Linear Algebra : Matrices

In linear algebra, a commonly encountered situation is where we need to solve matrix equation of the form

$$\mathrm{A} \cdot \mathrm{x} = \mathrm{b}$$

where A is an $n \times n$ matrix, x, b can either be $n \times 1$ vectors (as in solving system of linear equations) or $n \times n$ matrices (as in inverse).

The solution implies determining the vector x

$$\mathrm{x} = \mathrm{A}^{-1}\mathrm{b} \quad \text{where} \ \ \mathrm{A}^{-1} = \mathrm{A}_c^T/|\mathrm{A}|$$

where $|\mathrm{A}|$ is the determinant and A_c are the cofactors. Although numerically doable, far more efficient methods exist to calculate inverse.

- Direct elimination : Gauss-Jordan, LU decomposition, Cholesky decomposition which are used for small, dense matrices and are memory intensive, possibly slower and difficult to parallelize.
- Iterative methods : Gauss-Jacobi, Gauss-Seidel, Conjugate gradient, GMRES which are useful for large, particularly sparse matrices and lend themselves to parallelization.

Linear equations: Gauss-Jordon method

Consider solving a linear system of equations, say with 3 variables

$$\mathrm{A} \cdot \mathrm{X} = \mathrm{B} \Rightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

To solve (x_1, x_2, x_3), the Gauss-Jordon method suggests to write the above equation in an *augmented matrix* form,

$$\left[\mathrm{A}\middle|\mathrm{B}\right] \equiv \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

Using Gauss-Jordon elimination, convert the *augmented matrix* in *reduced row echelon form* (RREF) eventually yielding

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & \tilde{b}_1 \\ 0 & 1 & 0 & \tilde{b}_2 \\ 0 & 0 & 1 & \tilde{b}_3 \end{array}\right]$$

thus solving the linear equations : $x_1 = \tilde{b}_1,\ x_2 = \tilde{b}_2,\ x_3 = \tilde{b}_3$.

RREF

In *reduced row echelon form*,

1. all rows with only zero entries are at the bottom of the matrix
2. the first nonzero entry in a row (called **pivot**) of each nonzero row is to the right of the leading entry of the row above it
3. leading entry *i.e.* pivot in any nonzero row is 1
4. all other entries in row or column containing a leading 1 are zeros.

$$\text{Illustration :} \quad \begin{pmatrix} 1 & -2 & 3 \\ 0 & 1 & -4 \\ 0 & 0 & 1 \end{pmatrix}$$

The way to achieve RREF by Gauss-Jordon involves any one or a combination of three elementary row operations

- swapping two rows
- multiplying a row by a nonzero number
- adding or subtracting a multiple of one row to another row

Operation involving rows is called *partial pivoting*.

Full pivoting (recommended by Numerical Recipes) : both row and column operations.

Partial pivoting

Say the augmented matrix we want to reduce to RREF be

$$\left[\begin{array}{ccc|c} \boxed{0} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right]$$

a_{11} is either 0 or order(s) of magnitude small compared to a_{21}, a_{31} $\rightarrow a_{31}, a_{21} \gg a_{11}$ and let $a_{31} > a_{21}$. To get leftmost nonzero entry at the top, swap $R_3 \leftrightarrow R_1$

$$\left[\begin{array}{ccc|c} \boxed{a_{31}} & a_{32} & a_{33} & b_3 \\ a_{21} & a_{22} & a_{23} & b_2 \\ 0 & a_{12} & a_{13} & b_1 \end{array}\right] \equiv \left[\begin{array}{ccc|c} \boxed{a^0_{11}} & a^0_{12} & a^0_{13} & b^0_1 \\ a^0_{21} & a^0_{22} & a^0_{23} & b^0_2 \\ a^0_{31} & a^0_{32} & a^0_{33} & b^0_3 \end{array}\right]$$

where a^0_{11} is the *pivot element*, new row R^0_1 is the *pivot row* and new column C^0_1 is the *pivot column*. Reduce $R^0_1/a^0_{11} \rightarrow R^1_1$,

$$\left[\begin{array}{ccc|c} \boxed{1} & a^1_{12} & a^1_{13} & b^1_1 \\ a^0_{21} & a^0_{22} & a^0_{23} & b^0_2 \\ a^0_{31} & a^0_{32} & a^0_{33} & b^0_3 \end{array}\right], \quad a^1_{1,2,3} = a^0_{1,2,3}/a^0_{11},\ b^1_1 = b^0_1/a^0_{11}$$

Partial pivoting

Reduce the pivot column C_1 by adding / subtracting multiples of pivot row from the following rows

$$a^1_{2(1,2,3)} = a^0_{2(1,2,3)} - a^0_{21} * R^1_1, \qquad b^1_2 = b^0_2 - a^0_{21} * b^1_1$$
$$a^1_{3(1,2,3)} = a^0_{3(1,2,3)} - a^0_{31} * R^1_1, \qquad b^1_3 = b^0_3 - a^0_{31} * b^1_1$$

$$\Rightarrow \left[\begin{array}{ccc|c} 1 & a^1_{12} & a^1_{13} & b^1_1 \\ 0 & \boxed{a^1_{22}} & a^1_{23} & b^1_2 \\ 0 & a^1_{32} & a^1_{33} & b^1_3 \end{array}\right]$$

a^1_{22} is new *pivot element* and undergoes the same test of smallness. If needed, partial pivoting done; eventually yielding

$$\text{RREF} \Rightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & b^3_1 \\ 0 & 1 & 0 & b^3_2 \\ 0 & 0 & 1 & b^3_3 \end{array}\right]$$

This method is Gauss-Jordan elimination using *partial pivoting*.
No matter what steps and in which order they are applied, the final augmented matrix is unique.

Pivoting tryout

Consider the following set of linear equations :

$$\begin{aligned} y + z - 2w &= -3 \\ x + 2y - z &= 2 \\ 2x + 4y + z - 3w &= -2 \\ x - 4y - 7z - w &= -19 \end{aligned}$$

Write down the relevant matrices in a file. Construct the augmented matrix and perform partial pivoting of the first row only. Do necessary operations to convert the first column of the augmented matrix in the form $(1, 0, 0, 0)$ and write down the augmented matrix.

N.B. – Remember, while pivoting, swap rows that are below the new pivot row.

An example :

Consider the equations and its corresponding augmented matrix

$$\left.\begin{array}{rcr} 2y+5z &=& 1 \\ 3x-y+2z &=& -2 \\ x-y+3z &=& 3 \end{array}\right\} \Rightarrow \left[\begin{array}{rrr|r} 0 & 2 & 5 & 1 \\ 3 & -1 & 2 & -2 \\ 1 & -1 & 3 & 3 \end{array}\right]$$

Since $a_{11}=0$ and $a_{21}>a_{31}$, we begin by swapping $R_1 \leftrightarrow R_2$,

$$\left[\begin{array}{rrr|r} 3 & -1 & 2 & -2 \\ 0 & 2 & 5 & 1 \\ 1 & -1 & 3 & 3 \end{array}\right] \xrightarrow{R_1^0/3 \to R_1^1} \left[\begin{array}{rrr|r} 1 & -1/3 & 2/3 & -2/3 \\ 0 & 2 & 5 & 1 \\ 1 & -1 & 3 & 3 \end{array}\right]$$

$$\xrightarrow{R_3^0 - R_1^1 \to R_3^1} \left[\begin{array}{rrr|r} 1 & -1/3 & 2/3 & -2/3 \\ 0 & 2 & 5 & 1 \\ 0 & -2/3 & 7/3 & 11/3 \end{array}\right]$$

$$\xrightarrow{R_2^1/2 \to R_2^2} \left[\begin{array}{rrr|r} 1 & -1/3 & 2/3 & -2/3 \\ 0 & 1 & 5/2 & 1/2 \\ 0 & -2/3 & 7/3 & 11/3 \end{array}\right]$$

$$\xrightarrow{R_1^1 + R_2^2/3 \to R_1^2} \left[\begin{array}{rrr|r} 1 & 0 & 9/6 & -3/6 \\ 0 & 1 & 5/2 & 1/2 \\ 0 & -2/3 & 7/3 & 11/3 \end{array}\right]$$

$$\xrightarrow{R_3^1+2R_2^2/3\to R_3^2}\left[\begin{array}{ccc|c} 1 & 0 & 9/6 & -3/6 \\ 0 & 1 & 5/2 & 1/2 \\ 0 & 0 & 12/3 & 24/6 \end{array}\right]$$

$$\xrightarrow{3R_3^2/12\to R_3^3}\left[\begin{array}{ccc|c} 1 & 0 & 9/6 & -3/6 \\ 0 & 1 & 5/2 & 1/2 \\ 0 & 0 & 1 & 1 \end{array}\right] \xrightarrow{R_2^2-5R_3^3/2\to R_2^3}\left[\begin{array}{ccc|c} 1 & 0 & 9/6 & -3/6 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

$$\xrightarrow{R_1^2-9R_3^3/6\to R_1^3}\left[\begin{array}{ccc|c} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & -2 \\ 0 & 0 & 1 & 1 \end{array}\right]$$

Therefore, the solution is $x = -2$, $y = -2$, $z = 1$.

Gauss-Jordon can be trivially extend to obtain inverse of an invertible matrix. First to test invertibility, determine the determinant.

$$\det\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \det\begin{pmatrix} a'_{11} & a'_{12} & a'_{13} \\ 0 & a'_{22} & a'_{23} \\ 0 & 0 & a'_{33} \end{pmatrix} = (-1)^n a'_{11}\, a'_{22}\, a'_{33}$$

where n is the number of times the rows are interchanged, $a'_{ii} \neq 0$ and none of a'_{ii} has to be 1.

Matrix inversion with Gauss-Jordon

Matrix inversion follows similar system of linear equations as before

$$\mathbf{A}\cdot\mathbf{B} = \mathbb{1} \;\rightarrow \text{augmented matrix : } \left[\mathrm{A}\middle|\mathbb{1}\right] \overset{G-J}{\Longrightarrow} \left[\mathbb{1}\middle|\mathrm{A}^{-1}\right]$$

Memory requirement : $N \times (N+1)$ for solving linear system of equations and $N \times (N+N)$ for inverse, ignoring swapping operation (involving sorting).

Mathematical operations : $(N-1) \times 2$ (multiplication / division and addition / subtraction) for each row reduction.

Accuracy : Rounding errors, proportional to N, quickly built up in solution

For large N not very suitable – memory requirement $\propto N^2$ and slow. Typically restricted to $\sim N = 10$