

# 并行计算

2022年春季

## 第十章 线性方程组的求解

10.1 三角形方程组的求解

10.2 三对角方程组的求解

10.3 稠密线性方程组的求解

10.4 稀疏线性方程组的求解

## 10.1 三角形方程组的求解

### 10.1.1 基本术语

### 10.1.2 上三角方程组的求解

# 基本术语

- 线性方程组的定义和符号

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n = b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n = b_2 \\ \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \dots + a_{n,n}x_n = b_n \end{cases}$$

记为

$$AX=b$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix},$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix},$$

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

## 10.1 三角形方程组的求解

### 10.1.1 基本术语

### 10.1.2 上三角方程组的求解

# 上三角方程组的求解

- 上三角方程组的回代解法并行化

(1) SISD 上的回代算法

Begin

(1) for  $i=n$  downto 1 do

(1.1)  $x_i = b_i / a_{ii}$

(1.2) for  $j=1$  to  $i-1$  do

$b_j = b_j - a_{ji} x_i$

$a_{ji} = 0$

endfor

endfor

End

} 可并行化

# 上三角方程组的求解

- 上三角方程组的回代解法并行化

## (2) SIMD-CREW上的并行回代算法

- 划分:  $p$ 个处理器行循环带状划分

- 算法

Begin

for  $i=n$  downto 1 do

$x_i = b_i / a_{ii}$

for all  $P_j$ , where  $1 \leq j \leq p$  do

for  $k=j$  to  $i-1$  step  $p$  do

$b_k = b_k - a_{ki} x_i$

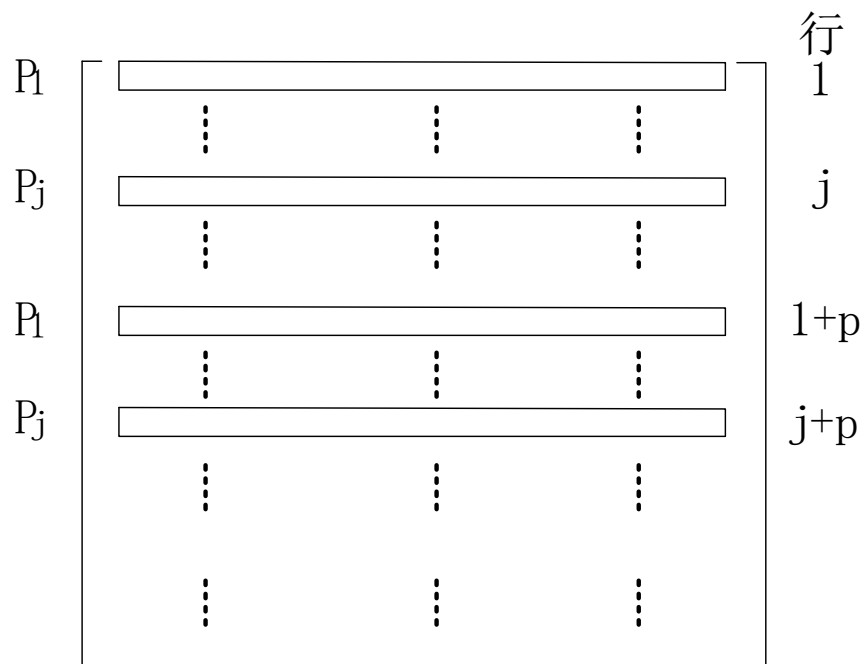
$a_{ki} = 0$

endfor

endfor

endfor

End //  $p(n)=n, t(n)=n$



## 第十章 线性方程组的求解

10.1 三角形方程组的求解

10.2 三对角方程组的求解

10.3 稠密线性方程组的求解

10.4 稀疏线性方程组的求解



## 10.2 三对角方程组的求解

### 10.2.1 直接求解法

### 10.2.2 奇偶规约法

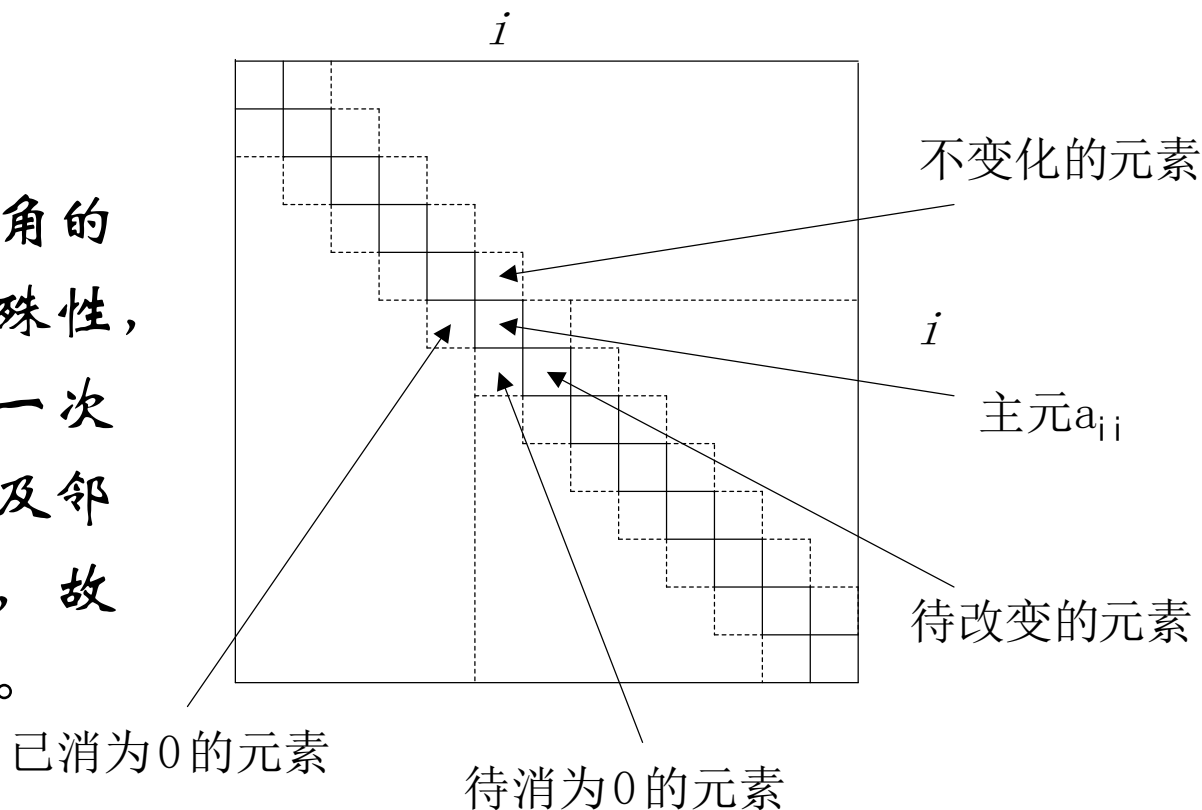
# 三对角方程组的直接求解法

- Gauss消去法(难以并行化)

①消元

②回代

注：由于三对角的方程组的特殊性，一次消元或一次回代，只涉及邻近一个方程，故难以并行化。



## 10.2 三对角方程组的求解

### 10.2.1 直接求解法

### 10.2.2 奇偶规约法

# 三对角方程组的直接求解法

- 奇偶规约求解法(可并行化)

三对角方程可以写成如下形式

$$f_i x_{i-1} + g_i x_i + h_i x_{i+1} = b_i \quad i=1 \sim n$$

$$f_1 = h_n = 0$$

- 串行算法描述

①利用上下相邻方程消去偶序号方程中的奇下标变量:

$$f_{2i-1} x_{2i-2} + g_{2i-1} x_{2i-1} + h_{2i-1} x_{2i} = b_{2i-1}$$

$$f_{2i} x_{2i-1} + g_{2i} x_{2i} + h_{2i} x_{2i+1} = b_{2i}$$

$$f_{2i+1} x_{2i} + g_{2i+1} x_{2i+1} + h_{2i+1} x_{2i+2} = b_{2i+1}$$

2i-1方程乘上某个数消去2i方程中的 $f_{2i} x_{2i-1}$ 项, 2i+1方程乘上某个数消去2i方程中的 $h_{2i} x_{2i+1}$ 项, 使2i方程变为

$$\alpha_i x_{2i-2} + \beta_i x_{2i} + \gamma_i x_{2i+2} = \eta_i \quad i=1, 2, \dots, n/2$$

# 三对角方程组的直接求解法

②重复①最终可得:

case 1:

$$g_1x_1 + h_1x_2 = b_1$$

$$f_2x_1 + g_2x_2 + h_2x_3 = b_2$$

$$f_3x_2 + g_3x_3 + h_3x_4 = b_3$$

$$f_4x_3 + g_4x_4 = b_4$$

可以分别得到

$$g_1x_1 + h_1x_2 = b_1$$

$$f_2x_1 + g_2x_2 = b_2$$

或

$$g_1x_1 + h_1x_2 = b_1$$

$$f_2x_1 + g_2x_2 + h_2x_3 = b_2$$

$$f_3x_2 + g_3x_3 = b_3$$

解得  $x_1, x_2$  或  $x_1, x_2, x_3$

③回代求解 $x$

• 并行化分析: ①、②消去奇下标可以并行化;

③回代求解可以并行化

## 第十章 线性方程组的求解

10.1 三角形方程组的求解

10.2 三对角方程组的求解

10.3 稠密线性方程组的求解

10.4 稀疏线性方程组的求解

## 10.3 稠密线性方程组的求解

### 10.3.1 有回代的高斯消去法

### 10.3.2 无回代的高斯-约旦法

### 10.3.3 迭代求解的高斯-赛德尔 法

# 有回代的高斯消去法

- 算法基本原理

- 求解过程分为消元和回代两个阶段，消元是将系数矩阵  $A$  化为上三角阵  $T$ ，然后对  $TX=c$  进行回代求解。
- 消元过程中可以应用选主元方法，增加算法的数值稳定性。

- 下面是消元过程图：

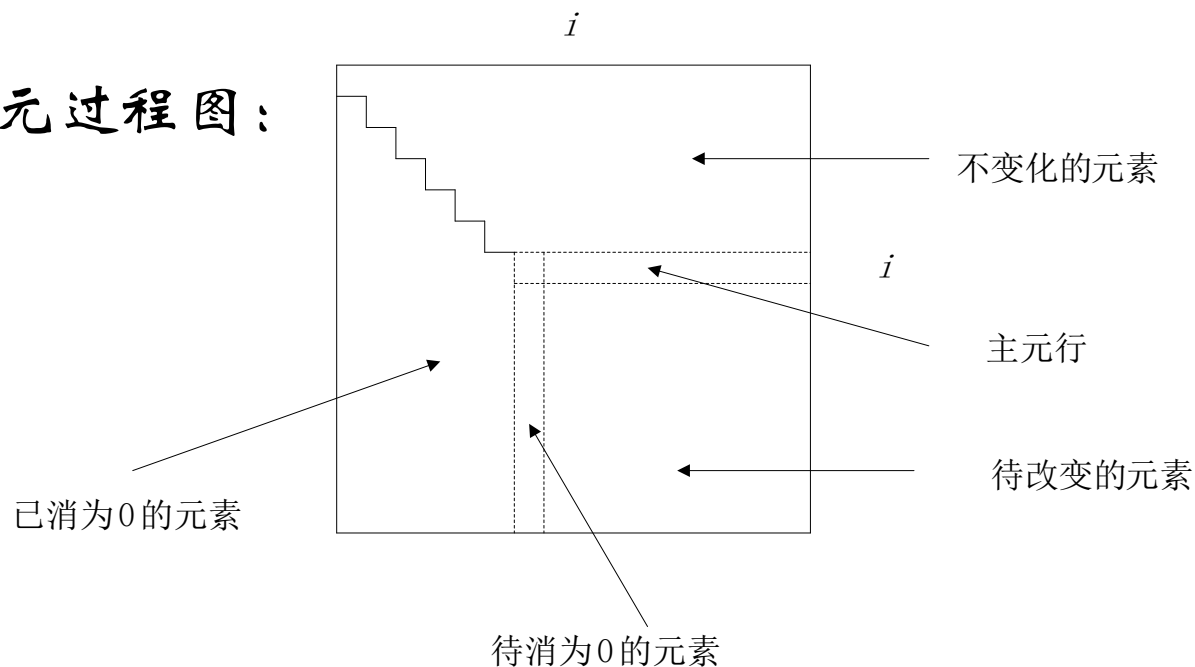


图10.1



# 有回代的高斯消去法

- 并行化分析

- 消元和回代均可以并行化;
- 选主元也可以并行化;
- 消元过程的并行化图示：处理器按行划分

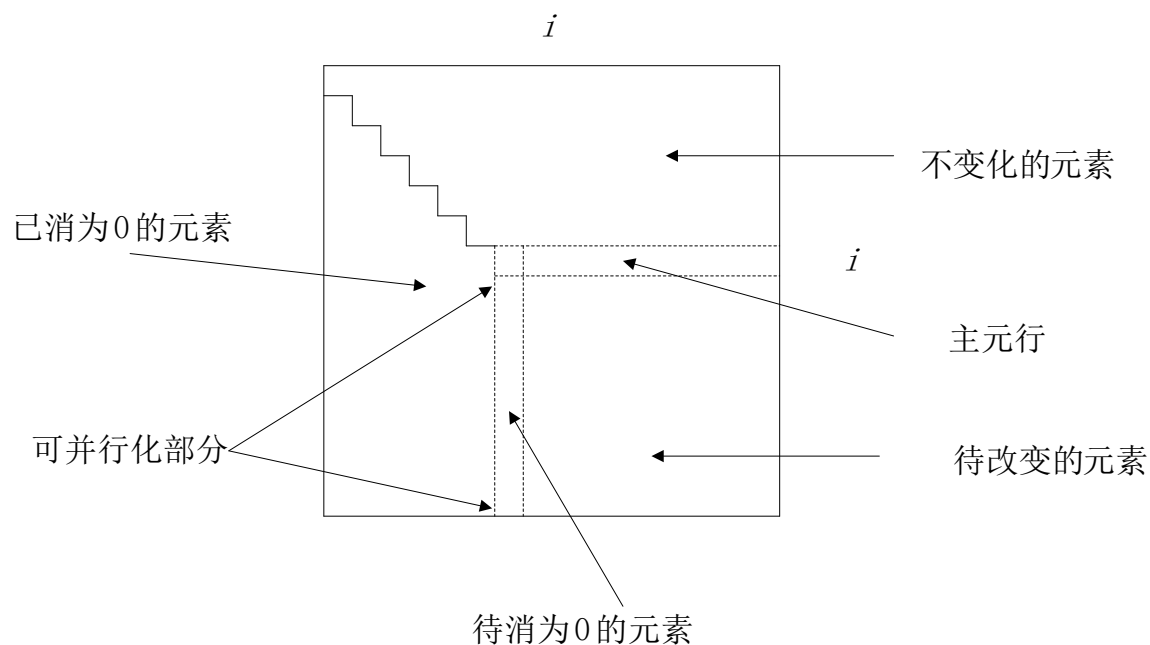


图10.1

## 10.3 稠密线性方程组的求解

10.3.1 有回代的高斯消去法

10.3.2 无回代的高斯-约旦法

10.3.3 迭代求解的高斯-赛德尔  
法

# 无回代的高斯-约旦法

- 串行算法原理

①消元: 通过初等行变换, 将  $(A, b)$  化为主对角线矩阵, 记  $b$  为  $A$  的第  $n+1$  列

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \cdots & a_{2n} & a_{2,n+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & a_{n,n+1} \end{bmatrix} \Rightarrow \begin{bmatrix} a'_{11} & & & & a'_{1,n+1} \\ & a'_{22} & & & a'_{2,n+1} \\ & & \ddots & & \vdots \\ & & & a'_{nn} & a'_{n,n+1} \end{bmatrix}$$

②求解:  $x_j = a'_{j,n+1} / a'_{jj}$

# 无回代的高斯-约旦法

- SIMD-CREW上的并行算法

(1) 处理器:  $n^2+n$ 个处理器, 这些处理器排成 $n \times (n+1)$ 的矩阵,  
处理器编号为 $P_{ik}$ ,  $i=1 \sim n$ ,  $k=1 \sim n+1$

(2) 并行化分析

① 消元的并行化:     //  $O(n)$

for  $j=1$  to  $n-1$ , each  $P_{ik}$  Par-do   // 第 $j$ 次消元

$P_{ij}(i \neq j): a_{ij} \leftarrow 0$

$P_{ik}(i \neq j, k=j+1 \sim n+1): a_{ik} \leftarrow a_{ik} - a_{jk}(a_{ij}/a_{jj})$

end for

② 求解: for each  $P_{jj}(j=1 \sim n)$  Par-do:  $x_j \leftarrow a_{j,n+1}/a_{jj}$  //  $O(1)$

(3) 时间分析:  $t(n)=O(n)$ ,  $p(n)=O(n^2)$ ,  $c(n)=O(n^3)$  成本最优?

# 无回代的高斯-约旦法

- 成本最优?

串行算法的最优时间：由于  $x = A^{-1}b$

①  $A^{-1}b$  (假设已有  $A^{-1}$ ):  $O(n^2)$

$$\textcircled{2} \text{求 } A^{-1}: A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix} \begin{bmatrix} A_{11} & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}$$
$$\Rightarrow A^{-1} = \begin{bmatrix} I & -A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix} \begin{bmatrix} A_{11}^{-1} & 0 \\ 0 & B^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1} & I \end{bmatrix} \quad \text{其中, } B = A_{22} - A_{21}A_{11}^{-1}A_{12}$$

$\therefore$  求  $A^{-1}$  需要: 2次  $n/2 \times n/2$  矩阵的逆  $i(n/2)$

6次  $n/2 \times n/2$  矩阵的乘  $m(n/2)$

2次  $n/2 \times n/2$  矩阵的加  $a(n/2)$

$$i(n) = i(n/2) + 6m(n/2) + 2a(n/2)$$

$$a(n/2) = n^2/2, \quad m(n/2) = O((n/2)^x) \quad 2 < x < 2.5$$

$$\Rightarrow i(n) = O(n^x) \quad \text{综上, 串行算法的最优时间为 } O(n^x) \quad 2 < x < 2.5$$

## 10.3 稠密线性方程组的求解

10.3.1 有回代的高斯消去法

10.3.2 无回代的高斯-约旦法

10.3.3 迭代求解的高斯-赛德尔

法

# 迭代求解的高斯-赛德尔法

- 串行算法原理

$$x_i^{k+1} = -\frac{1}{a_{ii}} \left[ \sum_{j<i} a_{ij} x_j^{k+1} + \sum_{j>i} a_{ij} x_j^k - b_i \right]$$

如果对某个k, 给定的误差允许值c有

$$\sum_{i=1}^n |x_i^{k+1} - x_i^k| < c$$

则认为迭代是收敛的。

- 并行化分析

由于每次迭代需要使用本次迭代的前面部分值, 因而难以到同步的并行算法, 下面给出一个异步的并行算法

# 迭代求解的高斯-赛德尔法

- MIMD异步并行算法

- $N$ 个处理器( $N \leq n$ )生成 $n$ 个进程, 每个进程计算 $x$ 的一个分量

Begin

(1)  $old_i \leftarrow x_i^0, new_i \leftarrow x_i^0$

(2) 生成进程 $i$

(3) 进程 $i$

repeat

(i)  $old_i \leftarrow new_i$

(ii)  $new_i \leftarrow (b_i - \sum_{k < i} a_{ik} \times old_k - \sum_{k > i} a_{ik} \times old_k) / a_{ii}$

until  $\sum_{i=1 \sim n} |old_i - new_i| < c$

$x_i \leftarrow new_i$

End



## 第十章 线性方程组的求解

10.1 三角形方程组的求解

10.2 三对角方程组的求解

10.3 稠密线性方程组的求解

10.4 稀疏线性方程组的求解

## 10.4 稀疏线性方程组的求解

### 10.4.1 线性方程组的并行化方法

### 10.4.2 稀疏线性方程组的迭代解法

### 10.4.3 高斯-赛德尔迭代法的并行化

# 线性方程方程的并行化方法

- 线性方程组选择算法的考虑因素
  - 系数矩阵A的结构
    - dense Gaussian elimination, etc
    - Sparse iterative method
    - triangular substitution, odd-even reduction
    - certain PDEs multigrid, etc
  - 计算精度要求
    - Gaussian elimination: more accurate, more expensive
    - Conjugate gradients: less accurate, less expensive
  - 计算环境要求
    - architecture, available languages, compiler quality
    - libraries?

# 线性方程方程的并行化方法

- 求解方法的并行化

(1) 直接解法的并行化(用于稠密线性方程组)

- Gauss消去法(包括选主元的Gauss消去法)
- Gauss-Jordan消去法
- LU分解法

(2) 迭代法的并行化(用于稠密和稀疏线性方程组)

- Jacobi
- Gauss-Seidel(可异步并行化)
- Jacobi OverRelaxation(JOR)
- Gauss-Seidel OverRelaxation(SOR)
- Conjugate Gradient

## 10.4 稀疏线性方程组的求解

10.4.1 线性方程组的并行化方法

10.4.2 稀疏线性方程组的迭代解法

10.4.3 高斯-赛德尔迭代法的并行化

# 稀疏线性方程方程的迭代解法

- 迭代解法

(1) *Jacobi*: 
$$x_i^{k+1} = (b_i - \sum_{j \neq i} a_{ij} x_j^k) / a_{ii}$$

(2) *Gauss-Seidel*: 
$$x_i^{k+1} = (b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k) / a_{ii}$$

(3) *JOR*: 
$$x_i^{k+1} = (1 - \omega) x_i^k + \omega (b_i - \sum_{j \neq i} a_{ij} x_j^k) / a_{ii}$$

(4) *SOR*: 
$$x_i^{k+1} = (1 - \omega) x_i^k + \omega (b_i - \sum_{j < i} a_{ij} x_j^{k+1} - \sum_{j > i} a_{ij} x_j^k) / a_{ii}$$

(5) *Conjugate Gradient*

## 10.4 稀疏线性方程组的求解

10.4.1 线性方程组的并行化方法

10.4.2 稀疏线性方程组的迭代解法

10.4.3 高斯-赛德尔迭代法的并行化

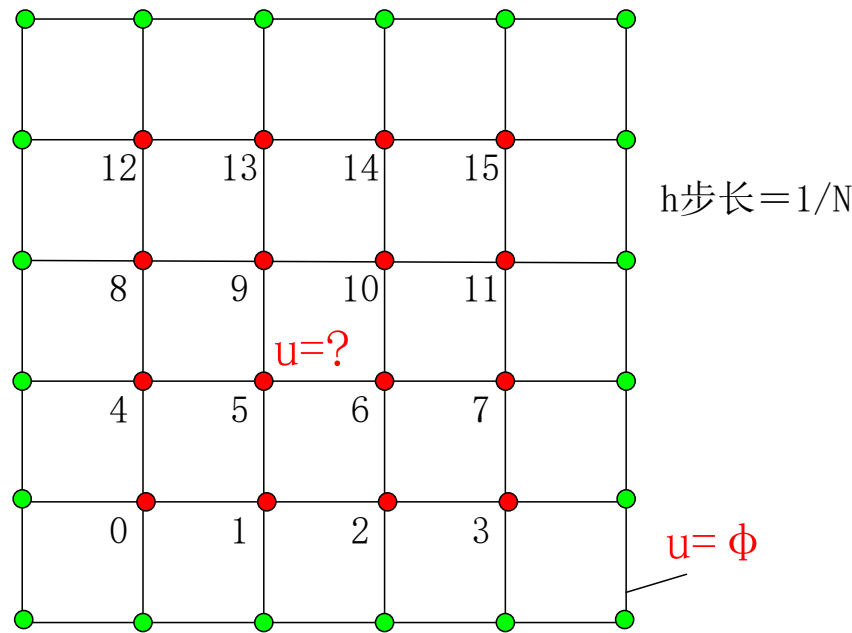
# 高斯-赛德尔迭代法的并行化

- 由PDE离散产生的稀疏线性方程组

## (1) Laplace方程

$$\frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2} = g(x, y) \quad (x, y) \in D = [0, 1] \times [0, 1]$$

求 $u(x, y)$ 满足上面的方程, 且满足边值 $\varphi(x, y)$





# 高斯-赛德尔迭代法的并行化

(2) 由五点格式的离散化(假设 $g(x,y)=0$ )

记 $u_{ij} = u(\frac{i}{N}, \frac{j}{N})$ ,  $0 \leq i, j \leq N$

用二阶中心差分逼近:  $\frac{\partial^2 u}{\partial x^2} \approx \frac{1}{h^2} [u(x+h, y) - 2u(x, y) + u(x-h, y)]$

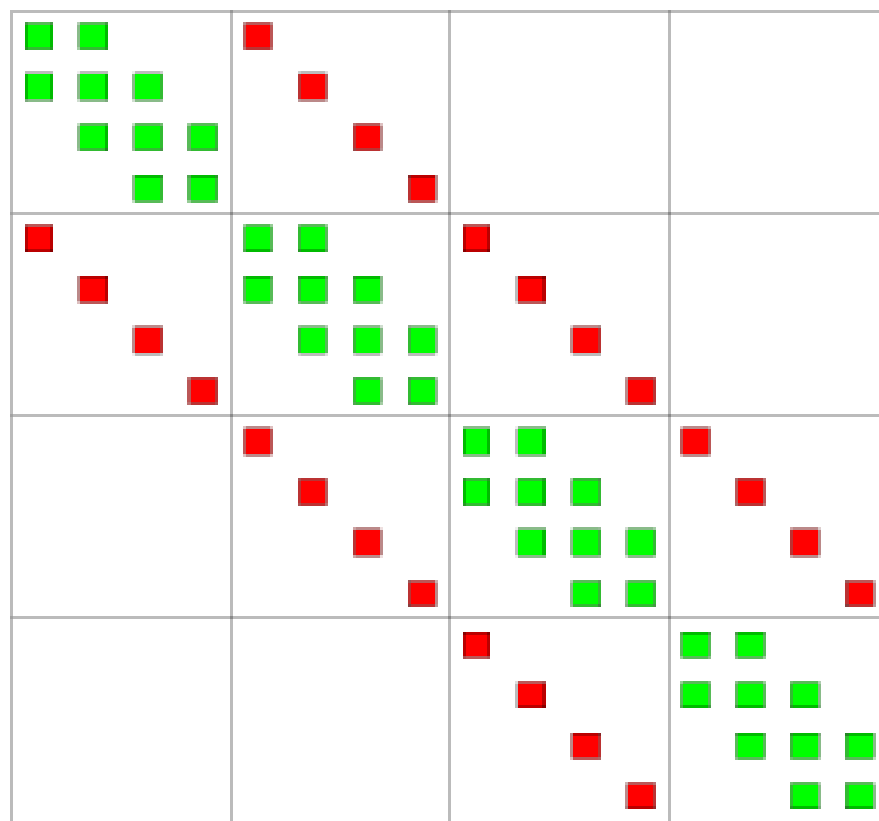
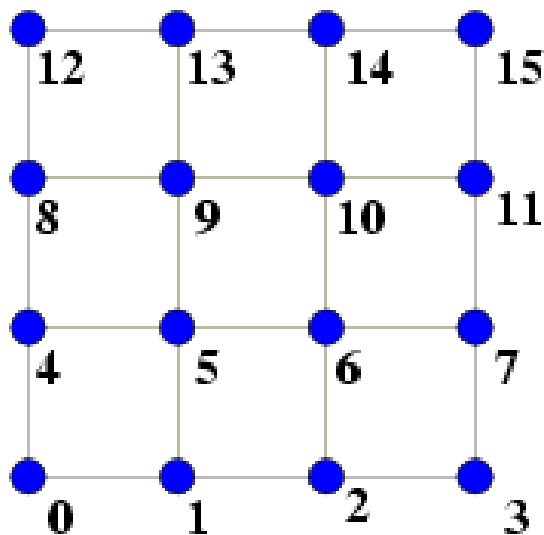
$$\frac{\partial^2 u}{\partial y^2} \approx \frac{1}{h^2} [u(x, y+h) - 2u(x, y) + u(x, y-h)]$$

代入Laplace方程  $\Rightarrow u_{ij} = \frac{1}{4} [u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1}]$   $0 < i, j < N$

$$\Rightarrow A = \begin{bmatrix} R & E & & & \\ E & R & E & & \\ & \ddots & \ddots & \ddots & \\ & & E & R & E \\ & & & E & R \end{bmatrix}, \quad R = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix}, \quad E \text{ 为单位阵}$$

# 高斯-赛德尔迭代法的并行化

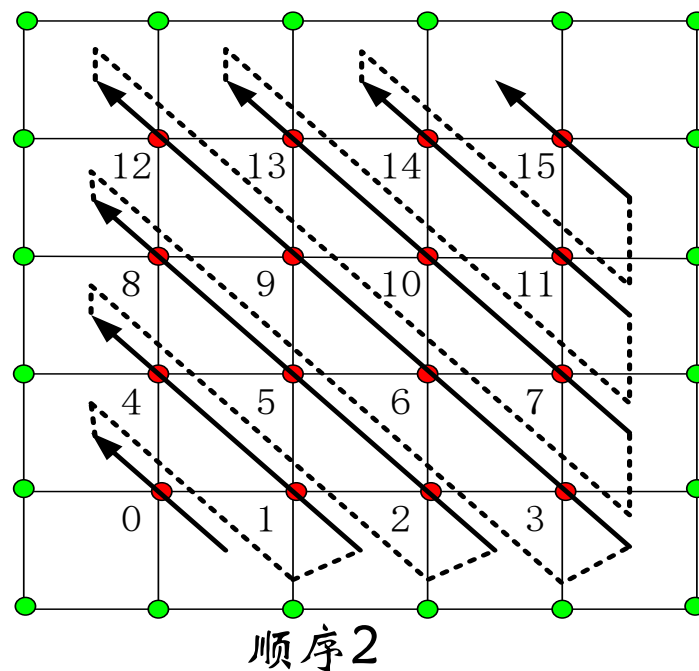
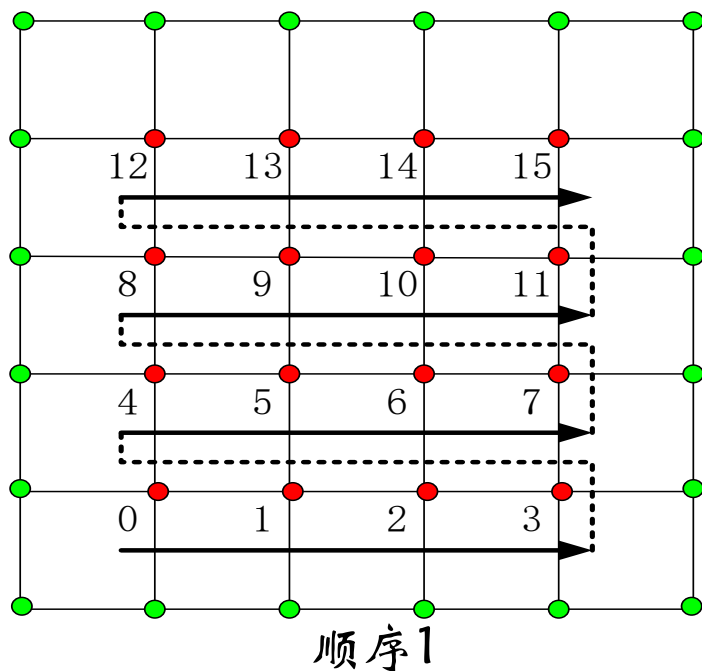
$A$ 为稀疏的块三对角矩阵



# 高斯-赛德尔迭代法的并行化

- Gauss-Seidel迭代解法的并行化

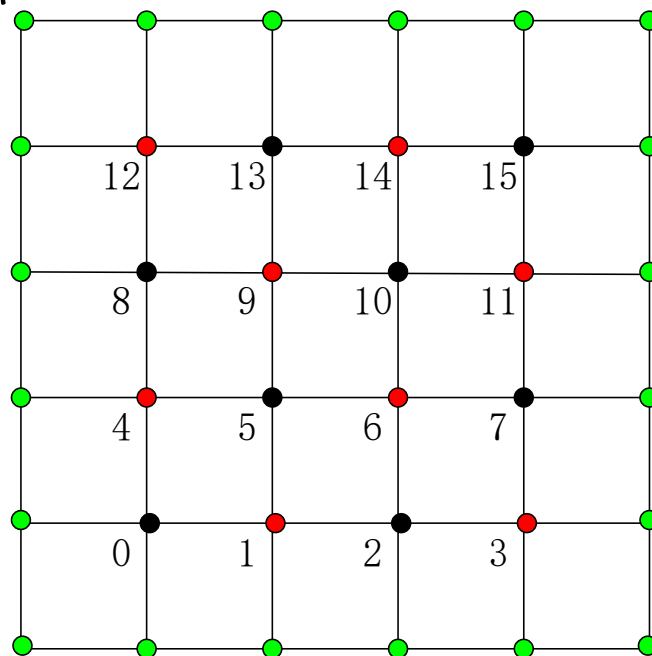
## (1) 两种串行算法的计算顺序及其并行化



注: 顺序1难以并行化; 顺序2可以小规模并行化

# 高斯-赛德尔迭代法的并行化

## (2) 红黑着色并行算法



- ① 并行计算所有的红点
- ② 并行计算所有的黑点
- ③ 重复①、②直至满足精度要求