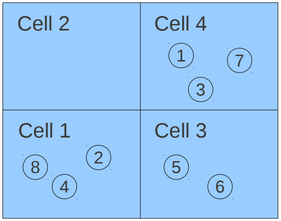
无网格计算方法的网格搜索方法

周吕文

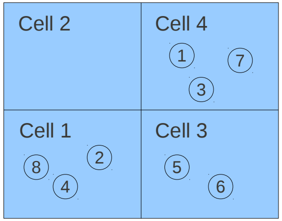
2010 年 11 月 5 日

Linked-list cells 方法

Linked-list cells 方法

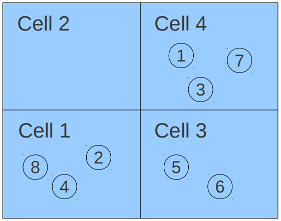


Linked-list cells 方法



	1	2	3	4				
Head	8	0	6	7				
	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

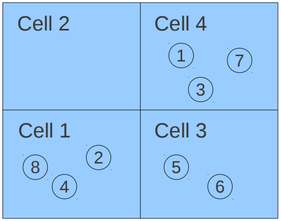
Linked-list cells 方法



	1	2	3	4
Head	8	0	6	7

	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

Linked-list cells 方法



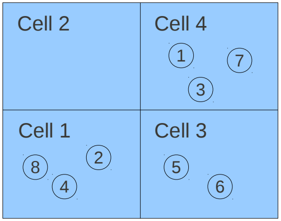
	1	2	3	4
Head	8	0	6	7

	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

通过 Head 和 Link 来访问某个格子中所有的粒子，如访问第 4 个格子中的粒子:

- Head(4) $=$ 7 找到第 4 个格子中最大号粒子即 7 号粒子

Linked-list cells 方法



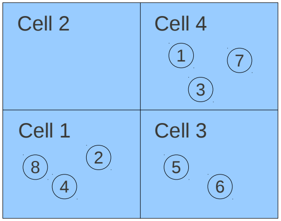
	1	2	3	4
Head	8	0	6	7

	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

通过 Head 和 Link 来访问某个格子中所有的粒子，如访问第 4 个格子中的粒子:

- Head(4) = 7 找到第 4 个格子中最大号粒子即 7 号粒子
- Link (7) = 3 找到第 4 个格子中次大号粒子即 3 号粒子

Linked-list cells 方法



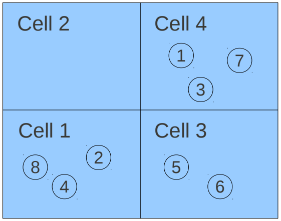
	1	2	3	4
Head	8	0	6	7

	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

通过 Head 和 Link 来访问某个格子中所有的粒子，如访问第 4 个格子中的粒子:

- Head(4) = 7 找到第 4 个格子中最大号粒子即 7 号粒子
- Link (7) = 3 找到第 4 个格子中次大号粒子即 3 号粒子
- Link (3) = 1 找到第 4 个格子中次大号粒子即 1 号粒子

Linked-list cells 方法



	1	2	3	4
Head	8	0	6	7

	1	2	3	4	5	6	7	8
Link	0	0	1	2	0	5	3	4

通过 Head 和 Link 来访问某个格子中所有的粒子，如访问第 4 个格子中的粒子:

- Head(4) $=$ 7 找到第 4 个格子中最大号粒子即 7 号粒子
- Link (7) $=$ 3 找到第 4 个格子中次大号粒子即 3 号粒子
- Link (3) $=$ 1 找到第 4 个格子中次大号粒子即 1 号粒子
- Link (1) $=$ 0 0 表示第 4 个格子中没有其它可搜索粒子

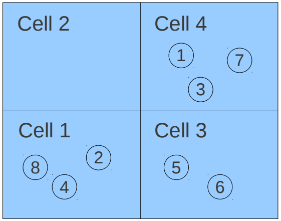
生成 Link 和 Head 数组的伪代码

```
Algorithm 1: Generate Head and Link arrays
Initialize Link and Head
Do for all particles i form 1 to n
    Icell = The number of cell which contain particle i

    ! Link to the previous occupant
    Link(i) = Head(Icell)

    ! The last one goes to the header
    Head(Icell) = i
End do
```

访问某一格子中粒子的伪代码

```
Algorithm 2: Access All Particle in One Cell
i = Head(Number of one cell)
Do while I not equal 0
    Access the i-th particle
    i = Link(i)
End do
```

Data locality 方法



	1	2	3	4	5	6	7	8
OldList	1	2	3	4	5	6	7	8
	1	2	3	4	5	6	7	8
NewList	2	4	8	5	6	1	3	7
	1	2	3	4	5			
CellEnd	0	3	3	5	8			

用 CellEnd 数组存放每个格子中所用粒子在 NewList 中的位置, 第 i 个格子中的所用粒子表示为

$$\mathit{CellEnd}(i)+1\cdots\mathit{CellEnd}(i+1)$$

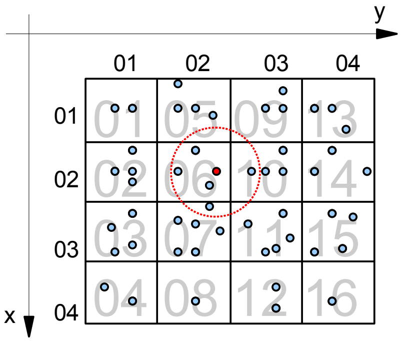
生成 NewList 和 CellEnd 数组的伪代码

```
Algorithm 3: Generate NewList and CellEnd arrays
Initialize NewList, CellEnd and Temp

! Generate CellEnd array
Do for all particles i form 1 to n
    Icell = The index of cell which contain particle i
    CellEnd(Icell+1:end) = CellEnd(Icell+1:end) +1
End do

! Generate NewList array
Temp = CellEnd
Do for all particles i from n to 1
    Icell = The index of cell which contain particle i
    NewList( Temp(Icell+1) ) = I
    Temp(Icell+1) = Temp(Icell+1) - 1
End do
```

访问某一格子中粒子的伪代码

```
Algorithm 4: Access All Particle in One Cell
Do for i form CellEnd(c)+1 to CellEnd(c+1)
        access particle NewList(i)
End do
```

格子的编号



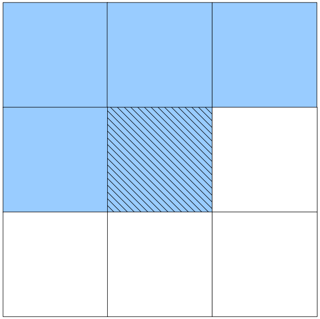
$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

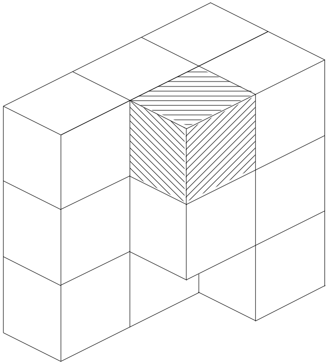
$$boxl = [L_x, L_y], Nc = [Nc_x, Nc_y]$$

$$CellSub_i = \left[\lfloor \frac{x_i}{Nc_x} \cdot L_x \rfloor, \lfloor \frac{y_i}{Nc_y} \cdot L_y \rfloor \right] + 1$$

$$CellInd_i = \left[CellSub_i - [0, 1] \right] \begin{bmatrix} 1 \\ Nc_x \end{bmatrix}$$

搜索网格





搜索网格

```
Algorithm 5: half neighbors search base on Linked-list Cells (3D)
Do for all cells i
    CellInd_i = i
    atom_i = Head(CellInd_i)
    Do while atom_i not equal 0
        Do for all 14 cells j in the neighborhood of cell i
            CellInd_j = Index of cells j
            If CellInd_i equal CellInd_j then
                atom_j = Link(atom_i)
            Else
                atom_j = head(CellInd_j)
            End if
            Do while atom_j not equal 0
                !****************************************
                compute the force between atom_i and atom_j in here
                !****************************************
                atom_j = link(atom_j)
            End do
        End do
        atom_i = Link(atom_i)
    End do
End do
```

THE END