Chapter 2

Solving Linear Equations

Po-Ning Chen, Professor

Department of Computer and Electrical Engineering

National Chiao Tung University

Hsin Chu, Taiwan 30010, R.O.C.

2.1 Vectors and linear equations

- What is this course **Linear Algebra** about?

Algebra (代數)

> *The part of mathematics in which letters and other general symbols are used to represent numbers and quantities in formulas and equations.*

Linear Algebra (線性代數)

> *To combine these algebraic symbols (e.g., vectors) in a linear fashion.*

So, we will not combine these algebraic symbols in a **nonlinear** fashion in this course!

Example of nonlinear equations for $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$: $\begin{cases} x_1 x_2 = 1 \\ x_1/x_2 = 1 \end{cases}$

Example of linear equations for $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$: $\begin{cases} x_1 + x_2 = 4 \\ x_1 - x_2 = 1 \end{cases}$

2.1 Vectors and linear equations

The linear equations can always be represented as *matrix operation*, i.e.,

$$A\boldsymbol{x} = \boldsymbol{b}.$$

Hence, the central problem of linear algorithm is to solve a system of **linear equations**.

Example of linear equations. $\begin{cases} x - 2y = 1 \\ 3x + 2y = 11 \end{cases} \Leftrightarrow \begin{bmatrix} 1 & -2 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \end{bmatrix}$ □

- A *linear equation problem* can also be viewed as a *linear combination problem* for column vectors (as referred by the textbook as the **column picture** view). In contrast, the original linear equations (the red-color one above) is referred by the textbook as the **row picture** view.

$$\text{Column picture view:} \quad x \begin{bmatrix} 1 \\ 3 \end{bmatrix} + y \begin{bmatrix} -2 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 11 \end{bmatrix}$$

We want to find the scalar coefficients of column vectors $\begin{bmatrix} 1 \\ 3 \end{bmatrix}$ and $\begin{bmatrix} -2 \\ 2 \end{bmatrix}$ to form another vector $\begin{bmatrix} 1 \\ 11 \end{bmatrix}$.

2.1 Matrix equation

From now on, we will focus on the **matrix equation** because it can be easily extended to higher dimension (≥ 2).

$$A\boldsymbol{x} = \boldsymbol{b}.$$

In terminologies,

- $A\boldsymbol{x} = \boldsymbol{b}$ will be referred to as **matrix equation**.
- A will be referred to as **coefficient matrix**.
- $\boldsymbol{x}$ will be referred to as **unknown vector**.

More terminologies:

The elements of a matrix A as well as its size are indexed first by row, and then by column.

Example.

$$A_{2\times 3} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$$

2.1 Other representations of matrix

In MATLAB:

MATLAB ("Matrix Laboratory") is a numerical computing environment, innovated by Cleve Moler, and developed by MathWorks. It allows matrix manipulation, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs in other languages.

- A matrix

$$A_{2,3} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & -2 \end{bmatrix}$$

is represented by

```
A=[1 2 3 ; 2 5 -2]   % Again, row first, and then column.
                     % Rows are separated by semicolon ";"
```

- A vector $\boldsymbol{x} = \begin{bmatrix} 1 & -1 & 1 \end{bmatrix}'$ is inputed as

```
x=[1 ; -1 ; 1]
```

2.1 Other representations of matrix

- The matrix multiplication $\boldsymbol{b} = A\boldsymbol{x}$ is then given by

```
b=A*x
```

Note: The textbook refers to this "$*$" as **dot product** (equivalently, **inner product**) in MATLAB, which is not entirely correct. MATLAB treated this "$*$" as the usual **matrix multiplication**.

- An element of a matrix A can be extracted by row and column indexes.

`A(1,2)` % This is exactly $a_{1,2}$ in $A_{2,3} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix}$.

- The row vector of a matrix A can be extracted by row index.

`A(1,:)` % This is exactly $\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \end{bmatrix}$ in $A_{2,3}$.

- The submatrix of a matrix A can be extracted by row and column index ranges.

`A(1:2,1:2)` % This is exactly $\begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$ in $A_{2,3}$.

2.1 Other representations of matrix

- The matrix multiplication $\boldsymbol{b} = A\boldsymbol{x}$ can then be rewritten by two forms:

```
                    % Inner product with row vectors
b=[A(1,:)*x ; A(2,:)*x]
                    % Linear combination of column vectors
b=A(:,1)*x(1)+A(:,2)*x(2)+A(:,3)*x(3)
```

- The above MATLAB rules for matrix manipulation can be easily extended:

`A(:,2)=[]`	% Eliminate the second column.
`A(1:2,:)=[]`	% Eliminate the first two rows.
`A=[A;` $a_{3,1}$ $a_{3,2}$ $a_{3,3}$`]`	% Add a third row to matrix A.
`v=[3:5]'`	% $\boldsymbol{v} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$

2.1 Other representations of matrix

- Some useful MATLAB functions appear in textbook Problems.

`A=eye(3)`	% Generate a 3-by-3 identity matrix $A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
`A=ones(3)`	% Generate a 3-by-3 all-one matrix $A = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
`A=zeros(3)`	% Generate a 3-by-3 all-zero matrix $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
`v=ones(3,1)`	% Generate a 3-by-1 all-one vector $\boldsymbol{v} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
`A=zeros(3,1)`	% Generate a 3-by-1 all-zero vector $A = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$

2.2 The idea of elimination

- Is there a **systematic** method to solve the **linear equations**?

Answer: Forward elimination and back(ward) substitution.

Example. $\begin{cases} x - 2y = 1 \\ 3x + 2y = 11 \end{cases}$

Step 1: Forward elimination.

Eliminate in a forward sequential order, i.e., x, y, ...

$$\begin{cases} x - 2y = 1 & \ldots(1) \\ 3x + 2y = 11 & \ldots(2) \end{cases}$$

$$\Rightarrow \begin{cases} x - 2y = 1 & \ldots(1) \\ (3x - 3x) + (2y - (-6y)) = (11 - 3) & \ldots(2) - \underbrace{3}_{\text{Multiplier}} \times (1) \end{cases}$$

$$\Rightarrow \begin{cases} x \; - \; 2y \; = \; 1 & \ldots(1) \\ 8y \; = \; 8 & \ldots(2) - 3 \times (1) \end{cases}$$

Step 2: Back(ward) substitution.

$\boxed{\text{Substitute in a backward sequential order, i.e., } \ldots, y, x.}$

$$\begin{cases} x - 2y = 1 \ \ldots (i) \\ \qquad 8y = 8 \ \ldots (ii) \end{cases}$$

$$\Rightarrow \begin{cases} x - 2y = 1 \ \ldots (i) \\ \qquad y = 1 \end{cases}$$

$$\Rightarrow \begin{cases} x - 2 = 1 \ \ldots \boxed{y = 1} \xrightarrow{\text{substitude}} (i) \\ \qquad y = 1 \end{cases}$$

$$\Rightarrow \begin{cases} x \qquad = 3 \\ \qquad y = 1 \end{cases}$$

- After forward elimination, the equations form an **upper triangle**.

$$\begin{cases} x - 2y = 1 \\ \qquad 8y = 8 \end{cases}$$

- In terminology, the element (e.g., x in the first equation) that is used to eliminate the same unknown in the remaining equations is called **pivot**. The **pivots** are on the **diagonal** of the upper triangle after elimination.

2.2 Failure of forward elimination

- Can the systematic method fail?

Answer: If $A\boldsymbol{x} = \boldsymbol{b}$ has no solution or has infinitely many solutions, then some **pivots** will **disappear** during the process of **forward elimination**.

Example.

$$\begin{cases} x - 2y = 1 & \ldots(1) \\ 2x - 4y = 11 & \ldots(2) \end{cases}$$

$$\Rightarrow \begin{cases} x - 2y = 1 & \ldots(1) \\ (2x - 2x) + (-4y - (-4y)) = (11 - 2) & \ldots(2) - \underbrace{2}_{\text{Multiplier}} \times (1) \end{cases}$$

$$\Rightarrow \begin{cases} x \;-\; 2y \;=\; 1 & \ldots(1) \\ \qquad\quad 0 \;=\; 9 & \ldots(2) - 2 \times (1) \Rightarrow \boxed{\text{No solution since } 0 \neq 9!} \end{cases}$$

Hence, the next **pivot** disappears (or was canceled). $\square$

If obtaining $0 = 0$, then there are infinitely many solutions for the linear equations.

2.2 Failure of forward elimination

- Can the systematic method fail if the solution exists and is unique?

Answer: Yes, if the **sequence** for forward elimination is not **properly ordered**.

Example. Sequence of choosing **pivots** in forward elimination $\{x$ from (1), y from (2), z from (3)$\}$.

$$\left\{\begin{array}{rcrcrcrl} x & - & 2y & + & z & = & 1 & \ldots(1) \\ 2x & - & 4y & + & z & = & 11 & \ldots(2) \\ x & + & y & - & z & = & 22 & \ldots(3) \end{array}\right.$$

This has a unique solution $(x, y, z) = (12, 1, -9)$.

$$\Rightarrow \left\{\begin{array}{rcrcrcrl} x & - & 2y & + & z & = & 1 & \ldots(1) \\ & & & - & z & = & 9 & \ldots(2) - 2 \times (1) \\ & & 3y & - & 2z & = & 21 & \ldots(3) - 1 \times (1) \end{array}\right.$$

$\Rightarrow$ **No y from (2) can be found!**

$\square$

In such case, one can simply switch (2) with any of the remaining equations (e.g., (3)) of which its y still exists (i.e., has a coefficient other than zero).

2.3 Forward elimination/back substitution in matrix form

- The forward elimination/back substitution can be easily done in matrix form.

Example. The **pivots** are the **diagonal** elements.

$$\begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix} \quad \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) \\ \ldots(3) \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} \quad \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) - 2 \times (1) \\ \ldots(3) - (-1) \times (1) \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} \quad \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{4} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix} \quad \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') - 1 \times (2') \end{matrix}$$

Based on the above, back substitution can be straightforwardly followed. □

The above procedure transforms $A\boldsymbol{x} = \boldsymbol{b}$ into $U\boldsymbol{x} = \boldsymbol{c}$, where U is an **upper triangular matrix**.

2.3 Forward elimination/back substitution in matrix form

We summarize the algorithmic procedure of **forward elimination** for $n \times n$ matrix A and n unknowns as follows.

- Use the first **diagonal** element as the first **pivot** to create zeros below it.
  - If the diagonal element is zero, switch the row with any row below it, for which its respective element at the same column is non-zero.
- Use the new second **diagonal** element as the second **pivot** to create zeros below it.
  - If the diagonal element is zero, switch the row with any row below it, for which its respective element at the same column is non-zero.
- Use the new third **diagonal** element as the third **pivot** to create zeros below it.
  - If the diagonal element is zero, switch the row with any row below it, for which its respective element at the same column is non-zero.
- $\ldots$
- Repeat the above procedure until either an upper-triangular U is resulted or the next pivot does not exist even with row switching.

2.3 Forward elimination/back substitution in matrix form

This algorithmic procedure can be graphically illustrated as:

$$\begin{bmatrix} \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \end{bmatrix} \Rightarrow \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & \blacksquare & \blacksquare & \cdots & \blacksquare \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \blacksquare & \blacksquare & \cdots & \blacksquare \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & 0 & \blacksquare & \cdots & \blacksquare \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \blacksquare & \cdots & \blacksquare \end{bmatrix} \Rightarrow \begin{bmatrix} \blacksquare & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & \blacksquare & \blacksquare & \cdots & \blacksquare \\ 0 & 0 & \blacksquare & \cdots & \blacksquare \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \blacksquare \end{bmatrix}$$

Terminology:

When a set of linear equations has either no solution or infinite many solutions, it is often referred to as a **singular** system.

Reminder of terminologies (in English)

- **Sigma notation**: The dot or inner product can be expressed using the so-called "sigma notation".

$$\boldsymbol{v} \cdot \boldsymbol{w} = \sum_{j=1}^{n} v_j w_j.$$

- m **by** n **matrix**: The terminology to describe the size of a matrix with m rows and n columns.

- **Entry of a matrix**: $a_{i,j}$ in the ith row and jth column of matrix A is referred to as an entry of A.

- **Component of a vector**: v_j in a vector $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}'$ is referred to as a component of $\boldsymbol{a}$.

2.3 The matrix form of forward elimination

- The **forward elimination** process can be implemented by multiplying the **elimination matrix**.

Example. The **pivots** are the **diagonal** elements.

$$\begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix} \quad \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) \\ \ldots(3) \end{matrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 1 & 1 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 1 & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} \quad \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) - 2 \times (1) \\ \ldots(3) - (-1) \times (1) \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix} \quad \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') \end{matrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{4} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix} \quad \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') - 1 \times (2') \end{matrix}$$

Based on the above, back substitution can be straightforwardly followed. $\square$

2.3 The matrix form of forward elimination

The forward elimination process can then be implemented by multiplying the elimination matrices $E_{2,1}$, $E_{3,1}$ and $E_{3,2}$ in sequence.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}}_{E_{3,2}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}}_{E_{3,1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{2,1}} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix} \boldsymbol{x}$$

$$= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}}_{E_{3,2}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}}_{E_{3,1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{2,1}} \begin{bmatrix} 2 \\ 8 \\ 10 \end{bmatrix}$$

which immediately gives

$$\begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{4} \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} 2 \\ 4 \\ 8 \end{bmatrix}.$$

□

For matrix multiplications, **associative law** is true (e.g., $(E_{3,1}E_{2,1})A = E_{3,1}(E_{2,1}A)$) but **commutative law** is false (e.g., $EA \neq AE$).

2.3 The matrix form of forward elimination

- We now understand why the **forward elimination** process can be implemented by multiplying the **elimination matrix**.
  - What if a row exchange is needed (because the next diagonal element is zero) during the process?

Answer: Use the **row exchange matrix**. Specifically, insert the row exchange matrix inbetween the elimination matrices.

* Denote by $P_{i,j}$, the **row exchange matrix** that exchanges rows i and row j by left-multiplying it.
* $P_{i,j}$ is an identity matrix with row i and j reversed. For example,

$$P_{2,3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{array}{l} \\ \longleftarrow \text{The original row 3.} \\ \longleftarrow \text{The original row 2.} \end{array}$$

2.3 The matrix form of forward elimination

Example. The **pivots** are the **diagonal** elements.

$$\begin{bmatrix}\mathbf{2} & 4 & -2\\ 4 & \boxed{8} & -3\\ -2 & -3 & 7\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}2\\8\\10\end{bmatrix}\quad\begin{matrix}\ldots(1)\text{ with }\mathbf{2}\text{ the first pivot}\\ \ldots(2)\text{ (Replace the middle 9 by 8)}\\ \ldots(3)\end{matrix}$$

$$\begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{2} & 4 & -2\\ 4 & 8 & -3\\ -2 & -3 & 7\end{bmatrix}$$

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -(-1) & 0 & 1\end{bmatrix}\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & 0 & 1\\ -2 & -3 & 7\end{bmatrix}$$

$$\Leftrightarrow\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & 0 & 1\\ 0 & 1 & 5\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}2\\4\\12\end{bmatrix}\quad\begin{matrix}\ldots(1)\text{ with }\mathbf{2}\text{ the first pivot}\\ \ldots(2)-2\times(1)\\ \ldots(3)-(-1)\times(1)\end{matrix}$$

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & 0 & 1\\ 0 & 1 & 5\end{bmatrix}$$

$$\Leftrightarrow\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & \mathbf{1} & 5\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}2\\12\\4\end{bmatrix}\quad\begin{matrix}\ldots(1)\\ \ldots(2')\text{ with }\mathbf{1}\text{ the second pivot}\\ \ldots(3')\end{matrix}$$

$$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -0 & 1\end{bmatrix}\begin{bmatrix}2 & 4 & -2\\ 0 & \mathbf{1} & 5\\ 0 & 0 & 1\end{bmatrix}$$

$$\Leftrightarrow\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & \mathbf{1} & 5\\ 0 & 0 & \mathbf{1}\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}=\begin{bmatrix}2\\12\\4\end{bmatrix}\quad\begin{matrix}\ldots(1)\\ \ldots(2')\text{ with }\mathbf{1}\text{ the second pivot}\\ \ldots(3')-0\times(2')\end{matrix}$$

Based on the above, back substitution can be straightforwardly followed. $\square$

2.3 The matrix form of forward elimination

The forward elimination process can then be implemented by multiplying $E_{2,1}$, $E_{3,1}$, $P_{2,3}$ and $E_{3,2}$ in sequence.

$$\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -0 & 1\end{bmatrix}}_{E_{3,2}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}}_{P_{2,3}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -(-1) & 0 & 1\end{bmatrix}}_{E_{3,1}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}_{E_{2,1}}\begin{bmatrix}2 & 4 & -2\\ 4 & 8 & -3\\ -2 & -3 & 7\end{bmatrix}\boldsymbol{x}$$

$$= \underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -0 & 1\end{bmatrix}}_{E_{3,2}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}}_{P_{2,3}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -(-1) & 0 & 1\end{bmatrix}}_{E_{3,1}}\underbrace{\begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}}_{E_{2,1}}\begin{bmatrix}2\\ 8\\ 10\end{bmatrix}$$

which immediately gives

$$\begin{bmatrix}\mathbf{2} & 4 & -2\\ 0 & \mathbf{1} & 5\\ 0 & 0 & \mathbf{1}\end{bmatrix}\boldsymbol{x} = \begin{bmatrix}2\\ 12\\ 4\end{bmatrix}.$$

□

2.3 The augmented matrix for forward elimination

- Since we need to multiply both A and $\boldsymbol{b}$ by $E = E_{3,2}P_{2,3}E_{3,1}E_{2,1}$ in order to solve $A\boldsymbol{x} = \boldsymbol{b}$, e.g.,

$$\underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & -0 & 1\end{bmatrix}}_{E_{3,2}}\underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}}_{P_{2,3}}\underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\-(-1) & 0 & 1\end{bmatrix}}_{E_{3,1}}\underbrace{\begin{bmatrix}1 & 0 & 0\\-2 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{E_{2,1}}\begin{bmatrix}2 & 4 & -2\\4 & 8 & -3\\-2 & -3 & 7\end{bmatrix}\boldsymbol{x}$$

$$= \underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & -0 & 1\end{bmatrix}}_{E_{3,2}}\underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 0 & 1\\0 & 1 & 0\end{bmatrix}}_{P_{2,3}}\underbrace{\begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\-(-1) & 0 & 1\end{bmatrix}}_{E_{3,1}}\underbrace{\begin{bmatrix}1 & 0 & 0\\-2 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{E_{2,1}}\begin{bmatrix}2\\8\\10\end{bmatrix}$$

we can introduce a so-called **augmented matrix** $\begin{bmatrix}A & \boldsymbol{b}\end{bmatrix}$ so that $\boldsymbol{x}$ can be solved by directly computing

$$E\begin{bmatrix}A & \boldsymbol{b}\end{bmatrix} = E\begin{bmatrix}2 & 4 & -2 & 2\\4 & 9 & -3 & 8\\-2 & -3 & 7 & 10\end{bmatrix} = \begin{bmatrix}\mathbf{2} & 4 & -2 & 2\\0 & \mathbf{1} & 5 & 12\\0 & 0 & \mathbf{1} & 4\end{bmatrix}.$$

- If we place $E_{2,1}$ on the right, what would happen then?

Answer: Row operations will become column operations. For example,

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ 0 & 0 & 1 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\text{where } \begin{cases} (\text{row } 1) = (\text{row } 1) \\ (\text{row } 2) = (\text{row } 2) - 2 \times (\text{row } 1) \\ (\text{row } 3) = (\text{row } 3) \end{cases}$$

becomes

$$\begin{bmatrix} 2 & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -6 & 4 & -2 \\ -12 & 8 & -3 \\ 4 & -3 & 7 \end{bmatrix}$$

$$\text{where } \begin{cases} (\text{column } 2) = (\text{column } 2) \\ (\text{column } 1) = (\text{column } 1) - 2 \times (\text{column } 2) \\ (\text{column } 3) = (\text{column } 3) \end{cases}$$

Exchange
$1 \leftrightarrow 2$
row$\leftrightarrow$column

- If we place $P_{2,3}$ on the right, what would happen then?

Answer: Row operations will become column operations. For example,

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix} = \begin{bmatrix} 2 & 4 & -2 \\ -2 & -3 & 7 \\ 4 & 8 & -3 \end{bmatrix}$$

$$\text{where } \begin{cases} (\text{row } 1) = (\text{row } 1) \\ (\text{row } 2) = (\text{row } 3) \\ (\text{row } 3) = (\text{row } 2) \end{cases}$$

becomes

$$\begin{bmatrix} 2 & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & -2 & 4 \\ 4 & -3 & 8 \\ -2 & 7 & -3 \end{bmatrix}$$

Exchange
$2 \leftrightarrow 3$
row$\leftrightarrow$column

$$\text{where } \begin{cases} (\text{column } 1) = (\text{column } 1) \\ (\text{column } 3) = (\text{column } 2) \\ (\text{column } 2) = (\text{column } 3) \end{cases}$$

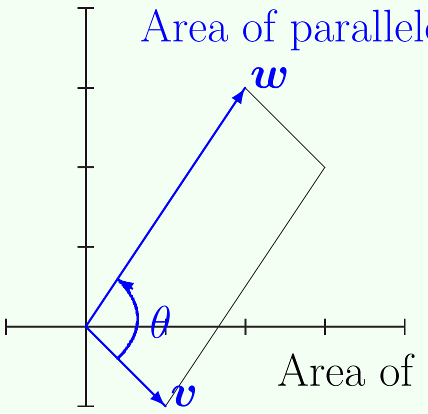
- The below definition gives what is required by *Problem 8*.

(Problem 8, Section 2.3) The **determinant** of $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ is $\texttt{det}(M) = ad - bc$. Subtract ℓ times row 1 from row 2 (i.e., row 2 - $\ell \times$ row 1) to produce a new M^*. Show that $\texttt{det}(M^*) = \texttt{det}(M)$ for every ℓ. When $\ell = c/a$, *the product of pivots equals the determinant*: $(a)(d - \ell b)$ equals $ad - bc$.

2.3 Appendix

Definition (Determinant 決定因子): A determinant, often denoted as $\texttt{det}(A)$ or simply $\texttt{det}A$, is a determinant factor associated with a square matrix. For example, for a matrix formed by $\begin{bmatrix} \boldsymbol{v}' \\ \boldsymbol{w}' \end{bmatrix}$, where $\boldsymbol{v}$ and $\boldsymbol{w}$ are 2-dimensional (column) vectors, the determinant is the (signed) area of the parallelogram formed by $\boldsymbol{v}$ and $\boldsymbol{w}$, and is given by

$$\texttt{det}A = \texttt{det}\begin{bmatrix} v_1 & v_2 \\ w_1 & w_2 \end{bmatrix} = \begin{vmatrix} v_1 & v_2 \\ w_1 & w_2 \end{vmatrix} = v_1 w_2 - v_2 w_1$$



$$\begin{aligned}\text{Area of parallelogram} &= \|\boldsymbol{v}\|\|\boldsymbol{w}\|\sin(\theta) \quad (\theta:\ \text{from } \boldsymbol{v} \text{ to } \boldsymbol{w} \text{ counterclockwisely}) \\ &= \|\boldsymbol{v}\|\|\boldsymbol{w}\|\sqrt{1-\cos^2(\theta)} = \|\boldsymbol{v}\|\|\boldsymbol{w}\|\sqrt{1-\left(\frac{\boldsymbol{v}\cdot\boldsymbol{w}}{\|\boldsymbol{v}\|\|\boldsymbol{w}\|}\right)^2} \\ &= \sqrt{\|\boldsymbol{v}\|^2\|\boldsymbol{w}\|^2-(\boldsymbol{v}\cdot\boldsymbol{w})^2}\end{aligned}$$

$$\text{Area of parallelogram} = \|\boldsymbol{v}\|\|\boldsymbol{w}\|\sin(2\pi-\theta) \text{ if } A = \begin{bmatrix} w_1 & w_2 \\ v_1 & v_2 \end{bmatrix}$$

2.3 Appendix

- We can say more about determinant.
  - The determinant of a matrix formed by $\begin{bmatrix} \boldsymbol{v}' \\ \boldsymbol{w}' \\ \boldsymbol{u}' \end{bmatrix}$, where $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{u}$ are 3-dimensional (column) vectors, the determinant is the (signed) volumn of the parallelepiped formed by $\boldsymbol{v}$, $\boldsymbol{w}$ and $\boldsymbol{u}$.
  - ...

(We will introduce the determinant in great detail in Chapter 5.)

- The below definition gives what is required by *Problem 30*.

(Problem 30, Section 2.3) Write $M = \begin{bmatrix} 3 & 4 \\ 5 & 7 \end{bmatrix}$ as a product of many factors $A = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$.

(a) What matrix E subtracts row 1 from row 2 to make row 2 of EM smaller?

(b) What matrix F subtracts row 2 of EM from row 1 to reduce row 1 of FEM?

(c) Continue E's and F's until (many E's and F's) times (M) is (A or B).

(d) E and F are the inverses of A and B! Moving all E's and F's to the right side will give you the desired result $M =$ product of A's and B's.

This is possible for all matrices $M = \begin{bmatrix} a & b \\ c & d \end{bmatrix} > 0$ that have $ad - bc = 1$.

2.3 Appendix

Definition (Positivity of a matrix): A positive matrix is a matrix in which all the elements are greater than zero. It is often denoted by $A > 0$.

Particularly note that a **positive matrix** is different from a **positive definite** matrix, which we will introduce in Chapter 6.

2.4 Rules for matrix operations

- Let us revisit **matrix operations** and summarize the rules that they obey.
  - Matrix addition is simply a term-by-term addition.
  - The definition of matrix multiplication is repeated below.

Definition (Product of matrices): The product of two **matrices** is defined as the inner products of row vectors and column vectors respectively from the first matrix and the second matrix, counted from the left. Specifically as an example,

$$A_{2\times 3}B_{3\times 2} = \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{a}_1'\boldsymbol{b}_1 & \boldsymbol{a}_1'\boldsymbol{b}_2 \\ \boldsymbol{a}_2'\boldsymbol{b}_1 & \boldsymbol{a}_2'\boldsymbol{b}_2 \end{bmatrix}_{2\times 2} = \begin{bmatrix} \boldsymbol{a}_1\cdot\boldsymbol{b}_1 & \boldsymbol{a}_1\cdot\boldsymbol{b}_2 \\ \boldsymbol{a}_2\cdot\boldsymbol{b}_1 & \boldsymbol{a}_2\cdot\boldsymbol{b}_2 \end{bmatrix}_{2\times 2},$$

where for the two matrices on the extreme left of the above equation, "2" is the number of vectors, and "3" is the dimension of the vectors.

Note that the dimension of the vectors shall be the same, otherwise the inner product operation cannot be performed.

2.4 Rules for matrix operations

- Rules or laws that matrix operations should obey.

  - Commutative law for addition: $A + B = B + A$.
  - ~~Commutative law for multiplication:~~ $\cancel{AB = BA}$.
  - Distributive law from the left: $A(B + C) = AB + AC$.
  - Distributive law from the right: $(A + B)C = AC + BC$.
  - Distributive law with respect to scaler: $\alpha(A + B) = \alpha A + \alpha B$.
  - Associative law for addition: $(A + B) + C = A + (B + C)$.
  - Associative law for multiplication: $(AB)C = A(BC)$.
  - Power law: $\begin{cases} A^pA^q = A^{p+q}, & \text{if } p \text{ and } q \text{ are positive integers} \\ A^pA^q = A^{p+q}, & \text{if } p \text{ or } q \text{ is a negative integer, and } A^{-1} \text{ exists} \\ A^0 = I, & \text{if } A \text{ is a square matrix,} \\ & \text{ where } I \text{ is an identity matrix} \\ & \text{Note: We can talk about the product } AA \\ & \text{only when } A \text{ is a square matrix.} \end{cases}$

2.4 Rules for matrix operations

- The way to prove these rules may require the below two pictures.
  - *Column picture of matrix product*: From the definition of matrix product, we can see that

$$A_{2\times 3}B_{3\times 2} = \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \boldsymbol{b}_1 & \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \boldsymbol{b}_2 \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{a}_1'\boldsymbol{b}_1 & \boldsymbol{a}_1'\boldsymbol{b}_2 \\ \boldsymbol{a}_2'\boldsymbol{b}_1 & \boldsymbol{a}_2'\boldsymbol{b}_2 \end{bmatrix}_{2\times 2}$$

    or we can simply write

$$A_{2\times 3}B_{3\times 2} = A \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} A\boldsymbol{b}_1 & A\boldsymbol{b}_2 \end{bmatrix}.$$

    In summary, the jth column of product AB is given by $A\boldsymbol{b}_j$, where $\boldsymbol{b}_j$ is the jth column of B.

– *Row picture of matrix product*: From the definition of matrix product, we can see that

$$A_{2\times 3}B_{3\times 2} = \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1' \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} \\ \boldsymbol{a}_2' \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{a}_1'\boldsymbol{b}_1 & \boldsymbol{a}_1'\boldsymbol{b}_2 \\ \boldsymbol{a}_2'\boldsymbol{b}_1 & \boldsymbol{a}_2'\boldsymbol{b}_2 \end{bmatrix}_{2\times 2}$$

or we can simply write

$$A_{2\times 3}B_{3\times 2} = \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} B = \begin{bmatrix} \boldsymbol{a}_1' B \\ \boldsymbol{a}_2' B \end{bmatrix}.$$

In summary, the ith row of product AB is given by $\boldsymbol{a}_i' B$, where $\boldsymbol{a}_i'$ is the ith row of A.

2.4 Block manipulation of matrices

- The above *column* or *row picture* indicates that we can manipulate the matrix in **blocks**. For example, matrix B can be divided into two blocks when calculating the matrix product.

$$AB = A \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} A\boldsymbol{b}_1 & A\boldsymbol{b}_2 \end{bmatrix}.$$

- Recall that we can solve the linear equations using the augmented matrix $\begin{bmatrix} A & \boldsymbol{b} \end{bmatrix}$, which consists of two blocks of different sizes.

- To generalize the idea, we can represent matrix A in the form

$$A = \left[\begin{array}{cc|cc|cc} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \end{array}\right] = \begin{bmatrix} I & I & I \\ I & I & I \end{bmatrix}$$

2.4 Block manipulation of matrices

- We can treat **blocks** as **number entries** in matrix operation, which is usually simpler.

Example.

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} + \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} A_{1,1} + B_{1,1} & A_{1,2} + B_{1,2} \\ A_{2,1} + B_{2,1} & A_{2,2} + B_{2,2} \end{bmatrix}$$

and

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{bmatrix} B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} = \begin{bmatrix} A_{1,1}B_{1,1} + A_{1,2}B_{2,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \end{bmatrix}$$

2.4 Block manipulation of matrices

- By **block manipulation**, the product AB does not have to be calculated/represented in terms of **rows** of A and **columns** of B, but can be done in a vice versa way.

$$AB = \begin{bmatrix} \boldsymbol{a}_1' \\ \boldsymbol{a}_2' \end{bmatrix} \begin{bmatrix} \boldsymbol{b}_1 & \boldsymbol{b}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{a}_1'\boldsymbol{b}_1 & \boldsymbol{a}_1'\boldsymbol{b}_2 \\ \boldsymbol{a}_2'\boldsymbol{b}_1 & \boldsymbol{a}_2'\boldsymbol{b}_1 \end{bmatrix} \quad \text{Rpresentation based on Inner product}$$

$$= \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_1' \\ \boldsymbol{d}_2' \end{bmatrix} = \boldsymbol{c}_1\boldsymbol{d}_1' + \boldsymbol{c}_2\boldsymbol{d}_2' \quad \text{Representation based on Outer product}$$

The representation using inner product is more convenient in the sense that we can have an entry-wise identification of elements in $A_{m\times n}B_{n\times p}$. Hence, the (i, j)th entry of $A_{m\times n}B_{n\times p}$ is the inner product of row i of A and column j of B, i.e.,

$$\boldsymbol{a}_i'\boldsymbol{b}_j = \boldsymbol{a}_i \cdot \boldsymbol{b}_j.$$

Hence, we need to perform $m \times p$ **inner products** to obtain $A_{m\times n}B_{n\times p}$.

The representation using outer product can represent AB as sum of all outer products of column i of A and row j of B, i.e.,

$$A_{m\times n}B_{n\times p} = \begin{bmatrix} \boldsymbol{c}_1 & \boldsymbol{c}_2 & \cdots & \boldsymbol{c}_n \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_1' \\ \boldsymbol{d}_2' \\ \vdots \\ \boldsymbol{d}_n' \end{bmatrix} = \sum_{i=1}^{n} \boldsymbol{c}_i \boldsymbol{d}_i'.$$

Hence, we need to perform n **outer products** to obtain $A_{m\times n}B_{n\times p}$.

2.4 Block manipulation of matrices

- By **block manipulation**, the **forward elimination** can be done in a block-by-block fashion.

Example.

$$E_{2,1}F = \begin{bmatrix} 1 & 0 \\ -ca^{-1} & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} a & b \\ 0 & d - ca^{-1}b \end{bmatrix}$$

and

$$\begin{aligned} E_{2,1}F &= \left[\begin{array}{c|c} I & 0 \\ \hline -CA^{-1} & I \end{array}\right] \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] \\ &= \left[\begin{array}{c|c} A & B \\ \hline C - CA^{-1}A & D - CA^{-1}B \end{array}\right] \\ &= \left[\begin{array}{c|c} A & B \\ \hline 0 & D - CA^{-1}B \end{array}\right] \end{aligned}$$

The matrix $D - CA^{-1}B$ is important in linear algebra; hence, it gets itself a terminology **Scher complement**, named after Issai Scher.

2.4 Appendix

What is the (i, j)th entry of ABC?

Solution:

- The (i, j)th entry of AD is the **inner product** of row i of A and column j of D, i.e. $\boldsymbol{a}_i' \boldsymbol{d}_j$.
- The column j of $D = BC$ is the **matrix multiplication** of matrix B and column j of C, i.e., $\boldsymbol{d}_j = B\boldsymbol{c}_j$.
- Therefore, the answer is $\boldsymbol{a}_i' B \boldsymbol{c}_j = \begin{bmatrix} a_{i,1} & a_{i,2} & \cdots \end{bmatrix} B \begin{bmatrix} c_{1,j} \\ c_{2,j} \\ \vdots \end{bmatrix}$.

Exercise: What is the (i, j)th entry of $ABCD$?

2.4 Appendix

Exercise: Problems 27, 28 and 37 in textbook.

(Problem 27, Section 2.4) Show that the product of upper triangular matrices is always upper triangular:

$$AB = \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix} \begin{bmatrix} x' & x' & x' \\ 0 & x' & x' \\ 0 & 0 & x' \end{bmatrix} = \begin{bmatrix} & & \\ 0 & & \\ 0 & 0 & \end{bmatrix}.$$

Proof using dot products (*Row times column*) (Row 2 of A)$\cdot$(column 1 of B)=0. Which other dot products gives zeros?

Proof using full matrices (*Column times row*) Draw x's and 0's in (column 2 of A) times (row 2 of B). Also show (column 3 of A) times (row 3 of B).

(Problem 28, Section 2.4) Draw the cuts in A (2 by 3) and B (3 by 4) and AB to show how each of the four multiplications rules is really a block multiplication:

(1) Matrix A times columns of B. **Columns of $\boldsymbol{AB}$**

(2) Rows of A times the matrix B. **Rows of $\boldsymbol{AB}$**

(3) Rows of A times columns of B. **Inner products** (numbers in AB)

(3) Columns of A times rows of B. **Outer products** (matrices add to AB)

2.4 Appendix

(Problem 37, Section 2.4) To prove $(AB)C = A(BC)$, use the column vectors $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n$ of B. First suppose that C has only one column $\boldsymbol{c}$ with entries $c_1, \ldots, c_n$:
AB has columns $A\boldsymbol{b}_1, \ldots, A\boldsymbol{b}_n$ and then $(AB)\boldsymbol{c} = c_1 A\boldsymbol{b}_1 + \cdots + c_n A\boldsymbol{b}_n$.
$B\boldsymbol{c}$ has one column $c_1\boldsymbol{b}_1 + \cdots + c_n\boldsymbol{b}_n$ and then $A(B\boldsymbol{c})$ equals $A(c_1\boldsymbol{b}_1 + \cdots + c_n\boldsymbol{b}_n)$.
Linearity gives equality of those two sums. This proves $(AB)\boldsymbol{c} = A(B\boldsymbol{c})$. The same is true for all other ____ of C. Therefore $(AB)C = A(BC)$. Apply to inverses:
If $BA = I$ and $AC = I$, prove that the left-inverse B equals the right-inverse C.

2.5 Inverse matices

Definition (Inverse matrix): The matrix A is *invertible* if there exists a matrix A^{-1} such that

$$A^{-1}A = I \quad \text{and} \quad AA^{-1} = I.$$

- A is *invertible* only when A is a square matrix.
- A is *invertible* if, and only if, $A\boldsymbol{x} = \boldsymbol{b}$ has a unique solution.
  - This unique solution must be $A^{-1}\boldsymbol{b}$.
  - If A is *invertible*, then $A\boldsymbol{x} = \mathbf{0}$ cannot have a non-zero solution.
- $A_{n\times n}$ is *invertible* if, and ony if, forward elimination (with possibly row exchange) should give n non-zero pivots. (This test is what you can use now!)
- If A is *invertible* if, and only if, the determinant of A (i.e., $\texttt{det}(A)$) is non-zero. (We will come back to this in Chapter 5.)
  - Hence, for a square matrix, either both left and right inverses do not exist ($\texttt{det}(A) = 0$) or both left and right inverses exist ($\texttt{det}(A) \neq 0$).

2.5 Inverse matices

- The inverse matrix of A is unique.

Proof: If L and R are both the inverse of A, then

$$L = LI = L(AR) = (LA)R = IR = R.$$

$\square$

  - In fact, the above proof also verifies that for a square matrix A, its left inverse L always equals its right inverse R.
  - Again, it is not possible for a square matrix without a left (respectively, right) inverse to have a right (respectively, left) inverse.
  - Again, it is not possible for a square matrix to have distinct left and right inverses.

Lemma. Suppose A and B are both square matrices. Then, AB has an inverse if, and only if, both A and B have inverses.

Proof:

1. *if-part*: If both A and B have inverses, then

$$(AB)(B^{-1}A^{-1}) = A(BB^{-1})A^{-1} = AIA^{-1} = AA^{-1} = I.$$

   Hence, AB has an inverse $B^{-1}A^{-1}$.

2. *only-if-part*: Suppose AB has an inverse C. Then, it is not possible that A or B has no inverse because if A had no inverse (meaning that both left and right inverses of A do not exist), then

$$ABC = I$$

   implies that the square matrix A has a right inverse BC, a contradiction.

   On the other hand, if B has no inverse (meaning that both left and right inverses of B do not exist), then

$$CAB = I$$

   implies that the square matrix B has a left inverse CA, a contradiction.

□

2.5 How to find the inverse matrix of A

- How to find A^{-1} if it exists?

Answer: Gauss-Jordan method.

- What is **Gauss-Jordan method**?

Answer: To repeat *forward elimination* and *back substitution* n times for an n-by-n matrix A.

Example. Suppose A is a 3-by-3 matrix.

– Find the solutions $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ for

$$A\boldsymbol{x}_1 = \boldsymbol{e}_1 \triangleq \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad A\boldsymbol{x}_2 = \boldsymbol{e}_2 \triangleq \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{and} \quad A\boldsymbol{x}_3 = \boldsymbol{e}_3 \triangleq \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

– Then,

$$A\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} A\boldsymbol{x}_1 & A\boldsymbol{x}_2 & A\boldsymbol{x}_3 \end{bmatrix} = \begin{bmatrix} \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \end{bmatrix} = I,$$

which means $\boxed{A^{-1} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 \end{bmatrix}}$.

2.5 How to find the inverse matrix of A

- **Gauss** said "We can solve the three linear equations in one time by using the augmented matrix."

$$\begin{cases} A\boldsymbol{x}_1 = \boldsymbol{e}_1 \\ A\boldsymbol{x}_2 = \boldsymbol{e}_2 \\ A\boldsymbol{x}_3 = \boldsymbol{e}_3 \end{cases} \quad \Longrightarrow \quad \begin{bmatrix} A & \boldsymbol{e}_1 & \boldsymbol{e}_2 & \boldsymbol{e}_3 \end{bmatrix} = \begin{bmatrix} A & I \end{bmatrix}.$$

Example. Suppose $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$.

$$\begin{aligned}
\begin{bmatrix} A & I \end{bmatrix} &= \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 1 & 0 \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \ldots(1) \\ \ldots(2) \\ \ldots(3) \end{matrix} \\
&\Rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 & 0 \\ 0 & -1 & 2 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \ldots(1) \\ \ldots(2) - \left(-\frac{1}{2}\right) \times (1) \\ \ldots(3) - 0 \times (1) \end{matrix} \quad \begin{matrix} \ldots(1') \\ \ldots(2') \\ \ldots(3') \end{matrix} \\
&\Rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{matrix} \ldots(1') \\ \ldots(2') \\ \ldots(3') - \left(-\frac{2}{3}\right) \times (2') \end{matrix}
\end{aligned}$$

$\square$

2.5 How to find the inverse matrix of A

- What is the next step after obtaining

$$\left[\begin{array}{ccc|ccc} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{array}\right]$$

Answer: Back substitution to solve

$$\boldsymbol{x}_1 = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \end{bmatrix}, \quad \boldsymbol{x}_2 = \begin{bmatrix} x_{1,2} \\ x_{2,2} \\ x_{3,2} \end{bmatrix} \quad \text{and} \quad \boldsymbol{x}_3 = \begin{bmatrix} x_{1,3} \\ x_{2,3} \\ x_{3,3} \end{bmatrix}.$$

Example (Continue).

$$\boldsymbol{x}_1 = \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ x_{3,1} \end{bmatrix} = \begin{bmatrix} \frac{1-(-1)x_{2,1}-(0)x_{3,1}}{2} \\ \frac{1/2-(-1)x_{3,1}}{3/2} \\ \frac{1/3}{4/3} \end{bmatrix} = \begin{bmatrix} \frac{3}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix}$$

and

$$\boldsymbol{x}_2 = \begin{bmatrix} x_{1,2} \\ x_{2,2} \\ x_{3,2} \end{bmatrix} = \begin{bmatrix} \frac{0-(-1)x_{2,2}-(0)x_{3,2}}{2} \\ \frac{1-(-1)x_{3,2}}{3/2} \\ \frac{2/3}{4/3} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{bmatrix}$$

and

$$\boldsymbol{x}_3 = \begin{bmatrix} x_{1,3} \\ x_{2,3} \\ x_{3,3} \end{bmatrix} = \begin{bmatrix} \frac{0-(-1)x_{2,3}-(0)x_{3,3}}{2} \\ \frac{0-(-1)x_{3,3}}{3/2} \\ \frac{1}{4/3} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{3}{4} \end{bmatrix}$$

$\square$

2.5 How to find the inverse matrix of A

- **Jordan** said "We can also solve $\begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \boldsymbol{x}_3 \end{bmatrix}$ by **backward elimination**."

Example. (Continue)

$$\begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 & 0 \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \\ \ldots(2) \\ \ldots(3) \end{array}$$

$$\Rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \\ \ldots(2) - (-3/4) \times (3) = (2') \\ \ldots(3) \end{array}$$

$$\Rightarrow \begin{bmatrix} 2 & -1 & 0 & 1 & 0 & 0 \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{array}{l} \ldots(1) - (0) \times (3) = (1') \\ \ldots(2') \\ \ldots(3) \end{array}$$

$$\Rightarrow \begin{bmatrix} 2 & 0 & 0 & \frac{3}{2} & 1 & \frac{1}{2} \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{bmatrix} \begin{array}{l} \ldots(1') - (-2/3) \times (2') = (1'') \\ \ldots(2') \\ \ldots(3) \end{array}$$

Based on this new augmented matrix, we can solve $\boldsymbol{x}_1$, $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ simply by dividing the pivot in each row (i.e., to make the pivot equal to 1). The resultant matrix is said to be in its **reduced echelon form**.

$$\Rightarrow \begin{bmatrix} 1 & 0 & 0 & \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ 0 & 1 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & 0 & 1 & \frac{1}{4} & \frac{1}{2} & \frac{3}{4} \end{bmatrix} \begin{array}{l} \ldots (1'')/2 \\ \ldots (2')/(3/2) \\ \ldots (3)/(4/3) \end{array}$$

Hence,

$$A^{-1} = \begin{bmatrix} \frac{3}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{3}{4} \end{bmatrix}.$$

$\square$

Gauss-Jordan method:

Multiply $[A \;\; I]$ by A^{-1} to get $[I \;\; A^{-1}]$ using

forward elimination and **backward elimination**.

2.5 Terminologies

- **Row echelon form**: A form of matrices satisfying
  - Every all-zero row is below non-all-zero rows.
  - The leading coefficient (the first nonzero number from the left, also called the **pivot**) of a non-all-zero row is always strictly to the right of the leading coefficient of the row above it.

  Example. Upper triangular matrix obtained from forward elimination.

- **(row) Reduced echelon form**: A form of matrices in **row echelon form** with the leading coefficient being one and also being the only non-zero entry in the column.

  Example. $\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$

2.5 MATLAB revisited

In MATLAB:

- The Gauss-Jordan method can be performed as:

```
I = eye(n);              % Define the n-by-n identity matrix.
R = rref([A I]);         % Obtain the row reduced echelon form.
X = R(:, n+1:n+n);       % Pick up the last n columns to form A^{-1}.
```

- You can also obtain the inverse matrix directly:

```
X=inv(A)   % A has to be a square matrix.
```

- Question: If `inv()` has already fulfilled the purpose of finding inverse, why we need `rref()`?

2.5 Gauss-Jordan method revisited

Example. Find the right inverse $R_{3\times 2}$ of a non-square matrix $A_{2\times 3} = \begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \end{bmatrix}$. (Note that this cannot be done by `inv()`.)

$$\begin{cases} A \begin{bmatrix} r_{1,1} \\ r_{2,1} \\ r_{3,1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \\ A \begin{bmatrix} r_{1,2} \\ r_{2,2} \\ r_{3,2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases} \quad \Longrightarrow \quad \text{Form the augumented matrix } \begin{bmatrix} A & \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \end{bmatrix}$$

$$\Longrightarrow \quad \text{Find } E_{2\times 2} \text{ such that } \begin{bmatrix} EA & E\begin{bmatrix} 1 \\ 0 \end{bmatrix} & E\begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} I_{2\times 2} & E\boldsymbol{a}_3 & E\begin{bmatrix} 1 \\ 0 \end{bmatrix} & E\begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{bmatrix}$$

$$\Longrightarrow \quad \begin{cases} \begin{bmatrix} r_{1,1} \\ r_{2,1} \end{bmatrix} = E\begin{bmatrix} 1 \\ 0 \end{bmatrix} - E\boldsymbol{a}_3 r_{3,1} \\ \begin{bmatrix} r_{1,2} \\ r_{2,2} \end{bmatrix} = E\begin{bmatrix} 0 \\ 1 \end{bmatrix} - E\boldsymbol{a}_3 r_{3,2} \end{cases}$$

- The right inverse R is not unique!

2.5 Gauss-Jordan method revisited

Further suppose $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \end{bmatrix}$. Then,

$$\begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 1 & 0 \\ -1 & 2 & -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & -1 & 0 & 1 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 \end{bmatrix} \quad \ldots \text{Forward elimination}$$

$$\begin{bmatrix} 1 & \frac{2}{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 0 & 1 & 0 \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & -\frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 \end{bmatrix} \quad \ldots \text{Backward elimination}$$

$$\begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 2 & 0 & -\frac{2}{3} & \frac{4}{3} & \frac{2}{3} \\ 0 & \frac{3}{2} & -1 & \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \quad \ldots \text{Pivot normalization}$$

Hence,

$$E = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{2}{3} \end{bmatrix} \begin{bmatrix} 1 & \frac{2}{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & 1 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \end{bmatrix}.$$

2.5 Gauss-Jordan method revisited

The problem is reduced to

$$\begin{cases} \textcolor{blue}{E}A\begin{bmatrix} r_{1,1} \\ r_{2,1} \\ r_{3,1} \end{bmatrix} = \textcolor{blue}{E}\begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ \\ \textcolor{blue}{E}A\begin{bmatrix} r_{1,2} \\ r_{2,2} \\ r_{3,2} \end{bmatrix} = \textcolor{blue}{E}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \end{cases} \implies \begin{cases} \begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & -\frac{2}{3} \end{bmatrix}\begin{bmatrix} r_{1,1} \\ r_{2,1} \\ r_{3,1} \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} \\ \\ \begin{bmatrix} 1 & 0 & -\frac{1}{3} \\ 0 & 1 & -\frac{2}{3} \end{bmatrix}\begin{bmatrix} r_{1,2} \\ r_{2,2} \\ r_{3,2} \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \end{cases}$$

We can let $r_{3,1} = r_{3,2} = 0$ and obtain

$$\begin{cases} \begin{bmatrix} r_{1,1} \\ r_{2,1} \end{bmatrix} = E\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{2}{3} \\ \frac{1}{3} \end{bmatrix} \\ \begin{bmatrix} r_{1,2} \\ r_{2,2} \end{bmatrix} = E\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \end{cases}$$

So, the right inverse is $R = \begin{bmatrix} E \\ 0 \quad 0 \end{bmatrix}$.

$\square$

Question: Can we choose $r_{3,1} = r_{3,2} = 1$? Hint: $R = \begin{bmatrix} E - E\boldsymbol{a}_3\begin{bmatrix} r_{3,1} & r_{3,2} \end{bmatrix} \\ r_{3,1} \qquad r_{3,2} \end{bmatrix}$.

2.5 Gauss-Jordan method revisited

In MATLAB:

The Gauss-Jordan method can be performed as:

```
I = eye(2);                  % Define the 2-by-2 identity matrix.
A = [2 -1 0; -1 2 -1];       % Define A.
S = rref([A I]);             % Obtain the row reduced echelon form.
```

$$\% \; S = \begin{bmatrix} 1 & 0 & -\frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ 0 & 1 & -\frac{2}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

```
R = [S(:, 4:5); 0 0 ];       % Pick up the last 2 columns to form R.
```

Important summary:

- In the process of Guass-Jordan method, we solve and obtain directly the **right inverse** in the above example.
- Using the same technique, we may obtain the **left inverse**, which is the product of the forward elimination matrices, the backward elimination matrices and the pivot normalization matrix.

  In the above example, $A_{2\times 3}$ does not have the left inverse because no matrix $L_{3\times 2}$ can make $L_{3\times 2}A_{2\times 3} = I_{3\times 3}$.

2.5 Diagonal and tridiagonal matrices

- (1) The inverse matrix of a *diagonal* square matrix (if the inverse exists) is a *diagonal* square matrix.
  (2) The inverse matrix of a *tridiagonal* square matrix may however be a *dense* square matrix (since a diagonal matrix is also a tridiagonal matrix).

  Definition (Diagonal and tridiagonal matrices): A square matrix is a diagonal matrix if all the off-diagonal entries are zero. Likewise, a tridiagonal matrix is one that all the entries are zero except those in the main diagonal and two adjacent diagonals (which can be zeros).

  – Some may define the tridiagonal matrix as one that all entries in the main diagonal and two adjacent diagonals are non-zeros. In such case, its inverse is always dense.

  – It is also worth knowing that **"antidiagonal" entries** of a matrix A are $a_{1,n}, a_{2,n-1}, a_{3,n-2}, \ldots, a_{n,1}$.

  Definition (Dense matrix): A matrix is *dense* if all the entries are non-zeros.

2.5 Triangular matrix and singular matrix

- If L is lower triangular with 1's on the diagonal, so is its inverse.

  A square triangular matrix is invertible if, and only if, all its diagonals are nonzeros.

Definition (Singular matrix): A square matrix is called *singular* if it has no inverse.

2.5 Symmetric and determinant

- The inverse matrix of a *symmetric* matrix (if the inverse exists) is *symmetric*.

  Definition (Symmetric matrix): A square matrix is *symmetric* if its (i, j)th entry is equal to its (j, i)th entry for every i and j.

- Recall that the *determinant* is the signed area (volume, etc) of the parallelogram (parallelepiped, etc) formed by the column vectors of a matrix A.

  – The *Gauss-Jordan method* can also be used to determine the *determinant*, which is equal to the **product of all the pivots before normalization**.

Example. Suppose $A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$.

$$[A \;\; I] \overset{G.J.Method}{\Longrightarrow} \left[\begin{array}{ccc|ccc} 2 & 0 & 0 & \frac{3}{2} & 1 & \frac{1}{2} \\ 0 & \frac{3}{2} & 0 & \frac{3}{4} & \frac{3}{2} & \frac{3}{4} \\ 0 & 0 & \frac{4}{3} & \frac{1}{3} & \frac{2}{3} & 1 \end{array}\right] \Rightarrow \det(A) = 2 \times \frac{3}{2} \times \frac{4}{3} = 4.$$

2.6 Elimination = Factorization : $A = LU$

Definition (Factorization): *Factorization* is the decomposition of an object (e.g., a number or a polynomial or a matrix) into a product of other objects, named factors, such that when being multiplied together, one gets the original.

Example.

$$\begin{aligned}
&15 = 3 \times 5\\
&x^2 - 3x + 2 = (x-1)(x-2)\\
&A = LU
\end{aligned}$$

□

- **Triangular factorization:** How to find *lower triangular* L and *upper triangular* U such that $A = LU$?

Answer: **Gauss-Jordan method** (Actually, just **forward elimination**. Why? See the below example).

- Forward elimination produces a matrix E such that $EA = U$.
- The inverse of *lower triangular* E is also *lower triangular* $L = E^{-1}$.
- This immediately gives $A = E^{-1}U = LU$.

2.6 Elimination = Factorization : $A = LU$

Example.

$$\begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 4 & 9 & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) \\ \ldots(3) \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 9 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 1 & 1 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & 1 & 1 & -2 & 1 & 0 \\ 0 & 1 & 5 & 1 & 0 & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) - 2 \times (1) \\ \ldots(3) - (-1) \times (1) \end{array}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & -2 & 1 & 0 \\ 0 & 1 & 5 & 1 & 0 & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') \end{array}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & -2 & 1 & 0 \\ 0 & 0 & \mathbf{4} & 3 & -1 & 1 \end{bmatrix} \begin{array}{l} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') - 1 \times (2') \end{array}$$

2.6 Elimination = Factorization : $A = LU$

The forward elimination process can then be implemented by multiplying the elimination matrices $E_{2,1}$, $E_{3,1}$ and $E_{3,2}$ in sequence.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}}_{E_{3,2}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}}_{E_{3,1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{2,1}} \begin{bmatrix} 2 & 4 & -2 & 1 & 0 & 0 \\ 4 & 9 & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 1 & -2 & 1 & 0 \\ 0 & 0 & \mathbf{4} & 3 & -1 & 1 \end{bmatrix}$$

Then we have that after **forward elimination**

$$\boxed{E_{3,2}E_{3,1}E_{2,1}\begin{bmatrix} A & I \end{bmatrix} = \begin{bmatrix} EA & E \end{bmatrix} = \begin{bmatrix} U & E \end{bmatrix}}$$

where $E = E_{3,2}E_{3,1}E_{2,1}$.

Next, we need to determine $\boxed{L = E^{-1} = E_{2,1}^{-1}E_{3,1}^{-1}E_{3,2}^{-1}}$. This is easy because every $E_{i,j}$ is formed by "changing" just one off-diagonal entry from an identity matrix.

What we need to do is to "negate" the only non-zero off-diagonal entry!

$$\underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & -1 & 1\end{bmatrix}^{-1}}_{E_{3,2}^{-1}} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 1 & 1\end{bmatrix}, \quad \underbrace{\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -(-1) & 0 & 1\end{bmatrix}^{-1}}_{E_{3,1}^{-1}} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ (-1) & 0 & 1\end{bmatrix}$$

$$\text{and} \quad \underbrace{\begin{bmatrix}1 & 0 & 0\\ -2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}^{-1}}_{E_{2,1}^{-1}} = \begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$$

More surprisingly, when they are multiplied in reverse order, the product is simply to place all these non-zero entries at their respective position!

$$\boxed{L = E^{-1} = E_{2,1}^{-1} E_{3,1}^{-1} E_{3,2}^{-1} = \begin{bmatrix}1 & 0 & 0\\ 2 & 1 & 0\\ -1 & 1 & 1\end{bmatrix}}$$

□

Intuitions behind the above example

- Why the inverse of $E_{i,j}$ is simply to negate the multiplier of $E_{i,j}$?
  Answer: The inverse action of $(i) - \ell_{i,j} \times (j)$ is simply $(i) + \ell_{i,j}(j)$.

- Why the product of $E_{2,1}^{-1}E_{3,1}^{-1}E_{3,2}^{-1}$ is to place all the non-zero entries in their respective position?
  Answer:

  $$\begin{bmatrix} 1 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare \\ 0 & \blacksquare & \blacksquare \end{bmatrix}$$

  – $E_{3,1}^{-1}$ = only updates row 3 of $E_{3,2}^{-1}$ by row 1, i.e., $\boxed{(3') = (3) + \ell_{3,1} \times (1)}$.
  Hence,

  $$(3') = \begin{cases} (3), & \text{if } (1) = 0 \text{ (such as the } (1,2)\text{th and } (1,3)\text{th entries of } E_{3,2}^{-1}) \\ \ell_{3,1}, & \text{if } (3) = 0 \text{ and } (1) = 1 \end{cases}$$

Hint of a formal proof:

$$[I + (E_{3,1}^{-1} - I)][I + (E_{3,2}^{-1} - I)] = I + (E_{3,1}^{-1} - I) + (E_{3,2}^{-1} - I) + (E_{3,1}^{-1} - I)(E_{3,2}^{-1} - I) = I + (E_{3,1}^{-1} - I) + (E_{3,2}^{-1} - I).$$

  – Similar interpretation can be done for $E_{2,1}^{-1}$ onto $E_{3,1}^{-1}E_{3,2}^{-1}$.

We summarize the observation in the below lemma.

2.6 Elimination = Factorization : $A = LU$

Lemma: For an invertible matrix A, if no row exchange is necessary during forward elimination, then its *lower triangular factor* L and *upper triangular factor* U satisfy

- $A = LU$;
- L has 1 on all main diagonal entries, and multipliers below the diagonal entries.
  - When a row of A starts with r zeros, so does the rows of L (because the multipliers are zeros).
- U has (non-normalized) pivots on its main diagonal.
  - When a column of A starts with c zeros, so does the columns of U (because forward eliminations will not change these zeros).

Example. Suppose $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$.

2.6 Elimination = Factorization : $A = LU$

Then

$$\begin{bmatrix} \boldsymbol{a}_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix}$$

$$\Leftrightarrow \begin{bmatrix} \boldsymbol{a}_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} - (a_{2,1}/a_{1,1})a_{1,2} & a_{2,3} - (a_{2,1}/a_{1,1})a_{1,3} \\ 0 & a_{3,2} - (a_{3,1}/a_{1,1})a_{1,2} & a_{3,3} - (a_{3,1}/a_{1,1})a_{1,3} \end{bmatrix} \begin{array}{l} \ldots(1) \\ \ldots(2) - (a_{2,1}/a_{1,1}) \times (1) \\ \ldots(3) - (a_{3,1}/a_{1,1}) \times (1) \end{array}$$

Then, we know

$$L = \begin{bmatrix} 1 & 0 & 0 \\ a_{2,1}/a_{1,1} & 1 & 0 \\ a_{3,1}/a_{1,1} & ? & 1 \end{bmatrix} \text{ and } U = \begin{bmatrix} \boldsymbol{a}_{1,1} & a_{1,2} & a_{1,3} \\ 0 & a_{2,2} - (a_{2,1}/a_{1,1})a_{1,2} & a_{2,3} - (a_{2,1}/a_{1,1})a_{1,3} \\ 0 & 0 & ? \end{bmatrix}$$

□

Remember the formula of multiplier: $\ell_{2,1} = a_{2,1}/a_{1,1}$ and $\ell_{3,1} = a_{3,1}/a_{1,1}$.

By LU decomposition, $A\boldsymbol{x} = \boldsymbol{b}$ can be solved by

$$\begin{cases} L\boldsymbol{c} = \boldsymbol{b} \text{ (forward substitution)} \\ U\boldsymbol{x} = \boldsymbol{c} \text{ (back substitution)} \end{cases}$$

2.6 Elimination = Factorization : $A = LU$

- What if **row exchange** is needed during **forward elimination**?

Example.

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 0 & 1 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 0 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & \mathbf{1} & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 4 & \boxed{8} & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) \text{ (Replace the middle 9 by 8)} \\ \ldots(3) \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 1 & 5 & 1 & 0 & 1 \end{bmatrix} \begin{matrix} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) - 2 \times (1) \\ \ldots(3) - (-1) \times (1) \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & 1 & -2 & 1 & 0 \end{bmatrix} \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') \end{matrix}$$

$$\Leftrightarrow \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{1} & -2 & 1 & 0 \end{bmatrix} \begin{matrix} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') - 0 \times (2') \end{matrix}$$

2.6 Elimination = Factorization : $A = LU$

The forward elimination process can then be implemented by multiplying $E_{2,1}$, $E_{3,1}$, $P_{2,3}$ and $E_{3,2}$ in sequence.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0 & 1 \end{bmatrix}}_{E_{3,2}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_{2,3}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}}_{E_{3,1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{2,1}} \begin{bmatrix} 2 & 4 & -2 & 1 & 0 & 0 \\ 4 & 9 & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{1} & -2 & 1 & 0 \end{bmatrix}$$

Then we have after **forward elimination**

$$\boxed{E_{3,2}P_{2,3}E_{3,1}E_{2,1} \begin{bmatrix} A & I \end{bmatrix} = \begin{bmatrix} EA & E \end{bmatrix} = \begin{bmatrix} U & E \end{bmatrix}}$$

where $E = E_{3,2}P_{2,3}E_{3,1}E_{2,1}$.

Next, we need to determine $\boxed{L = E^{-1} = E_{2,1}^{-1}E_{3,1}^{-1}P_{2,3}^{-1}E_{3,2}^{-1}}$. This is easy because every E_{ij} is formed by "changing" just one off-diagonal entry from an identity matrix, and the inverse of $P_{2,3}$ is itself!

2.6 Elimination = Factorization : $A = LU$

What we need to do is to "negate" the only non-zero off-diagonal entry!

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}^{-1}}_{E_{3,2}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}^{-1}}_{E_{3,1}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ (-1) & 0 & 1 \end{bmatrix}$$

$$\text{and} \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}}_{E_{2,1}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{L = E^{-1} = E_{2,1}^{-1} E_{3,1}^{-1} P_{2,3}^{-1} E_{3,2}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \quad \textbf{Not a lower triangular matrix!}}$$

$\square$

2.6 Elimination = Factorization : $A = LU$

How can we guarantee there is no row exchanges necessary for finding $A = LU$?

- No row exchanges during forward elimination if, and only if, all the upper-left square sub-matrices of A are invertible. (See *Problem 24* on p. 106.)

(Problem 24, Section 2.6) Which invertible matrices allow $A = LU$ (elimination without row exchanges)? *Good question*! Look at each of the square upper left submatrices of A.

All upper left k by k submatrices A_k must be invertible (sizes $k = 1, \ldots, n$).

Explain that answer: A_k factors into ____ because $LU = \begin{bmatrix} L_k & 0 \\ * & * \end{bmatrix} \begin{bmatrix} U_k & * \\ 0 & * \end{bmatrix}$.

- In such case $A_k = L_k U_k$, where A_k (similarly, L_k and U_k) consists of the first k rows and the first k columns of A.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{2,1} & 1 & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ 0 & u_{2,2} & u_{2,3} \\ 0 & 0 & u_{3,3} \end{bmatrix}$$

2.6 Elimination = Triple Factorization : $A = LDU$

- Now assume again no row exchange is needed during forward elimination.
- Sometimes, we wish to make the diagonal entries of U being all ones. Hence, a diagonal matrix D is introduced.

Example.

$$\begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{4} \end{bmatrix} \Rightarrow \underbrace{\begin{bmatrix} \mathbf{2} & 0 & 0 \\ 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{4} \end{bmatrix}}_{D} \underbrace{\begin{bmatrix} \mathbf{1} & 2 & -1 \\ 0 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{1} \end{bmatrix}}_{U}$$

□

The advantage of **triple factorization** is that for symmetric matrix $A = LDU$, we have

$$L = U',$$

where U' is the "transpose" of matrix U (i.e., the (i, j)th entry of U' is the (j, i)th entry of U). Hence,

$$A = LDL'.$$

2.6 The cost of elimination

Question: Again, assume with no row exchanges during elimination. What will be the complexity of using forward elimination and back substitution to solve $\boldsymbol{Ax} = \boldsymbol{b}$?

Answer:

i) Forward elimination for matrix A,

$\times$ operations	exact	approximation
$a_{i,1} \to 0$ for $i = 2, \ldots, n$	$n(n-1)$	n^2
$a_{i,2} \to 0$ for $i = 3, \ldots, n$	$(n-1)(n-2)$	$(n-1)^2$
$\vdots$	$\vdots$	$\vdots$
$a_{i,n-1} \to 0$ for $i = n$	$2 \cdot 1$	2^2
total	$\displaystyle\sum_{i=2}^{n} i(i-1) = \frac{1}{3}n(n^2-1)$	$\frac{1}{3}n^3$

Note: For example, $n(n-1) = n$ operations/row $\times$ $(n-1)$ rows.

Note: The blue-colored numbers are used for step ii) in the next page, which requires for example, 1 operations/row $\times$ $(n-1)$ rows.

2.6 The cost of elimination

ii) Forward elimination for vector $\boldsymbol{b}$, we need $\sum_{i=2}^{n}(i-1)=\frac{1}{2}n(n-1)$ computations.

iii) Back substitution $U\boldsymbol{x}=\boldsymbol{c}$ requires

$\times$ or $/$ operations	exact
$x_n = c_n/u_{n,n}$	1
$x_{n-1} = (c_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1}$	2
$\vdots$	$\vdots$
$x_1 = \cdots$	n
total	$\displaystyle\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$

Hence, *ii*) and *iii*) require n^2 multiplications.

- Using MATLAB, the above algorithm requires about **1** second to solve $A\boldsymbol{x}=\boldsymbol{b}$ with $n=1000$.
- Doubling n makes the algorithm $2^3=8$ times slower.
  Example. $n=100\times 1000 \to 100^3$ seconds $=11.574$ days. $\square$

2.6 The cost of elimination

Question: How about the complexity for calculating A^{-1} (without row exchange) using Gauss-Jordan? Does it require $n \times \left(\frac{1}{3}n(n^2-1)+n^2\right)$ multiplications as we need to solve $\boldsymbol{A}\boldsymbol{x}_i = \boldsymbol{e}_i$ for $i = 1, \ldots, n$?

Answer:

i) Forward elimination for augmented matrix $\begin{bmatrix} A & I \end{bmatrix}$,

$\times$ operations	exact
$a_{i,1} \to 0$ for $i = 2, \ldots, n$	$2n(n-1)$
$a_{i,2} \to 0$ for $i = 3, \ldots, n$	$(2n-1)(n-2)$
$\vdots$	$\vdots$
$a_{i,n-1} \to 0$ for $i = n$	$(n+2) \cdot 1$
total	$\displaystyle\sum_{i=2}^{n}(n+i)(i-1) = \frac{1}{6}n(n-1)(5n+2)$

2.6 The cost of elimination

ii) Backward elimination for augmented matrix $\begin{bmatrix} A & I \end{bmatrix}$,

$\times$ operations	exact
$a_{i,n} \to 0$ for $i = n-1, \ldots, 1$	$(n+1)(n-1)$
$a_{i,n-1} \to 0$ for $i = n-2, \ldots, 1$	$(n+2)(n-2)$
$\vdots$	$\vdots$
$a_{i,2} \to 0$ for $i = 1$	$(2n-1) \cdot 1$
total	$\displaystyle\sum_{i=1}^{n-1}(2n-i)i = \frac{1}{6}n(n-1)(4n+1)$

iii) Pivot normalization: n^2 divisions.

Three steps in total, $\frac{1}{2}n(n-1)(3n+1) + n^2 = \frac{1}{2}n(3n^2-1) \approx \frac{3}{2}n^3$.

- We can also find the inverse of A by solving $A\boldsymbol{x}_i = \boldsymbol{e}_i$ for $i = 1, \ldots, n$ separately. In such case, the forward elimination part in slide 2-71 (step *i*)) can be shared among all n equations, while steps *ii*) and *iii*) in slide 2-72 (steps *ii*) and *iii*)) must be repeated n times. This yields $\frac{1}{3}n(n^2-1) + n \cdot n^2 \approx \frac{4}{3}n^3$ multiplications.

- Backward substitution is more efficient than backward elimination.

2.6 The cost of elimination

Again, assume with no row exchanges during elimination. What will be the complexity of using forward elimination and back substitution to solve $\boldsymbol{A\boldsymbol{x}} = \boldsymbol{b}$, if A is a band matrix with w non-zero diagonals both below and above the main diagonal?

Answer:

i) Forward elimination for matrix A,

$\times$ operations	exact	approximation
$a_{i,1} \to 0$ for $i = 2, \ldots, w+1$	$w \cdot w$	w^2
$a_{i,2} \to 0$ for $i = 3, \ldots, w+2$	$w \cdot w$	w^2
$\vdots$	$\vdots$	$\vdots$
$a_{i,n-w} \to 0$ for $i = n-w+1, \ldots, n$	$w \cdot w$	w^2
$\vdots$	$\vdots$	$\vdots$
$a_{i,n-1} \to 0$ for $i = n$	$1 \cdot 1$	w^2
total	$\displaystyle\sum_{i=1}^{w-1} i^2 + w^2(n-w)$ $= \frac{1}{6}w(6nw - 4w^2 - 3w + 1)$	nw^2

Note: The blue-colored numbers are used for step ii).

2.6 The cost of elimination

ii) Forward elimination for vector $\boldsymbol{b}$, we need $\sum_{i=1}^{w-1} i + (n-w)w = \frac{1}{2}w(2n-w-1)$ computations.

iii) Back substitution $U\boldsymbol{x} = \boldsymbol{c}$ requires

$\times$ operations	exact
$x_n = c_n/u_{n,n}$	1
$x_{n-1} = (c_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1}$	2
$\vdots$	$\vdots$
$x_{n-w} = (c_{n-w} - u_{n-w,n}x_n - \cdots)/u_{n-w,n-w}$	$w+1$
$\vdots$	$\vdots$
$x_1 = \cdots$	$w+1$
total	$\frac{1}{2}(w+1)(2n-w)$

Hence, *ii*) and *iii*) require $n^2 + 2nw - w(1+w)$ computations.

A band-$(2w+1)$ matrix A (if no row exchange is needed during forward elimination) satisfies that its factors L and U are both band matrices with bandwidth of $w+1$.

2.7 Transpose and permutation

Definition (Matrix transpose): The transpose A' of a matrix A is simply to exchange the rows and columns. Hence, the (i, j)th entry of A' is the (j, i)th entry of A.

Note: A matrix transpose is sometimes denoted by A' or A^{T}. For example,

- A' is used by MATLAB.
- A^{T} is adopted by the textbook. (We will use this notation in the sequel!)

Properties of transpose

- **Sum Property**: $(A + B)^{\mathsf{T}} = A^{\mathsf{T}} + B^{\mathsf{T}}$

  Proof: This can be proved by listing all the matrix entries of both sides. □

2.7 Transpose and permutation

- **Product Property**: $(AB)^{\mathsf{T}} = B^{\mathsf{T}}A^{\mathsf{T}}$

Proof: First, prove using **Sum Property** of Transpose.

For a matrix A and vector $\boldsymbol{b}$,

$$
\begin{aligned}
(A\boldsymbol{b})^{\mathsf{T}} &= \left(\begin{bmatrix}\boldsymbol{a}_1 & \boldsymbol{a}_2 & \cdots\end{bmatrix}\boldsymbol{b}\right)^{\mathsf{T}} \\
&= (b_1\boldsymbol{a}_1 + b_2\boldsymbol{a}_2 + \cdots)^{\mathsf{T}} = b_1\boldsymbol{a}_1^{\mathsf{T}} + b_2\boldsymbol{a}_2^{\mathsf{T}} + \cdots = \boldsymbol{b}^{\mathsf{T}}\begin{bmatrix}\boldsymbol{a}_1^{\mathsf{T}} \\ \boldsymbol{a}_2^{\mathsf{T}} \\ \vdots\end{bmatrix} = \boldsymbol{b}^{\mathsf{T}}A^{\mathsf{T}}
\end{aligned}
$$

Then,

$$
\begin{aligned}
(AB)^{\mathsf{T}} &= \left(A\begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \cdots\end{bmatrix}\right)^{\mathsf{T}} \\
&= \begin{bmatrix}A\boldsymbol{b}_1 & A\boldsymbol{b}_2 & \cdots\end{bmatrix}^{\mathsf{T}} = \begin{bmatrix}\boldsymbol{b}_1^{\mathsf{T}}A^{\mathsf{T}} \\ \boldsymbol{b}_2^{\mathsf{T}}A^{\mathsf{T}} \\ \vdots\end{bmatrix} = \begin{bmatrix}\boldsymbol{b}_1^{\mathsf{T}} \\ \boldsymbol{b}_2^{\mathsf{T}} \\ \vdots\end{bmatrix}A^{\mathsf{T}} = B^{\mathsf{T}}A^{\mathsf{T}}.
\end{aligned}
$$

$\square$

- **Inverse Property**: $(A^{\mathsf{T}})^{-1} = (A^{-1})^{\mathsf{T}}$

  Proof: $A^{-1}A = I \Rightarrow (A^{-1}A)^{\mathsf{T}} = I^{\mathsf{T}} = I$;
  hence, the **Product Property** of Transpose gives $A^{\mathsf{T}}(A^{-1})^{\mathsf{T}} = I$, which implies $(A^{-1})^{\mathsf{T}}$ is the inverse of A^{T}. $\Box$

2.7 Inner product revisited

Recall the difference between inner product and outer product (for real-valued vectors)

$$\text{Inner product} \quad \boldsymbol{v} \cdot \boldsymbol{w} = \boldsymbol{v}^{\mathtt{T}}\boldsymbol{w} = \begin{bmatrix} v_1 & v_2 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = v_1 w_1 + v_2 w_2$$

and

$$\text{Outer product} \quad \boldsymbol{v} \otimes \boldsymbol{w} = \boldsymbol{v}\boldsymbol{w}^{\mathtt{T}} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \begin{bmatrix} w_1 & w_2 \end{bmatrix} = \begin{bmatrix} v_1 w_1 & v_1 w_2 \\ v_2 w_1 & v2 w_2 \end{bmatrix}$$

To memorize it,

- for inner product, $^{\mathtt{T}}$ is placed "inner" (inbetween $\boldsymbol{v}$ and $\boldsymbol{w}$)
- for outer product, $^{\mathtt{T}}$ is placed "outer" (outside $\boldsymbol{v}$ and $\boldsymbol{w}$).

2.7 Inner product revisited

- The "dot product" we have introduced in Chapter 1 should be only termed as a "product" between two vectors, which is denoted by a "dot".
- It **happens** to satisfy the three axioms of an **inner product**.

Definition: A mapping from $\mathcal{V} \times \mathcal{V}$ to $\mathbb{F}$, denoted by $\langle \cdot, \cdot \rangle$, is an inner product if for every $x, y, z \in \mathcal{V}$ and $a \in \mathbb{F}$,

1. Positive-definiteness: $\langle x, x \rangle \geq 0$ with equality only when $x = 0$
2. Symmetry: $\langle x, y \rangle = \langle y, x \rangle$
3. Linearity: $\begin{cases} \langle ax, y \rangle = a\langle x, y \rangle \\ \langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle \end{cases}$

  – Vector dot product is an inner product with $\mathcal{V} = \mathbb{F} = \Re$ (set of all real numbers).

- It is then possible to parallel linear algebra with calculus along this conception.

2.7 Inner product revisited

Example. What is the "calculus" for "function transpose operation", such as in $(A\boldsymbol{x})^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{x}^{\mathsf{T}}(A^{\mathsf{T}}\boldsymbol{y})$, where A is the difference matrix?

- Inner product of two vectors can be expressed in terms of "transpose" as

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{y}.$$

- Hence, $(A\boldsymbol{x})^{\mathsf{T}}\boldsymbol{y} = \boldsymbol{x}^{\mathsf{T}}(A^{\mathsf{T}}\boldsymbol{y})$ can be expressed as

$$\langle A\boldsymbol{x}, \boldsymbol{y} \rangle = \langle \boldsymbol{x}, A^{\mathsf{T}}\boldsymbol{y} \rangle.$$

- "Inner product" of two real functions can be defined as

$$\langle x, y \rangle \triangleq \int_{-\infty}^{\infty} x(t)y(t)dt.$$

- We then reasonably guess the calculus counterpart of difference matrix A is derivative. I.e., we guess

$$\left\langle \left(\frac{\partial}{\partial t}\right) x, y \right\rangle = \left\langle x, \left(\frac{\partial}{\partial t}\right)^{\mathsf{T}} y \right\rangle.$$

  Question is what $\left(\frac{\partial}{\partial t}\right)^{\mathsf{T}}$ is?

2.7 Inner product revisited

- Given that $\int_{-\infty}^{\infty} |x(t)|dt < \infty$ and $\int_{-\infty}^{\infty} |y(t)|dt < \infty$, integration by parts gives us

$$\begin{aligned}
\langle x', y\rangle &= \int_{-\infty}^{\infty} \frac{\partial x(t)}{\partial t} y(t)dt \\
&= x(t)y(t)|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} x(t)\frac{\partial y(t)}{\partial t}dt \quad \text{(Integration by parts)} \\
&= \int_{-\infty}^{\infty} x(t)\left(-\frac{\partial y(t)}{\partial t}\right) dt = \langle x, -y'\rangle
\end{aligned}$$

Hence,

$$\text{Transpose of differentiation is} \quad \left(\frac{\partial}{\partial t}\right)^{\mathtt{T}} = -\frac{\partial}{\partial t}.$$

- *Verification:*

$$A\boldsymbol{x} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 - x_1 \\ x_3 - x_2 \end{bmatrix} \text{ and } A^{\mathtt{T}}\boldsymbol{y} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{y} = \begin{bmatrix} -(y_2 - y_1) \\ -(y_3 - y_2) \\ y_3 \end{bmatrix}$$

$\square$

2.7 Inner product revisited

Summary

- Inner product is a more general concept, and has much more extensional use than what we have introduced here.

2.7 Symmetric matrix

We can re-define **symmetric matrix** (different from that given in slide 2-58) using **transpose**.

Definition (Symmetric matrix): A matrix A is said to be symmetric if

$$A^{\mathsf{T}} = A.$$

Example. Prove that $A^{\mathsf{T}}A$ is a symmetric matrix.

Proof: $(A^{\mathsf{T}}A)^{\mathsf{T}} = A^{\mathsf{T}}A$; hence, it is symmetric by definition. □

Remarks.

- Based on the above definition, we can even introduce *skew-symmetric* matrix defined as $\boxed{A^{\mathsf{T}} = -A}$.
- One advantage of symmetric matrices is that its **triple factorization** has a "symmetric" form. In other words, $A = LDL^{\mathsf{T}}$ if A is symmetric (and no row exchange during forward elimination).
- This can save half efforts in, e.g., forward elimination (from $n^3/3$ down to $n^3/6$). (See slide 2-71.)

2.7 Symmetric matrix

Question: Again, assume without row exchanges during elimination. What will be the complexity of forward elimination for a symmetric A?

Answer:

i) Forward elimination for matrix A,

$\times$ operations	exact
$a_{i,1} \to 0$ for $i = 2, \ldots, n$	$2 + 3 + \cdots + (n-1) + n = \frac{1}{2}(n-1)(n+2)$
$a_{i,2} \to 0$ for $i = 3, \ldots, n$	$2 + 3 + \cdots + (n-1) = \frac{1}{2}(n-2)(n+1)$
$\vdots$	$\vdots$
$a_{i,n-1} \to 0$ for $i = n$	$2 = \frac{1}{2} 1 \cdot 4$
total	$\displaystyle\sum_{i=2}^{n} \frac{1}{2}(i-1)(i+2) = \frac{1}{6} n(n-1)(n+4)$

ii) During the above eliminations, we only need to retain each multiplier (i.e., $\ell_{i,j}$) and pivot (i.e., $u_{i,i}$) to form L and D.

(Note that in the above process, we know that A will be reduced to LDL^{T}, which is of course known after knowing L and D.)

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \ell_{2,1} & 1 & 0 \\ \ell_{3,1} & \ell_{3,2} & 1 \end{bmatrix} \begin{bmatrix} u_{1,1} & - & - \\ 0 & u_{2,2} & - \\ 0 & 0 & u_{3,3} \end{bmatrix} L^{\mathrm{T}}$$

2.7 Factorization with row exchange: $PA = LU$

- What if **row exchange** is needed during **forward elimination**?

Example. (This is exactly the same example that appears on slide 2-66!)

$$\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 4 & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 0 & 1 \\ -2 & -3 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & 0 & 1 \\ 0 & 1 & 5 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 4 & -2 \\ 0 & \mathbf{1} & 5 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left[\begin{array}{ccc|ccc} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 4 & \boxed{8} & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{array}\right] \begin{array}{l} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) \text{ (Replace the middle 9 by 8)} \\ \ldots(3) \end{array}$$

$$\Leftrightarrow \left[\begin{array}{ccc|ccc} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 1 & 5 & 1 & 0 & 1 \end{array}\right] \begin{array}{l} \ldots(1) \text{ with } \mathbf{2} \text{ the first pivot} \\ \ldots(2) - 2 \times (1) \\ \ldots(3) - (-1) \times (1) \end{array}$$

$$\Leftrightarrow \left[\begin{array}{ccc|ccc} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & 1 & -2 & 1 & 0 \end{array}\right] \begin{array}{l} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') \end{array}$$

$$\Leftrightarrow \left[\begin{array}{ccc|ccc} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{1} & -2 & 1 & 0 \end{array}\right] \begin{array}{l} \ldots(1) \\ \ldots(2') \text{ with } \mathbf{1} \text{ the second pivot} \\ \ldots(3') - 0 \times (2') \end{array}$$

2.7 Factorization with row exchange: $PA = LU$

The forward elimination process can then be implemented by multiplying $E_{2,1}$, $E_{3,1}$, $P_{2,3}$ and $E_{3,2}$ in sequence.

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -0 & 1 \end{bmatrix}}_{E_{3,2}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}}_{P_{2,3}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}}_{E_{3,1}} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{E_{2,1}} \begin{bmatrix} 2 & 4 & -2 & 1 & 0 & 0 \\ 4 & 9 & -3 & 0 & 1 & 0 \\ -2 & -3 & 7 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{2} & 4 & -2 & 1 & 0 & 0 \\ 0 & \mathbf{1} & 5 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{1} & -2 & 1 & 0 \end{bmatrix}$$

Then we have that after **forward elimination**

$$\boxed{E_{3,2}P_{2,3}E_{3,1}E_{2,1}\begin{bmatrix} A & I \end{bmatrix} = \begin{bmatrix} EA & E \end{bmatrix} = \begin{bmatrix} U & E \end{bmatrix}}$$

where $E = E_{3,2}P_{2,3}E_{3,1}E_{2,1}$.

Next, we need to determine $\boxed{L = E^{-1} = E_{2,1}^{-1}E_{3,1}^{-1}P_{2,3}^{-1}E_{3,2}^{-1}}$. This is easy because every E_{ij} is formed by "changing" just one off-diagonal entry from an identity matrix, and the inverse of $P_{2,3}$ is itself!

2.7 Factorization with row exchange: $PA = LU$

What we need to do is to "negate" the only non-zero off-diagonal entry!

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}^{-1}}_{E_{3,2}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -(-1) & 0 & 1 \end{bmatrix}^{-1}}_{E_{3,1}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ (-1) & 0 & 1 \end{bmatrix}$$

$$\text{and} \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{-1}}_{E_{2,1}^{-1}} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\tilde{L} = E^{-1} = E_{2,1}^{-1} E_{3,1}^{-1} P_{2,3}^{-1} E_{3,2}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix}$$ **Not a lower triangular matrix!**

Then, we obtain

$$A = \tilde{L}U = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 5 \\ 0 & 0 & \mathbf{1} \end{bmatrix}$$

2.7 Factorization with row exchange: $PA = LU$

Hence,

$$PA = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} A = P\tilde{L}U = LU = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{2} & 4 & -2 \\ 0 & \mathbf{1} & 5 \\ 0 & 0 & \mathbf{1} \end{bmatrix}$$

□

Factorization with row exchange: Find permutation matrix P such that

$$PA = LU.$$

- The pre-permutation matrix that can force zero row exchange is not unique! For (another) example, check the below for the above example.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} A = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{bmatrix} \begin{bmatrix} 4 & 8 & -3 \\ 0 & 1 & \frac{11}{2} \\ 0 & 0 & -\frac{1}{2} \end{bmatrix}$$

2.7 Factorization with row exchange: $PA = LU$

- The MATLAB command for LU-factorization is

```
[L,U,P]=lu(A)
```

Note: The MATLAB will make row exchanges such that the next pivot is the largest! This is the reason why for

$$A = \begin{bmatrix} 2 & 4 & -2 \\ \mathbf{4} & 8 & -3 \\ -2 & -3 & 7 \end{bmatrix},$$

the first pivot that MATLAB chooses is $\mathbf{4}$!

2.7 Factorization with row exchange: $PA = LU$

Some useful notes

- A permutation matrix P (which has the rows of identity matrix I in any order) satisfies $P^{\mathsf{T}} = P^{-1}$.

  Proof: Permutation does not change the inner product; hence $(P\boldsymbol{x})^{\mathsf{T}}(P\boldsymbol{y}) = \boldsymbol{x}^T\boldsymbol{y}$. Since $\boldsymbol{x}$ and $\boldsymbol{y}$ are arbitrary, $P^{\mathsf{T}}P = I$. $\square$

- It is not necessary true that $P^{-1} = P$ as P may not be *symmetric*.

- There are $n!$ permutation matrices of size $n \times n$.

- The determinant of P is either 1 (when the number of row changes is even) or -1 (when the number of row changes is odd).

- We can also use **forward elimination** to make **symmtric** of a matrix A, i.e., $A = ES$ where E is an elementary row operation matrix, and S is symmetirc.

  - The idea is simply to make the lower triangle entry $a_{i,j}$ the same as $a_{j,i}$ (instead of making $a_{i,j} = 0$).

    Example (Problem 34 on page 119). $A = \begin{bmatrix} 1 & 2 \\ 4 & 9 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix} = ES$

    $\square$

2.7 Factorization with row exchange: $PA = LU$

(Problem 34, Section 2.7) Write $A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ as the product EH of an elementary row operation matrix E and a symmetric matrix H.

Definition (Elementary row operation matrix): A matrix E that only includes a combination of operations as

1. row exchanges;
2. row multiplications; and
3. row additions.

– In fact, we can use *triple factorization* to "symmetricize" a matrix.

$$A = LDU = \underbrace{L(U^{\mathsf{T}})^{-1}}_{E}\underbrace{(U^{\mathsf{T}}DU)}_{S} = ES.$$