Week 2

Solving Systems of Linear Equations

Given a set of equations,

$$\begin{aligned} x_1 - x_2 &= -2 \\ 2x_1 - 5x_2 &= -7 \end{aligned}$$

First check the determinant: (1)(-5) – (-1)(2) = -3

The following is what you did in high school

$$\begin{aligned} x_1 - x_2 &= -2 \\ 2x_1 - 5x_2 &= -7 \end{aligned} \xrightarrow{\text{Gaussian Elimination}} \begin{aligned} x_1 - x_2 &= -2 \\ -3x_2 &= -3 \end{aligned} \xrightarrow{\text{Back substitution}} \begin{aligned} x_1 &= -1 \\ x_2 &= 1 \end{aligned}$$

Matrix Form:

$$\underbrace{\begin{pmatrix} 1 & -1 \\ 2 & -5 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2 \\ -7 \end{pmatrix}}_{\mathbf{Ax}=\mathbf{b}} \Rightarrow \underbrace{\begin{pmatrix} 1 & -1 \\ 0 & -3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \end{pmatrix}}_{\mathbf{Ux}=\mathbf{b}'} \Rightarrow \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}}_{\mathbf{x}=\mathbf{b}''}$$

$\mathbf{U}$ is an upper triangular matrix.

Definition: $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ denotes the solution of $\mathbf{x}$ obtained using Gaussian Elimination

Importance: Mathematically $\mathbf{A} \backslash \mathbf{b} = \mathbf{A}^{-1}\mathbf{b}$, but computationally we do not need $\mathbf{A}^{-1}$ when calculating $\mathbf{A} \backslash \mathbf{b}$. So $\mathbf{A} \backslash \mathbf{b}$ is more efficient.

In Matlab, $\mathbf{A} \backslash \mathbf{b}$ is executed by `A\b` and $\mathbf{A}^{-1}\mathbf{b}$ is executed by `inv(A)*b`

LU decomposition

Purpose: Given a square $n \times n$ matrix **A**, decompose **A** into a product of two square matrices: $\mathbf{A} = \mathbf{LU}$, where **L** is lower triangular and **U** is upper triangular.

U is obtained using Gaussian Elimination. But how to get **L**? **L** is to record how we make a zero in **A**. First write

$$L = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Consider this intermediate step:

$$\begin{pmatrix} 1 & -1 \\ 2 & -5 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -2 \\ -7 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} 1 & -1 \\ 0 & -3 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -2 \\ -3 \end{pmatrix}$$

We have multiplied 2 to the first row to use it to subtract the second row, resulting a zero in the (2,1)-th element. So put 2 into the same location where **A** is made zero:

$$\mathbf{L} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$$

Check:

$$\mathbf{LU} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 1 & -1 \\ 0 & -3 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 2 & -5 \end{pmatrix} = \mathbf{A}$$

Then

$$\mathbf{Ax} = \mathbf{b} \quad \Rightarrow \quad (\mathbf{LU})\mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{Ux} = \mathbf{L} \backslash \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \mathbf{U} \backslash (\mathbf{L} \backslash \mathbf{b})$$

Here

$\mathbf{A} = \mathbf{LU}$ manifests Gaussian Elimination $\sim O\left(\frac{2}{3}n^3\right)$ float-point operations (flops)

$\mathbf{L} \backslash \mathbf{b}$ manifests forward substitution $\sim O(n^2)$ flops

$\mathbf{U} \backslash \mathbf{b}$ manifests back substitution $\sim O(n^2)$ flops

Pivoting and Permutation

There is situation where $\mathbf{A} = \mathbf{LU}$ cannot be done. Consider

$$\begin{aligned} x_2 &= 1 \\ 2x_1 - 5x_2 &= -7 \end{aligned}, \text{ or equivalently, } \begin{pmatrix} 0 & 1 \\ 2 & -5 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -7 \end{pmatrix}$$

This system cannot be made upper triangular by Gaussian Elimination. But one can swap the order of the rows of A by applying a permutation matrix

$$\mathbf{P} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

so that

$$\mathbf{A}' = \mathbf{PA} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} 0 & 1 \\ 2 & -5 \end{pmatrix} = \begin{pmatrix} 2 & -5 \\ 0 & 1 \end{pmatrix}$$

And $\mathbf{A}'$ is LU-decomposable (although not necessary in this example). In Matlab, when `A\b` is executed, the rows of **A** are first rearranged by a permutation matrix `P` to avoid zero pivots. Then `P*A` is decomposed into `L` and `U`. Finally, **x** is obtained by `U\(L\P*b)`.

$$\mathbf{Ax} = \mathbf{b} \;\Rightarrow\; (\mathbf{PA})\mathbf{x} = \mathbf{Pb} \;\Rightarrow\; (\mathbf{LU})\mathbf{x} = \mathbf{Pb} \;\Rightarrow\; \mathbf{Ux} = \mathbf{L}\backslash(\mathbf{Pb}) \;\Rightarrow\; \mathbf{x} = \mathbf{U}\backslash\left(\mathbf{L}\backslash(\mathbf{Pb})\right)$$

Matlab commands

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

`A\b` or `inv(A)*b`?

To obtain `A\b`, $O(2.67n^3)$ flops are taken. To obtain `inv(A)*b`, $O(5.67n^3)$ flops are taken. So always use `A\b`.

Given any matrices, check

`det(A)`: If $|\det(\mathbf{A})| >> 10^{-16}$, unique solution of **x** exists.

But does not mean accurate. Check

`cond(A)`: Fractional error $\dfrac{\delta\mathbf{x}}{\mathbf{x}} = cond(A) \times \varepsilon$, where $\varepsilon = 10^{-16}$ is the machine precision.