



Exercice

Ecrire une fonction algorithmique **HERMITE** permettant de calculer H_n (polynôme d'interpolation de Lagrange-Hermite associé aux $n + 1$ triplets $(x_i, y_i, z_i)_{i \in \llbracket 0, n \rrbracket}$ en $t \in \mathbb{R}$.

Correction Exercice

But : Calculer le polynôme $H_n(t)$ défini par (??)

Données : \mathbf{X} : vecteur/tableau de \mathbb{R}^{n+1} , $X(i) = x_{i-1} \ \forall i \in \llbracket 1, n+1 \rrbracket$ et $X(i) \neq X(j)$ pour $i \neq j$,
 \mathbf{Y} : vecteur/tableau de \mathbb{R}^{n+1} , $Y(i) = y_{i-1} \ \forall i \in \llbracket 1, n+1 \rrbracket$,
 \mathbf{Z} : vecteur/tableau de \mathbb{R}^{n+1} , $Z(i) = z_{i-1} \ \forall i \in \llbracket 1, n+1 \rrbracket$,
 t : un réel.

Résultat : pH : le réel $\text{pH} = H_n(t)$.

D'après la Définition ??, on a

$$H_n(t) = \sum_{i=0}^n y_i A_i(t) + \sum_{i=0}^n z_i B_i(t) = \sum_{i=0}^n (y_i A_i(t) + z_i B_i(t))$$

avec

$$A_i(t) = (1 - 2L'_i(x_i)(t - x_i))L_i^2(t) \quad \text{et} \quad B_i(t) = (t - x_i)L_i^2(t)$$

où

$$L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^n \frac{t - x_j}{x_i - x_j}.$$

2
 Pour rendre effectif le calcul de $H_n(t)$, il reste à déterminer $L'_i(x_i)$. On a

$$L'_i(x) = \sum_{\substack{k=0 \\ k \neq i}}^n \frac{1}{x_i - x_k} \prod_{\substack{j=0 \\ j \neq i \\ j \neq k}}^n \frac{t - x_j}{x_i - x_j}$$

d'où

$$L'_i(x_i) = \sum_{\substack{k=0 \\ k \neq i}}^n \frac{1}{x_i - x_k}. \quad (1)$$

La fonction que l'on va écrire use (et certains diront abuse) de fonctions.

Algorithme 1 Fonction **HERMITE** permettant de calculer le polynôme d'interpolation de Lagrange-Hermite $H_n(t)$ définit par (??)

```

1: Fonction pH  $\leftarrow$  HERMITE (  $X, Y, Z, t$  )
2:   pH  $\leftarrow$  0
3:   Pour  $i \leftarrow 0$  à  $n$  faire
4:     pH  $\leftarrow$  pH + POLYA( $i, X, t$ ) *  $Y(i + 1)$  + POLYB( $i, X, t$ ) *  $Z(i + 1)$ 
5:   Fin Pour
6: Fin Fonction
```

Les différentes fonctions utilisées pour la fonction **HERMITE** (directement ou indirectement) sont les suivantes :

POLYA : calcul du polynôme A_i en t , (données i, X, t)

POLYB : calcul du polynôme B_i en t , (données i, X, t)

POLYL : calcul du polynôme L_i en t , (données i, X, t)

POLYLP : calcul de $L'_i(x_i)$, (données i, X)

0.490.7

Algorithme 2 Fonction **POLYA** permettant de calculer le polynôme A_i en $t \in \mathbb{R}$ donné par $A_i(t) = (1 - 2L'_i(x_i)(t - x_i))L_i^2(t)$

```

1: Fonction  $y \leftarrow \text{POLYA} ( i, \mathbf{X}, t )$ 
2:    $y \leftarrow (1 - 2 * \text{POLYLP}(i, X) * (t - X(i + 1))) * (\text{POLYL}(i, X, t))^2$ 
3: Fin Fonction
```

0.490.7

Algorithme 3 Fonction **POLYB** permettant de calculer le polynôme B_i en $t \in \mathbb{R}$ donné par $B_i(t) = (t - x_i)L_i^2(t)$

```

1: Fonction  $y \leftarrow \text{POLYB} ( i, \mathbf{X}, t )$ 
2:    $y \leftarrow (t - X(i + 1)) * (\text{POLYL}(i, X, t))^2$ 
3: Fin Fonction
```

Algorithme 4 Fonction **POLYL** permettant de calculer le polynôme L_i en $t \in \mathbb{R}$ donné par

$$L_i(t) = \prod_{j=0, j \neq i}^n \frac{t - x_j}{x_i - x_j}$$

```

1: Fonction  $y \leftarrow \text{POLYL} ( i, \mathbf{X}, t )$ 
2:    $y \leftarrow 1$ 
3:   Pour  $j \leftarrow 0$  à  $n$ , ( $j \sim = i$ ) faire
4:      $y \leftarrow y * (t - X(j + 1)) / (X(i + 1) - X(j + 1))$ 
5:   Fin Pour
6: Fin Fonction
```

Algorithme 5 Fonction **POLYLP** permettant de calculer $L'_i(x_i) = \sum_{k=0, k \neq i}^n \frac{1}{x_i - x_k}$

```

1: Fonction  $y \leftarrow \text{POLYLP} ( i, \mathbf{X} )$ 
2:    $y \leftarrow 0$ 
3:   Pour  $k \leftarrow 0$  à  $n$ ,  $(k \sim i)$  faire
4:      $y \leftarrow y + 1/(X(i+1) - X(k+1))$ 
5:   Fin Pour
6: Fin Fonction

```

Bien évidemment une telle écriture est loin d'être optimale mais elle a l'avantage d'être facile à programmer et facile à lire car elle "colle" aux formules mathématiques.

On laisse le soin au lecteur d'écrire des fonctions plus performantes...

◇

