Lecture notes 6 – Introduction to Computational Science

Micheal Multerer
Copied by: Claudio Maggioni

April 19, 2020

2.5: Partial pivoting

Obviously, Algorithm 2.8 fails if one of the pivots becomes zero. In this case, we need to choose a different pivot element.

Simple approach: Column pivoting choose $|a_{k,i}^{(i)}| = \max_{i \leq l \leq n} |a_{l,i}^{(i)}|$ In order to move the pivot element and the corresponding row we switch row k and row i by a *primitive matrix*:

$$\bar{P}_i := \begin{bmatrix} 1_1 & & & & & & & \\ & \ddots & & & & & & \\ & & 1_{i-1} & & & & & \\ & & & 0 & \dots & 1_k & & \\ & & & \vdots & 1_{i+1} & & & \\ & & & 1_i & & 0 & & \\ & & & & & & \ddots & \\ & & & & & & & 1_n \end{bmatrix}$$

The following rules apply:

1. Multiplication by P_i from the left $\Rightarrow$ interchange rows i and k
2. Multiplication by P_i from the right $\Rightarrow$ interchange columns i and k
3. $(\bar{P}_i)^2 = \bar{P}_i \cdot \bar{P}_i = \bar{I}$

Then performing an LU decompisition with pivoting can be written in matrix notation as:

$$A_{i+1} = L_i \cdot \bar{P}_i \cdot A_i$$

Note: For $j < i$, it holds $\bar{P}_i \bar{L}_j = \widetilde{L}_j \bar{P}_i$ where $\widetilde{L}$ is the same matrix as $\bar{L}_j$ except that $[\widetilde{l}_j]_i$ and $[\widetilde{L}_j]_k$ are interchanged:

$$\bar{L}_i := \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{l}_2^{(j)} & & & \\ & & \hat{l}_i^{(j)} & 1 & & \\ & & \hat{l}_k^{(j)} & & \ddots & \\ & & \hat{l}_n^{(j)} & & & 1 \end{bmatrix} \qquad \widetilde{L}_i := \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \hat{l}_2^{(j)} & & & \\ & & \hat{l}_k^{(j)} & 1 & & \\ & & \hat{l}_i^{(j)} & & \ddots & \\ & & \hat{l}_n^{(j)} & & & 1 \end{bmatrix}$$

Resolving (*) then yields:

$$\bar{A}_{i+1} = \bar{L}_i \bar{P}_i \bar{L}_{i-1} \bar{P}_{i-1} \dots \bar{L}_1 \bar{P}_1 A$$

or

$$\bar{L}_{n-1} \bar{P}_{n-1} \bar{L}_{n-2} \bar{P}_{n-2} \dots \bar{L}_1 \bar{P}_1 \bar{A} = \bar{U}$$

Now we can exploit $P_2 \bar{L}_1 \bar{P}_1 = \widetilde{L}_1 \bar{P}_2 \bar{P}_1$ and so on. This yields:

$$\bar{P}\bar{A} = \bar{L}\bar{U}$$

with

$$\bar{P} = \bar{P}_{n-1} \bar{P}_{n-2} \dots \bar{P}_1$$

and

$$\widetilde{L} = \widetilde{L}_1^{-1} \dots \widetilde{L}_{n-1}^{-1}$$

and

$$\widetilde{L}_{n-1} = \bar{L}_{n-1}$$
$$\widetilde{L}_{n-2} = \bar{P}_{n-1} \bar{L}_{n-2} \bar{P}_{n-1}$$
$$\vdots$$
$$\widetilde{L}_1 = \bar{P}_{n-1} \bar{P}_{n-2} \dots \bar{P}_2 \bar{L}_1 \bar{P}_2 \dots \bar{P}_{n-2} \bar{P}_{n-1}$$

Note: If $\bar{A} \in R^{n\times n}$ is non-singular, the pivoted LU decomposition $\bar{P}\bar{A} = \bar{L}\bar{U}$ always exists.

We can easily add column priority in Algorithm 2.8:

Algorithm 2.9 (Outer product LU decomposition with column pivoting)
input: matrix $\bar{A} = [a_{i,j}]_{i,j=1}^n \in R^{n\times n}$
output: pivoted LU decomposition $\bar{L}\bar{U} = \bar{P}\bar{A}$

1. Set $\bar{A}_1 = \bar{A}, \bar{p} = [1, 2, \dots, n]$
2. For $i = 1, 2, \dots, n$
   - compute: k = $\arg\max_{1\leq j\leq n} |a_{p_j,i}^{(i)}|$ % find pivot
   - swap: $p_i \longleftrightarrow p_k$
   - $\bar{l}_i := \bar{a}_{:,i}^{(i)} / a_{p_i,i}^{(i)}$
   - $\bar{u}_i := a_{p_i,:}^{(i)}$
   - compute: $\bar{A}_{i+1} = \bar{A}_i - \bar{l}_i \cdot \bar{u}_i$
3. set $\bar{P} := [\bar{e}_{p_1}, \bar{e}_{p_2}, \dots, \bar{e}_{p_n}]^T$ % $\bar{e}_i$ is i-th unit vector
4. set $\bar{L} = \bar{P}[\bar{l}_1, \bar{l}_2, \cdots, \bar{l}_n]$

Example 2.10 *(omitted)*

2.6: Cholesky decomposition

If $\bar{A}$ is symmetric and positive definite, i.e. all eigenvalues of $\bar{A}$ are bigger than zero or equivalently $\bar{x}^T\bar{A}\bar{x} > 0$ for all $\bar{x} \neq 0$, we can compute a symmetric decomposition of $\bar{A}$.

Note: if $\bar{A}$ is symmetric and positive definite, then the *Schur complement* $\bar{S} := \bar{A}_{2:n,2:n} - (\bar{a}_{2:n}, 1/a_{1,1})\bar{a}^T_{2:n,1}$, is symmetric and positive definite as well. In particular, it holds $s_{i,i} > 0$ and $a_{i,i} > 0$!

Definition 2.11 A decomposition $\bar{A} = \bar{L}\bar{L}^T$ with a lower triangular matrix $\bar{L}$ with positive diagonal elements is called *Cholesky decomposition of* $\bar{A}$.

Note: A Cholesky decomposition exists, if $\bar{A}$ is symmetric and positive definite.

Algorithm 2.12 (outer product of Cholesky decomposition)
input: matrix $\bar{A}$ symmetric and positive definite
output: Cholesky decomposition $\bar{A} = \bar{L}\bar{L}^T = [\bar{l}_1, \bar{l}_2, \dots, \bar{l}_n][\bar{l}_1, \bar{l}_2, \dots, \bar{l}_n]^T$

1. set: $\bar{A}_1 := \bar{A}$
2. for $i = 1, 2, \dots, n$
   - set: $\bar{l}_i := a^{(i)}_{:,i} / \sqrt{a^{(i)}_{i,i}}$
   - set: $\bar{A}_{i+1} := \bar{A}_i - \bar{l}_i \bar{l}_i^T$
3. set: $\bar{L} = [\bar{l}_1, \bar{l}_2, \dots, \bar{l}_n]$

The computational cost is $\frac{1}{6}n^3 + O(n^2)$ and thus only half the cost of LU decomposition.

Example 2.13 *(omitted)*