

Durée 2 heures.

Les documents papier sont autorisés, tout matériel électronique est interdit.

Les quaternions de Hamilton sont des nombres hypercomplexes qui se représentent à l'aide de quatre coefficients réels et trois racines de -1 :  $i$ ,  $j$  et  $k$  qui ont les propriétés suivantes :

$$\begin{aligned} i^2 &= j^2 = k^2 = ijk = -1 \\ ij &= -ji \\ jk &= -kj \\ ki &= -ik \end{aligned}$$

Un quaternion est alors défini comme  $q = t + ui + vj + wk$ , avec  $t, u, v, w \in \mathbb{R}$ . On peut définir les opérations habituelles suivantes sur les quaternions. Soient  $q_1 = t_1 + u_1i + v_1j + w_1k$  et  $q_2 = t_2 + u_2i + v_2j + w_2k$ , avec  $t_l, v_l, u_l, w_l \in \mathbb{R}$ ,  $l = 1, 2$  deux quaternions alors :

$$\begin{aligned} q_1 + q_2 &= t_1 + t_2 + (u_1 + u_2)i + (v_1 + v_2)j + (w_1 + w_2)k \\ q_1 - q_2 &= t_1 - t_2 + (u_1 - u_2)i + (v_1 - v_2)j + (w_1 - w_2)k \\ q_1 q_2 &= t_1 t_2 - u_1 u_2 - v_1 v_2 - w_1 w_2 + \\ &\quad (t_1 u_2 + u_1 t_2 + v_1 w_2 - w_1 v_2)i + \\ &\quad (t_1 v_2 - u_1 w_2 + v_1 t_2 + w_1 u_2)j + \\ &\quad (t_1 w_2 + u_1 v_2 - v_1 u_2 + w_1 t_2)k \end{aligned}$$

On notera que la multiplication de quaternions n'est pas commutative. On définit le conjugué du quaternion  $q$  par :

$$q^* = t - ui - vj - wk$$

et sa norme par :

$$\|q\| = \sqrt{qq^*} = \sqrt{t^2 + u^2 + v^2 + w^2}$$

Ceci nous permet de définir l'inverse multiplicatif d'un quaternion :

$$q^{-1} = \frac{q^*}{\|q\|^2}$$

Un nombre réel  $t$  est aussi un quaternion  $t + 0i + 0j + 0k$ . De la même façon pour un complexe  $z = t + ui = t + ui + 0j + 0w$ .

#### Exercice 1

Écrire une classe C++ nommée **Quaternion**, représentant les quaternions de Hamilton ainsi que ses opérations simples :

- Les opérations composées  $+=$ ,  $-=$ ,  $*=$  et  $/=$ .
- Les opérations somme, différence, produit, quotient, conjugué et norme.

La classe doit appartenir à l'espace de noms `ensiie`. Vous devez pouvoir convertir des réels et des complexes vers des quaternions. Les erreurs seront traitées par des exceptions appropriées. On écrira la définition de la classe dans un fichier d'en-tête et la définition des fonctions dans un fichier source.

#### Exercice 2

Écrire le code qui permet aux flots standards `std::ostream` d'afficher des quaternions représentés par la classe de l'exercice 1.

#### Exercice 3

Généraliser la classe de l'exercice 1 aux quaternions à coefficients définis par un type numérique **quelconque** à simple ou double précision.

#### Exercice 4

Écrire la fonction template suivante (incomplète, à corriger et à implémenter) :

`bool equal<T>(Quaternion<T> q, T r, T s, T t, T u)`

qui renvoie `true` si le quaternion a pour composante les quatre nombres en paramètre, `false` sinon. Tester dans un programme chacune des opérations de l'exercice 1 en utilisant cette fonction template.

*Quaternion<T> q1*