
Solving linear systems

Systems of linear equations, such as

$$\begin{aligned} 4x_1 - 8x_3 &= 12 \\ x_1 + 2x_2 + 3x_3 &= 2 \\ -2x_1 + 4x_2 - x_3 &= 7 \end{aligned}$$

can be written in matrix form as $A\mathbf{x} = \mathbf{b}$.
In the above example, take

$$A = \begin{pmatrix} \qquad\qquad\qquad & & \\ & & \\ & & \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} \qquad \\ \\ \\ \end{pmatrix}$$

Gaussian elimination

This method of solving $A\mathbf{x} = \mathbf{b}$ proceeds by systematically zeroing out entries in the lower left corner of A (working downward and to the right) using the following "row operation":

Replace a row by itself minus a multiple of another row.

The system that results from such a row operation will be equivalent to the original one provided whatever we do to the left-hand side we do to the right-hand side as well. In other words, the same row operations must be performed on the right-hand side vector $\mathbf{b}$. Thus, before starting Gaussian elimination it is customary to make an "augmented matrix" $\big(A \mid \mathbf{b}\big)$ by placing A and $\mathbf{b}$ side by side. Then we select the row operations to do by comparing the diagonal elements of A (the "pivots") with the entries we wish to zero out.

After Gaussian elimination, the coefficient matrix becomes *upper triangular*, i.e. all entries below the main diagonal are zero. Then, starting with the last row, we can use *back substitution* to solve for the x_k's from the last component to the first one.

Tasks:

- Do Gaussian elimination on the above 3-component example by hand.
- Use Matlab to perform row operations and back substitution on the following system:

$$\begin{aligned} 4x_1 + 8x_2 - 4x_4 + x_5 &= -1 \\ 3x_1 + 2x_2 + x_3 + 2x_5 &= -4 \\ x_1 + 3x_2 + x_3 + x_4 - 3x_5 &= 11 \\ -2x_1 + x_2 + 5x_3 + 6x_4 &= 5 \\ 12x_1 - 5x_2 - 2x_3 - x_4 + 3x_5 &= 3 \end{aligned}$$

- Based on the calculations for the 5-component example, write a Matlab function to automate the back substitution process.

Although the choice of what row operations to do (i.e. what multiples of the pivot row to subtract from the current row) depends only on the coefficient matrix A, if the right-hand side $\mathbf{b}$ is changed, even just slightly, we would need to repeat all these operations on the new augmented matrix with the different $\mathbf{b}$—and we would have to remember them all first.

LU factorization

It turns out that the row multiples needed to reduce A to upper triangular form U can be stored in a *lower triangular* matrix L with ones along its main diagonal, e.g. for the 3-component example

$$L = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{4} & 1 & 0 \\ -\frac{1}{2} & 2 & 1 \end{pmatrix}$$

You can check that

$$LU = A$$

and, moreover, it can be proved that this is always the case. LU is called an "LU factorization of A".

Once the LU factorization of a matrix A has been found, given any right-hand side $\mathbf{b}$ we can solve the system $A\mathbf{x} = \mathbf{b}$ as follows:

- Since $A = LU$, the equation $A\mathbf{x} = \mathbf{b}$ becomes

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

- Let $U\mathbf{x} = \mathbf{y}$; then the equation becomes

  $$L\mathbf{y} = \mathbf{b}$$

  Since L is lower triangular, this system can be solved by *forward substitution* (starting with the first row, solve for the y_k's from the first component to the last one).

- Once $\mathbf{y}$ has been found, solve

  $$U\mathbf{x} = \mathbf{y}$$

  by backward substitution. This yields the solution $\mathbf{x}$ to the original system $A\mathbf{x} = \mathbf{b}$.

Use Matlab's `[L,U]=lu(A)` function to find an LU factorization of

$$A = \begin{pmatrix} -10 & 7 & 6 & 5 \\ 4 & 3 & -1 & -7 \\ 1 & 2 & 6 & 8 \\ -2 & -4 & 5 & 3 \end{pmatrix}$$