Matrices

Bioinformatics Degree
Algebra

Departament de Matemàtiques

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Outline

Outline

Definition

An $m \times n$ matrix is a collection of $m \times n$ (real or complex) numbers arranged into a rectangular array of m rows and n columns.

The entry $a_{i,j}$ is the element at row i and column j of A.

Notation: $A = (a_{i,j})$.

- If $m = n$, A is a square matrix of size n.
- The set of $m \times n$ matrices is denoted by $\mathcal{M}_{m,n}$.
- The elements of $\mathcal{M}_{n,1}$ are called vectors or column vectors.
- The elements of $\mathcal{M}_{1,n}$ are called row vectors.

Special matrices

- The matrix $\mathbf{0}$ is the matrix whose elements are all 0.
- A square matrix A is a diagonal matrix if $a_{i,j} = 0$ for all $i \neq j$.
- The identity matrix Id_n is the diagonal $n \times n$ matrix that has 1's at the diagonal entries.
- A square matrix A is a lower triangular matrix if $a_{i,j} = 0$ for all $i < j$.
- A square matrix A is an upper triangular matrix if $a_{i,j} = 0$ for all $i > j$.

Transpose

The transpose of $A \in \mathcal{M}_{m,n}$ is the $n \times m$ matrix A^t whose (i,j)-entry is $a_{j,i}$:

$$A = \begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix} \to A^t = \begin{pmatrix} a_{1,1} & \cdots & a_{m,1} \\ \vdots & \vdots & \vdots \\ a_{1,n} & \cdots & a_{m,n} \end{pmatrix}$$

- A square matrix is symmetric if $A^t = A$

Outline

Sum of matrices

If A, B are two $m \times n$ matrices, then the sum $A + B$ is the matrix whose (i,j)-entry is $c_{i,j} = a_{i,j} + b_{i,j}$:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} + \begin{pmatrix} b_{1,1} & \dots & b_{1,n} \\ \vdots & \vdots & \vdots \\ b_{m,1} & \dots & b_{m,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} + b_{1,1} & \dots & a_{1,n} + b_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} + b_{m,1} & \dots & a_{m,n} + b_{m,n} \end{pmatrix}$$

Properties: associative, commutative, neutral element $\mathbf{0}$, opposite element $-A = (-a_{i,j})$,

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} & \dots & a_{m,n} \end{pmatrix} - \begin{pmatrix} b_{1,1} & \dots & b_{1,n} \\ \vdots & \vdots & \vdots \\ b_{m,1} & \dots & b_{m,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} - b_{1,1} & \dots & a_{1,n} - b_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} - b_{m,1} & \dots & a_{m,n} - b_{m,n} \end{pmatrix}$$

$$(A + B)^t = A^t + B^t$$

Product by a scalar

Let $A \in \mathcal{M}_{m,n}$ and let $c \in \mathbb{R}$ be a number (scalar), then $c \cdot A$ is the $m \times n$ matrix whose (i, j)-element is $c\, a_{i,j}$ for all $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$:

$$c \cdot \begin{pmatrix} a_{1,1} & \ldots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{m,1} & \ldots & a_{m,n} \end{pmatrix} = \begin{pmatrix} c\, a_{1,1} & \ldots & c\, a_{1,n} \\ \vdots & \vdots & \vdots \\ c\, a_{m,1} & \ldots & c\, a_{m,n} \end{pmatrix}$$

Properties: $0 \cdot A = \mathbf{0}$, $c \cdot (A + B) = c \cdot A + c \cdot B$.

Multiplication of matrices

Let $A \in \mathcal{M}_{m,n}$ and $B \in \mathcal{M}_{n,p}$, then $A\,B$ is the matrix C such that

$$c_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,n}b_{n,j}.$$

Note that $c_{i,j} = (a_{i,1}\; a_{i,2} \ldots a_{i,n}) \begin{pmatrix} b_{1,j} \\ \vdots \\ b_{n,j} \end{pmatrix}$.

Example:

a) $$\begin{bmatrix} 1 & 2 & 4 \\ 3 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 & 2 \\ 4 & 6 & 1 \\ 3 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 22 & 23 & 24 \\ 29 & 41 & 16 \\ 17 & 18 & 20 \end{bmatrix}$$

b) $$\begin{bmatrix} 1 & 2 & 4 \\ 3 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 & 2 \\ 4 & 6 & 1 \\ 3 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 22 & 23 & 24 \\ 29 & 41 & 16 \\ 17 & 18 & 20 \end{bmatrix}$$

c) $$\begin{bmatrix} 1 & 2 & 4 \\ 3 & 5 & 1 \\ 2 & 1 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 & 2 \\ 4 & 6 & 1 \\ 3 & 2 & 5 \end{bmatrix} = \begin{bmatrix} 22 & 23 & 24 \\ 29 & 41 & 16 \\ 17 & 18 & 20 \end{bmatrix}$$

Properties of matrix multiplication

- $Id_n\, A = A\, Id_n = A$ (neutral element).
- $A\,(B\,C) = (A\,B)\,C$ (associative).
- $A\,(B + C) = A\,B + A\,C$ (distributive law).
- $(A + B)\,C = A\,C + B\,C$ (distributive law).
- $A\,B \neq B\,A$.
- $(A\,B)^t = B^t\,A^t$.

Given a matrix A, under which conditions does there exist a matrix B such that

$$A\,B = B\,A = Id_n\,?$$

Properties of matrix multiplication

- $Id_n\, A = A\, Id_n = A$ (neutral element).
- $A\,(B\,C) = (A\,B)\,C$ (associative).
- $A\,(B + C) = A\,B + A\,C$ (distributive law).
- $(A + B)\,C = A\,C + B\,C$ (distributive law).
- $A\,B \neq B\,A$.
- $(A\,B)^t = B^t\,A^t$.

Given a matrix A, under which conditions does there exist a matrix B such that

$$A\,B = B\,A = Id_n\,?$$

Inverse

Let A be an $n \times n$ matrix. If there exists a matrix B such that

$$A\,B = B\,A = Id_n$$

then B is called the inverse of A and is denoted as A^{-1}.

A matrix is called invertible (or non-singular) if it has an inverse and is called singular if it does NOT have an inverse.

Remark. Only $A\,B = Id_n$ or $B\,A = Id_n$ is necessary (the other comes for free).

Properties of the inverse

If A and B are $n \times n$ invertible matrices, then

- The inverse is unique.
- $(A^{-1})^{-1} = A$.
- $(A^t)^{-1} = (A^{-1})^t$.
- $(A\,B)^{-1} = B^{-1}A^{-1}$.
- $(A^k)^{-1} = (A^{-1})^k$ for $k \in \mathbb{N}$

Inverse in the 2×2 case

If $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ and $ad - bc \neq 0$, then

$$A^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}.$$

Computing the inverse for larger matrices: see the section "Determinant" and the next topic ("Linear Systems").

Outline

Elementary operations

Given an $m \times n$ matrix A, the following are called row elementary transformations

- E_1 Exchange two rows.
- E_2 Multiply a row by a nonzero constant.
- E_3 Add a multiple of one row to another row.

Similarly, we could define the column elementary transformations.

Elementary operations

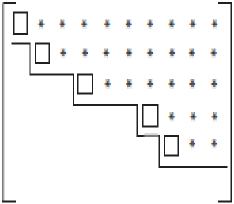
Given an $m \times n$ matrix A, the following are called row elementary transformations

- E_1 Exchange two rows.
- E_2 Multiply a row by a nonzero constant.
- E_3 Add a multiple of one row to another row.

Similarly, we could define the column elementary transformations.

Row echelon form

Gaussian elimination is an algorithm that uses row elementary transformations to transform a matrix to a matrix with row echelon form:



- □: first non-zero element of each row (pivots).
- $*$: can be 0 or not.
- Everything below the line is 0.
- Every pivot is further to the right than the pivot of the previous row.

Gaussian elimination:

Any non-zero matrix can be transformed into a matrix with row echelon form by using row elementary transformations to repeat these steps for each column from left to right:

1. If it is possible, choose a pivot and put it as high as possible (E1).
2. Put zeros below the pivot (E3).

Remark: We can transform a matrix into row echelon form by doing elementary transformations in many different ways. However, all of them lead to the same number of pivots.

Gaussian elimination:

Any non-zero matrix can be transformed into a matrix with row echelon form by using row elementary transformations to repeat these steps for each column from left to right:

1. If it is possible, choose a pivot and put it as high as possible (E1).
2. Put zeros below the pivot (E3).

Remark: We can transform a matrix into row echelon form by doing elementary transformations in many different ways. However, all of them lead to the same number of pivots.

Outline

Rank

The rank of a matrix A is the number of pivots (=the number of nonzero rows) in a row echelon form of A.

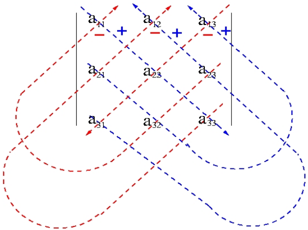
Properties:

- The *rank* does not change if we perform elementary operations on a matrix.
- $rank(A) = rank(A^t)$.

Determinant of a 3 × 3 matrix

Sarrus Rule:

$$\begin{vmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \end{vmatrix} = a_{1,1}a_{2,2}a_{3,3} + a_{2,1}a_{3,2}a_{1,3} + a_{3,1}a_{1,2}a_{2,3} - a_{1,3}a_{2,2}a_{3,1} - a_{2,3}a_{3,2}a_{1,1} - a_{3,3}a_{1,2}a_{2,1}$$



Warning: Not valid for $n \geq 4$.

Definition of determinant

Let A be an $n \times n$ matrix, we define the determinant of A, $\det(A)$, as follows (notation $|A| = det(A)$):

- If $n = 1$: $A = (a_{1,1})$, then $\det(A) = a_{1,1}$.
- If $n = 2$: $\det(A) = \begin{vmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{vmatrix} = a_{1,1}|a_{2,2}| - a_{1,2}|a_{2,1}|$.
- If $n = 3$,

$$\det(A) = a_{11} \begin{vmatrix} a_{2,2} & a_{2,3} \\ a_{3,2} & a_{3,3} \end{vmatrix} - a_{1,2} \begin{vmatrix} a_{2,1} & a_{2,3} \\ a_{3,1} & a_{3,3} \end{vmatrix} + a_{1,3} \begin{vmatrix} a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{vmatrix}$$

Definition of determinant

- Recursively, if $A_{i,j}$ is the matrix obtained by removing row i and column j from A,

  $$|A| = a_{11} \det A_{1,1} - a_{1,2} \det A_{1,2} + \cdots + (-1)^{n+1} a_{1,n} \det A_{1,n}.$$

  The expression above is called the Laplace expansion of the determinant by the first row.

Laplace expansion Theorem

Given a square matrix A, we define the cofactor matrix of A as the matrix $co(A)$ whose (i, j) entry is

$$C_{i,j} = (-1)^{i+j} \det A_{i,j},$$

where $A_{i,j}$ is the matrix obtained by removing the row i and the column j of A.

Theorem (Laplace expansion)

The determinant of an $n \times n$ matrix A can be computed as the cofactor expansion along the i-th row,

$$\det A = a_{i,1} C_{i,1} + \ldots + a_{i,n} C_{i,n}$$

and also as the cofactor expansion along the j-th column:

$$\det A = a_{1,j} C_{1,j} + \ldots + a_{n,j} C_{n,j}.$$

Laplace expansion Theorem

Given a square matrix A, we define the cofactor matrix of A as the matrix $co(A)$ whose (i,j) entry is

$$C_{i,j} = (-1)^{i+j} \det A_{i,j},$$

where $A_{i,j}$ is the matrix obtained by removing the row i and the column j of A.

Theorem (Laplace expansion)

The determinant of an $n \times n$ matrix A can be computed as the cofactor expansion along the i-th row,

$$\det A = a_{i,1} C_{i,1} + \ldots + a_{i,n} C_{i,n}$$

and also as the cofactor expansion along the j-th column:

$$\det A = a_{1,j} C_{1,j} + \ldots + a_{n,j} C_{n,j}.$$

Effect of elementary transformations on det

Let A be a square matrix.

E_1 If B is obtained by exchanging two rows/columns of A, then:

$$\det(B) = -\det(A)$$

E_2 If B is obtained by multiplying a row/column by $c \neq 0$, then

$$\det(B) = c\det(A).$$

E_3 If B is obtained by changing one row/column by itself plus a multiple of another row/column, then

$$\det(B) = \det(A).$$

Goal: Do transformations of type E_3 (and of type E_1 if necessary) to compute efficiently $det(A)$.

Effect of elementary transformations on det

Let A be a square matrix.

E_1 If B is obtained by exchanging two rows/columns of A, then:

$$\det(B) = -\det(A)$$

E_2 If B is obtained by multiplying a row/column by $c \neq 0$, then

$$\det(B) = c\det(A).$$

E_3 If B is obtained by changing one row/column by itself plus a multiple of another row/column, then

$$\det(B) = \det(A).$$

Goal: Do transformations of type E_3 (and of type E_1 if necessary) to compute efficiently $det(A)$.

Properties of the determinant

Properties of the determinant:

- If one row or column is 0, then $det(A) = 0$.
- If A is a triangular matrix, det(A) is the product of elements in the diagonal. In particular, $det(Id_n) = 1$.
- $det(A^t) = det(A)$.
- $det(c \cdot A) = c^n det(A)$ (where n is the number of rows/columns of A).
- $det(A\,B) = det(A)\,det(B)$.

Consequence

If A is invertible (non-singular) $\Rightarrow$ $det(A^{-1}) = 1/det(A)(\neq 0)$.

Properties of the determinant

Properties of the determinant:

- If one row or column is 0, then $det(A) = 0$.
- If A is a triangular matrix, $\det(A)$ is the product of elements in the diagonal. In particular, $det(Id_n) = 1$.
- $det(A^t) = det(A)$.
- $det(c \cdot A) = c^n det(A)$ (where n is the number of rows/columns of A).
- $det(A\,B) = det(A)\,det(B)$.

Consequence
If A is invertible (non-singular) $\Rightarrow$ $det(A^{-1}) = 1/det(A)(\neq 0)$.

Determinants and rank

A minor of A is the determinant of a square submatrix of A (obtained by selecting some rows and columns of A).

Proposition

The maximum size of non-zero minors of A is equal to $rank(A)$.

This can be used to compute $rank(A)$ without transforming it into a matrix in row echelon form:

- An $n \times n$ matrix A has rank n (full rank) if and only if $\det(A) \neq 0$.
- If all $m \times m$ minors of A are 0 then $rank(A) < m$.

Existence of inverse

The adjugate or adjoint matrix is the transpose of the cofactor matrix. We have that

$$A^{-1} = \frac{1}{det(A)} co(A)^t$$

Warning! This is not the optimal way to compute the inverse for $n \geq 4$.

Theorem

For any square matrix A the following are equivalent:

- *A is invertible.*
- $\det(A) \neq 0$.
- *A has full rank.*

Existence of inverse

The adjugate or adjoint matrix is the transpose of the cofactor matrix. We have that

$$A^{-1} = \frac{1}{det(A)} co(A)^t$$

Warning! This is not the optimal way to compute the inverse for $n \geq 4$.

Theorem

For any square matrix A the following are equivalent:

- *A is invertible.*
- $\det(A) \neq 0$.
- *A has full rank.*

Outline

Linear systems

Definition

A system of m linear equations with n variables is a collection of equations

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

where the coefficients a_{ij}, the constant terms $b_1, b_2, \ldots, b_m$ and the values that the unknowns $x_1, x_2, \ldots, x_n$ are real numbers.

A system is homogenous if $b_i = 0$ for $i = 1, \ldots, m$.

Linear systems

A particular solution is a list of values for the unknowns $s = (s_1, \dots, s_n) \in \mathbb{R}^n$ that is a solution to all the equations.
The general solution is the set of all the solutions to the system.

Geometric interpretation
From a geometric point of view, the general solution to a linear system describes a linear variety (a point, a line, a plane, etc.).
Each particular solution is a point of the linear variety.

Matrix expression of a linear system

Any linear system can be put as a matrix equation $Ax = b$ by taking

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix}$$

The matrix A is called the matrix of the system.
The augmented matrix is $(A \mid b)$.

Number of solutions

Theorem

Any linear system has either (i) a unique solution, (ii) no solution, or (iii) an infinite number of solutions.

A linear system is consistent if it has one or more solutions. If it does not have solutions, it is inconsistent.

Example

(i) $\begin{cases} x_1 = 1 \\ x_2 = 2 \end{cases}$ *(ii)* $\begin{cases} x_1 + x_2 = 0 \\ x_1 + x_2 = 1 \end{cases}$ *(iii)* $\begin{cases} x_1 - x_2 = 0 \\ x_2 - x_2 = 0 \end{cases}$

Rouché-Frobenius Theorem

The matrix expression of linear systems of equations allow us to know how many solutions the system has:

Theorem (Rouché-Frobenius)

- *$Ax = b$ is consistent* **if and only if** *$rank(A) = rank(A|b)$.*

  In this case, its set of solutions depends on $n - rank(A)$ free variables. This value is known as the degrees of freedom of the system.

- *In particular, if $n = rank(A)$ the solution is unique.*

Outline

Solving systems: Gaussian elimination

Goal: convert the system $Ax = b$ to a simpler system using elementary transformations.
Consider the augmented matrix $(A \mid b)$ and

1st step Reduce $(A \mid b)$ to **row echelon form**.

2nd step Solve the system by **back substitution** if it is consistent.

- The number of pivots (rank) of the row echelon form of A and $(A|b)$ tells us whether the system is consistent or not.
- If the system is consistent, then the **leading variables** corresponding to pivots can be written in terms of the other variables (called **free variables**).
- The number of free variables is the **degrees of freedom** of the system.

Back substitution and Gauss-Jordan elimination

The back substitution step can also be performed by elementary row operations on the row echelon form of $(A|b)$ by **Gauss-Jordan elimination**:

Once we have a matrix in *row echelon form*, do:

1. start with the rightmost pivot and use an operation of type E_2 to convert it to 1.
2. from bottom to top: make all the entries above the pivot equal to zero using type E_3.
3. Repeat the previous steps the next column to the left (so, from right to left).

Reduced row echelon form

In this way we obtain a matrix in **row reduced echelon form**, that is a matrix of the following form:

$$A = \begin{pmatrix} 1 & * & 0 & 0 & * & * & 0 & * & 0 \\ 0 & 0 & 1 & 0 & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 1 & * & * & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Definition

A matrix is in **row reduced echelon form** if it is in row echelon form and

- all pivots are 1
- the pivots are the only non-zero entries in its column.

Row reduced echelon form

- If A square and the row reduced echelon form is Id_n, then $Ax = b$ can be trivially solved: the solution is the new independent term

$$(A \mid b) \sim \cdots \sim (Id_n \mid b') \quad \text{so} \quad Ax = b \Leftrightarrow Id_n x = b' \Leftrightarrow x = b'$$

- Whereas the row echelon form of A is not unique, the row *reduced* echelon form is unique.

Solving simultaneous systems

Goal: solve systems with the same $m \times n$ matrix A but different independent terms,

$$Ax^{(1)} = b^{(1)}, Ax^{(2)} = b^{(2)}, \dots, Ax^{(r)} = b^{(r)}.$$

Equivalently: find X $m \times r$ matrix such that

$$AX = \underbrace{\begin{pmatrix} b^{(1)} \, b^{(2)} \, \dots \, b^{(r)} \end{pmatrix}}_{B}.$$

matrix equation $AX = B$

Efficient solution: Gauss-Jordan elimination to the following augmented matrix

$$\begin{pmatrix} A \,|\, b^{(1)} \, b^{(2)} \, \dots \, b^{(r)} \end{pmatrix}$$

Solving simultaneous systems

Goal: solve systems with the same $m \times n$ matrix A but different independent terms,

$$Ax^{(1)} = b^{(1)}, Ax^{(2)} = b^{(2)}, \ldots, Ax^{(r)} = b^{(r)}.$$

Equivalently: find X $m \times r$ matrix such that

$$AX = \underbrace{\begin{pmatrix} b^{(1)} \, b^{(2)} \, \ldots \, b^{(r)} \end{pmatrix}}_{B}.$$

matrix equation $AX = B$

Efficient solution: Gauss-Jordan elimination to the following augmented matrix

$$\left(A \,|\, b^{(1)} \, b^{(2)} \, \ldots \, b^{(r)} \right)$$

Application: finding the inverse of a matrix

The previous algorithm is useful to find the inverse of a matrix.
Input: a square matrix A.
Output: the inverse of A if A is nonsingular, or that the inverse does not exist (if A is singular).

1. Form the $n \times 2n$ matrix $M = (A \mid Id_n)$
2. Reduce M to row echelon form (*Gaussian elimination*). This process generates a zero row in the left half of M if and only if A has no inverse.
3. Reduce the matrix to its row reduced echelon form (*Gauss-Jordan*). In the end, we obtain $M \sim (Id_n \mid B)$, where the identity matrix Id_n has replaced A in the left half.
4. Then $A^{-1} = B$, the matrix that is now on the right.

Outline

Python: numpy and linalg

- The numpy package allows us to work with matrices in python:
  `import numpy as np`
- We can use `array` to create matrices introducing them by rows:
  $\mathtt{A} = \mathtt{np.array}([[a_{11}, \ldots, a_{1n}], [a_{21}, \ldots, a_{2n}], \ldots, [a_{m1}, \ldots, a_{mn}]])$
- To visualize: `print(A)`
- To operate with matrices we need the linalg submodule of numpy:

```
from numpy.linalg import *
```

Python: Matrix operations

Command	Output
`np.zeros((m,n))`	the $m \times n$ zero matrix.
`np.identity(n)`	the $n \times n$ identity matrix.
`A.T`	the transpose of A.
`A+B`	the sum of matrices A and B.
`A@B` or `np.matmul(A, B)`	the product of matrices A and B.
`c*A`	the product of the matrix A by $c \in \mathbb{R}$.
`inv(A)`	the inverse of A.
`matrix_rank(A)`	the rank of A.
`det(A)`	the determinant of A.

Python for Linear Systems

```
import numpy as np
from numpy.linalg import *
```

$\texttt{A} = \texttt{np.array}([[\texttt{a}_{11}, \ldots, \texttt{a}_{1n}], [\texttt{a}_{21}, \ldots, \texttt{a}_{2n}], \ldots, [\texttt{a}_{n1}, \ldots, \texttt{a}_{nn}]])$

$\texttt{b} = \texttt{np.array}([\texttt{b}_1, \texttt{b}_2, \ldots, \texttt{b}_m])$

If A is an invertible square matrix, we can solve the system by using:

```
solve(A,b)
```