

Exercice: Algorithmique

Q. 1 Ecrire une fonction **FACTQR** permettant de calculer la factorisation QR d'une matrice $A \in \mathcal{M}_n(\mathbb{C})$.
On pourra utiliser la fonction **HOUSEHOLDER** (voir Exercice 3.1.9, page 91).

Q. 2 Ecrire un programme permettant de tester cette fonction.

Correction Exercice

Q. 1 L'objectif est de déterminer les matrices Q , matrice unitaire, et R matrice triangulaire supérieure telle que $A = QR$.

Données : A : matrice de $\mathcal{M}_n(\mathbb{K})$.

Résultat : Q : matrice unitaire de $\mathcal{M}_n(\mathbb{K})$.

R : matrice triangulaire supérieure de $\mathcal{M}_n(\mathbb{K})$.

On rappelle la technique utilisée dans la correction de l'exercice 3.1.10 pour déterminer l'ensemble des matrices de Householder permettant de transformer la matrice A en une matrice triangulaire supérieure. On pose

$$A^{[0]} = A, \quad A^{[k+1]} = H^{[k+1]} A^{[k]}, \quad \forall k \in \llbracket 0, n-2 \rrbracket$$

où $H^{[k+1]}$ est soit une matrice de Householder soit la matrice identité. Plus précisément, on note $\underline{s} \in \mathbb{K}^{n-k}$ le vecteur composé des $n-k$ dernières composantes de la $k+1$ -ème colonne de $A^{[k]}$ et $\mathbf{a} = \begin{pmatrix} \mathbf{0}_k \\ \underline{s} \end{pmatrix}$.

- Si $\underline{s}_1 = 0$ ou \underline{s} colinéaire à \mathbf{e}_1^{n-k} premier vecteur de la base canonique de \mathbb{K}^{n-k} alors

$$H^{[k+1]} = I.$$

En notant \mathbf{e}_{k+1}^n le $k+1$ -ème vecteur de la base canonique de \mathbb{K}^n , cette matrice peut-être calculée avec la fonction **HOUSEHOLDER** par

$$[H^{[k+1]}, \alpha] \leftarrow \text{HOUSEHOLDER}(\mathbf{a}, \mathbf{e}_{k+1}^n, 1)$$

- sinon $H^{[k+1]} = I$.

On a vu que dans ce cas $\mathbb{A}^{[n-1]}$ est triangulaire supérieure. On pose $\mathbb{H} = \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$ qui est une matrice unitaire. On a alors $\mathbb{R} = \mathbb{A}^{[n-1]} = \mathbb{H}\mathbb{A}$ et $\mathbb{Q} = \mathbb{H}^*$.

Algorithme 1 \mathcal{R}_0

- 1: Calculer \mathbb{Q} et \mathbb{R}

Algorithme 1 \mathcal{R}_1

- 1: $\mathbb{H} \leftarrow \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$
- 2: $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$
- 3: $\mathbb{Q} \leftarrow \mathbb{H}^*$

Algorithme 1 \mathcal{R}_1

- 1: $\mathbb{H} \leftarrow \mathbb{H}^{[n-1]} \times \dots \times \mathbb{H}^{[1]}$
- 2: $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$
- 3: $\mathbb{Q} \leftarrow \mathbb{H}^*$

Algorithme 1 \mathcal{R}_2

- 1: $\mathbb{H} \leftarrow \mathbb{I}$
 - 2: $\mathbb{A}^{[0]} \leftarrow \mathbb{A}$
 - 3: **Pour** $k \leftarrow 0$ à $n - 2$ **faire**
 - 4: Calculer $\mathbb{H}^{[k+1]}$ à partir de $\mathbb{A}^{[k]}$
 - 5: $\mathbb{A}^{[k+1]} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{A}^{[k]}$
 - 6: $\mathbb{H} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{H}$
 - 7: **Fin Pour**
 - 8: $\mathbb{R} \leftarrow \mathbb{H} * \mathbb{A}$
 - 9: $\mathbb{Q} \leftarrow \mathbb{H}^*$
- \triangleright ou $\mathbb{R} \leftarrow \mathbb{A}^{[n-1]}$

Algorithme 1 $\boxed{\mathcal{R}_2}$

```
1:  $\mathbb{H} \leftarrow \mathbb{I}$ 
2: Pour  $k \leftarrow 0$  à  $n - 2$  faire
3:   Calculer  $\mathbb{H}^{[k+1]}$  à partir de  $\mathbb{A}^{[k]}$ 
4:    $\mathbb{A}^{[k+1]} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{A}^{[k]}$ 
5:    $\mathbb{H} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{H}$ 
6: Fin Pour
7:  $\mathbb{R} \leftarrow \mathbb{A}^{[n-1]}$ 
8:  $\mathbb{Q} \leftarrow \mathbb{H}^*$ 
```

Algorithme 1 $\boxed{\mathcal{R}_3}$

```
1:  $\mathbb{H} \leftarrow \mathbb{I}$ 
2: Pour  $k \leftarrow 0$  à  $n - 2$  faire
3:    $\mathbf{a} \leftarrow [\mathbf{0}_k; \mathbb{A}^{[k]}(k + 1 : n, k + 1)]$ 
4:    $[\mathbb{H}^{[k+1]}, \alpha] \leftarrow \text{HOUSEHOLDER}(\mathbf{a}, \mathbf{e}_{k+1}^n, 1)$ 
5:    $\mathbb{A}^{[k+1]} \leftarrow \mathbb{H}^{[k+1]} * \mathbb{A}^{[k]}$ 
6:    $\mathbb{H} \leftarrow \mathbb{H}^{[k+1]} \mathbb{H}$ 
7: Fin Pour
8:  $\mathbb{R} \leftarrow \mathbb{A}^{[n-1]}$ 
9:  $\mathbb{Q} \leftarrow \mathbb{H}^*$ 
```

Ici, l'opérateur $[\bullet; \bullet]$ est l'opérateur de concaténation de deux vecteurs.

Algorithme 1 Fonction **FACTQR**

Données : \mathbb{A} : matrice de $\mathcal{M}_n(\mathbb{K})$.

Résultat : \mathbb{Q} : matrice unitaire de $\mathcal{M}_n(\mathbb{K})$.

\mathbb{R} : matrice triangulaire supérieure de $\mathcal{M}_n(\mathbb{K})$.

```
1: Fonction [ $\mathbb{Q}, \mathbb{R}$ ]  $\leftarrow$  FACTQR (  $\mathbb{A}$  )
2:    $\mathbb{H} \leftarrow \mathbb{I}$ 
3:    $\mathbb{R} \leftarrow \mathbb{A}$ 
4:   Pour  $k \leftarrow 0$  à  $n - 2$  faire
5:      $\mathbf{a} \leftarrow [\mathbf{0}_k; \mathbb{R}(k + 1 : n, k + 1)]$ 
6:     [ $\mathbb{S}, \alpha$ ]  $\leftarrow$  HOUSEHOLDER( $\mathbf{a}, \mathbf{e}_{k+1}^n, 1$ )
7:      $\mathbb{R} \leftarrow \mathbb{S} * \mathbb{R}$ 
8:      $\mathbb{H} \leftarrow \mathbb{S} * \mathbb{H}$ 
9:   Fin Pour
10:   $\mathbb{Q} \leftarrow \mathbb{H}^*$ 
11: Fin Fonction
```

Q. 2

