

On s'intéresse au système linéaire :

$$Mu = d \quad (S)$$

où  $M \in \mathcal{M}_N(\mathbb{R})$  ( $N$  étant un entier naturel supérieur ou égal à 2) est une matrice tridiagonale définie par :

$$M = \begin{pmatrix} b_1 & c_1 & & & \\ a_2 & b_2 & \ddots & (0) & \\ & \ddots & \ddots & \ddots & \\ (0) & & \ddots & b_{N-1} & c_{N-1} \\ & & & a_N & b_N \end{pmatrix}$$

et  $u$  et  $d$  les deux matrices colonnes de  $\mathcal{M}_{N,1}(\mathbb{R})$  :

$$u = \begin{pmatrix} u_1 \\ \vdots \\ u_N \end{pmatrix} \quad \text{et} \quad d = \begin{pmatrix} d_1 \\ \vdots \\ d_N \end{pmatrix}$$

Dans la suite, on suppose que toutes les fonctions Python demandées sont écrites dans un script comportant la ligne :

```
import numpy as np
```

## Partie I : construction de la matrice $M$

On note  $L_a$ ,  $L_b$  et  $L_c$  les trois listes Python de longueurs respectives  $N-1$ ,  $N$  et  $N-1$  :

```
L_a = [a_2, a_3, ..., a_N]
L_b = [b_1, b_2, ..., b_N]
L_c = [c_1, c_2, ..., c_{N-1}]
```

Ecrire une fonction Python `BuildM` qui reçoit en arguments les trois listes  $L_a$ ,  $L_b$  et  $L_c$  et renvoie un tableau numpy `M` correspondant à la matrice tridiagonale  $M$  ci-dessus.

## Partie II : résolution de $(S)$ par l'algorithme de Thomas

Dans cet algorithme, on calcul d'abord les coefficients suivants :

$$\begin{cases} c'_1 = \frac{c_1}{b_1} \\ c'_i = \frac{c_i}{b_i - a_i c'_{i-1}} \end{cases} \quad \text{pour } i = 2, 3, \dots, N-1$$

et :

$$\begin{cases} d'_1 = \frac{d_1}{b_1} \\ d'_i = \frac{d_i - a_i d'_{i-1}}{b_i - a_i c'_{i-1}} \end{cases} \quad \text{pour } i = 2, 3, \dots, N$$

Les inconnues sont alors obtenues par les formules :

$$\begin{cases} u_N = d'_N \\ u_i = d'_i - c'_i u_{i+1} \end{cases} \quad \text{pour } i = N-1, N-2, \dots, 2, 1$$

Ecrire une fonction Python `ThomasSolver` qui reçoit en argument deux tableaux numpy `M` et `d` correspondant respectivement aux matrices  $M$  et  $d$  et renvoie un tableau numpy `u` correspondant à la matrice colonne  $u$ , solution de (S).

### Partie III : complexité

Déterminer, pour la résolution du système (S) à l'aide de l'algorithme de Thomas :

- Le nombre total NS de soustractions.
- Le nombre total NM de multiplications.
- Le nombre total ND de divisions.

### Partie IV : une famille de matrices et une suite de déterminants

Pour tout entier naturel non nul on définit  $M_n \in \mathcal{M}_n(\mathbb{R})$  :

$$M_n = \begin{pmatrix} b & c & & \\ a & b & \ddots & (0) \\ & \ddots & \ddots & \ddots \\ (0) & & \ddots & b & c \\ & & & a & b \end{pmatrix}$$

où  $a$ ,  $b$  et  $c$  sont trois réels.

On pose :  $\Delta_n = \det(M_n)$ .

1. Calculer  $\Delta_1$ ,  $\Delta_2$  et  $\Delta_3$ .
2. Montrer que pour tout entier naturel  $n$  non nul, on a :  $\Delta_{n+2} = b \Delta_{n+1} - ac \Delta_n$ .
3. Ecrire une fonction Python non récursive `detMn_nr` qui reçoit en argument trois flottants  $a$ ,  $b$  et  $c$  (correspondant aux réels  $a$ ,  $b$  et  $c$ ) et un entier  $n$  (correspondant à l'entier  $n$ ) et renvoie le flottant `detMn` (correspondant au réel  $\Delta_n$ ).
4. Proposer une version récursive du calcul précédent en écrivant une fonction récursive Python `detMn_r` (les variables  $a$ ,  $b$  et  $c$  seront déclarées globales).