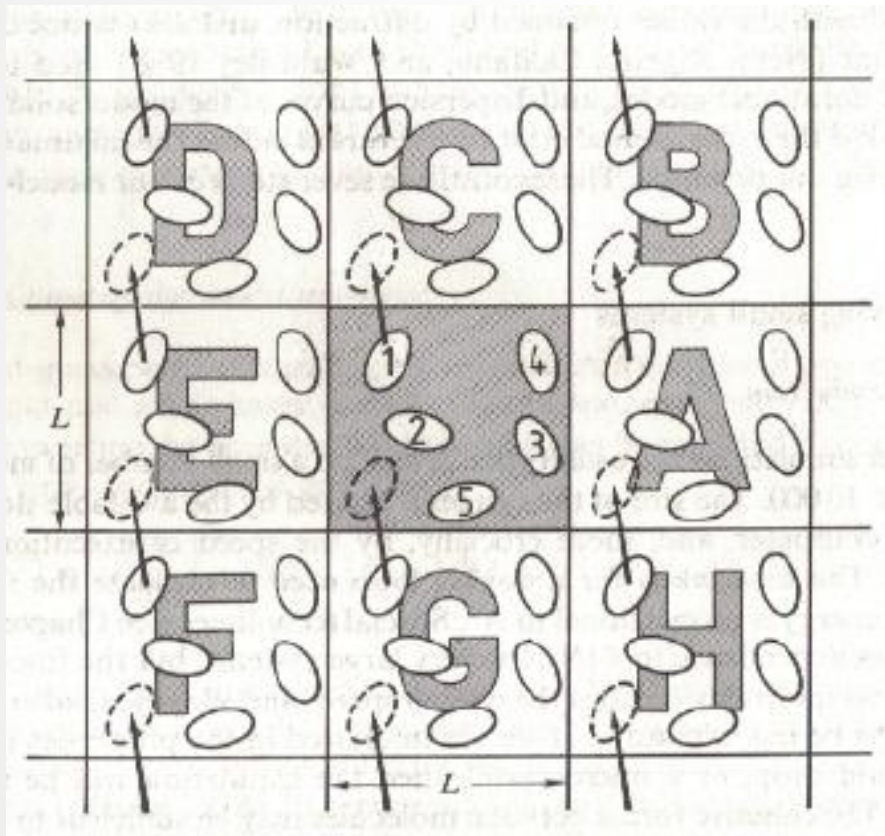
- O valor do **raio do cutoff**,  $r_c$ , é definido pelo **alcance do potencial**:
  - O potencial deve ir para zero em  $r \geq r_c$
  - Para LJ podemos usar  $r_c = 2.5\sigma$
- Para **consistência com MI** devemos usar um **raio do cutoff** limitado por

$$r_c < \frac{L}{2}$$

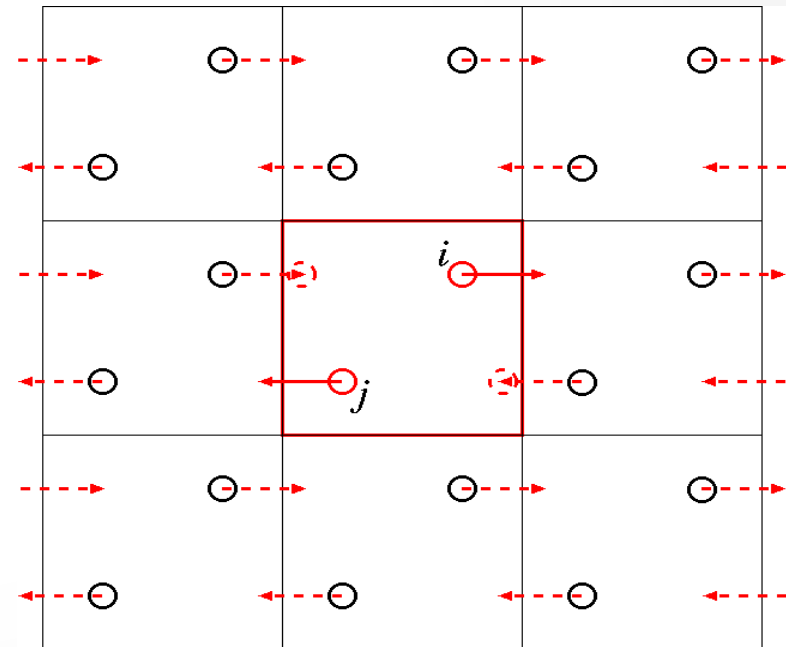
**Apenas** as **partículas dentro da caixa central** de **simulação** devem ser **considerada na evolução do sistema**.

# Sistemas macroscópicos

## Implementando PBC e MI em MD

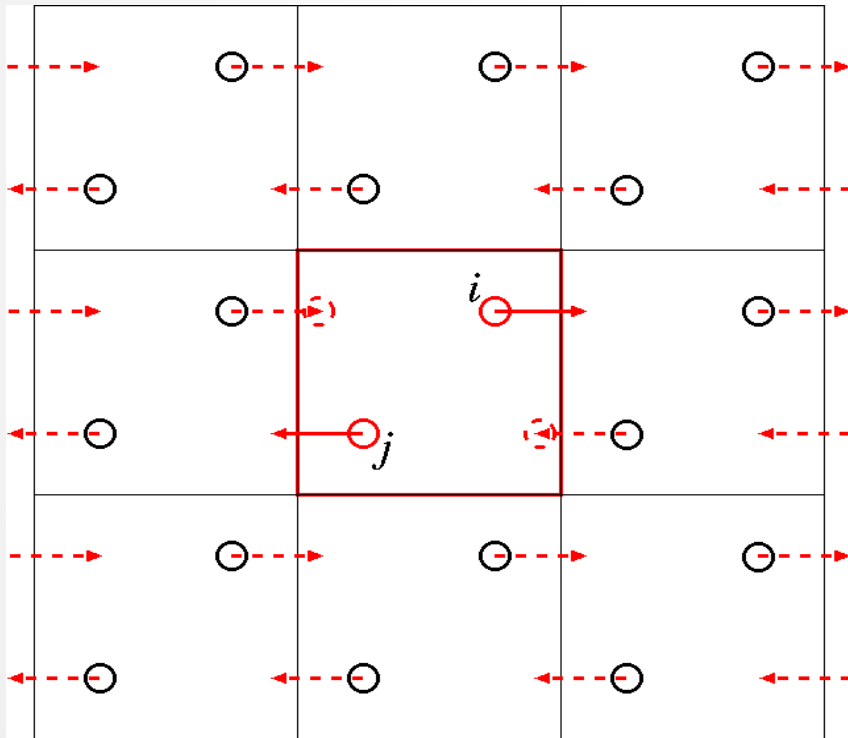


**PBC:** Partículas  $i$  e  $j$  “saindo” da caixa central de simulação, após um **ciclo de MD**

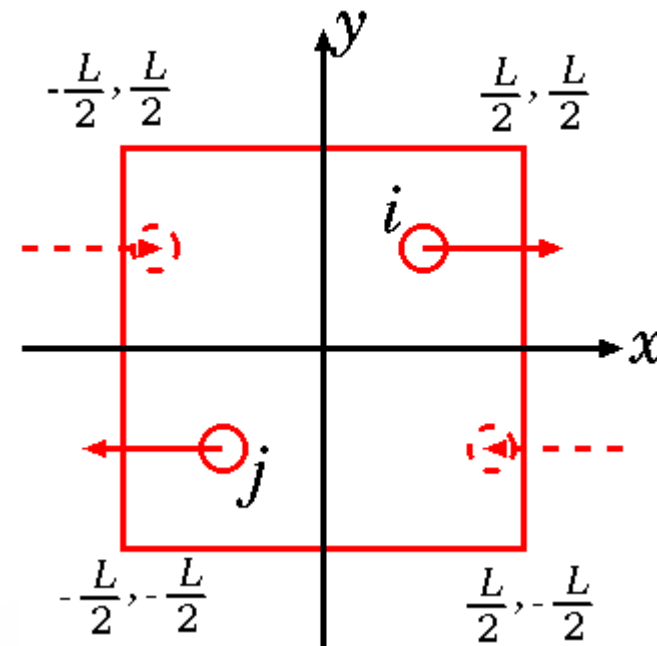


# Sistemas macroscópicos

## Implementando PBC e MI em MD



**PBC:** Partículas *i* e *j* “saindo” da caixa central de simulação, após um **ciclo de MD**

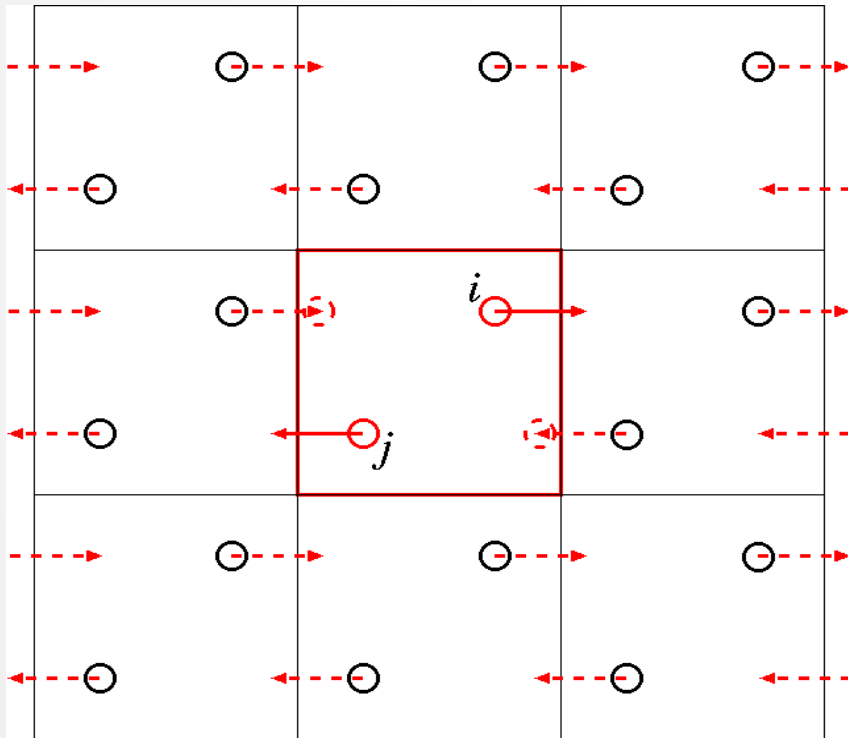


$$\text{if } \left( x(i) > \frac{L}{2} \right) x(i) = x(i) - L$$

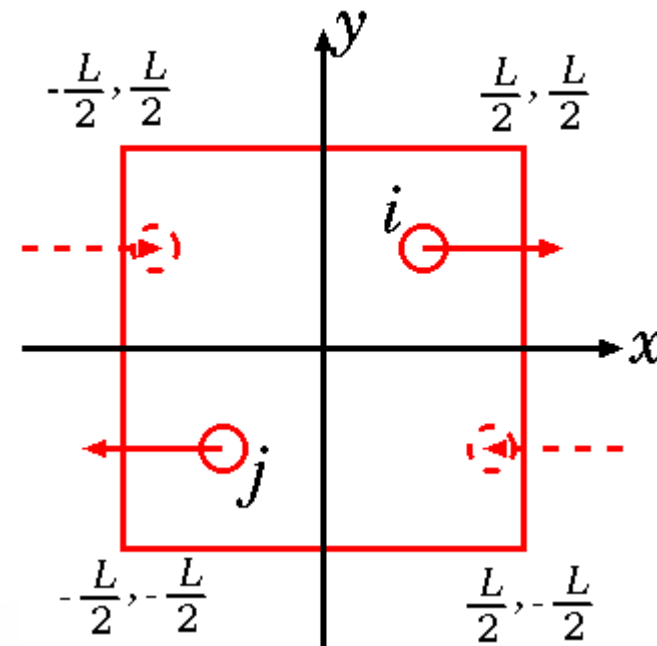


# Sistemas macroscópicos

## Implementando PBC e MI em MD



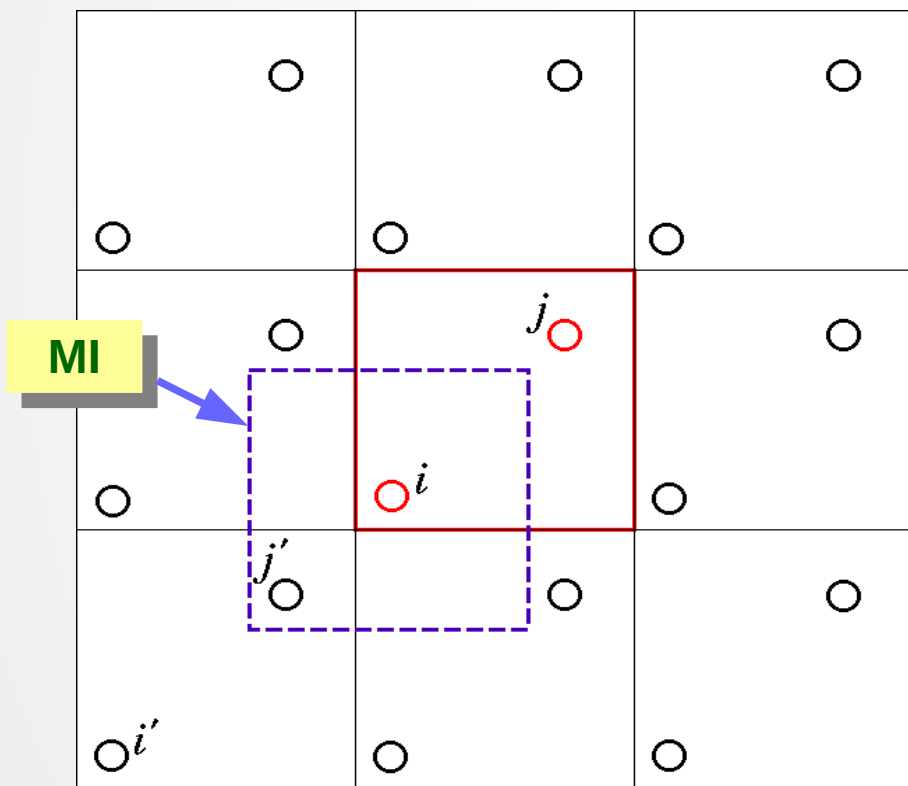
**PBC:** Partículas *i* e *j* “saindo” da caixa central de simulação, após um **ciclo de MD**



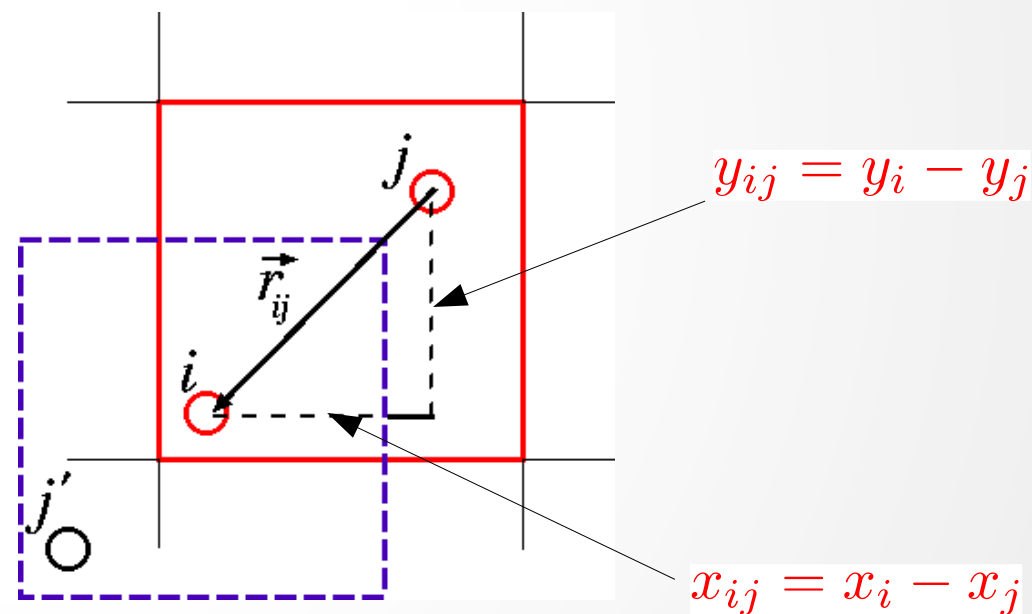
$$\text{if } \left( x(j) < -\frac{L}{2} \right) x(j) = x(j) + L$$

# Sistemas macroscópicos

## Implementando PBC e MI em MD



**MI:** Partículas  $i$  e  $j$  na caixa central de simulação, e suas imagens  $i'$  e  $j'$



$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$

# Sistemas macroscópicos

## Implementando PBC e MI em MD

### Força sobre $i$

Calculamos a distância entre  $i$  e  $j$

☀ se  $x_{ij} < \frac{L}{2}$

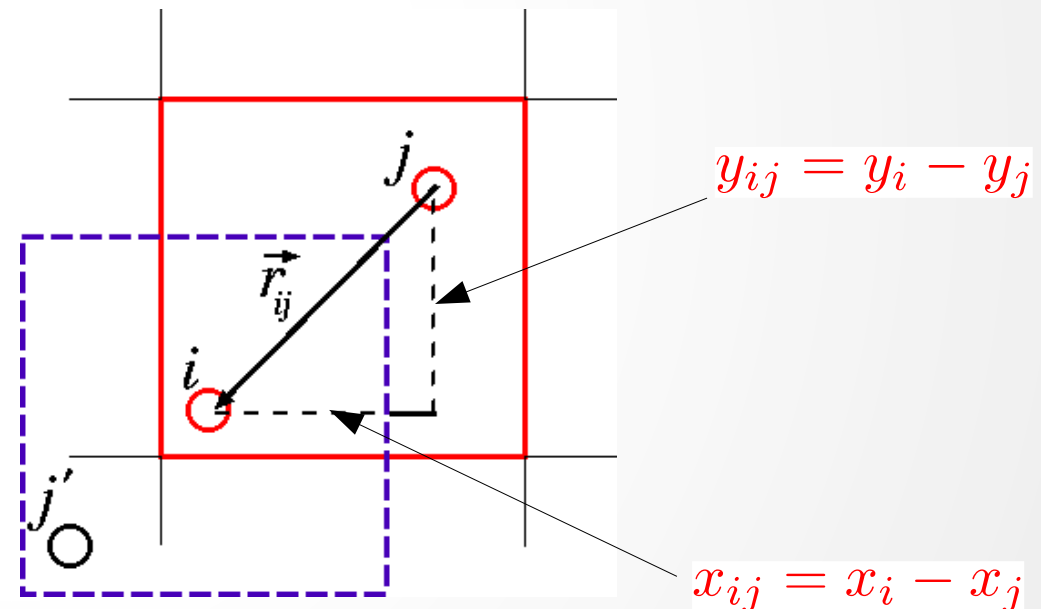
➡  $i$  interage com  $j$

☀ se  $x_{ij} > \frac{L}{2}$

➡  $i$  interage com  $j'$

Fazemos o mesmo para as componentes  $y$  e  $z$ .

**MI:** Partículas  $i$  e  $j$  na caixa central de simulação, e suas imagens  $i'$  e  $j'$



$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$$

# Sistemas macroscópicos

## Implementando PBC e MI em MD (em FORTRAN 90)

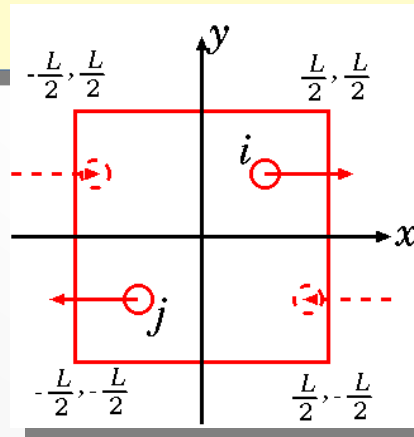
### PBC

Após as **partículas** terem sido **movidas** (por exemplo, usando **velocity-Verlet**)

```
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
    if (xyz(k,i) < -L/2.0) xyz(k,i) = xyz(k,i) + L ! PBC
    if (xyz(k,i) > L/2.0) xyz(k,i) = xyz(k,i) - L ! PBC
  end do
end do
```

Dimensão do espaço

Origem no  
centro da caixa  
de simulação



Considere usar **ANINT**

```
xyz(k,i) = xyz(k,i) - L*ANINT(xyz(k,i)/L)
```

# Sistemas macroscópicos

## Implementando PBC e MI em MD (em FORTRAN 90)

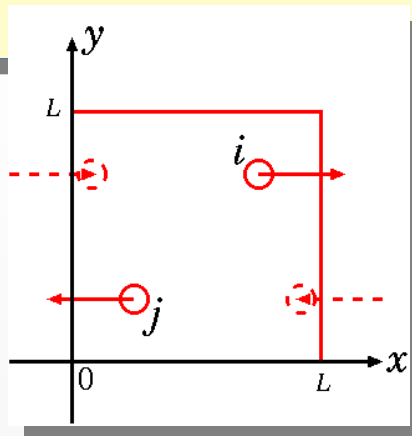
### PBC

Após as **partículas** terem sido **movidas** (por exemplo, usando **velocity-Verlet**)

```
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
    if (xyz(k,i) < 0.0) xyz(k,i) = xyz(k,i) + L ! PBC
    if (xyz(k,i) > L) xyz(k,i) = xyz(k,i) - L ! PBC
  end do
end do
```

Dimensão do espaço

Origem no  
vértice da caixa  
de simulação



# Sistemas macroscópicos

## Implementando PBC e MI em MD (em FORTRAN 90)

### MI

Dentro da **rotina de forças** ou da **energia potencial total**

```
do i = 1, Npart-1
  do j = i+1, Npart
    r2 = 0.0
    do k = 1, dim
      dr(k) = xyz(k, i) - xyz(k, j)
      dr(k) = dr(k) - L*ANINT(dr(k)/L) ! MI
      r2 = r2 + dr(k)*dr(k)
    enddo
    (...)
  end do
end do
```

Função ANINT(a)

$r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$

# Sistemas macroscópicos

## Implementando PBC e MI em MD (em FORTRAN 90)

### MI

Dentro da **rotina de forças** ou da **energia potencial total**

```
do i = 1,Npart-1
  do j = i+1, Npart
    r2 = 0.0
    do k = 1,dim
      dr(k) = xyz(k,i) - xyz(k,j)
      dr(k) = MIN(abs(dr(k)),L-abs(dr(k))) ! MI
      r2 = r2 + dr(k)*dr(k)
    enddo
    (...)
  end do
end do
```

Função MIN(a,b)

$r_{ij}^2 = (x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2$

# Sistemas macroscópicos

## Implementando PBC e MI em MD (em FORTRAN 90)

### MI com *cutoff* $r_c$

Dentro da **rotina de forças** ou da **energia potencial total**

```
do i = 1,Npart-1
  do j = i+1, Npart
    r2 = 0.0
    do k = 1,dim
      dr(k) = xyz(k,i) - xyz(k,j)
      dr(k) = dr(k) - L*ANINT(dr(k)/L)    ! MI
      r2 = r2 + dr(k)*dr(k)
    enddo
    if (r2 < rcut2) then
      (...)
    end if
  end do
end do
```

$rcut2 = r_c^2$

Cálculo da força (e da energia potencial)