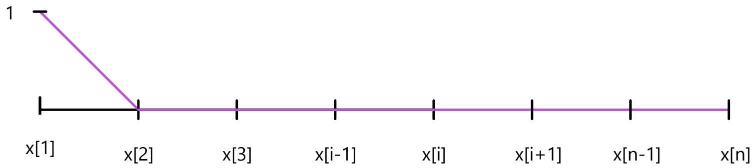
INTERPOLACIÓN POR TRAMOS DE PRIMER GRADO

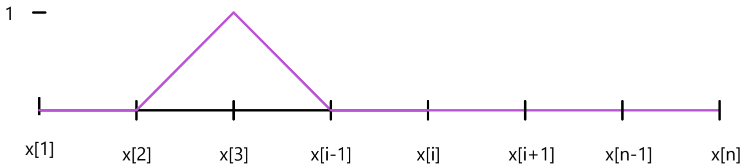
Cuando tenemos que aplicar la interpolación a un soporte que tiene muchos puntos, encontramos que con los métodos habituales obtenemos un solo polinomio de un grado muy elevado. Este polinomio es oscilante, por lo que generalmente nos proporciona un valor interpolado erróneo y no nos resulta útil.

En estos casos, aplicamos la **interpolación por tramos**, que consiste en dividir el soporte en pequeños subintervalos y calcular los polinomios de base en cada intervalo. Los dos tipos principales de interpolación por tramos son la de primer grado (en la que nos centraremos en estos apuntes) y la de segundo grado.

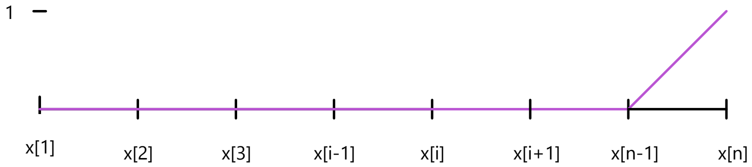
En la interpolación por tramos, tendremos que calcular las **funciones de base**, que son equivalentes a los polinomios de base, una para cada punto. Siguiendo con la definición de función de base, debe tomar como valor 1 si el valor interpolado coincide con el punto del soporte para el que calculamos la función de base, y 0 para el resto de puntos del soporte. Al ser de primer grado, las funciones serán rectas entre los puntos consecutivos del soporte; es decir, los subintervalos estarán compuestos por dos puntos. Representándolo gráficamente:



Función de base φ (1), asociada al primer punto



Función de base φ (2), asociada al segundo punto (sería igual para los demás puntos internos)



Función de base φ (2), asociada al último punto

El cálculo de las funciones de base es muy sencillo, y se corresponde con el cálculo de los polinomios de base, teniendo en cuenta que las funciones de base en este caso tendrán grado 1. Las funciones de base están definidas por tramos, para que podamos especificar que valen 0 para cualquier otro punto del soporte; como vemos, las funciones de base se obtendrán de manera diferente si se trata de un punto interno del soporte o si es el punto inicial/final.

Tomando el valor para el que interpolamos como x, las funciones de base son:

Para el punto inicial:

$$\varphi_1(x) = \begin{cases} \frac{x-x_2}{x_1-x_2} & \text{si } x \in [x_1, x_2] \\ 0 & \text{si } x \notin [x_1, x_2] \end{cases}$$

Para el punto final:

$$\varphi_n(x) = \begin{cases} \frac{x-x_{n-1}}{x_n-x_{n-1}} & \text{si } x \in [x_{n-1}, x_n] \\ 0 & \text{si } x \notin [x_{n-1}, x_n] \end{cases}$$

Para un punto intermedio cualquiera, i:

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}} & \text{si } x \in [x_{i-1}, x_i] \\ \frac{x-x_{i+1}}{x_i-x_{i+1}} & \text{si } x \in [x_i, x_{i+1}] \\ 0 & \text{si } x \notin [x_{i-1}, x_{i+1}] \end{cases}$$

En el caso de un punto intermedio, tenemos que definir la función como 0 si el valor para el que interpolamos está comprendido entre el punto por debajo y el punto por encima; y otras dos ramas, una para cuando esté entre el punto del soporte que estamos tratando (x_i) y el punto por debajo, y otro para cuando esté entre el punto x_i y el punto por encima

Para obtener el valor interpolado, simplemente tenemos que multiplicar el valor obtenido de la función de base de cada punto por el valor conocido que toma la función en dicho punto. Es decir, la función U también estará definida por tramos, y cada rama se obtendrá multiplicando $f_i \cdot \varphi_i$.

Este planteamiento nos resulta muy útil cuando lo que queremos hacer es el algoritmo que nos dé un vector phi que contenga los valores de las funciones de base y una variable U con el valor interpolado. Sin embargo, puede resultar un poco raro aplicarlo a un ejercicio en el que no tenemos que hacer algoritmos sino calcular. Por ello, más abajo os explicamos cómo debe hacerse un ejercicio de interpolación por tramos si lo que queremos es el dato numérico:

- Obtenemos la expresión de cada función de base.

a) Si estamos usando el primer o el último punto del soporte, la función tendrá dos ramas. Una de ellas se obtiene como $\frac{x-a}{b-a}$, donde b es el punto del soporte para el que estamos calculando la función, que coincide con uno de los límites del intervalo, y a es el otro límite del intervalo (el segundo punto de soporte si es la primera función, o el penúltimo punto si se trata de la última función), y se aplicará cuando x (el valor interpolado) pertenezca al intervalo [a,b], y si no pertenece la función vale 0.

b) Si estamos usando un punto intermedio del soporte, la función tendrá tres ramas. Una se obtiene como $\frac{x-a}{b-a}$, donde b es el punto de soporte para el que calculamos la función, y a el anterior punto de soporte (el que queda a la izquierda en una representación gráfica), si x está dentro del intervalo [a,b]; otra se obtiene como $\frac{x-c}{b-c}$, donde b sigue siendo igual que en la anterior rama y c el siguiente punto de soporte (el de la derecha), si x está en el intervalo [b,c]; y la última es 0, cuando x no pertenece a [a,c].

Mi recomendación es sustituir por el momento solamente a, b y c, y obtener TODAS las funciones de base (una por cada punto), indicando también sus intervalos.

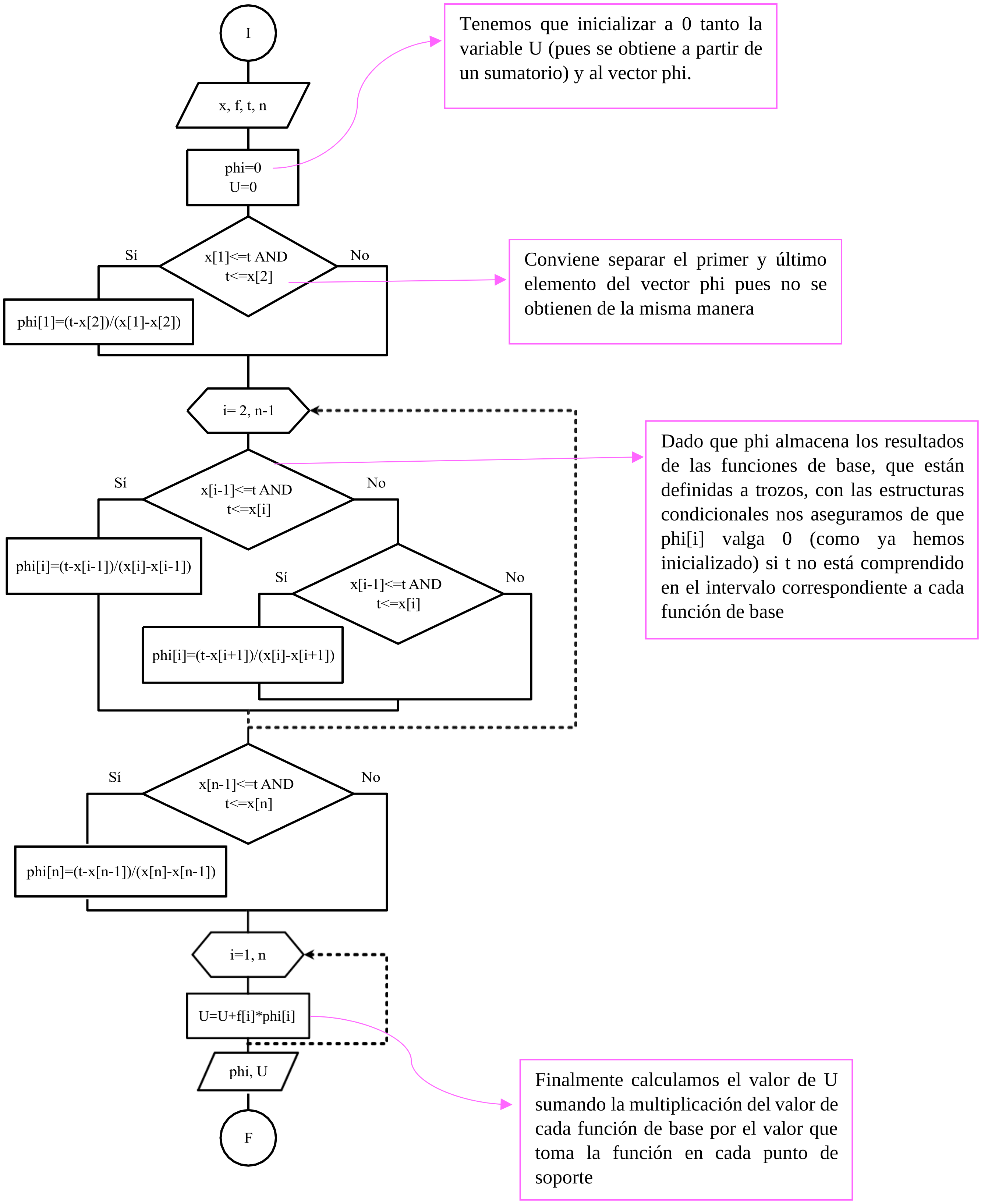
- Multiplicamos cada función de base (todas sus ramas) por el valor conocido de la función del enunciado en cada punto.

- Agrupamos aquellas ramas de las funciones de base (ya multiplicadas por el valor de la función) que tengan los mismos intervalos, y las sumamos. Escribimos la función polinómica U, cuyas ramas serán dichas sumas, y escribimos también los intervalos.

- Vemos cuál es el valor que queremos interpolar y a qué intervalo de la función U pertenece; SOLAMENTE lo sustituimos en las x de la rama de U con dicho intervalo. El valor resultante será el valor interpolado.

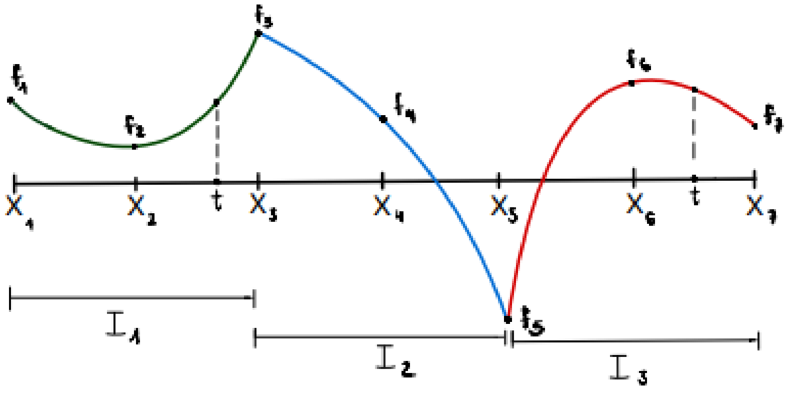
ALGORITMO PARA LA INTERPOLACIÓN POR TRAMOS DE PRIMER GRADO

Aquí se propone el algoritmo para obtener el vector phi, que contiene los distintos valores de las funciones de base para un valor interpolado t (para el que queremos aproximar el valor de la función, y la variable U, que contiene el valor interpolado para dicho punto t. Es recomendable repasar primero los apuntes de interpolación antes de enfrentarse al algoritmo.



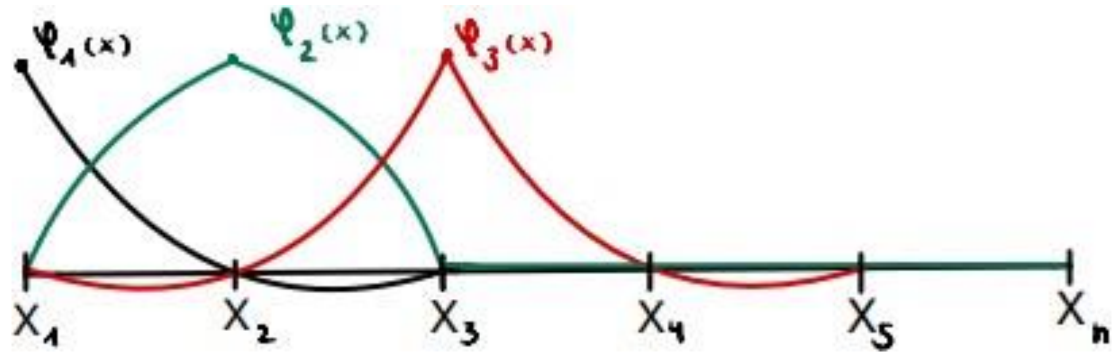
INTERPOLACIÓN POR TRAMOS DE SEGUNDO GRADO

En este caso dividiremos el soporte en intervalos de tres puntos, y las funciones de base tendrán grado 2, es decir, serán parábolas.

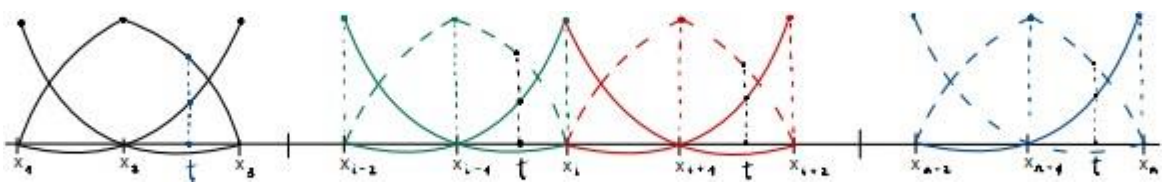


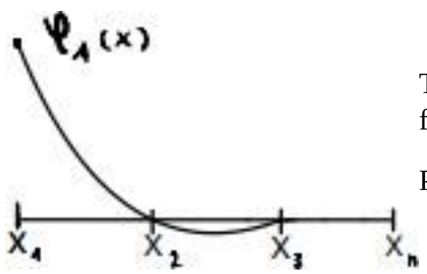
El número de subintervalos se calcula como (n-1)/2, y por tanto, es posible ver que para esta interpolación es necesario tener un número impar de puntos de soporte. Por ejemplo, con un soporte de 7 puntos se formarán 3 subintervalos.

Las funciones de base cumplen las mismas condiciones que antes; poseen valor 1 en el punto de soporte al que están asociadas, y 0 en el resto de puntos del soporte. Es necesario tener en cuenta la posición del valor t para el que queremos interpolar dentro del soporte. Por ejemplo, si t está en el primer intervalo, tendrá un valor para phi[1], phi[2] y phi[3].



Para un soporte de n puntos:





Tanto el primer como el último punto del soporte tienen funciones de base distintas:

Para el primer punto:

$$\varphi_1(x) = \begin{cases} \dfrac{x-x_2}{x_1-x_2}\,\dfrac{x-x_3}{x_1-x_3} & x \in [x_1, x_3] \\ 0 & x \notin [x_1, x_3] \end{cases}$$

Para el último punto:

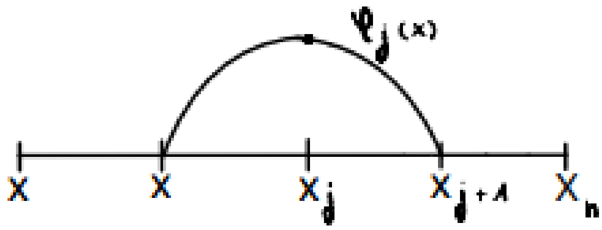
$$\varphi_n(x) = \begin{cases} \frac{x-x_{n-2}}{x_n-x_{n-2}} \frac{x-x_{n-1}}{x_n-x_{n-1}} & x \in [x_{n-2}, x_n] \\ 0 & x \notin [x_{n-2}, x_n] \end{cases}$$

Para el resto de puntos, es preciso comprobar si son puntos “internos” del intervalo, como x_2, o puntos que conectan intervalos, como x_3, pues la función de base asociada a cada uno de ellos será distinta.

Para el primer caso, las funciones de base alcanzan su máximo valor (1) en un punto intermedio del intervalo y no en sus extremos. En este caso, la función de base correspondiente tendrá dos ramas, y se obtendrá como:

$$\varphi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}} \frac{x-x_{j+1}}{x_j-x_{j+1}} & x \in [x_{j-1}, x_{j+1}] \\ 0 & x \notin [x_{j-1}, x_{j+1}] \end{cases}$$

j=2, n-1, 2.

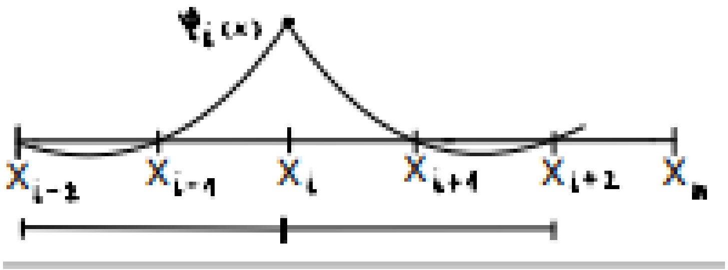


El primer punto para el que se aplica sería el segundo (el primero está calculando previamente), y a partir de ahí, los siguientes puntos “intermedios” son el x_4, x_6…, es decir, con un incremento de 2, y el último punto para el que se obtiene es el x_{n-1} (ver dibujo).

En el caso de los puntos intermedios, es importante ver que tendrán tres ramas, una para el intervalo que cierran, otra para el que abren y otra con valor 0 si el valor para el que interpolamos no está contenido en ninguno de los dos intervalos. Su función de base se obtendría como:

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-2}}{x_i-x_{i-2}} \frac{x-x_{i-1}}{x_i-x_{i-1}} & x \in [x_{i-2}, x_i] \\ \frac{x-x_{i+2}}{x_i-x_{i+2}} \frac{x-x_{i+1}}{x_i-x_{i+1}} & x \in [x_i, x_{i+2}] \\ 0 & x \notin [x_{i-2}, x_{i+2}] \end{cases}$$

i=3, n-2, 2



El primer punto para el que se aplica sería el tercero, y a partir de ahí, para x_5, x_7… (al igual que antes, con un incremento de 2). El último punto para el que nos serviría sería el x_{n-2}, pues es el último punto que conecta intervalos (ver dibujo).

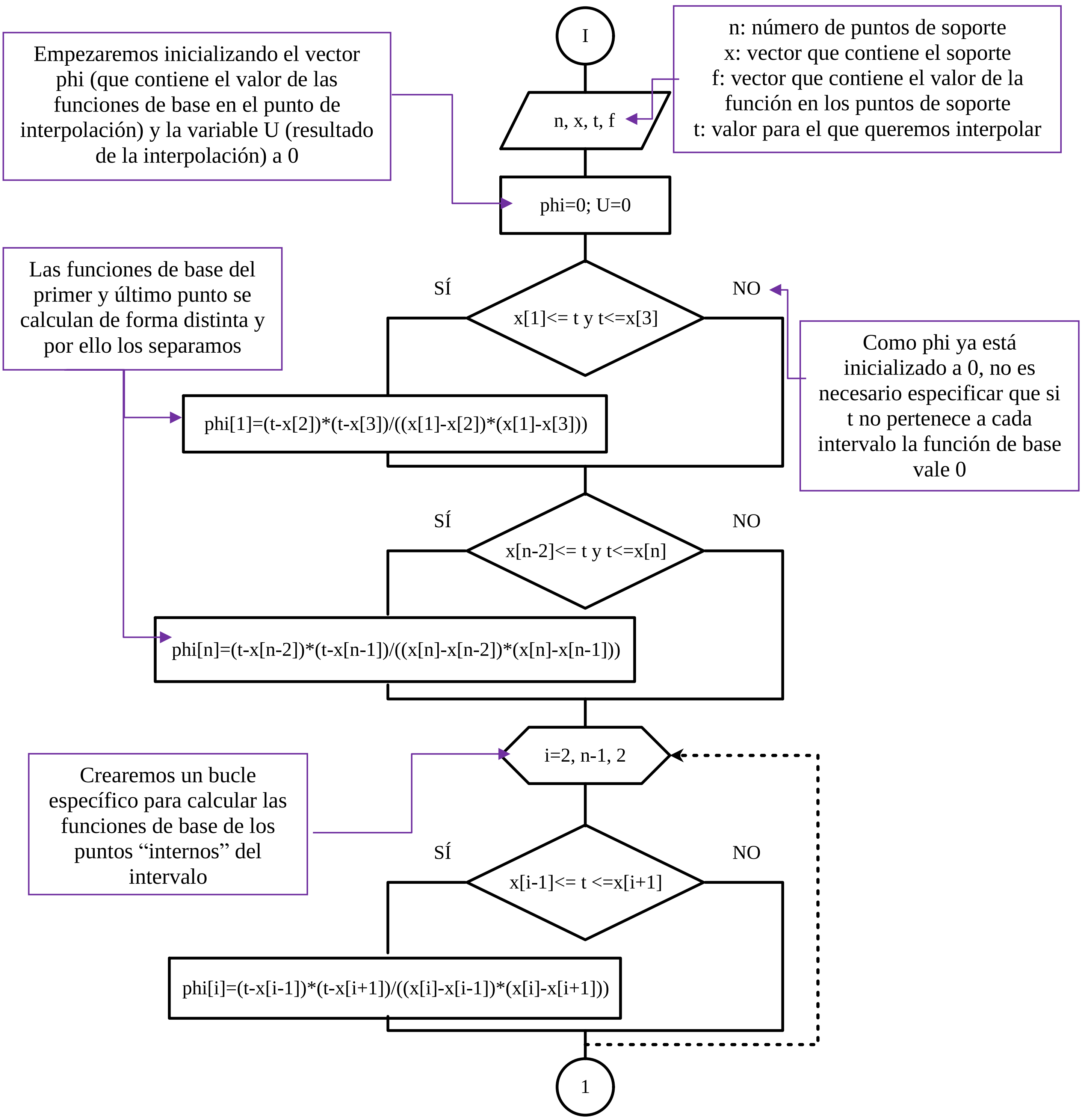
IDEA DEL ALGORITMO

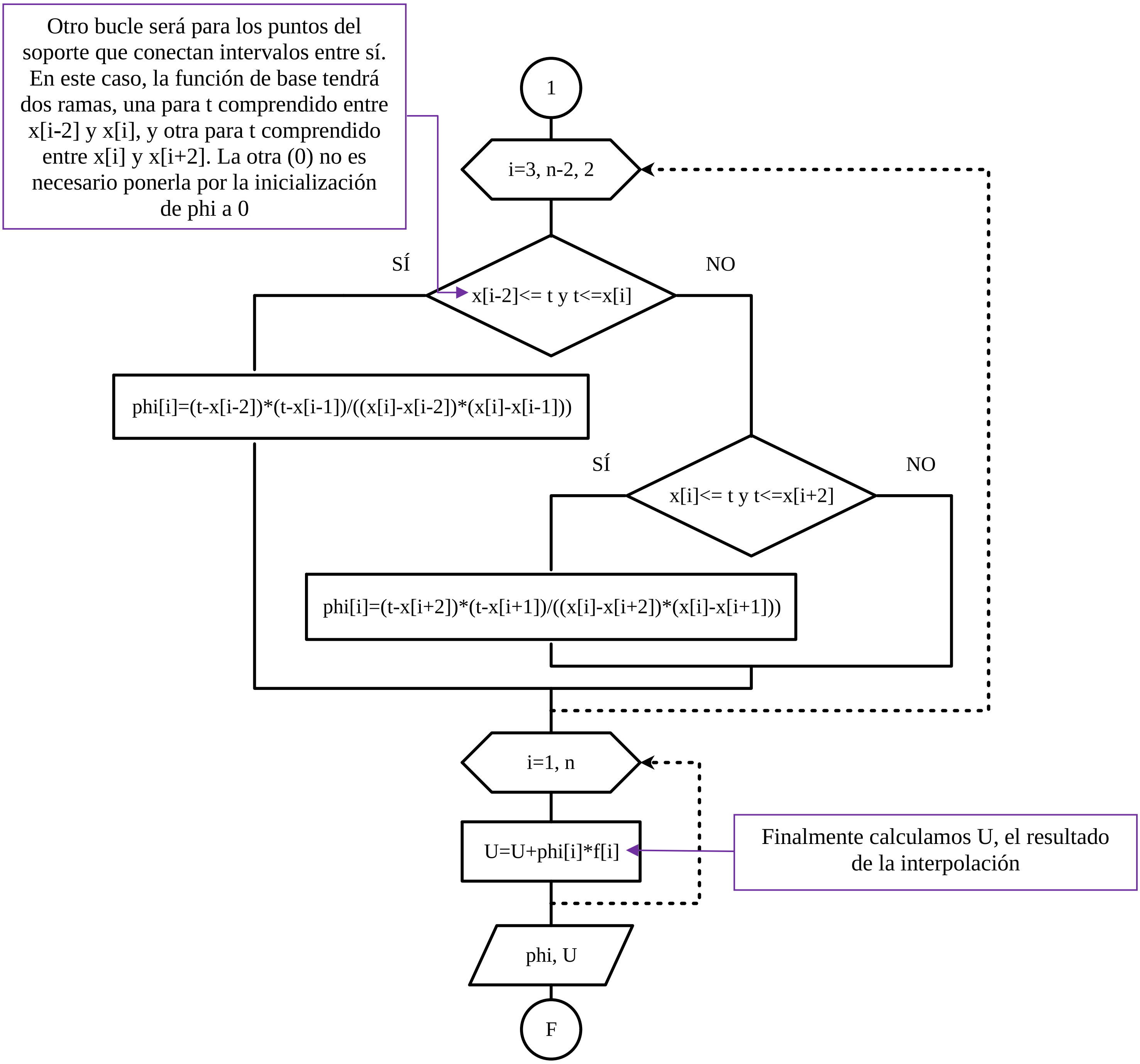
Este algoritmo es algo más complicado que para las funciones de base de primer grado, así que se propone el pseudocódigo para facilitar la realización del algoritmo.

Empezaremos calculando el vector phi, que contiene los valores de las funciones de base asociadas a cada punto, para después calcular el valor interpolado U, que se obtiene como el sumatorio $\sum_{i=1}^{n} phi_i \cdot f_i$. Los datos conocidos serán t (el valor para el que queremos interpolar), el vector x, que contiene los puntos del soporte, n (número de puntos del soporte), y f, vector que contiene los valores de la función para los puntos del soporte.

```
Datos: n, f, x, t

phi=0

Si x[1]<= t y t<=x[3]
    phi[1]=(t-x[2])*(t-x[3])/((x[1]-x[2])*(x[1]-x[3]))
Fin condición

Si x[n-2]<= t y t<=x[n]
    phi[n]=(t-x[n-2])*(t-x[n-1])/((x[n]-x[n-2])*(x[n]-x[n-1]))
Fin condición

Para i=2, n-1, 2
    Si x[i-1]<= t <=x[i+1]
        phi[i]=(t-x[i-1])*(t-x[i+1])/((x[i]-x[i-1])*(x[i]-x[i+1]))
    Fin condición
Fin bucle

Para i=2, n-2, 2
    Si x[i-2]<= t y t<=x[i]
        phi[i]=(t-x[i-2])*(t-x[i-1])/((x[i]-x[i-2])*(x[i]-x[i-1]))
    si no, si x[i]<= t y t<=x[i+2]
        phi[i]=(t-x[i+2])*(t-x[i+1])/((x[i]-x[i+2])*(x[i]-x[i+1]))
    Fin condición
Fin bucle

U=0

Para i=1, n
    U=U+phi[i]*f[i]
Fin bucle

Escribir phi, escribir U
```

ALGORITMO INTERPOLACIÓN POR TRAMOS DE SEGUNDO GRADO




Otro bucle será para los puntos del soporte que conectan intervalos entre sí. En este caso, la función de base tendrá dos ramas, una para t comprendido entre x[i-2] y x[i], y otra para t comprendido entre x[i] y x[i+2]. La otra (0) no es necesario ponerla por la inicialización de phi a 0
1
i=3, n-2, 2
SÍ
NO
x[i-2]<= t y t<=x[i]
phi[i]=(t-x[i-2])*(t-x[i-1])/((x[i]-x[i-2])*(x[i]-x[i-1]))
SÍ
NO
x[i]<= t y t<=x[i+2]
phi[i]=(t-x[i+2])*(t-x[i+1])/((x[i]-x[i+2])*(x[i]-x[i+1]))
i=1, n
U=U+phi[i]*f[i]
Finalmente calculamos U, el resultado de la interpolación
phi, U
F