SI231b: Matrix Computations

Lecture 5: Solving Linear Equations (Direct Methods)

Yue Qiu

qiuyue@shanghaitech.edu.cn

School of Information Science and Technology

ShanghaiTech University

Sept. 22, 2020

Direct Solution of Linear Systems

- LU Factorization with Pivoting
- Implementation on Computers
- Computational Complexity of LU Factorization
- General Procedure of Direct Methods

LU Factorization with Pivoting

Step k of LU factorization

$$\begin{bmatrix} a_{11}^{(0)} & \times & \times & \cdots & \times & \times \\ 0 & a_{22}^{(1)} & \times & \cdots & \times & \times \\ 0 & 0 & \times & \cdots & \times & \times \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & a_{kk}^{(k-1)} & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times & \times \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11}^{(0)} & \times & \times & \cdots & \times & \times \\ 0 & a_{22}^{(1)} & \times & \cdots & \times & \times \\ 0 & 0 & \times & \cdots & \times & \times \\ \vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\ 0 & 0 & 0 & a_{kk}^{(k-1)} & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix}$$

- $\mathrm{A}^{(k)} = \mathrm{M}_k \mathrm{A}^{(k-1)}$
- Require: $a_{kk}^{(k-1)} \neq 0$
  - under which condition?
  - if unsatisfied, what to do?

LU Factorization with Pivoting

$$
\begin{bmatrix}
a_{11}^{(0)} & \times & \times & \cdots & \times & \times \\
0 & a_{22}^{(1)} & \times & \cdots & \times & \times \\
0 & 0 & \times & \cdots & \times & \times \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \times & \times \\
0 & 0 & 0 & \times & \times & \times \\
0 & 0 & 0 & S & \times & \times
\end{bmatrix}
\xrightarrow{\text{pivoting}}
\begin{bmatrix}
a_{11}^{(0)} & \times & \times & \cdots & \times & \times \\
0 & a_{22}^{(1)} & \times & \cdots & \times & \times \\
0 & 0 & \times & \cdots & \times & \times \\
\vdots & \vdots & \vdots & \ldots & \vdots & \vdots \\
0 & 0 & 0 & S & \times & \times \\
0 & 0 & 0 & \times & \times & \times \\
0 & 0 & 0 & 0 & \times & \times
\end{bmatrix}
$$

- partial pivoting
  - finding $p = \arg\max_{k \le i \le n} \left| a_{ik}^{(k-1)} \right|$
  - let $a_{kk}^{(k-1)} = a_{pk}^{(k-1)}$ (row exchange)
- complete pivoting
  - finding $[p_r, p_c] = \arg\max_{k \le i,\ j \le n} \left| a_{ij}^{(k-1)} \right|$
  - let $a_{kk}^{(k-1)} = a_{p_r p_c}^{(k-1)}$ (row and column exchange)

Permutation Matrix

A square matrix with exactly one entry of 1 in each row and column and 0 elsewhere.

Example

$$P = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \qquad Px = \begin{bmatrix} x_3 \\ x_2 \\ x_1 \end{bmatrix}, \qquad x^T P = \begin{bmatrix} x_3 & x_2 & x_1 \end{bmatrix}$$

PA: exchange rows of A

AP: exchange columns of A

Properties:

- P is an orthogonal matrix, i.e., $P^T P = PP^T = I$.
- $P^{-1} = P^T$

LU Factorization with Partial Pivoting

Step k of LU factorization

1. row exchange: $\tilde{A}^{(k-1)} = P_k A^{(k-1)}$
2. Gaussian elimination: $A^{(k)} = M_k \tilde{A}^{(k-1)}$

In general, the procedure follows

$$M_{n-1}P_{n-1}M_{n-2}P_{n-2}\cdots M_1P_1A = U.$$

Denote

$$\begin{aligned}
\tilde{M}_{n-1} &= M_{n-1}, \\
\tilde{M}_{n-2} &= P_{n-1}M_{n-2}P_{n-1}^T, \\
\vdots \quad &= \quad \vdots \\
\tilde{M}_k &= P_{n-1}P_{n-2}\cdots P_{k+1}M_kP_{k+1}^T\cdots P_{n-2}^TP_{n-1}^T
\end{aligned}$$

Note: $\tilde{M}_k$ has the same structure with M_k (recall the structure of M_k)

LU Factorization with Partial Pivoting

Following the aforementioned procedure,

$$PA = LU,$$

where

- $P = P_{n-1}P_{n-2}\cdots P_1$ is again a permutation matrix (why?)
- $L = \left(\tilde{M}_{n-1}\tilde{M}_{n-2}\cdots\tilde{M}_1\right)^{-1}$ is a lower-triangular matrix with unit diagonals

- sometimes called LUP factorization
- always exists for any square A, no matter A is nonsingular or not[1]

Another Interpretation

1. permute the rows of A according to P
2. compute the LU factorization without pivoting to PA

Note: LU factorization with partial pivoting is not carried out in this way, since P is unknown in advance.

[1] https://arxiv.org/abs/math/0506382

A Simple Example

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

Step 1, 1st row $\longleftrightarrow$ 3rd row of A, then perform Gaussian elimination

$$\tilde{A}^{(0)} = P_1 A = \begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

$$A^{(1)} = M_1 \tilde{A}^{(0)} = \begin{bmatrix} 1 & & & \\ -\frac{1}{2} & 1 & & \\ -\frac{1}{4} & & 1 & \\ -\frac{3}{4} & & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix}$$

A Simple Example

Step 2: 2nd row $\longleftrightarrow$ 4th row of $A^{(1)}$, then repeat Gaussian elimination

$$\tilde{A}^{(1)} = P_2 A^{(1)} = \begin{bmatrix} 1 & & & \\ & & & 1 \\ & & 1 & \\ & 1 & & \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix}$$

$$A^{(2)} = M_2 \tilde{A}^{(1)} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} \\ & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{2}{7} & \frac{4}{7} \\ & & -\frac{6}{7} & -\frac{2}{7} \end{bmatrix}$$

Now, it's your turn to give P_3, M_3 and the final P, L, and U

$$\underbrace{\begin{bmatrix} & & 1 & \\ & & & 1 \\ & 1 & & \\ 1 & & & \end{bmatrix}}_{\mathsf{P}} \underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{\mathsf{A}} = \underbrace{\begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}}_{\mathsf{L}} \underbrace{\begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}}_{\mathsf{U}}$$

In practice, the permutation matrix P

- is not represented explicitly as a matrix or the product of permutation matrices
- an equivalent effect can be achieved via a permutation vector

Note: $|\ell_{ij}| \leq 1$ for $i \geq j$

LU with complete pivoting:

In matrix form, at each stage before Gaussian elimination

- permutation of rows with P_k on the left
- permutation of columns with Q_k on the right

$$\mathrm{M}_{n-1}\mathrm{P}_{n-1}\mathrm{M}_{n-2}\mathrm{P}_{n-2}\cdots\mathrm{M}_1\mathrm{P}_1\mathrm{A}\mathrm{Q}_1\mathrm{Q}_2\cdots\mathrm{Q}_{n-1}=\mathrm{U}.$$

By

- using the same definition of L, P with LU factorization with partial pivoting,
- denoting $\mathrm{Q}=\mathrm{Q}_1\mathrm{Q}_2\cdots\mathrm{Q}_{n-1}$,

the LU factorization with complete pivoting can be represented by

$$\mathrm{PAQ}=\mathrm{LU}$$

Too computationally expensive, why?

LU Factorization without Pivoting:

```
U = A, L = I;
for k = 1 :  n-1
     for j = k+1 :  n
          ℓ_jk = u_jk / u_kk
          u_{j,k:n} = u_{j,k:n} − ℓ_jk u_{k,k:n}
     end
end
U = triu(U)
```

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops

Please give your own explanation

Naive Implementation on Computers

LU Factorization with Partial Pivoting:

```
U = A, L = I, P = I;
for k = 1 :  n-1
     select i ≥ k to maximize |u_ik|
     u_{k,k:m} ↔ u_{i,k:m} (exchange of rows)
     ℓ_{k,1:k-1} ↔ ℓ_{i,1:k-1}
     p_{k,:} ↔ p_{i,:}
     for j = k+1 :  n
         ℓ_jk = u_jk / u_kk
         u_{j,k:n} = u_{j,k:n} - ℓ_jk u_{k,k:n}
     end
end
U = triu(U)
```

Operations count:

- $\mathcal{O}\left(\frac{2}{3}n^3\right)$ flops, flops count of partial pivoting?

General Procedure of Direct Methods

1. compute the LU factorization with partial pivoting, $PA = LU$, $\mathcal{O}(\frac{2}{3}n^3)$ flops
2. solve $Lz = Pb$ using forward substitution, $\mathcal{O}(n^2)$ flops
3. solve $Ux = z$ using backward substitution, $\mathcal{O}(n^2)$ flops

Variant of LU Factorization: LDU Factorization

For LU factorization with partial pivoting $PA = LU$

- denote $D = \mathrm{diag}(u_{11}, u_{22}, \cdots, u_{nn})$
- $\bar{U} = D^{-1}U$: upper-triangular matrix with unit diagonal entries, i.e., $\bar{u}_{ij} = u_{ij}/u_{ii}$ for $i \leq j$

Then $PA = LD\bar{U}$ gives an LDU factorization of A