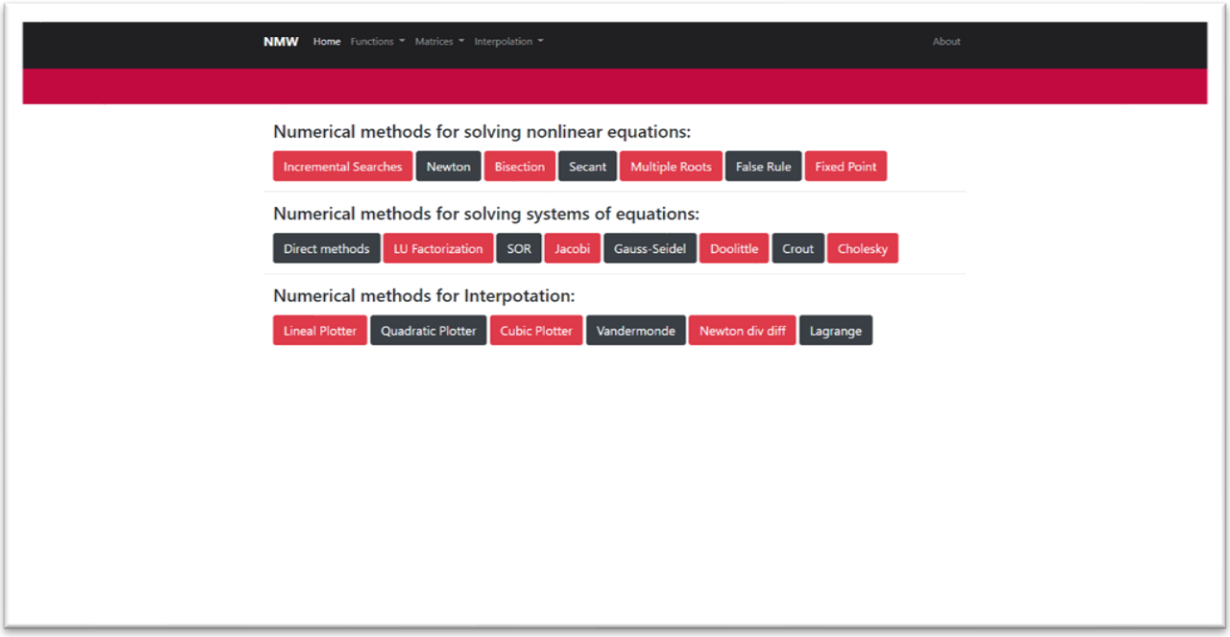
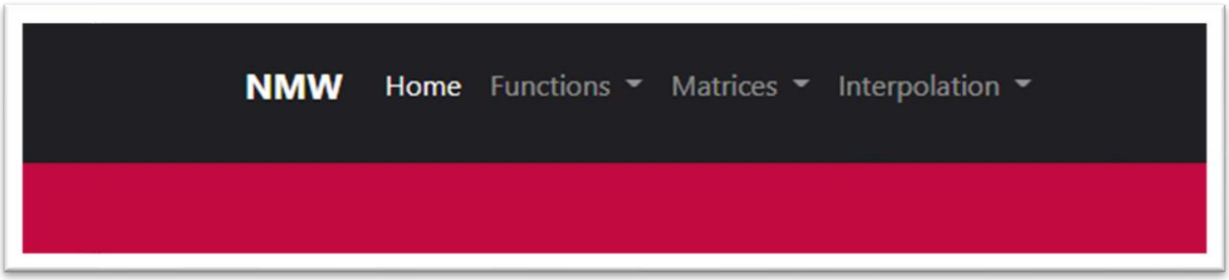
Welcome to Numerical Methods Web



To navigate in the page, you can use the navigation bar:



Data inputs:

$$10^{-7}$$

To represent x to the power of y in your inputs, you must use the following:

10**-7

These are some of the functions you can use:

exp() - to represent the exponential function

sin() - to represent sine function

cos() - to represent cosine function

tan() - to represent tangent function

cot() - to represent cotangent function

sec() - to represent secant function

csc() - to represent cosecant function

sinc() - to represent sine cardinal

log() - to represent natural logarithm

Let's see an example of a correct input of a function and Its derivative:

f: log(sin(x)**2+1)-1/2

f ': 2*((sin(x)**2+1)**-1)*sin(x)*cos(x)

To see all the functions you can use, you should visit:

https://docs.sympy.org/latest/modules/functions/index.html

Direct methods:

This section contains 3 different methods: Simple Gaussian Elimination, Simple Gaussian Elimination with partial pivoting and Simple Gaussian Elimination with total pivoting.

First, you can select the method you want to execute, if you don't, Simple Gaussian Elimination will be selected by default. After this, you can input the number of variables available in the system of equations. By clicking 'save', the page will display a matrix of nxn blank spaces and a vector of nx1. There, you can introduce the matrix A and the vector b to be used in the execution. After you fill in all the blanks you can click 'Solve' and the solution will be displayed. If you get an error message, you can look at the matrix you introduced and read the 'Help' section at the bottom of the page to understand what happened. Or you can simply try again by introducing the number of variables and clicking 'save'.

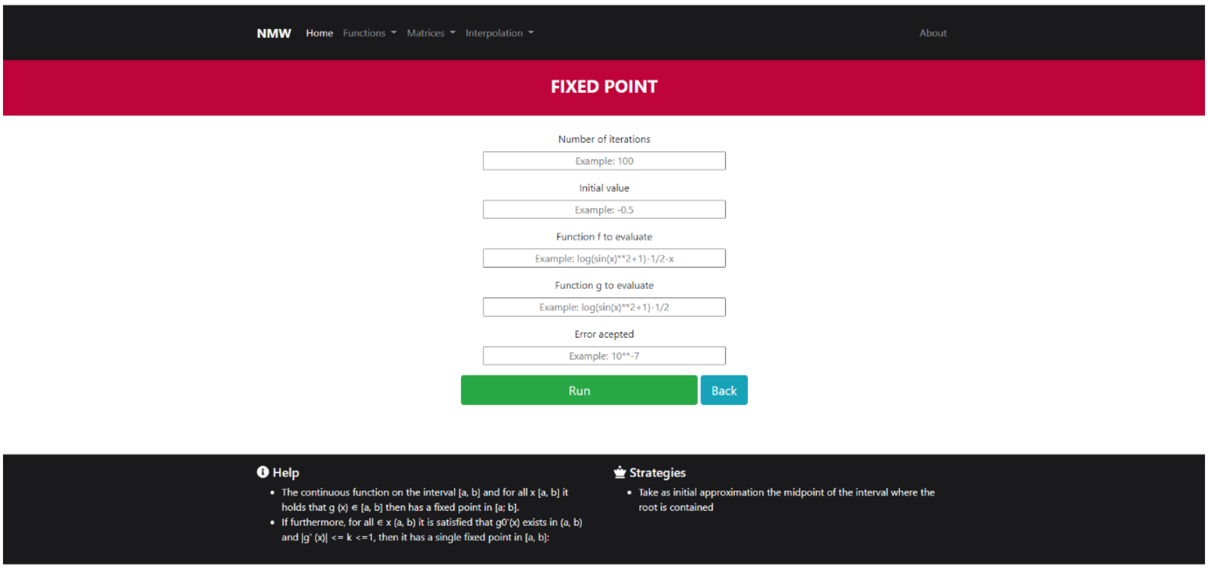
LU factorization:

This section contains 2 different methods: Simple LU Factorization and LU Factorization with partial pivoting.

First, you can select the method you want to execute, if you don't, Simple LU Factorization will be selected by default. After this, you can input the number of variables available in the system of equations. By clicking 'save', the page will display a matrix of nxn blank spaces and a vector of nx1. There, you can introduce the matrix A and the vector b to be used in the execution. After you fill in all the blanks you can click 'Solve' and the solution will be displayed. If you get an error message, you can look at the matrix you introduced and read the 'Help' section at the bottom of the page to understand what happened. Or you can simply try again by introducing the number of variables and clicking 'save'.

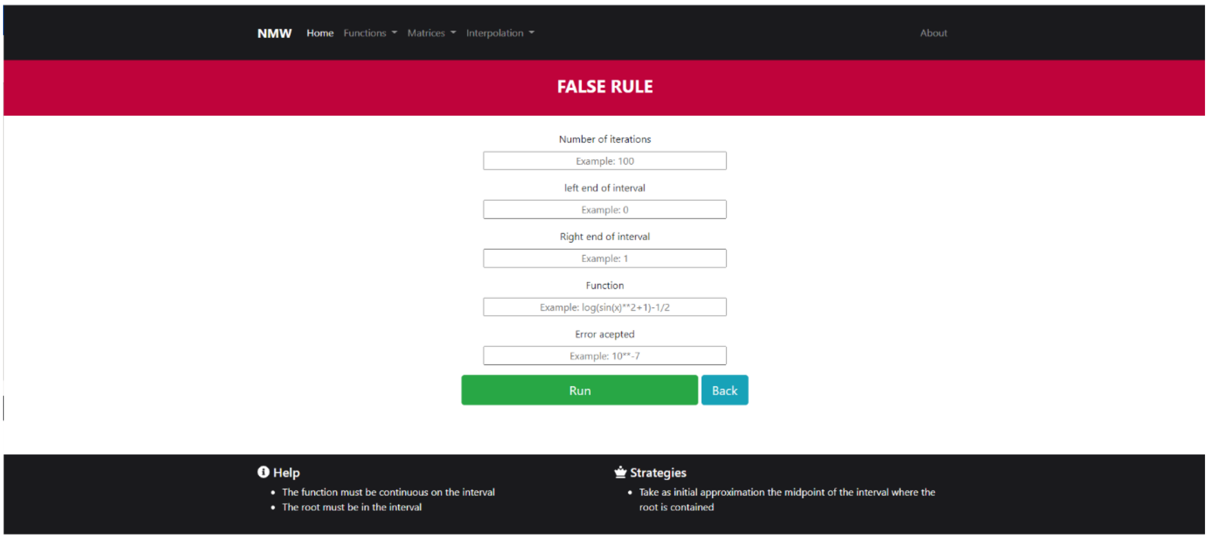
Fixed Point:

This section contains 5 inputs. The first one corresponds to the maximum number of iterations that the user wants in case the method does not reach the convergence. The second is the initial value, the third and fourth are the f and g function, respectively. Finally, the error accepted.



False rule:

The false rule needs five values to run. The first one corresponds to the maximum number of iterations that the user wants in case the method does not reach the convergence. The second and third are the first and second initial values, respectively. In the fourth field you must enter the function you want to be evaluated using the format described above, and finally the error accepted.



Secant method:

The Secant method needs five values to run. The first one corresponds to the maximum number of iterations that the user wants in case the method does not reach the convergence. The second and third are the first and second initial values, respectively. In the fourth field you must enter the function you want to be evaluated using the format described above, and finally the error accepted.

Multiple Roots:

The multiple roots method needs five values to work. The first one corresponds to the maximum number of iterations that the user wants in case the method does not reach convergence. The second and third are the first and second initial value, respectively. In the fourth field you must enter the function you want to be evaluated using the format described above, the program automatically calculates the first and second derivative of the entered function, and finally enter the accepted error.

SOR:

For this method you need to enter 8 values. In the first and second fields you must enter the size of the array you want to be evaluated, remember that it must be square. In the third field you must enter each of the elements of the array separated by a space. This input of values must be done per row (See Figure 1).

$$A = \begin{pmatrix} 4 & -1 & 0 & 3 \\ 1 & 15.5 & 3 & 8 \\ 0 & -1.3 & -4 & 1.1 \\ 14 & 5 & -2 & 30 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Figure 1. In this case, if you would want to enter the values of array A, the entry should be: 4 -1 0 3 1 15.5 3 8 0 -1.3 -4 1.1 14 5 -2 30

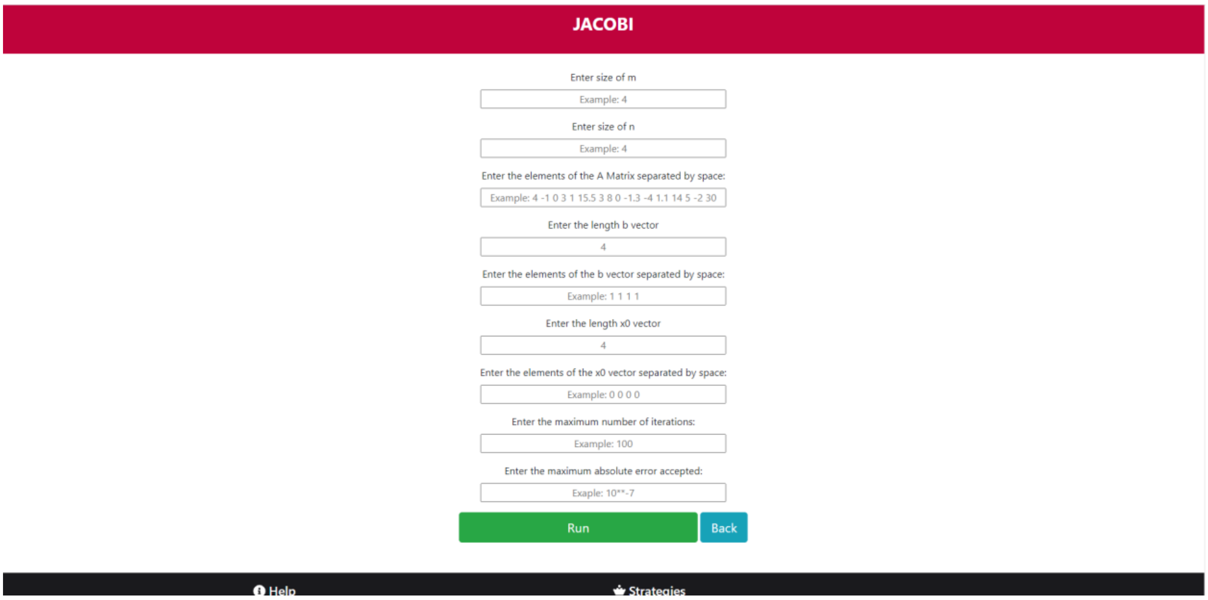
In the fourth and fifth fields, you must enter the value of vector b and the initial value vector, respectively, using the convention seen above. In the next three fields you must enter the maximum number of iterations, the maximum absolute error accepted and lambda of relaxation.

Jacobi:

$$A = \begin{pmatrix} 4 & -1 & 0 & 3 \\ 1 & 15.5 & 3 & 8 \\ 0 & -1.3 & -4 & 1.1 \\ 14 & 5 & -2 & 30 \end{pmatrix}, \qquad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad x_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

For this method you need to enter 9 values. In the first and second fields you must enter the size of the array you want to be evaluated, remember that it must be square. In the third field you must enter each of the elements of the array separated by a space. This input of values must be done per row (See Figure 1).

In the fourth, you must enter the number of elements of b. In the fifth fields, you must enter the value of vector b, using the convention seen above. In the next four fields you must enter the length x0 vector, the initial value vector (x0), the maximum number of iterations and the maximum absolute error accepted.



Numerical methods for Interpolation:

The methods of interpolation are

- Vandermonde.
- Newton with Divided Differences.
- Lineal Plotter.
- Quadratic Plotter.

To execute these methods, we have three fields. In the first one, we must introduce the quantity of elements that are going to have the vectors X and Y. In the second field must go each one of the elements of X separated by a space, the same must do the in the third field, but this time the values corresponding to the vector Y. The next image represents an example of the field that you will find.