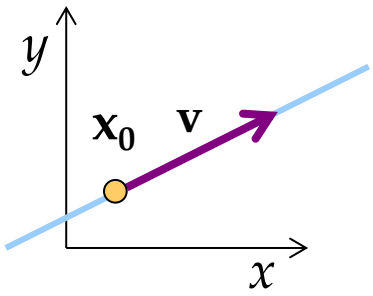
SYSTEMS OF LINEAR EQUATIONS

A line in $\mathbb{R}^2$ or $\mathbb{R}^3$ can be uniquely determined by specifying a point $\mathbf{x_0}$ on the line and a nonzero vector $\mathbf{v}$ that is parallel to the line:



$$\mathbf{x} = \mathbf{x_0} + t\mathbf{v} \qquad -\infty \le t \le \infty$$

As the parameter t varies from -∞ to ∞ the point $\mathbf{x}$ traces the line.

If $\mathbf{v} = (v_x, v_y)$, equating components yields the parametric equations:

$$x = x_0 + tv_x$$
$$y = y_0 + tv_y$$

isolate t from (2)

and substitute in (1)

$$v_y x - v_x y = v_y x_0 - v_x y_0$$

General equation for a line in $\mathbb{R}^2$:

$$a_1 x + a_2 y = b$$

The equation for a line in $\mathbb{R}^2$:

$$a_1x + a_2y = b$$

can be easily generalized to that of a plane in $\mathbb{R}^3$:

$$a_1x + a_2y + a_3z = b$$

or an n-1 dimensional hyperplane in $\mathbb{R}^n$:

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = b$$

These are all examples of linear equations in n variables $x_1, \ldots, x_n$ where $a_1, \ldots, a_n$ and b are constants and the a's are not all 0.

The special case with $b = 0$:

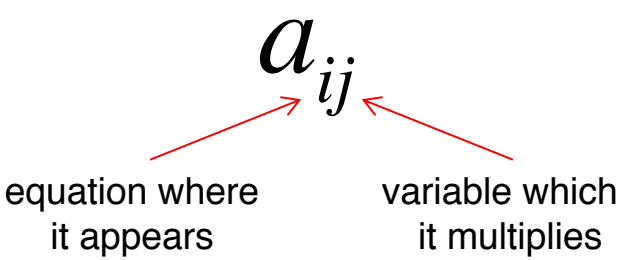
$$a_1x_1 + a_2x_2 + \ldots + a_nx_n = 0$$

is called a homogeneous linear equation.

A finite set of m linear equations in n variables x_i is called a system of linear equations or, for short, a linear system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

The double subscripting in the a_{ij} coefficients is used to indicate their position:

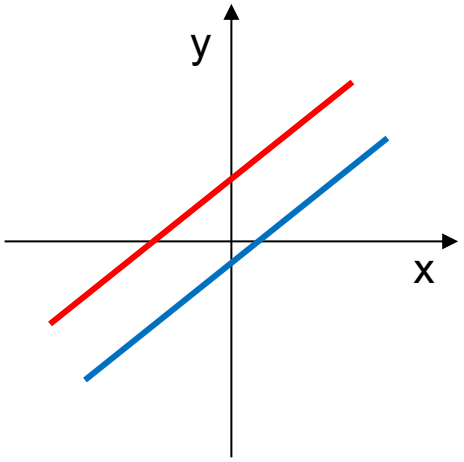


A solution for the linear system is a set of values for $x_1, \ldots, x_n$ that satisfies simultaneously all m equations.

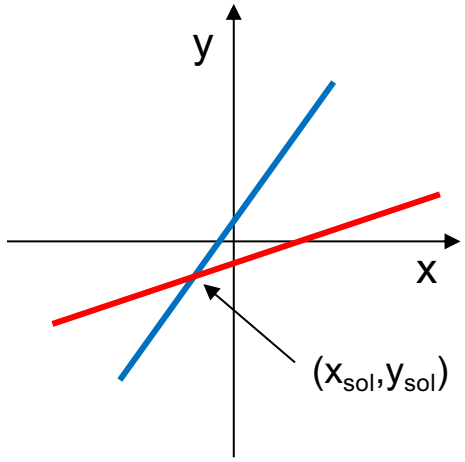
Linear systems in two unknowns x and y arise in connection with intersections of lines in R^2

$$a_{11}x + a_{12}y = b_1$$
$$a_{21}x + a_{22}y = b_2$$

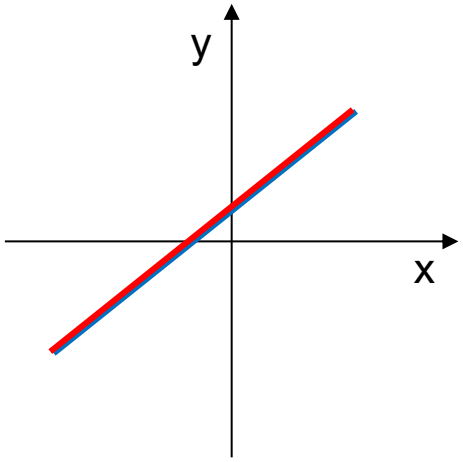
Each solution (x,y) for this system corresponds to a point of intersection of these lines and there are, in general, three possibilities:



No solution



One solution

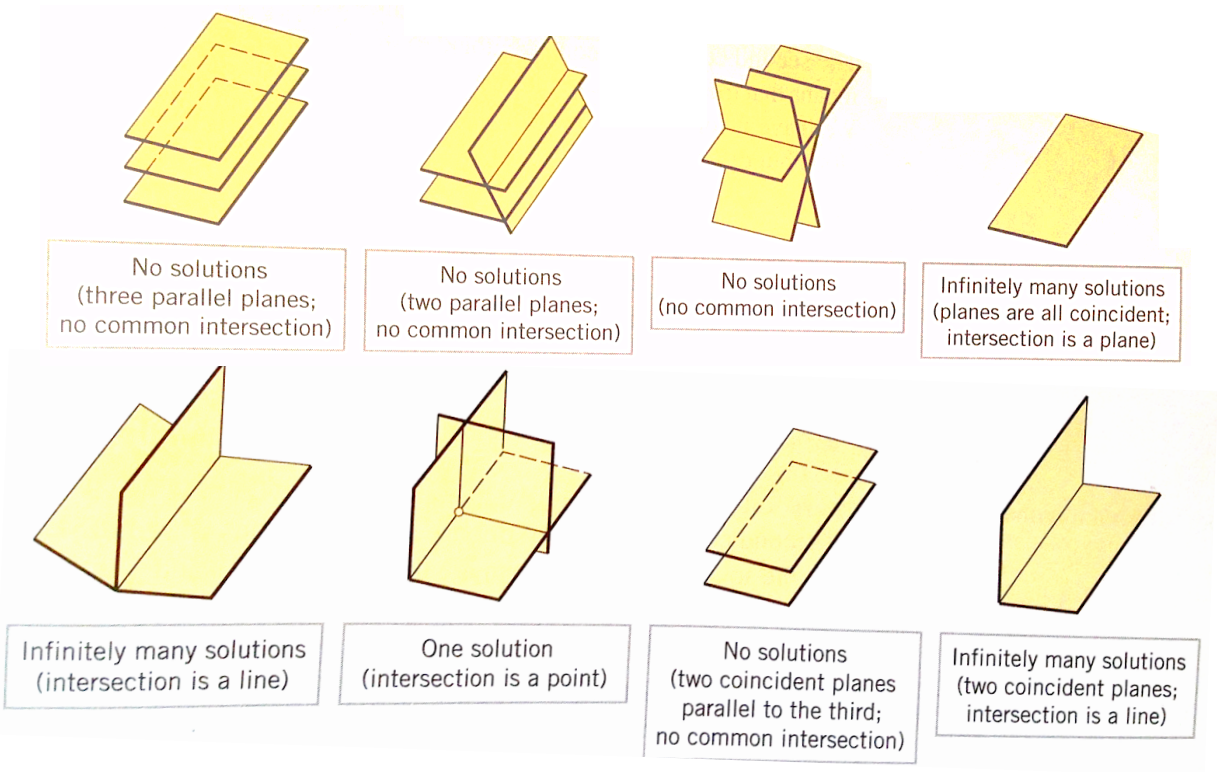


Infinite solutions

In linear systems in three unknowns x, y, z we are concerned with intersections of planes in R^3

$$a_{11}x + a_{12}y + a_{13}z = b_1$$
$$a_{21}x + a_{22}y + a_{23}z = b_2$$
$$a_{31}x + a_{32}y + a_{33}z = b_3$$

There are six possibilities now:



In general it can be shown that:

> Every linear system has zero, one, or infinitely many solutions; there are no other possibilities.

A linear system is said to be consistent if there is at least one solution and inconsistent if it has no solutions.

The basic method for solving a linear system is to perform appropriate algebraic operations to produce a succession of increasingly simpler systems that have the same solution as the original system until a point is reached where it is evident if the system is consistent or not, and if so, what the solutions are:

$$\begin{aligned} x+2y+z&=5\\ 3x+2y+4z&=17\\ 4x+4y+3z&=26 \end{aligned}$$

eliminate x from eq. 2
by eq. 2' = eq. 2 – 3 x (eq.1)

$$\begin{aligned} x+2y+z&=5\\ -4y+z&=2\\ 4x+4y+3z&=26 \end{aligned}$$

eliminate x from eq. 3
by eq. 3' = eq. 3 – 4 x (eq.1)

$$\begin{aligned} x+2y+z&=5\\ -4y+z&=2\\ -4y-z&=6 \end{aligned}$$

eliminate y from eq. 3'
by eq. 3" = eq. 3' – eq. 2'

$$\begin{aligned} x+2y+z&=5\\ -4y+z&=2\\ -2z&=4 \end{aligned}$$

The linear system is now in upper-triangular form because eq. 3 depends now only on z and eq. 2 depends only on y and z:

$$\begin{aligned} x+2y+z &= 5 \\ -4y+z &= 2 \\ -2z &= 4 \end{aligned}$$

The last equation can be directly solved to get

$$z = -2$$

Inserting this result into the second equation we arrive to

$$-4y = 4 \quad \Rightarrow \quad y = -1$$

and inserting the values of y and z into the first equation we get

$$x - 4 - 1 = 5 \quad \Rightarrow \quad x = 9$$

This process of computing the unknowns from a system that is in upper-triangular form is called back substitution.

In order to develop efficient algorithms to solve linear systems on a computer it is very convenient to describe them using the so-called augmented matrix

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots & \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned} \quad \Rightarrow \quad \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & & \ddots & & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right]$$

in the previous example:

$$\begin{aligned} x + 2y + z &= 5 \\ 3x + 2y + 4z &= 17 \\ 4x + 4y + 3z &= 26 \end{aligned} \quad \Rightarrow \quad \left[\begin{array}{ccc|c} 1 & 2 & 1 & 5 \\ 3 & 2 & 4 & 17 \\ 4 & 4 & 3 & 26 \end{array}\right]$$

The process of eliminating variables from the equations, or, equivalently, zeroing entries of the corresponding matrix, in order to reduce the system to upper-triangular form is called row reduction.

For example, for a system of n equations with n unknowns:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & & \ddots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{bmatrix} \Rightarrow \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} & b_1' \\ 0 & u_{22} & \cdots & u_{2n} & b_2' \\ 0 & 0 & \ddots & & b_3' \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} & b_n' \end{bmatrix}$$

The process of solving a linear system implies reducing the augmented matrix into a special form called row echelon form:

A matrix is in row echelon form when:

- If a row does not consist entirely of zeros, then the first number in the row is a 1 which we call a leading 1
- If there are any rows that contain only zeros, then they are grouped together at the bottom of the matrix
- In any two successive rows that do not consist entirely of zeros, the leading 1 in the lower row occurs farther to the right than the leading 1 in the higher row

$$\begin{bmatrix} 1 & 4 & -3 & 7 \\ 0 & 1 & 6 & 2 \\ 0 & 0 & 1 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 2 & 6 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

If a matrix in row echelon form has also the following property:

- Each column that contains a leading 1 has zeros everywhere else

it is said to be in reduced echelon form.

$$\begin{bmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 5 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The following two facts about row echelon forms and reduced echelon forms are important for the computational procedures for solving linear systems:

- Every matrix has a unique reduced row echelon form
- Row echelon forms are not unique, however all of the row echelon forms have their leading 1's in the same position and all have the same number of zero rows at the bottom

The positions of the leading 1's are called the pivot positions in the augmented matrix, and the columns that contain the leading 1's are called the pivot columns.

Suppose the following reduced augmented matrix for a linear system in the unknowns x_1, x_2, x_3, x_4:

$$\left[\begin{array}{ccccc} 1 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 5 \end{array}\right] \Rightarrow \begin{aligned} x_1 &= 3 \\ x_2 &= -1 \\ x_3 &= 0 \\ x_4 &= 5 \end{aligned}$$

From this example it is evident that finding the reduced echelon form of the augmented matrix for a linear system is equivalent to finding its solution.

The steps used in solving a linear system involve performing basic algebraic operations on the equations to produce a succession of increasingly simpler systems that have the same solution as the original one:

- Multiply an equation through by a nonzero constant
- Interchange two equations
- Add a multiple of one equation to another

Since the rows of an augmented matrix correspond to equations in the associated system, the three algebraic manipulations above correspond to the following operations on rows of the augmented matrix:

- Multiply a row through by a nonzero constant
- Interchange two rows
- Add a multiple of one row to another

These three operations are called elementary row operations on a matrix.

There are two basic algorithms to solve linear systems, the Gaussian elimination and the Gauss-Jordan algorithm:

Gaussian elimination	reduction of augmented matrix to row echelon form
Gauss – Jordan algorithm	reduction of augmented matrix to reduced row echelon form

Gaussian elimination does not give the final solution and it must be complemented by an algebraic procedure called back substitution.

The Gauss-Jordan algorithm has two phases: a forward phase (Gaussian elimination) to reduce the augmented matrix to row echelon form, and a backward phase to introduce 0's above the pivots to reach the reduced row echelon form for the augmented matrix.

For the system presented earlier as an example to solve linear systems, Gaussian elimination proceeds as follows :

$$\begin{aligned} x+2y+z &= 5 \\ 3x+2y+4z &= 17 \\ 4x+4y+3z &= 26 \end{aligned} \quad \Rightarrow \quad \left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 3 & 2 & 4 & 17 \\ 4 & 4 & 3 & 26 \end{array}\right] \quad r_2 - 3r_1 \Rightarrow$$

$$\left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 0 & -4 & 1 & 2 \\ 4 & 4 & 3 & 26 \end{array}\right] \quad -¼ * r_2 \Rightarrow \quad \left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2} \\ 4 & 4 & 3 & 26 \end{array}\right] \quad r_3 - 4r_1 \Rightarrow \quad \left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2} \\ 0 & -4 & -1 & 6 \end{array}\right] \quad r_3 + 4r_2 \Rightarrow$$

$$\left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2} \\ 0 & 0 & -2 & 4 \end{array}\right] \quad -½ * r_3 \Rightarrow \quad \left[\begin{array}{cccc} 1 & 2 & 1 & 5 \\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2} \\ 0 & 0 & 1 & -2 \end{array}\right] \quad \Rightarrow \quad \begin{aligned} x+2y+z &= 5 \\ y-\frac{1}{4}z &= -\frac{1}{2} \\ z &= -2 \end{aligned}$$

The linear system is now in upper-triangular form with unit coefficients on the leading variables in each equation:

$$\begin{aligned} x+2y+z &= 5 \\ y-\frac{1}{4}z &= -\frac{1}{2} \\ z &= -2 \end{aligned}$$

Starting from below and introducing the results in the equation immediately above we get:

$$z=-2$$

$$y=-1$$

$$x=9$$

The first phase of the Gauss-Jordan algorithm is Gaussian elimination to reduce the augmented matrix to row echelon form:

$$\begin{aligned} x+2y+z&=5\\ 3x+2y+4z&=17\\ 4x+4y+3z&=26 \end{aligned} \Rightarrow \left[\begin{array}{cccc} 1 & 2 & 1 & 5\\ 3 & 2 & 4 & 17\\ 4 & 4 & 3 & 26 \end{array}\right] \xrightarrow{\text{Gaussian elimination}} \left[\begin{array}{cccc} 1 & 2 & 1 & 5\\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2}\\ 0 & 0 & 1 & -2 \end{array}\right]$$

In the backward phase we introduce 0 in the columns of the pivots

$$\left[\begin{array}{cccc} 1 & 2 & 1 & 5\\ 0 & 1 & -\frac{1}{4} & -\frac{1}{2}\\ 0 & 0 & 1 & -2 \end{array}\right] \xrightarrow{r_2+4r_3} \left[\begin{array}{cccc} 1 & 2 & 1 & 5\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & -2 \end{array}\right] \xrightarrow{r_1-r_3} \left[\begin{array}{cccc} 1 & 2 & 0 & 7\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & -2 \end{array}\right] \xrightarrow{r_1-2r_2}$$

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 9\\ 0 & 1 & 0 & -1\\ 0 & 0 & 1 & -2 \end{array}\right] \Rightarrow \begin{aligned} x &= 9\\ y &= -1\\ z &= -2 \end{aligned}$$

In the following I will introduce some homework assignments that are due at the end of the course (check the course schedule for the exact date). During the first week of the intensive course you will have time to work on these problems under my advise, but feel free to start before if you want. Many of the programs I propose in the slides are closely related, so that when you finish one you may reuse it to write the next one by adding slight modifications. A wise way to proceed is to write reusable parts of the programs as subroutines.

During the intensive course I will propose you two extra assignments where you will have to to solve some applied problems that need the solution of a linear system and the diagonalization of a matrix, respectively. These two extra assignments will be the only ones that you will have to upload to the MOODLE platform, the ones proposed in the slides are just for practice (although you will need to use some of them in the extra assignments).

For each of the two assignments, please include a pdf document explaining the details of your algorithm with a version of the program in pseudocode, and the corresponding FORTRAN program (only source code) with comment lines indicating your name, the function of the variables you use, and any important information you consider that is necessary to follow your code.

Please check before submitting your assignments that your subroutines really work by testing them with some examples. Include also some example input and output files for each program.

1a) Gaussian elimination

Write a program to reduce the augmented matrix of a linear system to row echelon form using the Gaussian elimination procedure.

1b) Back substitution algorithm

Write a program to solve an upper triangular linear system by the algebraic back substitution procedure and couple it to program 1a to solve the linear system.

2) Gauss - Jordan algorithm

Write a program to find a solution of a linear system by the Gauss – Jordan algorithm. Use your program 1a to perform the forward phase and complete it with the code to perform the backward phase.

Recall that we have used three type of elementary row operations on the augmented matrix to solve a linear system

- Multiply a row through by a nonzero constant
- Interchange two rows
- Add a multiple of one row to another

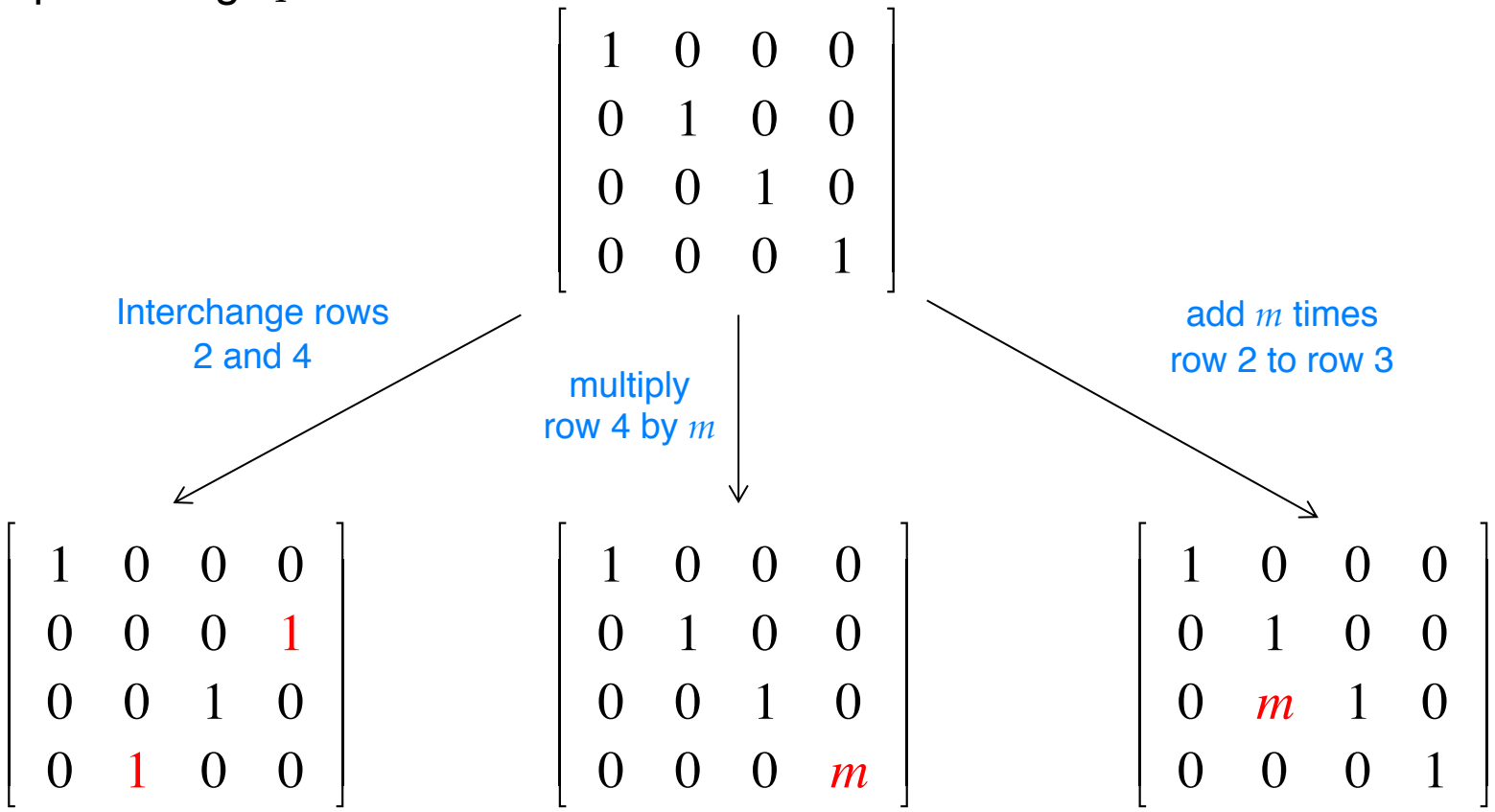
These operations can be performed by multiplying the augmented matrix $\mathbf{A}$ by an appropriate elementary matrix $\mathbf{E}$:

$$\mathbf{M = EA}$$

$$\begin{bmatrix} 1 & 2 & 1 & 5 \\ 0 & -4 & 1 & 2 \\ 4 & 4 & 3 & 26 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 & 5 \\ 3 & 2 & 4 & 17 \\ 4 & 4 & 3 & 26 \end{bmatrix}$$

An elementary matrix is the result of applying a single elementary row operation to an identity matrix.

Examples using $\mathbf{I}_4$:



- Elementary matrices are always square.
- If A is an $m \times n$ matrix and $\mathbf{E}$ an elementary matrix obtained by performing an elementary row operation on the $m \times m$ identity matrix $\mathbf{I_m}$, then the product $\mathbf{EA}$ is the matrix that results when the same row operation is performed on $\mathbf{A}$.
- An elementary matrix is invertible, and the inverse is also an elementary matrix that can be obtained from the following row operations:

Row operation on $\mathbf{I}$ that produces $\mathbf{E}$		Row operation on $\mathbf{E}$ that produces $\mathbf{I}$
Multiply row i by $m \neq 0$	→	Multiply row I by $1/m$
Interchange rows i and j	→	Interchange rows i and j
Add m times row i to row j	→	Add -m times row i to row j

If $\mathbf{A}$ is an $n \times n$ matrix, then the following statements are equivalent, that is, they are all false or true:

- $\mathbf{A}$ is invertible
- The reduced echelon form of $\mathbf{A}$ is $\mathbf{I_n}$
- $\mathbf{A}$ is expressible as a product of elementary matrices

If a matrix $\mathbf{B}$ can be obtained from matrix $\mathbf{A}$ by performing a finite sequence of k row operations then:

$$\mathbf{B} = \mathbf{E_k} \ldots \mathbf{E_2}\ \mathbf{E_1}\mathbf{A}$$

Since elementary matrices are invertible we can recover $\mathbf{A}$ from $\mathbf{B}$ by:

$$\mathbf{A} = \mathbf{E_1}^{-1}\mathbf{E_2}^{-1}\ldots\mathbf{E_k}^{-1}\ \mathbf{B}$$

and $\mathbf{A}$ and $\mathbf{B}$ are said to be row equivalent.

A square matrix $\mathbf{A}$ is invertible if and only if it is row equivalent to the identity matrix of the same size.

To find the inverse of an invertible $n \times n$ matrix $\mathbf{A}$, find a sequence of elementary row operations that reduces $\mathbf{A}$ to $\mathbf{I_n}$, and then perform the same sequence on $\mathbf{I_n}$ to obtain $\mathbf{A^{-1}}$

Indeed, if

$$\mathbf{I_n = (E_k \ldots E_2\ E_1)A}$$

and by definition of the inverse of a matrix:

$$\mathbf{(E_k \ldots E_2\ E_1) = A^{-1} = (E_k \ldots E_2\ E_1)\ I_n}$$

In practice, to find the inverse of a $n \times n$ matrix $\mathbf{A}$ we create a partitioned $n \times 2n$ matrix $[\mathbf{A} \mid \mathbf{I_n}]$ and perform elementary row operations until the left side is reduced to $\mathbf{I_n}$. Those operations will produce the inverse of $\mathbf{A}$ in the right side of the expanded matrix that will turn into $[\mathbf{I_n} \mid \mathbf{A^{-1}}]$.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 5 & 3 \\ 1 & 0 & 8 \end{bmatrix} \Rightarrow \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 5 & 3 & 0 & 1 & 0 \\ 1 & 0 & 8 & 0 & 0 & 1 \end{array}\right] \xrightarrow{r_2-2r_1,\ r_3-r_1} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & -2 & 5 & -1 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{r_3+2r_2} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & -1 & -5 & 2 & 1 \end{array}\right] \xrightarrow{-1r_3} \left[\begin{array}{ccc|ccc} 1 & 2 & 3 & 1 & 0 & 0 \\ 0 & 1 & -3 & -2 & 1 & 0 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\xrightarrow{r_2+3r_3,\ r_1-3r_3} \left[\begin{array}{ccc|ccc} 1 & 2 & 0 & -14 & 6 & 3 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right] \xrightarrow{r_1-2r_1} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -40 & 16 & 9 \\ 0 & 1 & 0 & 13 & -5 & -3 \\ 0 & 0 & 1 & 5 & -2 & -1 \end{array}\right]$$

$$\Rightarrow A^{-1} = \begin{bmatrix} -40 & 16 & 9 \\ 13 & -5 & -3 \\ 5 & -2 & -1 \end{bmatrix}$$

3) Matrix inversion via Gaussian elimination

Write a program to invert a square $n \times n$ matrix using the inversion algorithm.

The idea behind Gaussian elimination is to convert a linear system $\mathbf{Ax=b}$ into an equivalent triangular system $\mathbf{Ux=b'}$ by performing elementary row operations on the augmented matrix $\mathbf{[A \mid b]}$.

Often we have to solve a set of related linear systems $\mathbf{Ax=b}$; $\mathbf{Ax=c}$; $\mathbf{Ax=d}$ … where applying Gaussian elimination to each system is a waste of time. For this reason, the algorithm of choice is the so-called LU-decomposition:

1. Rewrite the system $\mathbf{Ax = b}$ as $\mathbf{LUx = b}$ where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular matrices, respectively
2. Define a new unknown y by letting $\mathbf{Ux = y}$ and rewrite the system as $\mathbf{Ly = b}$
3. Solve $\mathbf{Ly = b}$ for $\mathbf{y}$
4. Substitute $\mathbf{y}$ in $\mathbf{Ux = y}$ and solve for $\mathbf{x}$

Although we have now to solve two linear systems $\mathbf{Ly=b}$ and $\mathbf{Ux=y}$, both are triangular and can be solved easily by forward and back substitution, respectively.

Moreover, the LU-decomposition (step 1) is valid for all systems involving matrix $\mathbf{A}$ and we must perform this time consuming step only once.

Let us solve the following system using the LU-decomposition:

$$\begin{aligned} 2x_1 + 6x_2 + 2x_3 &= 2 \\ -3x_1 - 8x_2 &= 2 \\ 4x_1 + 9x_2 + 2x_3 &= 3 \end{aligned} \quad \Rightarrow \quad \mathbf{A} = \begin{bmatrix} 2 & 6 & 2 \\ -3 & -8 & 0 \\ 4 & 9 & 2 \end{bmatrix} = \underbrace{\begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{U}}$$

Step 1:

$$\begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 2 & 0 & 0 \\ -3 & 1 & 0 \\ 4 & -3 & 7 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 3 \end{bmatrix}$$

$\mathbf{Ux = y}$ $\mathbf{Ly = b}$

forward substitution $\Rightarrow$ $y_1 = 1$, $y_2 = 5$, $y_3 = 2$ ($\mathbf{y}$)

Step 2:

$$\begin{bmatrix} 1 & 3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 5 \\ 2 \end{bmatrix}$$

$\mathbf{Ux = y}$

backward substitution $\Rightarrow$ $x_1 = 2$, $x_2 = -1$, $x_3 = 2$

A factorization of a square matrix $\mathbf{A}$ as $\mathbf{A} = \mathbf{LU}$ where $\mathbf{L}$ is lower triangular and $\mathbf{U}$ is upper triangular is called a LU-decomposition of $\mathbf{A}$. In general, not every matrix has an LU-decomposition, nor is an LU-decomposition unique if it exists.

However, if $\mathbf{A}$ can be reduced to row echelon form without row interchanges, then A must have an LU-decomposition.

Gaussian elimination may be thought as a product of elementary matrices:

$$\mathbf{U} = \mathbf{E}_k \ldots \mathbf{E}_2 \, \mathbf{E}_1 \mathbf{A}$$

Since these are invertible

$$\mathbf{A} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \ldots \mathbf{E}_k^{-1} \, \mathbf{U}$$

If we do not perform row interchanges it can be shown that

$$\mathbf{L} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \ldots \mathbf{E}_k^{-1}$$

is a lower triangular matrix

As shown in the following example, $\mathbf{L}$ can be constructed during the Gaussian elimination process just by clever bookeeping of the multipliers used in it.

$$\mathbf{A}=\begin{bmatrix}6&-2&0\\9&-1&1\\3&7&5\end{bmatrix}\qquad\begin{bmatrix}\bullet&0&0\\\bullet&\bullet&0\\\bullet&\bullet&\bullet\end{bmatrix}$$

multiplier = 1/6

$$\begin{bmatrix}1&-1/3&0\\9&-1&1\\3&7&5\end{bmatrix}\qquad\begin{bmatrix}6&0&0\\\bullet&\bullet&0\\\bullet&\bullet&\bullet\end{bmatrix}$$

multiplier = -9

multiplier = -3

$$\begin{bmatrix}1&-1/3&0\\0&2&1\\0&8&5\end{bmatrix}\qquad\begin{bmatrix}6&0&0\\9&\bullet&0\\3&\bullet&\bullet\end{bmatrix}$$

multiplier = 1/2

$$\begin{bmatrix}1&-1/3&0\\0&1&1/2\\0&8&5\end{bmatrix}\qquad\begin{bmatrix}6&0&0\\9&2&0\\3&\bullet&\bullet\end{bmatrix}$$

multiplier = -8

$$\begin{bmatrix}1&-1/3&0\\0&1&1/2\\0&0&1\end{bmatrix}\qquad\begin{bmatrix}6&0&0\\9&2&0\\3&8&\bullet\end{bmatrix}$$

multiplier = 1

$$\mathbf{U}=\begin{bmatrix}1&-1/3&0\\0&1&1/2\\0&0&1\end{bmatrix}\qquad\begin{bmatrix}6&0&0\\9&2&0\\3&8&1\end{bmatrix}=\mathbf{L}$$

$$\mathbf{A}=\begin{bmatrix}6&-2&0\\9&-1&1\\3&7&5\end{bmatrix}$$

$$=\begin{bmatrix}6&0&0\\9&2&0\\3&8&1\end{bmatrix}\begin{bmatrix}1&-1/3&0\\0&1&1/2\\0&0&1\end{bmatrix}$$

$$=\mathbf{LU}$$

Many of the best algorithms for inverting matrices use LU-decomposition.

Let $\mathbf{A}$ be an invertible matrix partitioned into column vectors

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_n \end{bmatrix}$$

and let

$$\mathbf{I}_n = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_n \end{bmatrix}$$

be the $n \times n$ identity matrix partitioned into column vectors.

Then, finding the inverse of $\mathbf{A}$:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}_n$$

Is equivalent to solve the following n linear systems:

$$\mathbf{A}\mathbf{x}_1 = \mathbf{e}_1; \quad \mathbf{A}\mathbf{x}_2 = \mathbf{e}_2; \quad \cdots \quad \mathbf{A}\mathbf{x}_n = \mathbf{e}_n;$$

a task that can be easily performed by using the LU-decomposition of $\mathbf{A}$.

Computing the determinant of a square $n \times n$ matrix $\mathbf{A}$ using its definition requires a number of arithmetic operations that is exponential in n.

However more practical methods can be obtained considering that:

- If $\mathbf{A}'$ is obtained by adding a multiple of a row of $\mathbf{A}$ to another row, then $\det(\mathbf{A}')=\det(\mathbf{A})$
- If $\mathbf{B}$ is a $n \times n$ matrix, then $\det(\mathbf{AB}) = \det(\mathbf{A}) \det(\mathbf{B})$
- If $\mathbf{A}$ is triangular, then $\det(\mathbf{A}) = \prod_{i=1}^{n} a_{ii}$

It follows from these properties that if a LU-decomposition of $\mathbf{A}$ is possible, then

$$\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L}) \det(\mathbf{U}) = \prod_{i=1}^{n} l_{ii} \prod_{i=1}^{n} u_{ii}$$

Calculating a determinant via an LU-decomposition requires $O(n^3)$ operations.

4a) LU-decomposition

Write a subroutine to perform the LU-decomposition of a square $n \times n$ matrix.
Suggestion: use your program in exercise (1a) introducing the code to construct matrix $\mathbf{L}$ during the Gaussian elimination process.

4b) Solving linear systems via LU-decomposition

Write a program that uses your LU decomposition routine to solve linear systems with n equations in n variables.

4c) Matrix inversion via LU-decomposition

Write a program that uses your LU decomposition routine to invert a square $n \times n$ matrix.

4d) Determinant via LU-decomposition

Write a program that uses your LU decomposition routine calculate the determinant of a square $n \times n$ matrix.

Let us consider the following system and solve it via Gaussian elimination:

$$\begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} 1 \\ 2 \end{pmatrix} \Rightarrow \begin{pmatrix} \varepsilon & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} \xrightarrow{r_2-\varepsilon^{-1}r_1} \begin{pmatrix} \varepsilon & 1 & 1 \\ 0 & 1-\varepsilon^{-1} & 2-\varepsilon^{-1} \end{pmatrix}$$

Backward substitution yields:

$$x_2 = \frac{2-\varepsilon^{-1}}{1-\varepsilon^{-1}}$$

$$x_1 = (1-x_2)\varepsilon^{-1}$$

For $\varepsilon = 0.1$

$$x_2 = \frac{2-10}{1-10} = \frac{-8}{-9} = 0.88889$$

$$x_1 = (1-0.88889)\times 10 = 1.11111$$

For $\varepsilon = 0.0001$

$$x_2 = \frac{2-10{,}000}{1-10{,}000} = \frac{-9{,}998}{-9{,}999} = 0.9999 \approx 1$$

$$x_1 = (1-0.9999)\times 10{,}000 = 1.0001 \approx 1$$

finite precision arithmetic

$$x_1 \approx (1-1)\times 10{,}000 = 0$$

In Gaussian elimination, small pivots (large multipliers) lead to numerical instability

The problem created by small pivots can be avoided by swapping rows:

$$\begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}=\begin{pmatrix} 2 \\ 1 \end{pmatrix} \Rightarrow \begin{pmatrix} 1 & 1 & 2 \\ \varepsilon & 1 & 1 \end{pmatrix} \xrightarrow{r_2-\varepsilon r_1} \begin{pmatrix} 1 & 1 & 2 \\ 0 & 1-\varepsilon & 1-2\varepsilon \end{pmatrix}$$

Backward substitution yields:

$$x_2 = \frac{1-2\varepsilon}{1-\varepsilon}$$

$$x_1 = (2-x_2) = \frac{1}{1-\varepsilon}$$

For $\varepsilon = 0.1$

$$x_2 = \frac{1-2\times 0.1}{1-0.1} = \frac{0.8}{0.9} = 0.88889$$

$$x_1 = \frac{1}{(1-0.1)} = 1.11111$$

For $\varepsilon = 0.0001$

$$x_2 = \frac{1-2\times 0.0001}{1-0.0001} = \frac{0.9998}{0.9999} = 0.9999$$

$$x_1 = \frac{1}{(1-0.0001)} = 1.0001$$

finite precision arithmetic

$$x_1 \approx x_1 = \frac{1}{(1-0)} = 1.000$$

After (k-1) steps, Gaussian elimination leads to:

$$\mathbf{A}^{(k-1)} = \begin{pmatrix} a_{11} & a_{12} & \cdots & & \cdots & & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & & \cdots & & a_{2n}^{(1)} \\ \vdots & & \ddots & & & & \vdots \\ 0 & \cdots & 0 & a_{k-1.k-1}^{(k-2)} & a_{k-1.k}^{(k-2)} & \cdots & a_{k-1.n}^{(k-2)} \\ 0 & \ddots & & 0 & a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & a_{nk}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{pmatrix}$$

pivot for k^{th} step

Pivoting consists in avoiding small pivots for a given step by exchanging rows (or columns) below (or on the right) of the pivot.

Warning: Gaussian elimination with pivoting does not guarantee an LU-decomposition of $\mathbf{A}$

Partial pivoting: $\left|a_{mk}^{(k-1)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k-1)}\right|$ ⇒ Swap rows k and m if $k \neq m$

$$\mathbf{A^{(k-1)}} = \begin{pmatrix} a_{11} & a_{12} & \cdots & & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & & \cdots & a_{2n}^{(1)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & a_{k-1.k-1}^{(k-2)} & a_{k-1.k}^{(k-2)} \cdots & a_{k-1.n}^{(k-2)} \\ 0 & \ddots & & 0 & a_{kk}^{(k-1)} \cdots & a_{kn}^{(k-1)} \\ \vdots & & \ddots & \vdots & \vdots \ddots & \vdots \\ 0 & \cdots & 0 & 0 & a_{nk}^{(k-1)} \cdots & a_{nn}^{(k-1)} \end{pmatrix}$$

pivot for k^{th} step

search for largest $|a_{mk}|$ value below pivot and swap rows if it is not a_{kk}

Complete pivoting: $\left|a_{mp}^{(k-1)}\right| = \max_{\substack{k\le i\le n \\ k\le j\le n}} \left|a_{ij}^{(k-1)}\right|$ ⇒ Swap rows k *and* m and columns p and k if $k \neq m$ *or* $p \neq k$

$$\mathbf{A^{(k-1)}} = \begin{pmatrix} a_{11} & a_{12} & \cdots & & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & & \cdots & a_{2n}^{(1)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & a_{k-1.k-1}^{(k-2)} & a_{k-1.k}^{(k-2)} \cdots & a_{k-1.n}^{(k-2)} \\ 0 & \ddots & & 0 & a_{kk}^{(k-1)} \cdots & a_{kn}^{(k-1)} \\ \vdots & & \ddots & \vdots & \vdots \ddots & \vdots \\ 0 & \cdots & 0 & 0 & a_{nk}^{(k-1)} \cdots & a_{nn}^{(k-1)} \end{pmatrix}$$

pivot for k^{th} step

search for largest $|a_{mp}|$ value and swap rows and columns if it is not a_{kk}

Warning: swapping columns means changing variable names!

5) Gaussian elimination with partial pivoting

Write a program to solve linear systems using Gaussian elimination with partial pivoting

Howard Anton, Robert C. Busby
Contemporary Linear Algebra, Wiley, Hoboken, NJ (2003), Chapters 1-3

Gene H. Golub, Charles F. Van Loan
Matrix Computations, 3rd Ed., The Johns Hopkins University Press, Baltimore (1996), Chapter 3