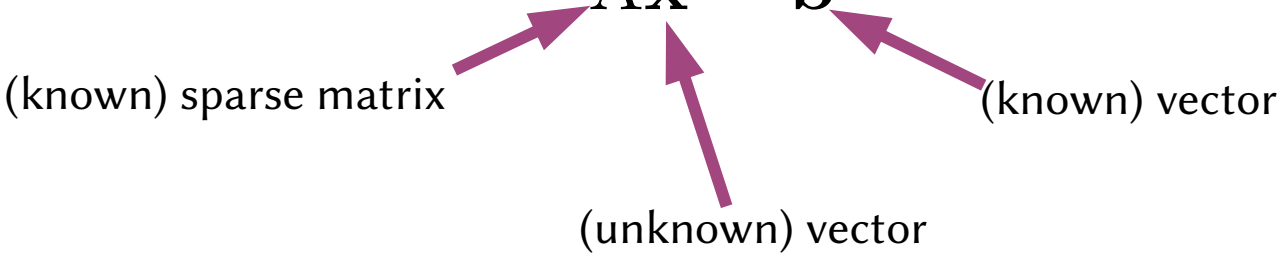
PHAS0102: Techniques of High-Performance Computing

Sparse solvers

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$



Direct solvers

An iterative solver finds a series of approximate solutions $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \ldots$

A direct solver finds the exact solution $\mathbf{x}$

Solving a triangular system

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

Solving a triangular system

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

Solving a triangular system

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

Solving a triangular system

$$\begin{pmatrix} 2 & 0 & 0 \\ 2 & 3 & 0 \\ 1 & -1 & 5 \end{pmatrix} \begin{pmatrix} 5 \\ 2 \\ -1 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

Solving a triangular system

$$\begin{pmatrix} 2 & 3 & 2 \\ 0 & 3 & 1 \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 11 \\ 0 \\ 15 \end{pmatrix}$$

Solving an LU system

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & 3 & -2 \end{pmatrix} \begin{pmatrix} 2 & 4 & 8 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

Solve

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 2 & 0 \\ -1 & 3 & -2 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 10 \\ 26 \\ -2 \end{pmatrix}$$

then solve

$$\begin{pmatrix} 2 & 4 & 8 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}$$

LU decomposition

$$\begin{pmatrix} -2 & 4 & 3 & 4 \\ -2 & 3 & 5 & 3 \\ -2 & 5 & 3 & -2 \\ -2 & 4 & 2 & 2 \end{pmatrix} = \begin{pmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{pmatrix} \begin{pmatrix} 1 & * & * & * \\ 0 & 1 & * & * \\ 0 & 0 & 1 & * \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

LU decomposition

$$\begin{pmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{pmatrix}$$

$$\begin{pmatrix} -2 & 4 & 3 & 4 \\ -2 & 3 & 5 & 3 \\ -2 & 5 & 3 & -2 \\ -2 & 4 & 2 & 2 \end{pmatrix} \div -2$$

LU decomposition

$$\begin{pmatrix} -2 & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & -\frac{3}{2} & -2 \\ -2 & 3 & 5 & 3 \\ -2 & 5 & 3 & -2 \\ -2 & 4 & 2 & 2 \end{pmatrix} \begin{matrix} = a \\ -(-2)a \\ -(-2)a \\ -(-2)a \end{matrix}$$

LU decomposition

$$\begin{pmatrix} -2 & 0 & 0 & 0 \\ -2 & * & 0 & 0 \\ -2 & * & * & 0 \\ -2 & * & * & * \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & -\frac{3}{2} & -2 \\ 0 & -1 & 2 & -1 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & -1 & -2 \end{pmatrix} \div(-1)$$

LU decomposition

$$\begin{pmatrix} -2 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 \\ -2 & * & * & 0 \\ -2 & * & * & * \end{pmatrix}$$

$$\begin{pmatrix} 1 & -2 & -\frac{3}{2} & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 1 & 0 & -6 \\ 0 & 0 & -1 & -2 \end{pmatrix} \begin{matrix} \\ = b \\ -1b \\ -0b \end{matrix}$$

LU decomposition

$$\begin{pmatrix} -2 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 \\ -2 & 1 & * & 0 \\ -2 & 0 & * & * \end{pmatrix} \qquad \begin{pmatrix} 1 & -2 & -\frac{3}{2} & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 2 & -7 \\ 0 & 0 & -1 & -2 \end{pmatrix}$$

[live Python example]

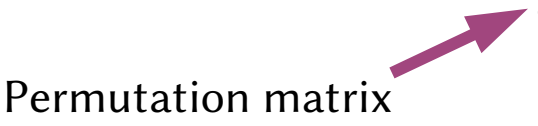
LU decomposition

$$\begin{pmatrix} -2 & 4 & 3 & 4 \\ -2 & 3 & 5 & 3 \\ -2 & 5 & 3 & -2 \\ -2 & 4 & 2 & 2 \end{pmatrix} = \begin{pmatrix} -2 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 \\ -2 & 1 & 2 & 0 \\ -2 & 0 & -1 & -\frac{11}{2} \end{pmatrix} \begin{pmatrix} 1 & -2 & -\frac{3}{2} & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & 1 & -\frac{7}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

What could go wrong?

- Division by 0
  - LU with pivoting moves the largest number to the diagonal to avoid this in most cases

$$PA = LU$$



Cost of LU

- Computing LU factorisation is $O(n^3)$
- Solving using the LU factors is $O(n^2)$

- LU is not good for general large matrices
- … so why are we learning about it?

LU for tridiagonal matrices

$$\begin{pmatrix} * & * & 0 & 0 & 0 & 0 \\ * & * & * & 0 & 0 & 0 \\ 0 & * & * & * & 0 & 0 \\ 0 & 0 & * & * & * & 0 \\ 0 & 0 & 0 & * & * & * \\ 0 & 0 & 0 & 0 & * & * \end{pmatrix}$$

Incomplete LU

- Incomplete LU is a variant of LU that throws away some data in the decomposition to make it faster.
- ILU can make a good preconditioner

[live Python example]