并行计算
Parallel Computing

主讲 孙经纬

2024年 春季学期

概要

- 第二篇 并行算法
  - 第九章 稠密矩阵运算
  - **第十章 线性方程组的求解**
  - 第十一章 快速傅里叶变换
  - 第十二章 数值计算的基本支撑技术

第十章 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
- 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

基本术语

- 线性方程组的定义和符号

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

记为 AX=b

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \qquad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \qquad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

上三角方程组的求解

- 上三角方程组的回代解法

先看一个例子。假定欲求解以下上三角方程组

$$
\begin{aligned}
x_1+x_2-\ x_3+4x_4&=8\\
-2x_2-3x_3+\ x_4&=5\\
2x_3-3x_4&=0\\
2x_4&=4
\end{aligned}
$$

先求解最后一个方程，得 $x_4=2$，将其值代入其他方程，清去 x_4 这一项，从而有

$$
\begin{aligned}
x_1+x_2-\ x_3\qquad&=0\\
-2x_2-3x_3\qquad&=3\\
2x_3\qquad&=6\\
2x_4&=4
\end{aligned}
$$

再求解第 3 个方程，得 $x_3=3$，将其代入第 1 和 2 个方程，消去 x_3 这一项，从而有

$$
\begin{aligned}
x_1+x_2\qquad&=3\\
-2x_2\qquad&=12\\
2x_3\qquad&=6\\
2x_4&=4
\end{aligned}
$$

同样，消去 x_2，最后将欲求解的上三角方程化成了如下形式的对角方程：

$$
\begin{aligned}
x_1\qquad&=9\\
-2x_2\qquad&=12\\
2x_3\qquad&=6\\
2x_4&=4
\end{aligned}
$$

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

时间复杂度：$O(n^2)$

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 $1\leqslant j\leqslant p$ do

            for k=j to i-1 step p do

                $b_k=b_k-a_{ki}x_i$

                $a_{ki}=0$

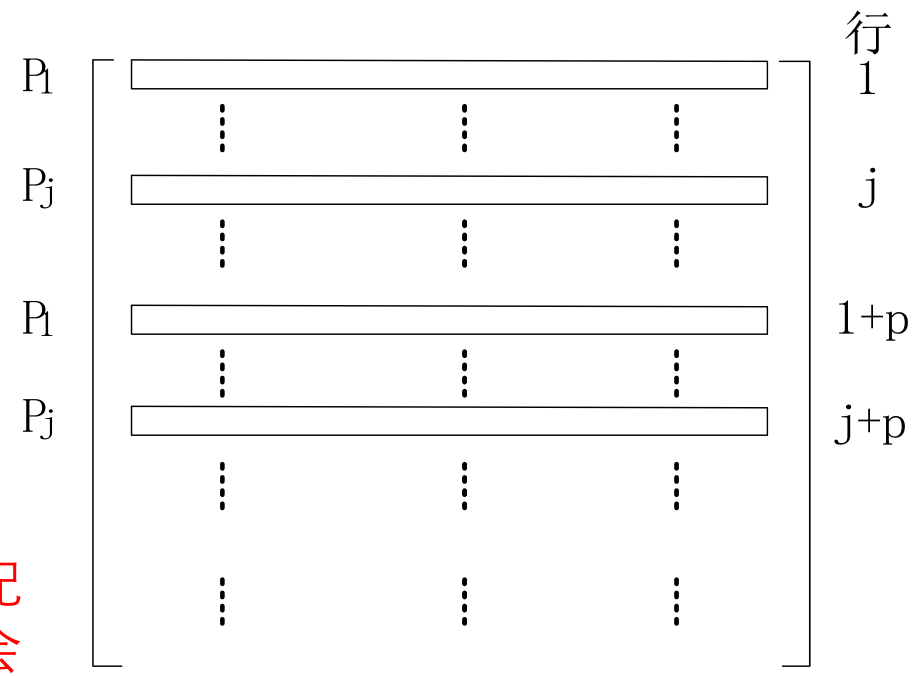
            endfor

        endfor

    endfor

End   // p(n)=n，t(n)=n

这里用的循环分配
如果用连续分配会
如何？



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**
  - 10.2.1 直接求解法
  - 10.2.2 奇偶规约法
- 10.3 稠密线性方程组的求解
- 10.4 稀疏线性方程组的求解

三对角方程组的直接求解

- 三对角方程组中的系数矩阵，除了三对角线上的元素非零，其余元素均为零
- 通常没有只含一个变量的方程，因此不能用求解上三角方程组的方法直接求解。

$$\begin{aligned}
16x_1 + 4x_2 \qquad\qquad\qquad &= 8 \quad ① \\
4x_1 + 11x_2 - 5x_3 \qquad\quad &= 7 \quad ② \\
2x_2 + 14x_3 - 6x_4 &= 13 \quad ③ \\
5x_3 + 18x_4 &= 24 \quad ④
\end{aligned}$$

消去

首先，②$-\frac{1}{4}$①，消去第②个方程中的 x_1：

$$\begin{aligned}
16x_1 + 4x_2 \qquad\qquad\qquad &= 8 \quad ① \\
10x_2 - 5x_3 \qquad\quad &= 5 \quad ② \\
2x_2 + 14x_3 - 6x_4 &= 13 \quad ③ \\
5x_3 + 18x_4 &= 24 \quad ④
\end{aligned}$$

然后，③$-\frac{1}{5}$②，消去第③个方程中的 x_2：

$$\begin{aligned}
16x_1 + 4x_2 \qquad\qquad\qquad &= 8 \quad ① \\
10x_2 - 5x_3 \qquad\quad &= 5 \quad ② \\
15x_3 - 6x_4 &= 12 \quad ③ \\
5x_3 + 18x_4 &= 24 \quad ④
\end{aligned}$$

最后，④$-\frac{1}{3}$③，消去第④个方程中的 x_3：

$$\begin{aligned}
16x_1 + 4x_2 \qquad\qquad\qquad &= 8 \\
10x_2 - 5x_3 \qquad\quad &= 5 \\
15x_3 - 6x_4 &= 12 \\
20x_4 &= 20
\end{aligned}$$

三对角方程组的直接求解

算法 10.3　SISD 上直接求解三对角方程组算法

输入：$A_{n\times n}, b=(b_1,\cdots,b_n)^T$

输出：$x=(x_1,\cdots,x_n)^T$

Begin

(1) **for** $i=1$ **to** $n-1$ **do**

$$\boxed{\begin{aligned} g_{i+1} &= g_{i+1}-(f_{i+1}/g_i)h_i \\ b_{i+1} &= b_{i+1}-(f_{i+1}/g_i)b_i \end{aligned}}$$

endfor

(2) **for** $i=n$ **to** 2 **do**

$$\boxed{\begin{aligned} x_i &= b_i/g_i \\ b_{i-1} &= b_{i-1}-x_i h_{i-1} \end{aligned}}$$

endfor

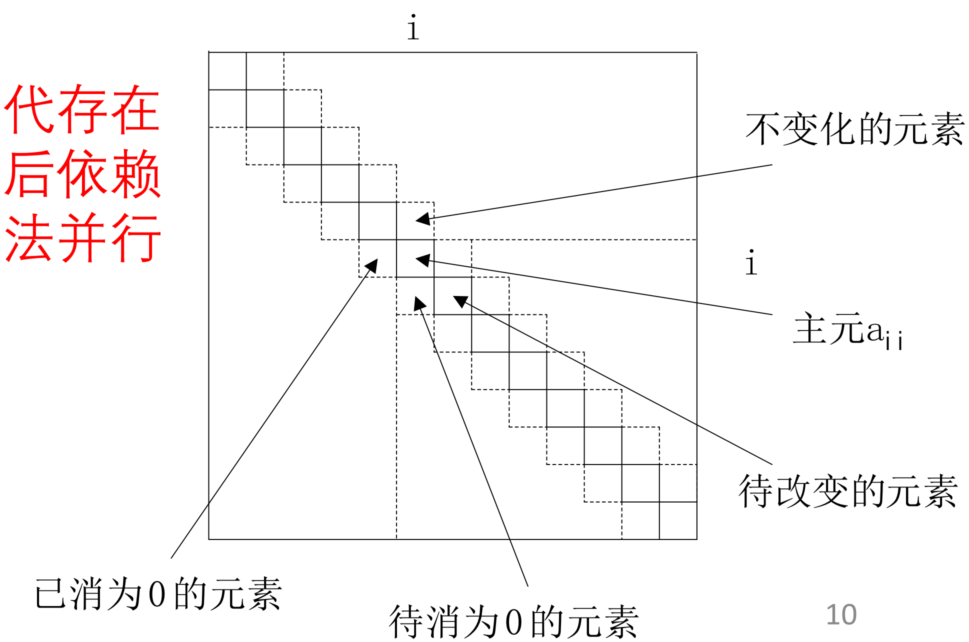
(3) $x_1=b_1/g_1$

End

迭代存在
前后依赖
无法并行

三对角方程组一般形式:

$$\begin{aligned} g_1x_1+h_1x_2 \qquad\qquad &= b_1 \\ f_ix_{i-1}+g_ix_i+h_ix_{i+1} &= b_i, \quad 2\leqslant i\leqslant n-1 \\ f_nx_{n-1}+g_nx_n \qquad\qquad &= b_n \end{aligned}$$



显然，算法 10.3 的复杂度为 $O(n)$。

奇偶规约求解法

- 三对角方程可以写成如下形式

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

$$f_1=h_n=0$$

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，或者奇序号方程中的偶下标变量:

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

$$f_i x_{i-1} + g_i x_i + h_i x_{i+1} \qquad =b_i$$

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

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

$$\alpha_i x_{i-2}+\beta_i x_i+\gamma_i x_{i+2}=\eta_i \qquad i=1,2,\cdots,n$$

奇偶规约求解法

①将方程组划分为奇下标和偶下标的2组n/2变量方程组

②重复①最终可得仅含2个或者3个变量的方程组:

$$\begin{aligned} g_1x_1+h_1x_2 &= b_1 \\ f_2x_1+g_2x_2 &= b_2 \end{aligned} \quad \text{或} \quad \begin{aligned} g_1x_1+h_1x_2 \quad &= b_1 \\ f_2x_1+g_2x_2+h_2x_3 &= b_2 \\ f_3x_2+g_3x_3 &= b_3 \end{aligned}$$

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消去下标可以并行化；

  ③回代求解可以并行化

奇偶规约求解法

假定如下三对角方程组，它有 4 个方程、4 个变量：

$$4x_1 + x_2 = 2 \qquad ①$$

$$4x_1 + 11x_2 - 5x_3 = 7 \qquad ②$$

$$2x_2 + 14x_3 - 6x_4 = 13 \qquad ③$$

$$5x_3 + 18x_4 = 24 \qquad ④$$

首先，将方程①中的 x_2，用方程②中的 $x_2 = (7 - 4x_1 + 5x_3)/11$ 代替之，于是方程①就变成了

$$8x_1 + x_3 = 3 \qquad ⑤$$

其次，将方程②中的 x_1 和 x_3，分别用方程①中的 $x_1 = (2 - x_2)/4$ 和方程③中的 $x_3 = (13 - 2x_1 + 6x_3)/14$ 代替之，于是方程②就变成了

$$10x_2 - 2x_4 = 9 \qquad ⑥$$

然后，将方程③中的 x_2 和 x_4，分别用方程②中 $x_2 = (7 - 4x_1 + 5x_3)/11$ 的和方程④中的 $x_4 = (24 - 5x_3)/18$ 代替之，于是方程③就变成了

$$-24x_1 + 547x_3 = 651 \qquad ⑦$$

最后，将方程④中 x_3 的，用方程③中的 $x_3 = (13 - 2x_2 + 6x_4)/14$ 代替之，于是方程④就变成了

$$-10x_2 + 282x_4 = 271 \qquad ⑧$$

奇偶规约求解法

这样，原来的三对角方程组①、②、③、④就构成了两个子方程组Ⅰ和Ⅱ，它们分别是只包含奇下标变量 x_1 和 x_3 的子方程⑤和⑦与只包含偶下标变量 x_2 和 x_4 的子方程⑥和⑧，即子方程组Ⅰ

$$\begin{cases} 8x_1 + x_3 = 3 \\ -24x_1 + 547x_3 = 651 \end{cases}$$

和子方程组Ⅱ

$$\begin{cases} 10x_2 - 2x_4 = 9 \\ -10x_2 + 282x_4 = 271 \end{cases}$$

由子方程组Ⅰ可解得 $x_1 = 9/40$ 和 $x_3 = 6/5$，由子方程组Ⅱ可解得 $x_2 = 11/10$ 和 $x_4 = 1$。

奇偶规约求解法

算法 10.4　SISD 上三对角方程组奇偶归约求解算法

输入：$A_{n\times n}, b=(b_1,\cdots,b_n)^T$

输出：$x=(x_1,\cdots,x_n)^T$

Begin

(1) **for** $i=0$ **to** $\log n-1$ **do**

(1.1) $d=2^i$

(1.2) **for** $j=2i+1$ **to** $n-1$ **step** $2d$ **to**

$r_j=f_j/g_{j-d}, \delta_j=h_j/g_{j+d}, f'_j=-r_jf_{j-d},$

$g'_j=-\delta_jf_{j+d}-r_jh_{j-d}, h'_j=\delta_jh_{j+d}$

$b'_j=b_j+r_jb_{j-d}-\delta_jb_{j+d}$

endfor

(1.3) $r_n=f_n/g_{n-d}$

(1.4) $f_n=-r_nf_{n-d}$

(1.5) $g_n=g_n-r_nh_{n-d}$

(1.6) $b_n=b_n+r_nb_{n-d}$

(1.7) **for** $j=2i+1$ **to** $n-1$ **step** $2d$ **do**

$f_j=f'_j, g_j=g'_j, h_j=h'_j, b_j=b'_j$

endfor

endfor

(2) $x_n=b_n/g_n$

(3) **for** $i=\log n-1$ **to** 0 **step** -1 **do**

(3.1) $d=2^i$

(3.2) $x_d=(b_d-h_dx_{2d})/g_d$

(3.3) **for** $j=3d$ **to** n **step** $2d$ **do**

$x_j=(b_j-f_jx_{j-d}-h_jx_{j+d})/g_j$

endfor

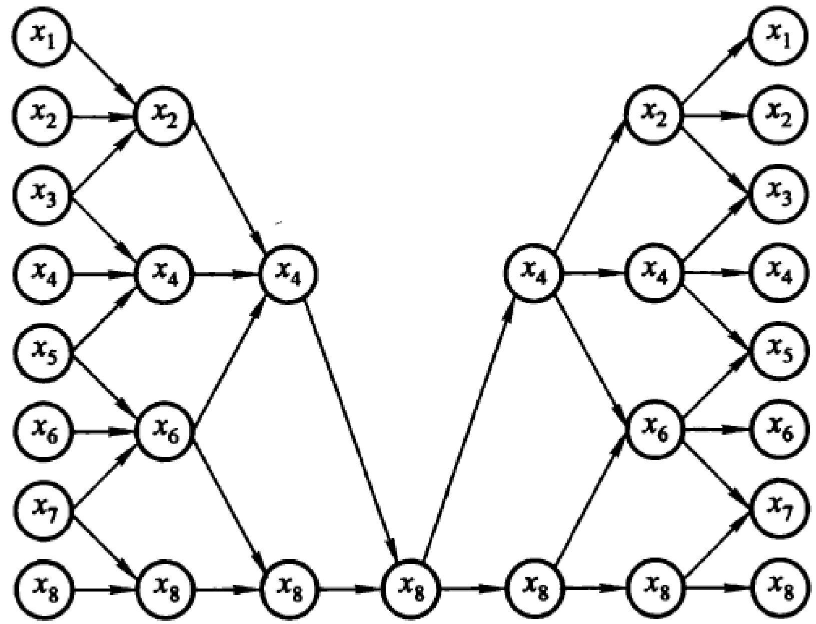
endfor

End

p(n)=O(n), t_1=t_3=O(logn)*O(1), t(n)=(logn)

奇偶规约求解法

为什么奇偶规约法可以并行?



欲求解含 8 个变量 $x_1, x_2, \cdots, x_8$ 的三对角方程组。第①步：消去方程组中含变量 x_1, x_3, x_5 和 x_7 的方程，剩下仅含有变量 x_2, x_4, x_6 和 x_8 的方程组。第②步：消去方程组中含变量 x_2 和 x_6 的方程，剩下仅有变量 x_4 和 x_8 的方程组。第③步：消去含变量 x_4 的方程，剩下只含有变量 x_8 的方程，它可立即求解。一旦 x_8 求解出，就可回代求解出 x_4；再由 x_4 和 x_8，就可求解出 x_2 和 x_6；然后由 x_2、x_4、x_6 和 x_8，就可回代求解出 x_1、x_3、x_5 和 x_7。

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**
  - 10.3.1 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
- 10.4 稀疏线性方程组的求解

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。
  - 消元过程中可以选则主元，即在第i步选择第i列第i~n行中绝对值最大的元素，增加算法的数值稳定性。
  - 消元过程图：

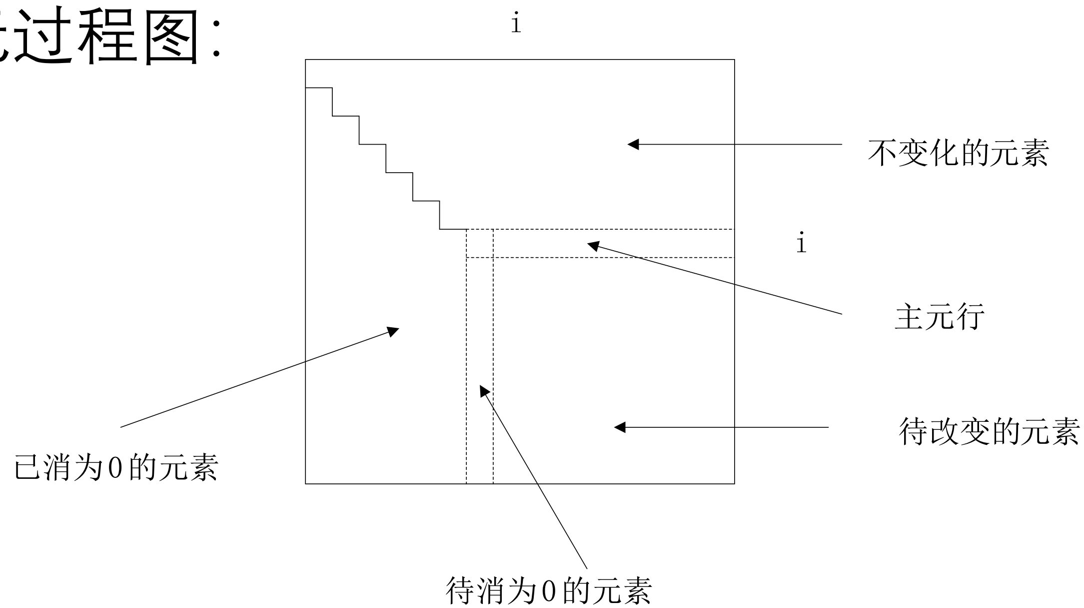


图10.1

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

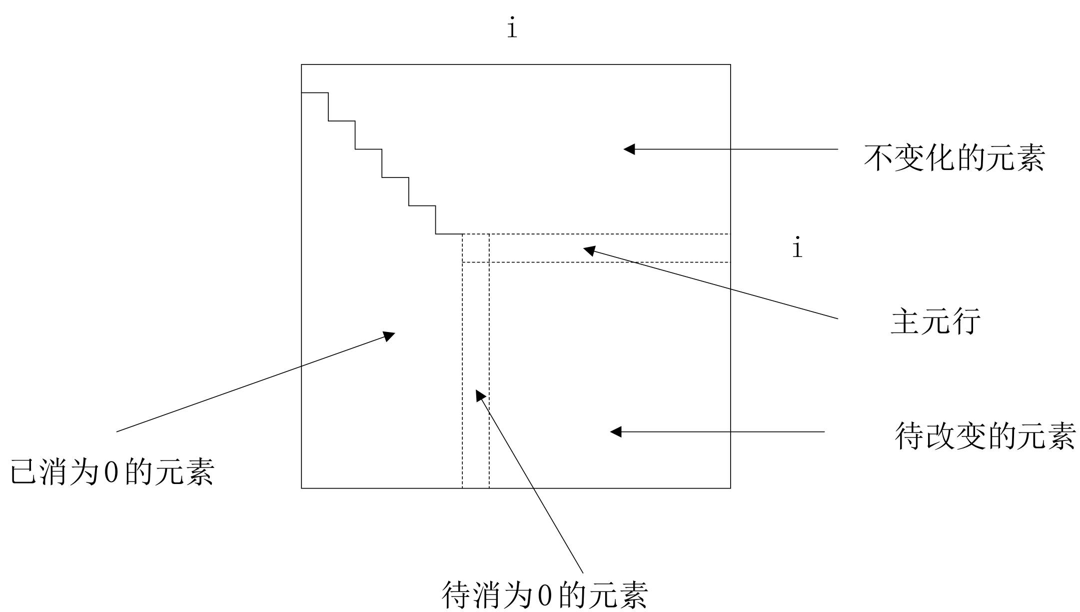


图10.1

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

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix}$$

(1)

$$\begin{pmatrix} \boxed{a_{11}} & a_{12} & a_{13} & a_{14} \\ 0 & b_{22} & b_{23} & b_{24} \\ 0 & b_{32} & b_{33} & b_{34} \\ 0 & b_{42} & b_{43} & b_{44} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} a_{15} \\ b_{25} \\ b_{35} \\ b_{45} \end{pmatrix}$$

(2)

$$\begin{pmatrix} \boxed{a_{11}} & 0 & c_{13} & c_{14} \\ 0 & \boxed{b_{22}} & b_{23} & b_{24} \\ 0 & 0 & c_{33} & c_{34} \\ 0 & 0 & c_{43} & c_{44} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} a_{15} \\ b_{25} \\ c_{35} \\ c_{45} \end{pmatrix}$$

(3)

$$\begin{pmatrix} \boxed{a_{11}} & 0 & 0 & 0 \\ 0 & \boxed{b_{22}} & 0 & 0 \\ 0 & 0 & \boxed{c_{33}} & 0 \\ 0 & 0 & 0 & \boxed{d_{44}} \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} a_{15} \\ b_{25} \\ c_{35} \\ d_{45} \end{pmatrix}$$

(4)

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

$P_{ij}(i<>j)$: $a_{ij} = 0$

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

end for

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

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

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$A^{-1}b$(假设存在A^{-1})：$O(n^2)$

②求A^{-1}：

$$A=\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}=\begin{bmatrix} I & 0 \\ A_{21}A_{11}^{-1} & I \end{bmatrix}\begin{bmatrix} A_{11} & 0 \\ 0 & B \end{bmatrix}\begin{bmatrix} I & A_{11}^{-1}A_{12} \\ 0 & I \end{bmatrix}$$

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

- 矩阵乘法不比矩阵求逆更难
- 矩阵求逆不比矩阵乘法更难

——《算法导论》

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　　$a(n/2)$

主定理：$T(n)=aT\left(\frac{n}{b}\right)+c(n^d)$

$$\begin{cases} T(n)=\Theta(n^d log_2(n)) & \text{if } (a=b^d) \\ T(n)=\Theta(n^d) & \text{if } (a<b^d) \\ T(n)=\Theta(n^{log_b(a)}) & \text{if } (a>b^d) \end{cases}$$

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

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

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

和串行矩阵乘法相同

迭代求解的高斯-赛德尔法

- 串行算法原理

求解 $\boldsymbol{Ax}=\boldsymbol{b}$ 的另一种方法是 Gauss-Seidel 方法。先将系数矩阵 $\boldsymbol{A}$ 分解为 $\boldsymbol{A}=\boldsymbol{E}+\boldsymbol{D}+\boldsymbol{F}$,其中 $\boldsymbol{D}$、$\boldsymbol{E}$ 和 $\boldsymbol{F}$ 均为 $n\times n$ 的矩阵,它们的元素可分别定义如下:

$$d_{ij}=\begin{cases}a_{ij}, & i=j\\ 0, & \text{否则}\end{cases}\quad e_{ij}=\begin{cases}a_{ij}, & i>j\\ 0, & \text{否则}\end{cases}\quad f_{ij}=\begin{cases}a_{ij}, & i<j\\ 0, & \text{否则}\end{cases}$$

这样,$\boldsymbol{Ax}=(\boldsymbol{D}+\boldsymbol{E}+\boldsymbol{F})\boldsymbol{x}=\boldsymbol{b}$,从而 $\boldsymbol{Dx}=\boldsymbol{b}-\boldsymbol{Ex}-\boldsymbol{Fx}$。

例如,$n=3$,则有

$$\begin{pmatrix}a_{11} & 0 & 0\\ 0 & a_{22} & 0\\ 0 & 0 & a_{33}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\boldsymbol{b}-\begin{pmatrix}0 & 0 & 0\\ a_{21} & 0 & 0\\ a_{31} & a_{32} & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}-\begin{pmatrix}0 & a_{12} & a_{13}\\ 0 & 0 & a_{23}\\ 0 & 0 & 0\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}$$

给定初始值 $\boldsymbol{x}^0$,则第 k 次迭代(Iteration)为

$$\boldsymbol{Dx}^k=\boldsymbol{b}-\boldsymbol{Ex}^k-\boldsymbol{Fx}^{k-1}$$

迭代求解的高斯-赛德尔法

- 串行算法原理

例如，$n=4$，$k=1$ 时，

$$a_{11}x_1^1 = b_1 - 0 - (a_{12}x_2^0 + a_{13}x_3^0 + a_{14}x_4^0)$$

$$a_{22}x_2^1 = b_2 - (a_{21}x_1^1) - (a_{23}x_3^0 + a_{24}x_4^0)$$

$$a_{33}x_3^1 = b_3 - (a_{31}x_1^1 + a_{32}x_2^1) - (a_{34}x_4^0)$$

$$a_{44}x_4^1 = b_4 - (a_{41}x_1^1 + a_{42}x_2^1 + a_{43}x_3^1) - 0$$

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

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

则认为迭代是收敛的。

- 并行化分析

第k次迭代时，下三角部分存在数据依赖，迭代内难并行。
上三角部分依赖于第k-1次迭代，迭代间难并行。

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**
  - 10.4.1 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

线性方程组的并行化方法

- 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?

线性方程组的并行化方法

- 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

稀疏线性方程组

- 稀疏线性方程组系数矩阵中绝大多数元素均为零。
- 绝大多数科学和工程计算问题均为由数学模型所表示的一个物理系统。为了适用于计算机求解，必须将连续物理系统的定义域离散化，其常用的方法就是将定义域网格化，这样在离散域上求解数学模型，就可在各格点上获得某些所要求的物理量。
- 通常离散物理域上每个格点的模拟计算，都是基于域上邻近格点的相互作用与影响的。典型地，单格点的模拟就会产生一组线性方程。由于每一格点仅与其近邻有关，所以只有邻近格点的变量的系数是非零，而线性方程组的其余系数均为零，因而线性方程组就变成稀疏的了。

稀疏矩阵存储格式

- Coordinate list (COO)

$$\begin{pmatrix} 1 & 0 & 0 & 2 & 0 & 3 \\ 4 & 5 & 0 & 0 & 0 & 0 \\ 0 & 6 & 7 & 0 & 0 & 8 \\ 9 & 0 & 0 & 10 & 11 & 12 \\ 0 & 13 & 0 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 15 \end{pmatrix}$$

VAL	8	6	12	1	15	14	9	2	3	5	13	4	11	7	10
I	3	3	4	1	6	5	4	1	1	2	5	2	4	3	4
J	6	2	6	1	6	5	1	4	6	2	2	1	5	3	4

- Compressed Sparse Row (CSR)

  记录元素列号和每行的首地址

VAL	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
J	1	4	6	1	2	2	3	6	1	4	5	6	2	5	6

I	1	4	6	9	13	15

稀疏线性方程组的迭代解法

- 迭代解法

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

Jacobi迭代很容易并行化

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

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

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

$$(5)\ Conjugate\ \ Gradient$$

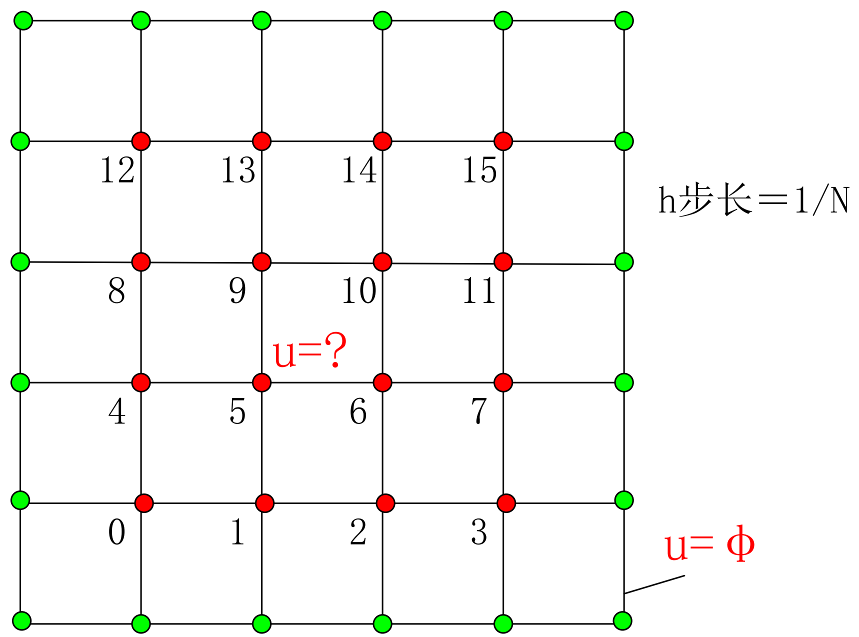
高斯-赛德尔迭代法的并行化

- 由PDE离散产生的稀疏线性方程组

(1)泊松方程

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

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 (设g(x,y)=0，此时为拉普拉斯方程)

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

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

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

高斯-赛德尔迭代法的并行化

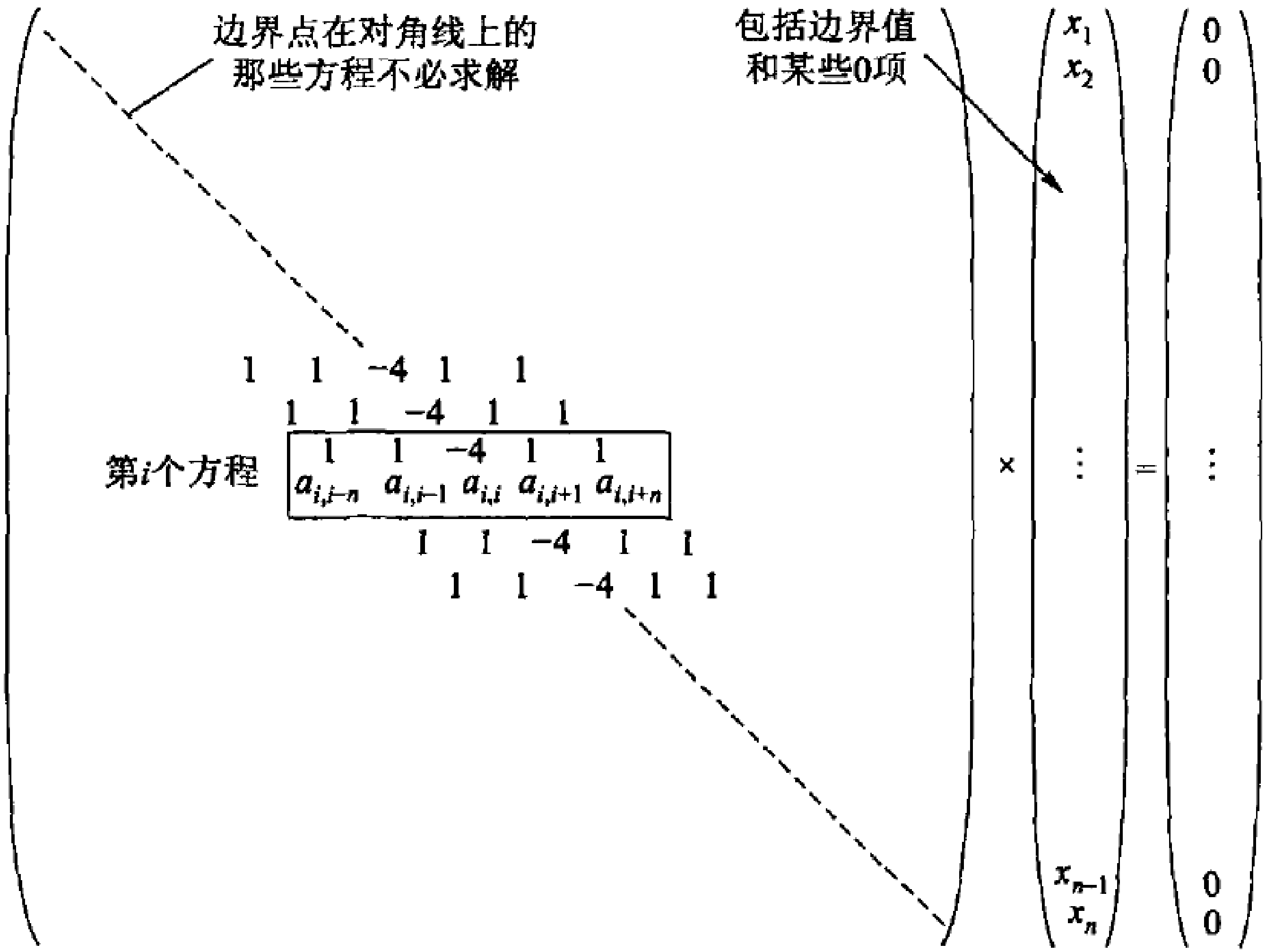


图 10.9 求解拉普拉斯方程的稀疏矩阵

高斯-赛德尔迭代法的并行化

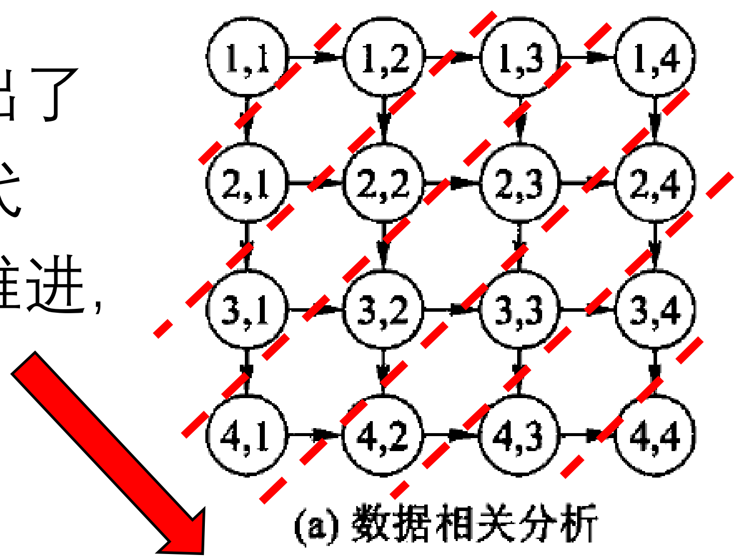
- 迭代公式

$$x_i(k) = \left[b_i - \sum_{j<i} a_{ij}\, x_j(k) - \sum_{j>i} a_{ij} x_j(k-1)\right] / a_{ii}$$

- 代入五点格式

$$x_{i,j}(k) = [x_{i-1,j}(k) + x_{i,j-1}(k) + x_{i+1,j}(k-1) + x_{i,j+1}(k-1)]/4$$

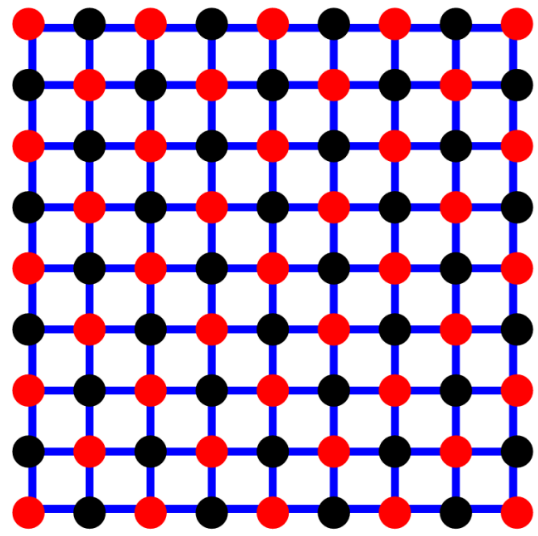
- 上轮迭代的值都已经求出了
- 数据依赖来自于本轮迭代
- 可以从左上往右下并行推进，但并行度并不高



(a) 数据相关分析

高斯-赛德尔迭代法的并行化

- 红黑着色并行算法



- 先同时计算所有红点

$$x_{i,j}(k)=\frac{1}{4}[x_{i-1,j}(k-1)+x_{i,j-1}(k-1)+x_{i+1,j}(k-1)+x_{i,j+1}(k-1)]$$

- 再同时计算所有黑点

$$x_{i,j}(k)=\frac{1}{4}[x_{i-1,j}(k)+x_{i,j-1}(k)+x_{i+1,j}(k)+x_{i,j+1}(k)]$$