HANDS ON Ax=b -> LUx=b

LU decomposition – study of the computational cost

Descomposición LU ¿Cuánto cuesta?

¿Cuánto tiempo lleva resolver un sistema de 100, 1000, o 10^6 de ecuaciones?

¿Cuántas operaciones para factorizar una matriz A de tamaño nxn?

Lo más importante es saber cómo escala el coste (n, n^2, n^3, n! ...)

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \ddots & a_{2n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ 0 & \ddots & \tilde{a}_{2n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ 0 & \ddots & \tilde{a}_{2n} \\ \vdots & & \vdots \\ 0 & \cdots & \tilde{a}_{nn} \end{bmatrix}$$

n operaciones*

n operaciones x (n-1) filas = n^2 operaciones

*Por simplicidad, consideramos "una operacion" como una multiplicación y una suma.

Descomposición LU ¿Cuánto cuesta?

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \ddots & a_{2n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ 0 & \ddots & \tilde{a}_{2n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ 0 & \ddots & \tilde{a}_{2n} \\ \vdots & & \vdots \\ 0 & \cdots & \tilde{a}_{nn} \end{bmatrix}$$

n operaciones*

n operaciones x n filas = n^2 operaciones

*Por simplicidad, consideramos "una operacion" como una multiplicación y una suma.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & & \tilde{a}_{2n} \\ 0 & a_{23} & \ddots & \tilde{a}_{3n} \\ & \vdots & & \vdots \\ 0 & a_{n2} & \cdots & \tilde{a}_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & & \tilde{a}_{2n} \\ 0 & 0 & \ddots & \tilde{a}_{3n} \\ & \vdots & & \vdots \\ 0 & a_{n2} & \cdots & \tilde{a}_{nn} \end{bmatrix} \longrightarrow \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22} & & \tilde{a}_{2n} \\ 0 & 0 & \ddots & \tilde{a}_{3n} \\ & \vdots & & \vdots \\ 0 & 0 & \cdots & \tilde{a}_{nn} \end{bmatrix}$$

n-1 operaciones*

n-1 operaciones x n-1 **filas** = (n-1)^2 operaciones

Descomposición LU ¿Cuánto cuesta?

(Explicación intuitiva) $\int n^2 = \frac{n^3}{3}$

$$n^2 + (n-1)^2 + (n-2)^2 + \cdots + 3^2 + 2^2 + 1^2 \sim \frac{1}{3} n^3$$

Lo más importante es saber cómo escala el coste (n, n^2, **n^3**, n! ...)

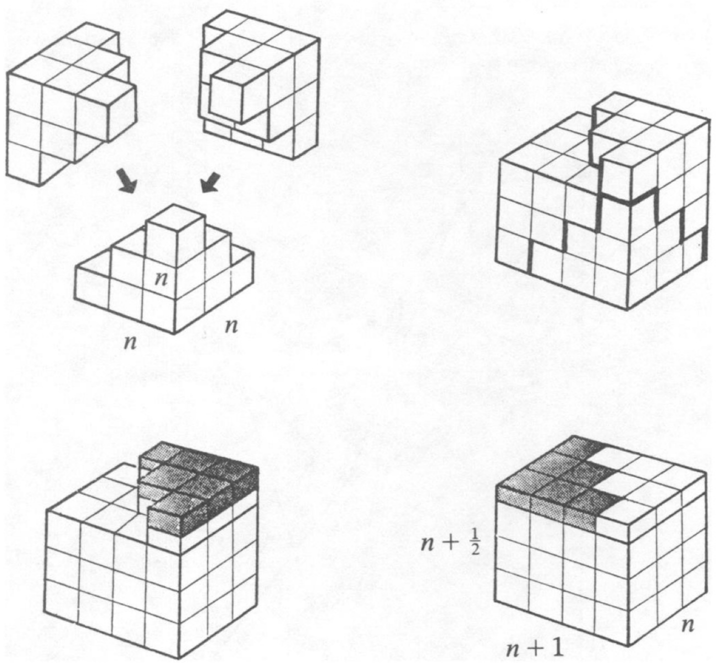
Si resolvemos $Ax = b$ el coste para cada b es aproximadamente n^2

Factorización LU es muy eficiente para resolver muchas veces el mismo sistema

¿Cuánto cuesta?

Proof without words:
Sum of squares

$$1^2 + 2^2 + \cdots + n^2 = \frac{1}{3}n(n+1)\left(n+\frac{1}{2}\right)$$



—Man-Keung Siu
University of Hong Kong

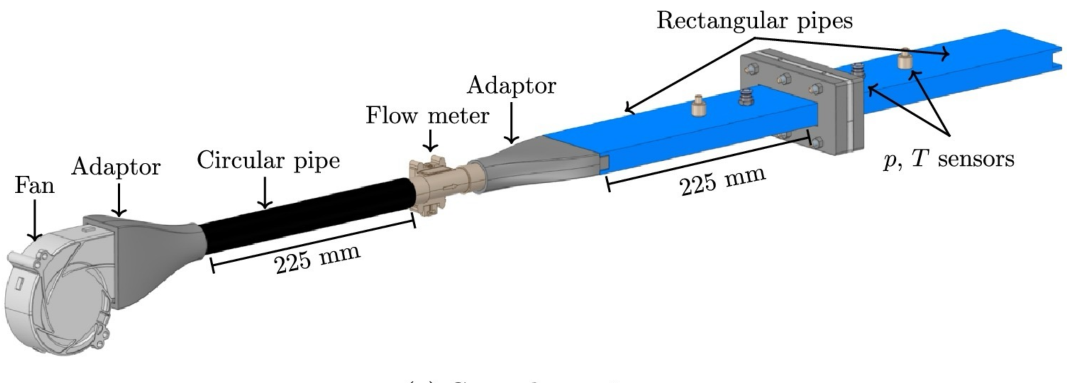
March 1984 issue of *Mathematics Magazine*

Example 2: Plane Poiseuille Flow (Channel Flow)

Consider the flow of a viscous Newtonian fluid between two solid boundaries at $y = \pm h$ driven by a constant pressure gradient $\nabla p = [-P, 0, 0]$. Show that

$$u = \frac{P}{2\,\mu}(h^2 - y^2), \quad v = w = 0.$$

$$\mu = \rho\nu$$



Aitor Amatriain, Corrado Gargiulo, Gonzalo Rubio, Numerical and experimental study of open-cell foams for the characterization of heat exchangers, International Journal of Heat and Mass Transfer, Volume 217, 2023,

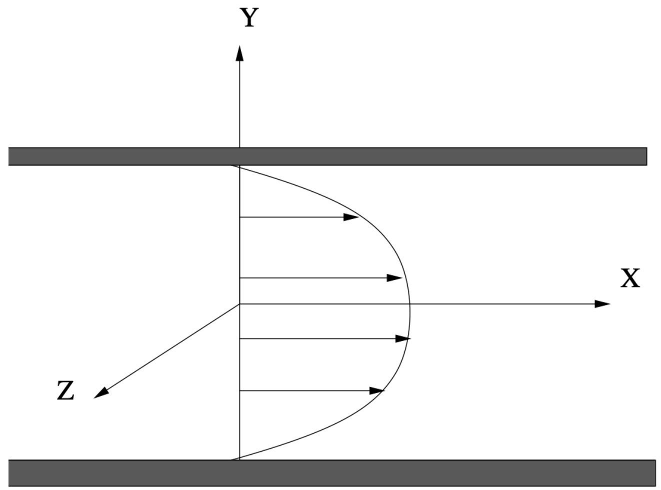


Figure 1: Coordinate system for plane Poiseuille flow.

https://www.mech.kth.se/~luca/Smak/OLD_REC/rec5.pdf

Navier-Stokes equations:

$$\begin{cases} \dfrac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} = -\dfrac{1}{\rho}\nabla p + \nu \nabla^2 \bar{u} \\ \nabla \cdot \bar{u} = 0. \end{cases}$$

Boundary conditions:

$$\bar{u}(y = \pm h) = 0$$

- We are considering stationary flow and thus $\dfrac{\partial \bar{u}}{\partial t} = 0$.
- The constant pressure gradient implies $\bar{u} = \bar{u}(y)$. Changes of $\bar{u}$ in x, z would require a changing pressure gradient in x, z.
- The continuity equation $\nabla \cdot \bar{u}$ reduces to $\dfrac{\partial v}{\partial y} = 0$. The boundary condition $v(y = \pm h) = 0$ then implies $v = 0$.

We can conclude that $\bar{u} = [u(y), 0, 0]$.

https://www.mech.kth.se/~luca/Smak/OLD_REC/rec5.pdf

Consider now the streamwise (x) component of the Navier-Stokes equations:

$$0 = \frac{P}{\rho} + \nu \frac{\partial^2 u}{\partial y^2} \quad \Longrightarrow \quad D^2 U = -\frac{P}{\rho\nu}$$

boundary conditions at $y = \pm h$ $\qquad \bar{u}(y = \pm h) = 0$

$$D^2 = \frac{d^2}{dx^2} = \frac{1}{(\Delta x)^2} \begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix} \qquad \begin{bmatrix} -\frac{P}{\rho\nu} \\ -\frac{P}{\rho\nu} \\ \vdots \\ -\frac{P}{\rho\nu} \end{bmatrix}$$

https://www.mech.kth.se/~luca/Smak/OLD_REC/rec5.pdf

Consider now the streamwise (x) component of the Navier-Stokes equations:

$$0 = \frac{P}{\rho} + \nu \frac{\partial^2 u}{\partial y^2} \qquad \Rightarrow \qquad D^2 U = -\frac{P}{\rho\nu}$$

boundary conditions at $y = \pm h$ $\qquad \bar{u}(y = \pm h) = 0$

$$\frac{1}{(\Delta x)^2}\begin{bmatrix} 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 1 & -2 & 1 \end{bmatrix}\begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_{n-1} \end{bmatrix} = \begin{bmatrix} -\frac{P}{\rho\nu} \\ -\frac{P}{\rho\nu} \\ \vdots \\ -\frac{P}{\rho\nu} \end{bmatrix}$$

Consider now the streamwise (x) component of the Navier-Stokes equations:

$$0 = \frac{P}{\rho} + \nu \frac{\partial^2 u}{\partial y^2} \quad \Longrightarrow \quad D^2 U = -\frac{P}{\rho \nu}$$

boundary conditions at $y = \pm h$ $\qquad \bar{u}(y = \pm h) = 0$

$$\frac{1}{(\Delta x)^2}\begin{bmatrix} (\Delta x)^2 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & (\Delta x)^2 \end{bmatrix} \begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_n \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{P}{\rho \nu} \\ \vdots \\ 0 \end{bmatrix}$$

Añadimos dos nuevas filas a la matriz, y los valores U_0 y U_n al vector de incógnitas para tener en cuenta las condiciones de contorno

THIS IS OUR MATHEMATICAL MODEL

$$\frac{1}{(\Delta x)^2}\begin{bmatrix} (\Delta x)^2 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & (\Delta x)^2 \end{bmatrix}\begin{bmatrix} U_0 \\ U_1 \\ \vdots \\ U_n \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{P}{\rho\nu} \\ \vdots \\ 0 \end{bmatrix}$$

Ax=b → LUx=b

PRÁCTICA LU

1.- Crear la matriz A y el vector b en función de la resolución del problema (dx) y los parámetros físicos P, ρ, ν.
2.- Utilizando `scipy.linalg` realizar la descomposición LU y resolver el problema `la.lu(A), la.solve_triangular`
3.- Comparar la solución obtenida con la solución exacta norm(U_num-U_exact) del problema para distintas resoluciones (cambiar espaciado malla dx). NOTA: el vector U_exact se puede crear evaluando la solución exacta del problema (Slide 6) en los puntos de la malla).
4.- Medir el tiempo necesario para resolver el problema, para distintos tamaños de matriz. Comparar el escalado con el exacto (n^3).

```
start_time = time.time()
x_LU = solve_with_LU_decomposition(A, b)
LU_time = time.time() - start_time
```