线性方程组并行求解

（OpenMP）

—— LU 分解

—— 三角矩阵求解

目录页

Contents

1 LU 分解

2 LU 分解并行计算

3 三角线性方程组的并行求解

1 LU 分解

- LU 分解算法
- 部分选主元 LU 分解算法

1 LU 分解
2 LU分解并行算法
3 三角方程并行求解

LU 分解

定理 (LU 分解的存在性和唯一性) 设 $A \in \mathbb{R}^{n\times n}$. 则存在唯一的单位下三角矩阵 L 和非奇异上三角矩阵 U, 使得 $A = LU$ 的充要条件是 A 的所有顺序主子矩阵 $A_k = A(1:k, 1:k)$ 都非奇异, $k = 1, 2, \ldots, n$.

LU 分解

LU 分解的实现 — 矩阵初等变换

给定一个矩阵

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

- 第一步: 假定 $a_{11} \neq 0$, 构造矩阵

$$L_1 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}, \quad \text{其中} \quad l_{i1} = \frac{a_{i1}}{a_{11}}, i = 2, 3, \ldots, n.$$

LU 分解

易知 L_1 的逆为

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -l_{21} & 1 & 0 & \cdots & 0 \\ -l_{31} & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & \\ -l_{n1} & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

用 L_1^{-1} 左乘 A, 并将所得到的矩阵记为 $A^{(1)}$, 则

$$A^{(1)} = L_1^{-1}A \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{bmatrix}.$$

即左乘 L_1^{-1} 后, A 的第一列中除第一个元素外其它都变为 0.

- 第二步: 将上面的操作作用在 $A^{(1)}$ 的子矩阵 $A^{(1)}(2:n,2:n)$ 上, 将其第一列除第一个元素外都变为 0: 假定 $a_{22}^{(1)} \neq 0$, 构造矩阵

$$L_2 = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ 0 & l_{n2} & 0 & \cdots & 1 \end{bmatrix}, \quad \text{其中} \quad l_{i2} = \frac{a_{i2}^{(1)}}{a_{22}^{(1)}}, i = 3, 4, \ldots, n.$$

用 L_2^{-1} 左乘 $A^{(1)}$, 并将所得到的矩阵记为 $A^{(2)}$, 则

$$A^{(2)} = L_2^{-1} A = L_2^{-1} L_1^{-1} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix}.$$

- 依此类推, 假定 $a_{kk}^{(k-1)} \neq 0\,(k=3,4,\ldots,n-1)$, 则我们可以构造一系列的矩阵 $L_3, L_4, \ldots, L_{n-1}$, 使得

$$L_{n-1}^{-1} \cdots L_2^{-1} L_1^{-1} A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \cdots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & \cdots & a_{nn}^{(n-1)} \end{bmatrix} \triangleq U \;\rightarrow \text{上三角}$$

于是可得 $\boxed{A = LU}$ 其中

$$L = L_1 L_2 \cdots L_{n-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ l_{21} & 1 & 0 & \cdots & 0 \\ l_{31} & l_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & & \ddots & \\ l_{n1} & l_{n2} & l_{n3} & \cdots & 1 \end{bmatrix}$$

算法 1.2 LU 分解

for $k = 1$ to $n-1$ **do**
  for $i = k+1$ to n **do**
    $l_{ik} = a_{ik}/a_{kk}$ % 计算 L 的第 k 列
  end for
  for $j = k$ to n **do**
    $u_{kj} = a_{kj}$ % 计算 U 的第 k 行
  end for
  for $i = k+1$ to n **do**
    for $j = k+1$ to n **do**
      $a_{ij} = a_{ij} - l_{ik}u_{kj}$ % 更新 $A(k+1:n, k+1:n)$
    end for
  end for
end for

LU 分解

算法 1.3 LU 分解

for $k = 1$ to $n-1$ **do**
    for $i = k+1$ to n **do**
        $a_{ik} = a_{ik}/a_{kk}$
        for $j = k+1$ to n **do**
            $a_{ij} = a_{ij} - a_{ik}a_{kj}$
        end for
    end for
end for

LU 分解

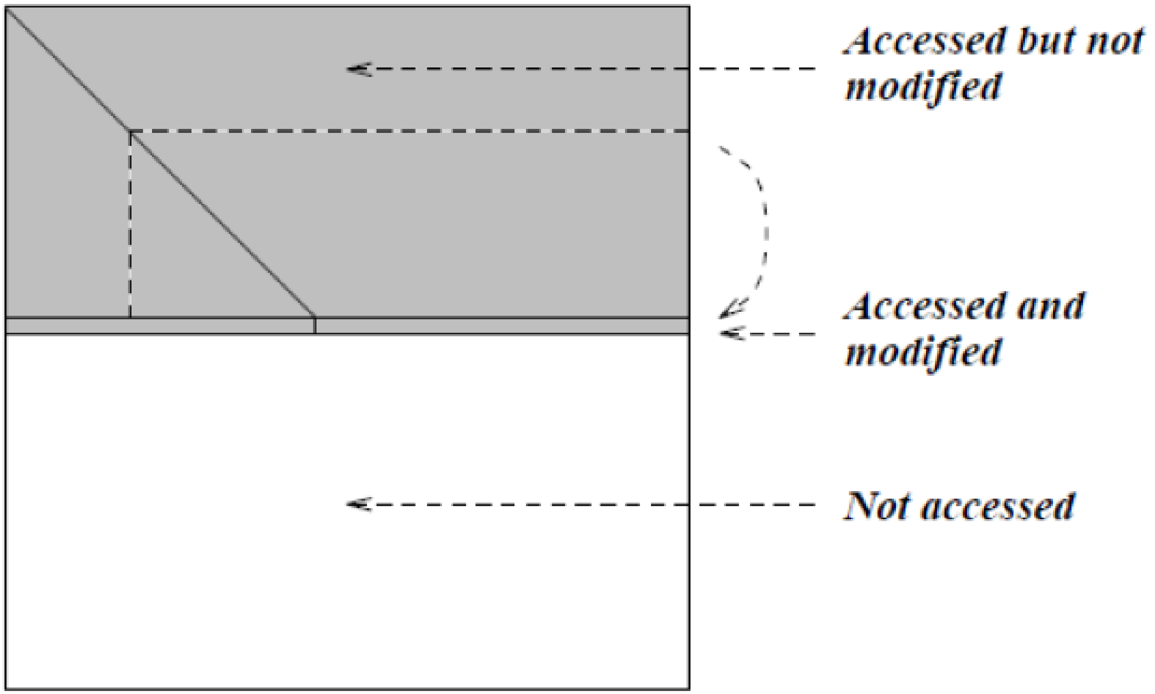
如果数据是按行存储的, 如 C/C++, 我们一般采用下面的 IKJ 型 LU 分解.

算法 1.4 LU 分解 (IKJ 型)

```
for i = 2 to n do
    for k = 1 to i − 1 do
        a_ik = a_ik / a_kk
        for j = k + 1 to n do
            a_ij = a_ij − a_ik a_kj
        end for
    end for
end for
```

LU 分解

上述算法可以用下图来描述.



举例

编写程序，实现 LU分解，其中

$$A=\left[a_{ij}\right]\in\mathbb{R}^{n\times n}=\mathbf{tridiag(-1,2-1)}$$

示例程序：C_LU.c

列主元 LU 分解

定理：若 A 非奇异，则存在排列矩阵 P，使得

$$PA = LU$$

其中 L 为单位下三角矩阵，U 为上三角矩阵。

算法 1.8 部分选主元 LU 分解

$p = 1 : n$ % 用于记录置换矩阵

for $k = 1$ to $n-1$ **do**

  $[a_{\max}, l] = \max_{k \le i \le n} |a_{ik}|$ % 选列主元, 其中 l 表示主元所在的行

  if $l \neq k$ **then**

    for $j = 1$ to n **do**

      $tmp = a_{kj}, a_{kj} = a_{lj}, a_{lj} = tmp$ % 交换第 k 行与第 l 行

    end for

    $tmp = p(k), p(k) = p(l), p(l) = tmp$ % 更新置换矩阵

  end if

  for $i = k+1$ to n **do**

    $a_{ik} = a_{ik} / a_{kk}$ % 计算 L 的第 k 列

  end for

  for $i = k+1$ to n **do**

    for $j = k+1$ to n **do**

      $a_{ij} = a_{ij} - a_{ik} * a_{kj}$ % 更新 $A(k+1:n, k+1:n)$

    end for

  end for

end for

2 LU 分解并行算法

- LU 分解算法
- 部分选主元 LU 分解算法

1 LU 分解
2 LU分解并行算法
3 三角方程并行求解

LU 分解并行算法

```
#pragma omp parallel num_threads(nThreads)
{
  for(int i=0;i<n;i++)
  {
      div=1/A[i][i];
      #pragma omp for schedule(static)
      for(j=i+1;j<n;j++)
         L[j][i] = A[j][i]*div;

      #pragma omp for schedule(static)
      for(int j=i+1;j<n;j++){
        for(int k=i;k<n;k++)
           U[j][k] = U[j][k] - L[j][i]*U[i][k];
      }
}
```

共享内存并行算法

基于 OpenMP 的部分选主元 LU 分解并行算法（自行练习）

手工任务划分（结合MPI实现，后面介绍）

算法　矩阵 LU 分解的并行算法

$icol = 0$

$for\ j = 0\ to\ n-2\ do$

　$if\ myid = j \bmod p\ then$

　　$find\ l : |a_{l,icol}| = \max\{a_{i,icol}, i = j, \ldots, n-1\}$

　　$if\ l \neq j,\ swap\ a_{j,icol}\ and\ a_{l,icol}$

　　$if\ a_{j,icol} = 0,\ A\ is\ singular\ and\ kill\ all\ processes$

　　$a_{i,icol} = a_{i,icol}/a_{j,icol}, i = j+1, \ldots, n-1$

　　$f_{i-j-1} = a_{i,icol}, i = j+1, \ldots, n-1$

　　$send(l, myid+1 \bmod p)\ and\ send(f, myid+1 \bmod p)$

　　$icol + 1 \to icol,\ swap\ a_{j,icol}\ and\ a_{l,icol}$

　$else$

　　$recv(l, myid-1 \bmod p)\ and\ recv(f, myid-1 \bmod p)$　　// 广播 l 和 f

　　$if\ myid+1 \neq j \bmod p,\ send(l, myid+1 \bmod p)\ and\ send(f, myid+1 \bmod p)$

　$end\{if\}$

　$if\ l \neq j,\ swap\ A_j\ and\ A_l$

　$for\ k = icol\ to\ m-1\ do$

　　$a_{ik} = a_{ik} - f_{i-j-1} * a_{jk}, i = j+1, \ldots, n-1$

　$end\{for\}$

$end\{for\}$

华东师范大学 数学科学学院
School of Mathematical Sciences, ECNU

3 三角方程并行求解

1 LU 分解
2 LU分解并行算法
3 三角方程并行求解

三角方程并行求解

$$\begin{pmatrix} l_{00} & & & \\ l_{10} & l_{11} & & \\ \vdots & & \ddots & \\ l_{n-1,0} & l_{n-1,1} & \cdots & l_{n-1,n-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{n-1} \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

$$\boxed{\begin{aligned} x_0 &= b_0/l_{00} \\ x_1 &= (b_1 - l_{10}x_0)/l_{11} \\ x_2 &= (b_2 - l_{20}x_0 - l_{21}x_1)/l_{22} \\ x_3 &= (b_3 - l_{30}x_0 - l_{31}x_1 - l_{32}x_2)/l_{33} \\ x_4 &= (b_4 - l_{40}x_0 - l_{41}x_1 - l_{42}x_2 - l_{43}x_3)/l_{44} \\ &\vdots \\ x_i &= (b_i - l_{i0}x_0 - l_{i1}x_1 - l_{i2}x_2 - \cdots - l_{i,i-1}x_{i-1})/l_{i,i} \\ &\vdots \end{aligned}}$$

下三角方程组 $Lx = b$ 的串行算法（列扫描法）

$$
\begin{aligned}
& for\ i = 0\ to\ n-1\ do \\
& \qquad x_i = b_i / l_{ii} \\
& \qquad for\ j = i+1\ to\ n-1\ do \\
& \qquad\qquad b_j = b_j - l_{ji} * x_i \\
& \qquad end\{for\} \\
& end\{for\}
\end{aligned}
$$

基于列扫描的下三角线性方程组的并行算法

- 串行计算 x_0，然后所有线程并行更新 $b_i = b_i - l_{i0}x_0$，$i = 1,2,\ldots,n-1$
- 串行计算 x_1，然后所有线程并行更新 $b_i = b_i - l_{i1}x_1$，$i = 2,3,\ldots,n-1$
- 串行计算 x_2，然后所有线程并行更新 $b_i = b_i - l_{i2}x_2$，$i = 3,4,\ldots,n-1$
- …… ……


谢谢
THANK YOU