Linear Algebra for Computer Science

An incremental document:

From Systems and Matrices to Eigenvalues and Eigenvectors

Francisco Escolano

Notes based on the text
"Elementary Linear Algebra" by Larson and Falvo, 6th edition

Formulating & Solving Linear Systems

Gauss-Jordan

Linear vs non-Linear

Homogeneous Systems

Least squares
Curve fitting
Traffic problems
Dirichlet problems
Neural Networks

Echelon form

Vectors & Matrices

Matrices and Systems

Properties of matrices
Product
Transpose

Elementary Matrices

Inverse of a matrix

Application to Graphs

Linear Transformations

Properties
Matricial rep
Kernel and Range
Isomorphism
Geometric
Isometry

Robotics
Vision
Graphics

Vector Spaces & Matrices

Polynomials Lines, planes, hyperplanes

Spaces and subspaces
Linear combinations
Bases and dimension
Change of basis

Rank & Nullity

Dot & Cross products
Norms and projections

Stochastic matrices
PCA

Eigenvalues and Eigenvectors

Eigenvectors/values and transformations

Finding eigenpairs

Eigenspaces

Quadratic forms and their rotation

Similarity & Diagonalization

Graph characterization & PageRank

Matrix exponentiation

Systems of Differential equations

Solving an Homogeneous System per eigenvalue

For largest eigenvalue NO NEED OF system solving

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

Each eigenvalue determines a subspace and the dimensions indicate whether A is diagonalizable

Eigenpairs allow both lossless and lossy changes of basis (PCA)

Eigenvalues and Eigenvectors

Eigenvectors/values and transformations

Finding eigenpairs

Eigenspaces

Quadratic forms and their rotation

Similarity & Diagonalization

Graph characterization & PageRank

Matrix exponentiation

Systems of Differential equations

Eigenvectors and eigenvalues Define rotation matrices/axes In 2D and 3D

Symmetric Matrices have Real eigenvalues and are diagonalizable

Diagonalization enables new operations In matrices , e.g. expm,() logm(), sin(), some of them are useful in graphs

Spectra define the DNA of graphs & eigenvectors give the steady state of random walks

1. Formulating and solving linear systems

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Francisco Escolano

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Linear Systems are Geometric configurations

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

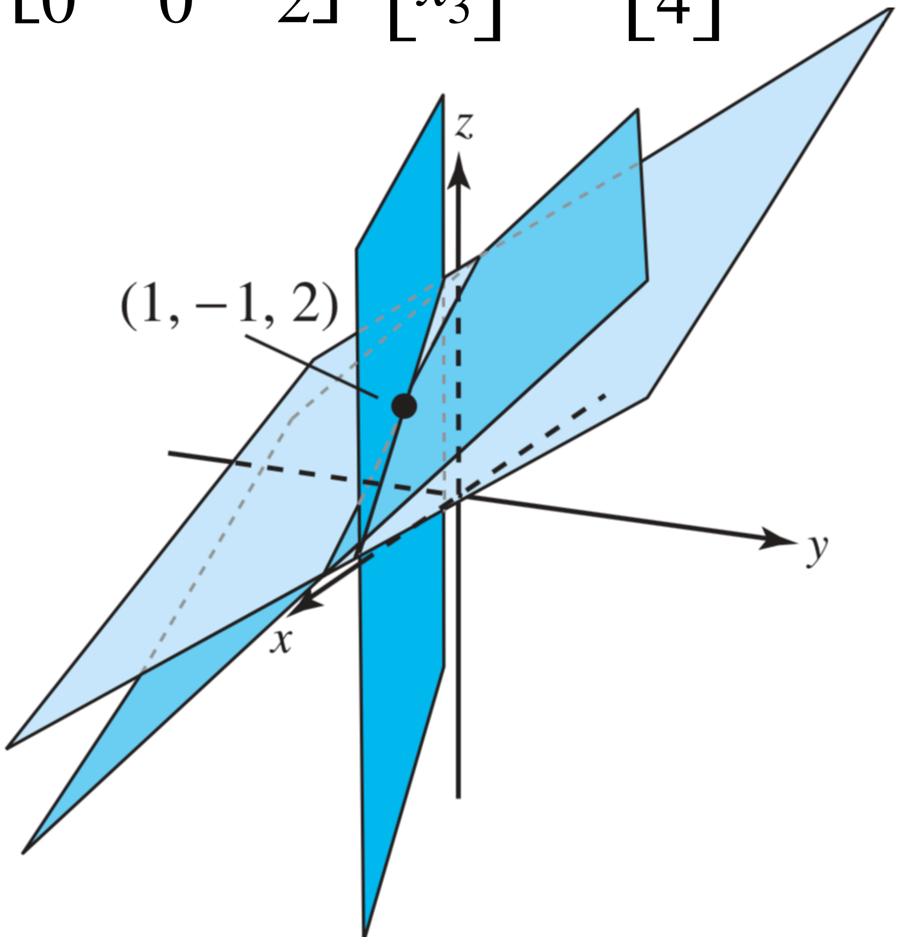
Functions are QUADRATIC FORMS

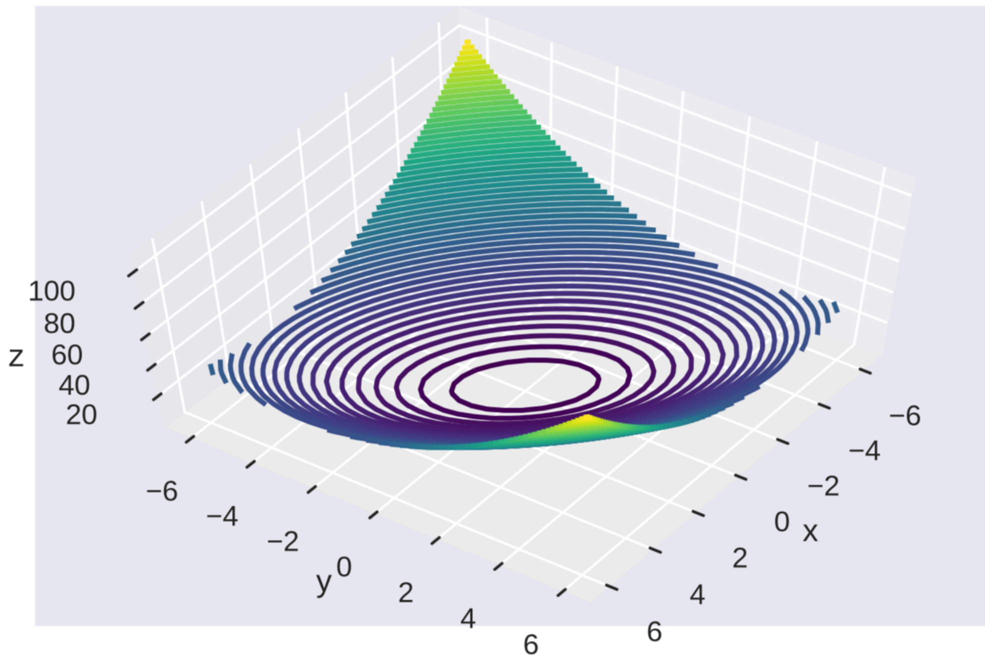
$$f(\mathbf{x}) = \mathbf{x}^T A\mathbf{x}$$

$$\begin{bmatrix} 1 & -2 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 9 \\ 5 \\ 4 \end{bmatrix}$$

$$A = [x\ y] \begin{bmatrix} 1 & 1/2 \\ 1/2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f(x, y) = x^2 + xy + y^2$$

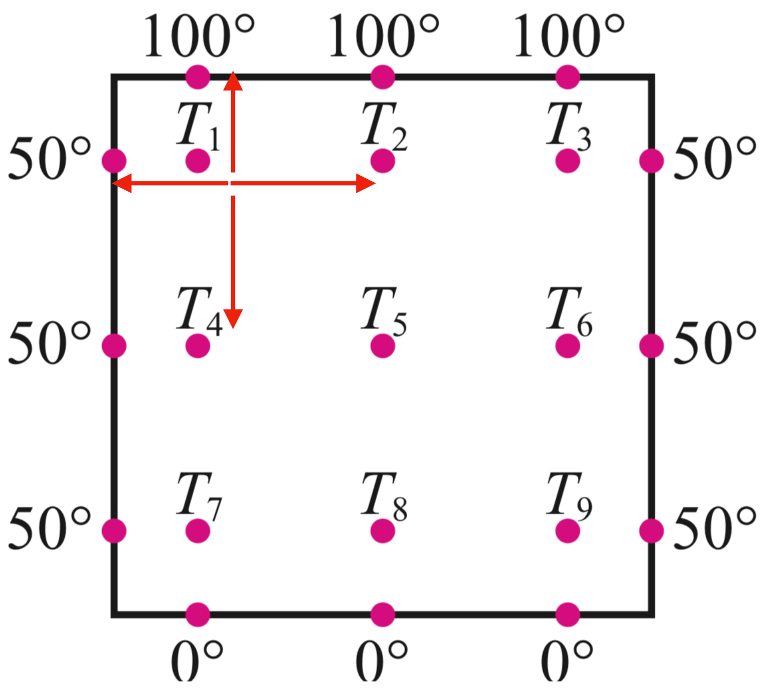




Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Harmonic Analysis

When the UNKNOWNS are the INNER NODES GIVEN THE KNOWN ONES



Harmonic (Linear) hypothesis:

$$T_i = \frac{1}{|N_i|} \sum_{j \in N_i} T_j$$

Find the Temperature in T_i
Given those of the border

$$T_1 = \frac{1}{4}(50 + 100 + T_2 + T_4)$$
$$T_2 = \frac{1}{4}(T_1 + 100 + T_3 + T_5)$$
$$T_3 = \frac{1}{4}(T_2 + 100 + 50 + T_6)$$
$$T_4 = \frac{1}{4}(50 + T_1 + T_5 + T_7)$$
$$T_5 = \frac{1}{4}(T_4 + T_2 + T_6 + T_8)$$
$$T_6 = \frac{1}{4}(T_5 + T_3 + 50 + T_9)$$
$$T_7 = \frac{1}{4}(50 + T_4 + T_8 + 0)$$
$$T_8 = \frac{1}{4}(T_7 + T_5 + T_9 + 0)$$
$$T_9 = \frac{1}{4}(T_8 + T_6 + 50 + 0)$$

Unique solution in this case. Why?

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Curve Fitting

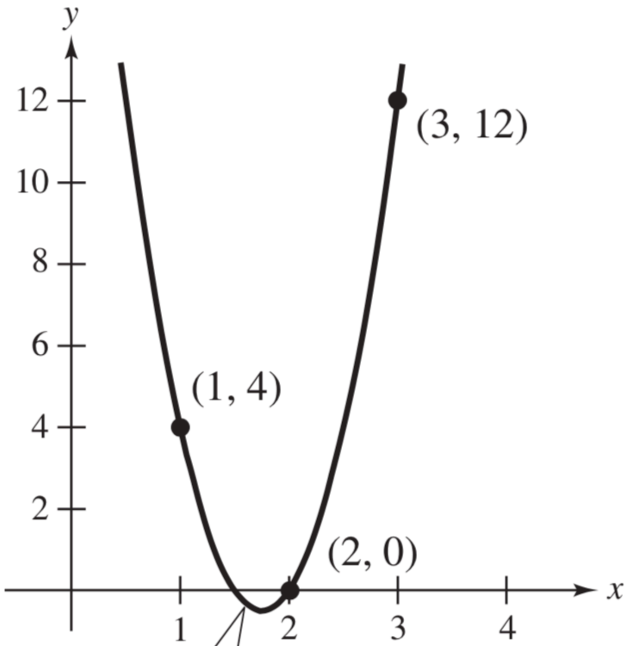
When the UNKNOWNS are the COEFICIENTS OF A CURVE

$$p(x) = a_0 + a_1 x + a_2 x^2$$

$$x_i, p(x_i) = \{(1,4), (2,0), (3,12)\}$$

$$\begin{aligned} p(x_1) &= a_0 + a_1 x_1 + a_2 x_1^2 \\ p(x_2) &= a_0 + a_1 x_2 + a_2 x_2^2 \\ p(x_3) &= a_0 + a_1 x_3 + a_2 x_3^2 \end{aligned}$$

$$\begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} p(x_1) \\ p(x_2) \\ p(x_3) \end{bmatrix}$$



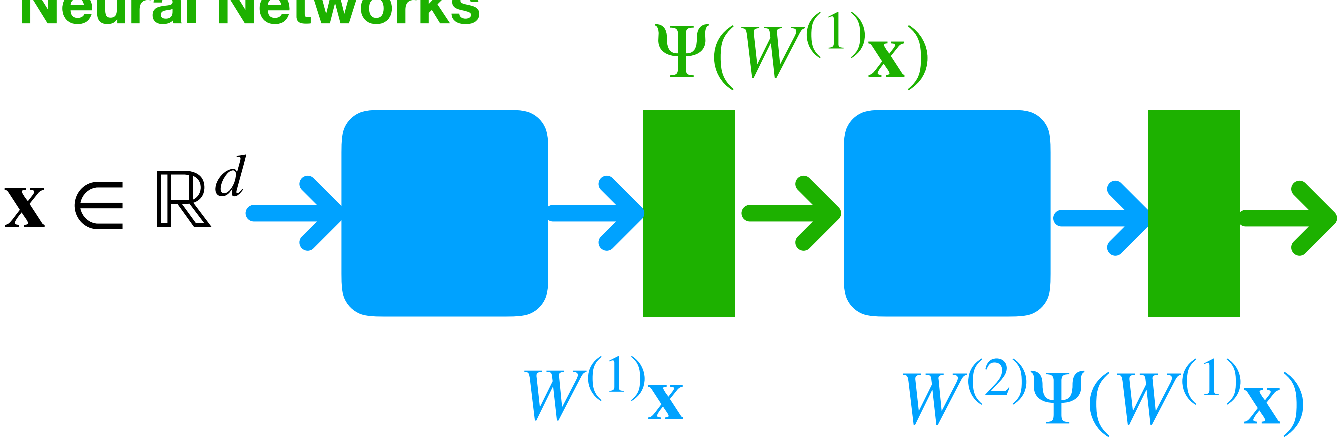
$p(x) = 24 - 28x + 8x^2$

$$\begin{bmatrix} 1 & 1 & 1^2 \\ 1 & 2 & 2^2 \\ 1 & 3 & 3^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 12 \end{bmatrix}$$

Solution=(24, -28, 8)

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

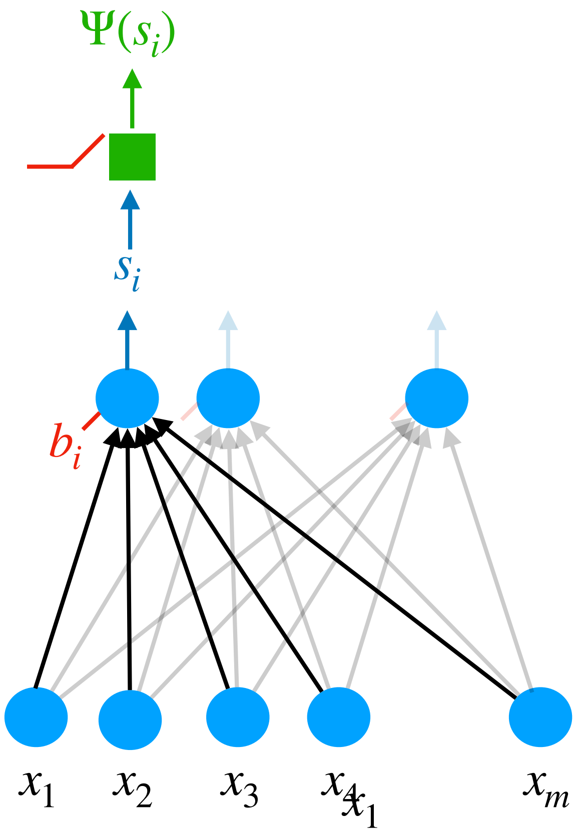
Neural Networks



$\mathbf{z} \in \mathbb{R}^o$

$$\mathbf{y} = \Psi(W^{(2)}\Psi(W^{(1)}\mathbf{x}))$$

$$Min_{W^{(1)},W^{(2)}}\|\mathbf{z}-\mathbf{y}\|^2$$



$$\Psi(S) = \Psi(W\mathbf{x})$$

$$\Psi(s_i) = max(0,s_i)$$

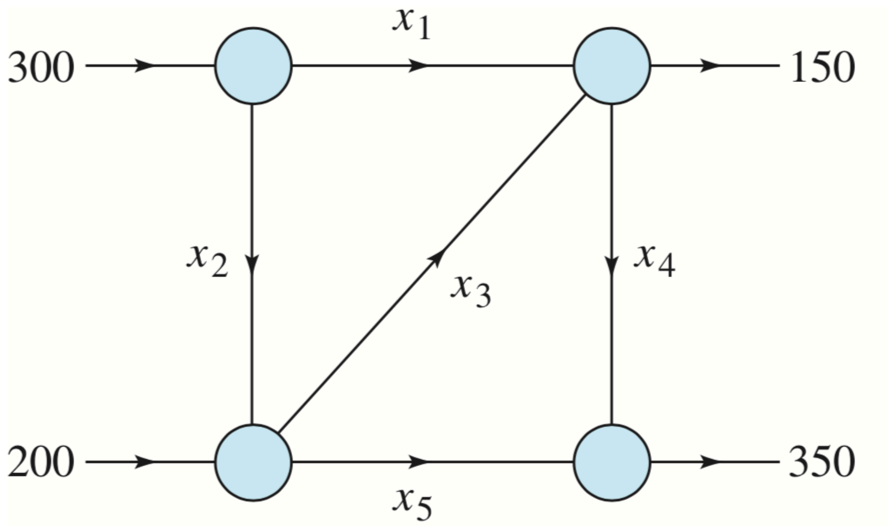
$$s_i = W_{i1}x_1 + W_{i2}x_1 + \ldots + W_{im}x_m + b_i$$

$$S = \begin{bmatrix} W_{11} & W_{12} & \ldots & W_{1m} & b_1 \\ W_{21} & W_{22} & \ldots & W_{2m} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ W_{n1} & W_{n2} & \ldots & W_{nm} & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ 1 \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix}$$

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Network flows

When the EQUATIONS are GIVEN BY JUNCTIONS



$$
\begin{aligned}
300 &= x_1 + x_2 \\
200 &= x_2 - x_3 - x_5 \\
150 &= x_1 + x_3 + x_4 \\
350 &= x_4 + x_5
\end{aligned}
$$

$$
\begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}
=
\begin{bmatrix} 300 \\ 200 \\ 150 \\ 350 \end{bmatrix}
$$

Solution in Sympy:

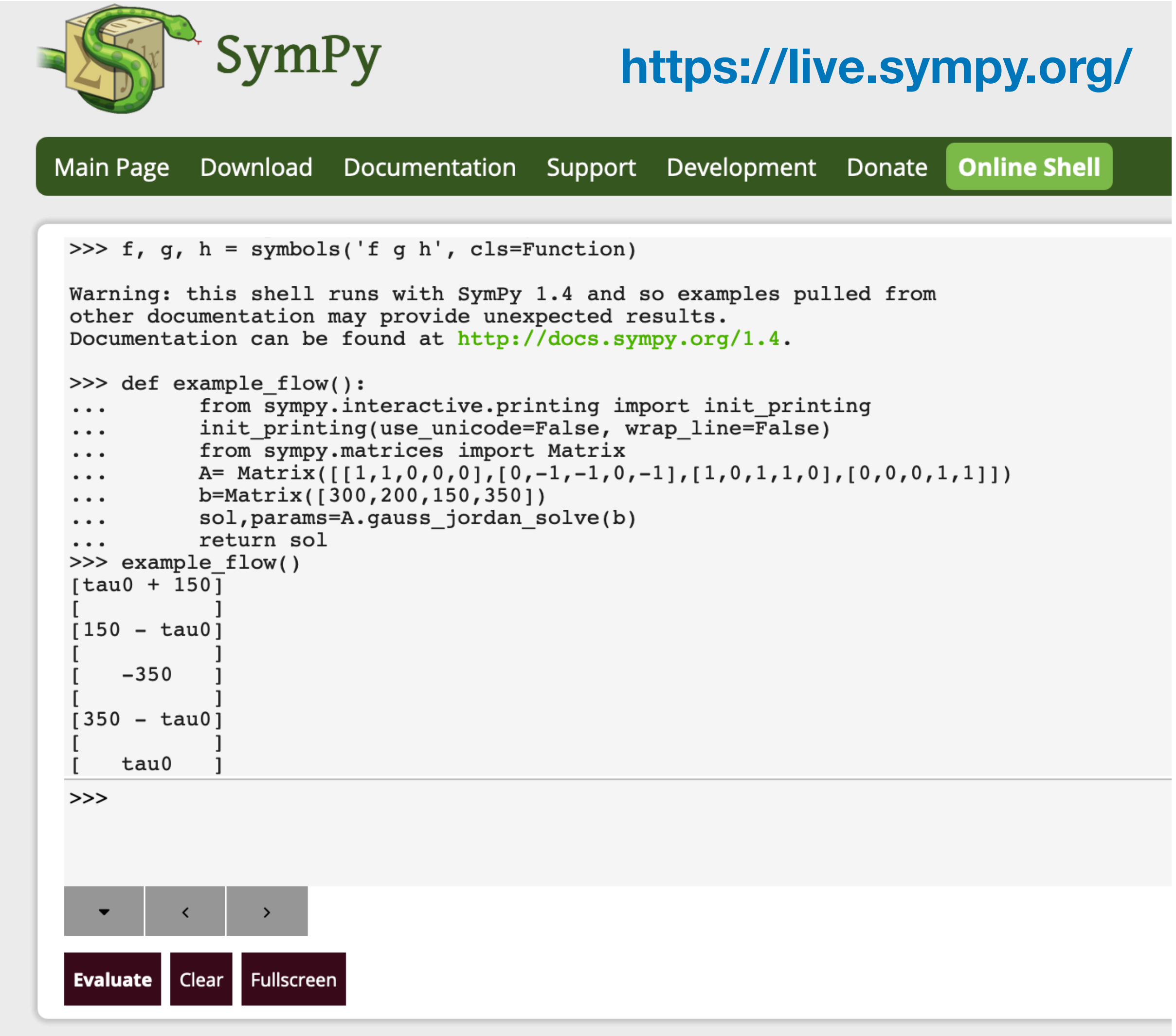
A.gauss_jordan_solve(b)

$$
\left(\begin{bmatrix} \tau_0 + 150 \\ 150 - \tau_0 \\ -350 \\ 350 - \tau_0 \\ \tau_0 \end{bmatrix}, \ \begin{bmatrix} \tau_0 \end{bmatrix} \right)
$$

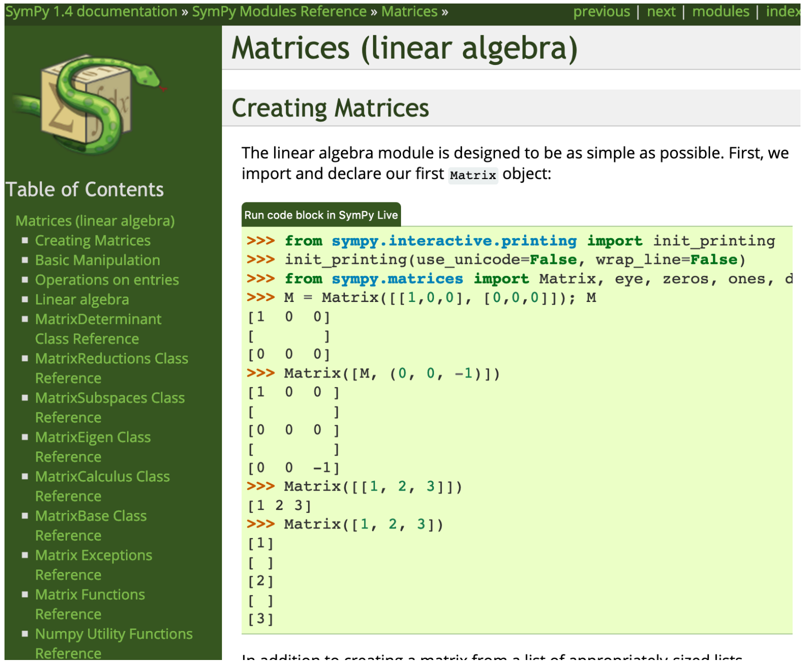
Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

Solving with SymPy

Desarrollos optativos para subir nota (NOTA EXTRA)




```
def example_flow():
    from sympy.interactive.printing import init_printing
    init_printing(use_unicode=False, wrap_line=False)
    from sympy.matrices import Matrix
    A= Matrix([[1,1,0,0,0],[0,-1,-1,0,-1],[1,0,1,1,0],[0,0,0,1,1]])
    b=Matrix([300,200,150,350])
    sol,params=A.gauss_jordan_solve(b)
    return sol
>>>> example_flow()
```



https://docs.sympy.org/latest/modules/matrices/matrices.html#linear-algebra

Linearity vs non-linearity, examples of applications, Gauss and Gauss-Jordan reduction, Echelon form, Homogeneous systems :

TO BE CONTINUED…