INTERPOLACIÓN DE LAGRANGE POR APLICACIÓN DE LA DEFINICIÓN

La interpolación polinómica de Lagrange nos sirve para estimar el valor de una función desconocida en un punto cualquiera, conociendo un conjunto de determinados puntos, a los que llamaremos x_i (**soporte**) y los valores conocidos de la función, f_i, en cada uno de esos puntos. Para ello, debemos obtener **polinomios**, tantos como puntos de soporte haya y de grado menor o igual a n-1, siendo n el número de puntos de soporte

Este primer método se basa en la aplicación de la definición de interpolación mediante la resolución de un sistema y es posible realizar un algoritmo que nos permita obtener la matriz de los coeficientes.

Desarrollando el sistema:

$$\begin{cases} a_1 + a_2x_1 + a_3{x_1}^2 + \cdots + a_n{x_1}^{n-1} = f_1 \\ a_1 + a_2x_2 + a_3{x_2}^2 + \cdots + a_n{x_2}^{n-1} = f_2 \\ \dots \\ a_1 + a_2x_n + a_3{x_n}^2 + \cdots + a_n{x_n}^{n-1} = f_n \end{cases}$$

Podemos expresarlo en forma de ecuación matricial, $AX = f$, donde X son nuestras incógnitas (es decir, los términos a_i, que no calcularemos) y A es la matriz de los coeficientes, que obtenemos a partir de los puntos del soporte x_i (importante no confundir los términos x_i, que son los puntos del soporte conocidos, con los elementos de la matriz A, que son las incógnitas).

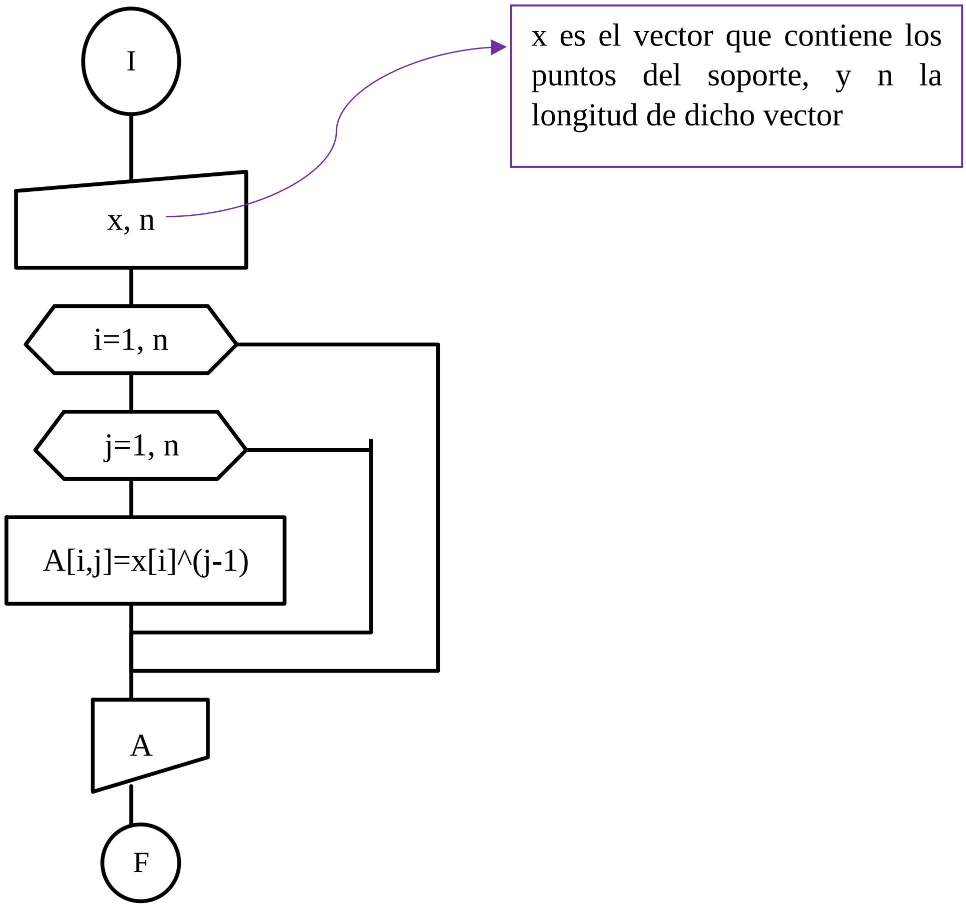
La matriz A queda:

$$A = \begin{pmatrix} 1 & x_1 & {x_1}^2 & \dots & {x_1}^{n-1} \\ 1 & x_2 & {x_2}^2 & & {x_2}^{n-1} \\ \dots & & & & \\ 1 & x_n & {x_n}^2 & \dots & {x_n}^{n-1} \end{pmatrix}$$

y por otro lado,

$$X = \begin{pmatrix} a_1 \\ a_2 \\ \dots \\ a_n \end{pmatrix}, \text{ y } f = \begin{pmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{pmatrix}$$

Es evidente que para el algoritmo de la matriz A tendremos que usar dos bucles, en ambos casos variando desde 1 hasta n. Es muy fácil ver que cada elemento de A se puede obtener como $A_{i,j} = {x_i}^{j-1}$ (algoritmo en la siguiente página)



Si quisiéramos obtener los valores de X (las incógnitas), deberíamos resolver el sistema a mano (por Gauss preferiblemente) o bien, en R, utilizando el comando “solve”.