Algèbre de Grassmann-Cayley

Projet de mathématiques pour l'informatique imac2

Résumé

Ce projet consiste à implémenter une bibliothèque en `C++` permettant d'utiliser l'algèbre de Grassmann-Cayley. Il s'agit d'une algèbre permettant de représenter et manipuler très simplement les points, les droites et les plans dans un espace projectif. L'un des avantages de cette algèbre est de rester tout à fait compatible avec l'algèbre linéaire et n'est donc qu'un outil supplémentaire n'imposant pas de contraintes particulières quant à son utilisation.

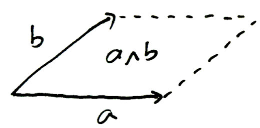
1 Introduction

L'aglèbre de Grassmann, aussi appelée algèbre extérieure, a été introduite au milieu du 19e siècle par le mathématicien Grassmann. Il s'agit d'une algèbre introduisant le produit extérieur, ou *wedge product* qui servira de base à de nombreuses algèbres géométriques. Malheureusement pour lui, Grassmann ne connaîtra jamais le succès de son vivant. Cayley, contemporain de Grassmann, voit dans ces travaux une application dans l'espace projectif. Il s'agit de représenter les points, les droites et les plans d'une façon extrêmement simple et compacte, fonctionnant quelle que soit la dimension de l'espace traité. Par ailleurs, cette représentation reste tout à fait compatible avec l'aglèbre linéaire et n'est donc qu'un outil supplémentaire n'imposant pas de contraintes particulières quant à son utilisation. L'algèbre de Grassmann-Cayley commence en ce moment à se populariser et a indéniablement de l'avenir dans l'industrie du jeu vidéo et de la vision par ordinateur.

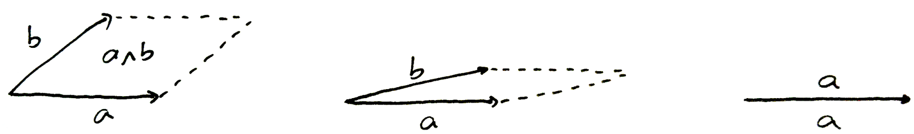
2 Algèbre de Grassmann

2.1 Le wedge product

L'algèbre de Grassmann est principalement définie par le produit extérieur noté $\wedge$ (lire "*wedge*"). Il s'agit d'un produit tel que $\mathbf{a}\wedge\mathbf{b}$ représente la surface orientée du parallélogramme défini par les vecteurs $\mathbf{a}$ et $\mathbf{b}$.



Le caractère orienté se traduit par $\mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a}$, ce qui implique que $\mathbf{a} \wedge \mathbf{a} = 0$.



Plus généralement, voici les propriétés du wedge product, pour $\alpha, \beta \in \mathbb{R}$ et $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^n$:

$$\begin{aligned}
\textbf{Multiplication scalaire :}\quad & \alpha \wedge \beta = \beta \wedge \alpha = \beta\alpha \\
& \beta \wedge \mathbf{a} = \mathbf{a} \wedge \beta = \beta\mathbf{a} \\[1ex]
\textbf{Anticommutativité :}\quad & \mathbf{a} \wedge \mathbf{b} = -\mathbf{b} \wedge \mathbf{a} \\
& \Rightarrow \mathbf{a} \wedge \mathbf{a} = 0 \\[1ex]
\textbf{Mise à l'échelle :}\quad & \mathbf{a} \wedge (\beta\mathbf{b}) = \beta(\mathbf{a} \wedge \mathbf{b}) \\[1ex]
\textbf{Distributivité :}\quad & \mathbf{a} \wedge (\mathbf{b} + \mathbf{c}) = (\mathbf{a} \wedge \mathbf{b}) + (\mathbf{a} \wedge \mathbf{c}) \\
& (\mathbf{a} + \mathbf{b}) \wedge \mathbf{c} = (\mathbf{a} \wedge \mathbf{c}) + (\mathbf{b} \wedge \mathbf{c}) \\[1ex]
\textbf{Associativité :}\quad & (\mathbf{a} \wedge \mathbf{b}) \wedge \mathbf{c} = \mathbf{a} \wedge (\mathbf{b} \wedge \mathbf{c})
\end{aligned}$$

2.2 Les bivecteurs

Un bivecteur est une entité construite par le wedge de deux vecteurs de $\mathbb{R}^n$. La construction du bivecteur se fait en utilisant les propriétés présentées dans la section 2.1. Voici un exemple dans le cas de $\mathbb{R}^2$, les vecteurs $\mathbf{a}$ et $\mathbf{b}$ se décomposent selon les vecteurs de bases $(\mathbf{e_1}, \mathbf{e_2})$ de la façon suivante :

$$\mathbf{a} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} \quad \text{et} \quad \mathbf{b} = \begin{pmatrix} b_x \\ b_y \end{pmatrix}$$

soit

$$\mathbf{a} = a_x\mathbf{e_1} + a_y\mathbf{e_2} \quad \text{et} \quad \mathbf{b} = b_x\mathbf{e_1} + b_y\mathbf{e_2}$$

Le wedge de deux vecteurs $\mathbf{a}$ et $\mathbf{b}$ de $\mathbb{R}^2$ donne **un bivecteur** dont les composantes sont :

$$\begin{aligned}
\mathbf{a} \wedge \mathbf{b} &= (a_x\mathbf{e_1} + a_y\mathbf{e_2}) \ \wedge \ (b_x\mathbf{e_1} + b_y\mathbf{e_2}) \\
&= (a_xb_x)(\mathbf{e_1} \wedge \mathbf{e_1}) \ + \ (a_xb_y)(\mathbf{e_1} \wedge \mathbf{e_2}) + (a_yb_x)(\mathbf{e_2} \wedge \mathbf{e_1}) + (a_yb_y)(\mathbf{e_2} \wedge \mathbf{e_2}) \\
&= (a_xb_y - a_yb_x)(\mathbf{e_1} \wedge \mathbf{e_2}) \\
&= (a_xb_y - a_yb_x)\mathbf{e_{12}}
\end{aligned}$$

A noter que dans $\mathbb{R}^2$, le bivecteur $\mathbf{a} \wedge \mathbf{b}$ n'a qu'une seule composante selon $\mathbf{e_{12}}$, ce qui n'est pas nécessairement le cas pour des vecteurs de $\mathbb{R}^n$ $(n > 2)$.

2.3 Les grades

Le wedge de trois vecteurs $\mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c}$ donne un **trivecteur**. Dans $\mathbb{R}^3$, un trivecteur représente un volume. Dans $\mathbb{R}^2$, on peut voir facilement que les trivecteurs sont tous nuls. Par extension, on appelle ***k*-blade** un élément dont la construction peut être faite par le wedge de k vecteurs. Le nombre k est appelé **le grade** de la blade.

$$\begin{array}{lcl}
\text{scalaire} & \rightarrow & \text{0-blades} \\
\text{vecteurs} & \rightarrow & \text{1-blades} \\
\text{bivectors} & \rightarrow & \text{2-blades} \\
\text{trivectors} & \rightarrow & \text{3-blades} \\
\vdots & & \vdots
\end{array}$$

Par exemple, voici les vecteurs définissant les bases de chaque k-blade dans $\mathbb{R}^4$:

1 scalaire	$(\mathbf{1})$	grade 0
4 vecteurs	$(\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, \mathbf{e_4})$	grade 1
6 bivecteurs	$(\mathbf{e_{12}}, \mathbf{e_{13}}, \mathbf{e_{14}}, \mathbf{e_{23}}, \mathbf{e_{24}}, \mathbf{e_{34}})$	grade 2
4 trivecteurs	$(\mathbf{e_{123}}, \mathbf{e_{124}}, \mathbf{e_{134}}, \mathbf{e_{234}})$	grade 3
1 quadvecteur	$(\mathbf{e_{1234}})$	grade 4

2.4 Les bases duales

Une base duale est définie par la relation : $\mathbf{e_{\{i\}}} \wedge \overline{\mathbf{e}}_{\mathbf{\{i\}}} = \mathbf{e_{1\cdots n}}$. Autrement dit, la base duale $\overline{\mathbf{e}}_{\mathbf{\{i\}}}$ d'une base $\mathbf{e_{\{i\}}}$ contient tous les vecteurs unitaires de dimension 1 que $\mathbf{e_{\{i\}}}$ ne contient pas. Par ailleurs, $\mathbf{e_{\{i\}}} \wedge \overline{\mathbf{e}}_{\mathbf{\{i\}}}$ doit générer le le k-vecteur unitaire contenant tous les vecteurs de dimension 1, dans l'ordre. Par exemple, la base duale de $\overline{\mathbf{e}}_{\mathbf{24}}$ dans $\mathbb{R}^4$ est $-\mathbf{e_{13}}$ car on a :

$$\mathbf{e_{24}} \wedge \overline{\mathbf{e}}_{\mathbf{24}} = \mathbf{e_{24}} \wedge (-\mathbf{e_{13}}) = -\mathbf{e_2} \wedge \mathbf{e_4} \wedge \mathbf{e_1} \wedge \mathbf{e_3} = \mathbf{e_1} \wedge \mathbf{e_2} \wedge \mathbf{e_3} \wedge \mathbf{e_4} = \mathbf{e_{1234}}$$

On a donc bien $\mathbf{e_{24}} \wedge \overline{\mathbf{e}}_{\mathbf{24}} = \mathbf{e_{1234}}$ avec $\mathbf{e_{13}} = -\overline{\mathbf{e}}_{\mathbf{24}}$.

Voici d'autres bases duales dans $\mathbb{R}^4$:

$$\begin{array}{lll} \mathbf{e_1} = -\overline{\mathbf{e}}_{\mathbf{234}} & \mathbf{e_{12}} = \ \ \overline{\mathbf{e}}_{\mathbf{34}} & \mathbf{e_{123}} = -\overline{\mathbf{e}}_{\mathbf{4}} \\ \mathbf{e_2} = \ \ \overline{\mathbf{e}}_{\mathbf{134}} & \mathbf{e_{13}} = -\overline{\mathbf{e}}_{\mathbf{24}} & \mathbf{e_{124}} = \ \ \overline{\mathbf{e}}_{\mathbf{3}} \end{array}$$

Les k-blades peuvent alors être exprimées aussi bien selon leur base d'origine que selon leur base duale, comme spécifié dans le tableau suivant :

1 **scalaire** $(\mathbf{1})$	1 **anti-quadvecteur** $(\overline{\mathbf{e}}_{\mathbf{1234}})$	(grade 0)
4 **vecteurs** $(\mathbf{e_1}, \mathbf{e_2}, \mathbf{e_3}, \mathbf{e_4})$	4 **anti-trivecteurs** $(\overline{\mathbf{e}}_{\mathbf{123}}, \overline{\mathbf{e}}_{\mathbf{124}}, \overline{\mathbf{e}}_{\mathbf{134}}, \overline{\mathbf{e}}_{\mathbf{234}})$	(grade 1)
6 **bivecteurs** $(\mathbf{e_{12}}, \mathbf{e_{13}}, \mathbf{e_{14}}, \mathbf{e_{23}}, \mathbf{e_{24}}, \mathbf{e_{34}})$	6 **anti-bivecteurs** $(\overline{\mathbf{e}}_{\mathbf{12}}, \overline{\mathbf{e}}_{\mathbf{13}}, \overline{\mathbf{e}}_{\mathbf{14}}, \overline{\mathbf{e}}_{\mathbf{23}}, \overline{\mathbf{e}}_{\mathbf{24}}, \overline{\mathbf{e}}_{\mathbf{34}})$	(grade 2)
4 **trivecteurs** $(\mathbf{e_{123}}, \mathbf{e_{124}}, \mathbf{e_{134}}, \mathbf{e_{234}})$	4 **anti-vecteurs** $(\overline{\mathbf{e}}_{\mathbf{1}}, \overline{\mathbf{e}}_{\mathbf{2}}, \overline{\mathbf{e}}_{\mathbf{3}}, \overline{\mathbf{e}}_{\mathbf{4}})$	(grade 3)
1 **quadvecteur** $(\mathbf{e_{1234}})$	1 **anti-scalaire** $(\overline{\mathbf{1}})$	(grade 4)

Le vecteur $\overline{\mathbf{a}}$ correspond ainsi au vecteur $\mathbf{a}$, exprimé dans sa base duale.

2.5 Anti-wedge

L'algèbre de Grassmann introduit un second opérateur dit anti-wedge, noté $\mathbf{a} \vee \mathbf{b}$. Il possède les mêmes propriétés que le wedge product mais sur les bases dualles, à savoir :

$$\mathbf{a} \vee \mathbf{b} = \overline{\mathbf{a}} \wedge \overline{\mathbf{b}}$$

Le grade des k-vecteurs de dimension n évoluent alors de la façon suivante :

$$\begin{aligned} \text{wedge :} \quad & k\text{-blade} \wedge j\text{-blade} = (k+j)\text{-blade} \\ \text{anti-wedge :} \quad & k\text{-blade} \vee j\text{-blade} = (k+j-n)\text{-blade} \end{aligned}$$

3 Algèbre de Grassmann-Cayley

3.1 Géométrie projective

L'algèbre de Grassmann-Cayley consiste à utiliser l'algèbre de Grassmann sur des vecteurs de l'espace projectif. Pour rappel, le vecteur $\mathbf{x} \in \mathbb{R}^3$ dont les coordonnées cartésiennes sont $\mathbf{x} = (x, y, z)^\top$ deviennent $\mathbf{x} = (x, y, z, w)^\top$ dans $\mathbb{P}^3$, où la composante w vaut 0 pour un point à l'infini et $w \neq 0$ pour un point fini. Par ailleurs, si $w = 1$, les composantes x, y et z ont les mêmes valeurs que dans l'espace cartésien. Enfin, les vecteurs $\mathbf{x} \in \mathbb{P}^3$ et $\alpha\mathbf{x}$ ($\alpha \in \mathbb{R}^*$) représentent la même entité.

3.2 Propriétés dans $\mathbb{P}^3$

Appliquées aux vecteurs de $\mathbb{P}^3$, les opérateurs de wedge et anti-wedge ont des propriétés particulièrement intéressantes.

Propriétés :

point $\wedge$ point $=$ droite	droite passant par les 2 points
point $\wedge$ point $\wedge$ point $=$ plan	plan contenant les 3 points
point $\wedge$ droite $=$ plan	contenant la droite et le plan
plan $\vee$ plan $=$ droite	intersection de 2 plans
plan $\vee$ plan $\vee$ plan $=$ point	intersection de 3 plans
plan $\vee$ droite $=$ point	intersection entre 1 droite et 1 plan

Relations entre ces entités :

point $\in$ droite	$\leftrightarrow$	point $\wedge$ droite $= 0$ / point $\vee$ droite $= 0$
point $\in$ plan	$\leftrightarrow$	point $\wedge$ plan $= 0$ / point $\vee$ plan $= 0$

4 Droites de Plücker

Comme présenté dans la section précédente, la droite $\mathbf{l}$ passant par les deux points $\mathbf{a}$ et $\mathbf{b}$ est définie par $\mathbf{l} = \mathbf{a} \wedge \mathbf{b}$. Dans $\mathbb{P}^3$ une telle droite a alors 6 composantes :

$$\mathbf{l} = \mathbf{a} \wedge \mathbf{b} = l_1\mathbf{e_{12}} + l_2\mathbf{e_{13}} + l_3\mathbf{e_{14}} + l_4\mathbf{e_{23}} + l_5\mathbf{e_{24}} + l_6\mathbf{e_{34}}$$

Il se trouve que ces 6 composantes correspondent aux coefficients de Plücker définissant une droite dans $\mathbb{P}^3$. D'une façon générale, une droite de Plücker est constituée de 2 vecteurs $\mathbf{u}$ et $\mathbf{v}$ de $\mathbb{R}^3$. Le vecteur $\mathbf{u}$ correspond au vecteur directeur de la droite et le vecteur $\mathbf{v}$ à son moment, c'est-à-dire à un vecteur orthogonal à $\mathbf{u}$ et porté par le plan passant par la droite et par l'origine. L'amplitude du moment correspond à la distance entre l'origine et le point de la droite le plus proche de l'origine, dans la métrique de la norme de $\mathbf{u}$. En pratique, la décomposition d'une droite $\mathbf{l}$ en deux vecteurs $\mathbf{u}$ et $\mathbf{v}$ est :

$$\mathrm{L} = \{\mathbf{u} : \mathbf{v}\} = \{(l_3, l_5, l_6)^\top : (-l_4, l_2, -l_1)^\top\}$$

Les droites de Plücker sont ainsi définies par :

- $\mathrm{L} = \{\mathbf{u} : \mathbf{v}\}$ avec $\mathbf{u}$ et $\mathbf{v} \in \mathbb{R}^3$ et $\mathbf{u} \cdot \mathbf{v} = 0$.
- $\mathbf{u}$: vecteur support de la droite.
- $\mathbf{v}$: moment de la droite.

Plus généralement, les propriétés des droites de Plücker sont :

- $\{\mathbf{u} : \mathbf{v}\}$ et $\alpha\{\mathbf{u} : \mathbf{v}\}$ $(\alpha \neq 0)$ représentent la même droite.
- $\mathrm{L} = \{\mathbf{p} - \mathbf{q} : \mathbf{p} \times \mathbf{q}\}$: droite passant par les points $\mathbf{p}$ et $\mathbf{q} \in \mathbb{R}^3$.
- $\mathrm{L} = \{q_w\mathbf{p}_{xyz} - p_w\mathbf{q}_{xyz} : \mathbf{p}_{xyz} \times \mathbf{q}_{xyz}\}$: droite passant par $\mathbf{p}$ et $\mathbf{q} \in \mathbb{P}^3$.
- $\mathrm{L} = \{\mathbf{u} : \mathbf{u} \times \mathbf{p}\}$: droite de vecteur support $\mathbf{u}$ passant par le point $\mathbf{p} \in \mathbb{R}^3$.
- $\mathbf{x} = (\mathbf{v} \times \mathbf{u}, \mathbf{u} \cdot \mathbf{u})^\top \in \mathbb{P}^3$: point de la droite le plus proche de l'origine.
- distance entre la droite L et un point $\mathbf{x} \in \mathbb{R}^3$: $\mathrm{d} = \|(\mathbf{x} \times \mathbf{u} - \mathbf{v})^\top\|_2 / \|\mathbf{u}\|_2^2$.

5 Forme Hessienne d'un plan

De même, un plan π passant par trois points $\mathbf{a}$, $\mathbf{b}$ et $\mathbf{c}$ comprend 4 composantes :

$$\pi = \mathbf{a} \wedge \mathbf{b} \wedge \mathbf{c} = \pi_1\mathbf{e_{123}} + \pi_2\mathbf{e_{124}} + \pi_3\mathbf{e_{134}} + \pi_4\mathbf{e_{234}}$$

Un point $\mathbf{x} = (x, y, z, w)^\top$ appartient au plan π si $\mathbf{x} \wedge \pi = 0$. Ce produit se décompose $\mathbf{x} \wedge \pi = (x\pi_4 - y\pi_3 + z\pi_2 + -w\pi_1)\mathbf{e_{1234}} = 0$. On reconnaît la forme Hessienne d'un plan qui se note aussi $\mathbf{n}^\top \cdot (x, y, z)^\top = w\pi_1$, avec $\mathbf{n} = (\pi_4, -\pi_3, \pi_2)^\top$ correspondant au vecteur normal au plan et π_1 à la distance entre l'origine et le plan dans la métrique de $\|\mathbf{n}\|_2$. A noter que π ou $\alpha\pi$ ($\alpha \neq 0$) représentent le même plan. Ainsi, dans le cas où $\|\mathbf{n}\|_2 = 1$ et $w = 1$, on obtient la forme Hessienne normalisée $\mathbf{n}^\top \cdot \mathbf{x} = \pi_1$ où π_1 représente la distance algébrique (signée) entre l'origine et le plan.

Remarque : cette forme Hessienne se généralise pour tous les hyperplans de dimension n et en particulier dans $\mathbb{P}^2$ où il représente une droite. Par conséquent, dans $\mathbb{P}^2$, la doite $\mathbf{l}$ passant par 2 points $\mathbf{a}$ et $\mathbf{b}$ se caclule par $\mathbf{l} = \mathbf{a} \wedge \mathbf{b}$ mais on a aussi l'intersection $\mathbf{x}$ de 2 droites $\mathbf{l}_1$ et $\mathbf{l}_2$ qui se calcule par $\mathbf{x} = \mathbf{l}_1 \wedge \mathbf{l}_2$ ce qui n'est pas vrai dans $\mathbb{P}^3$.

6 Travail demandé

L'objectif de ce projet est d'implémenter une bibliothèque permettant de gérer les éléments de l'algèbre de Grassmann-Cayley dans $\mathbb{P}^3$. Plus précisément, vous deverez :

- développer une bibliothèque en `C++` (voir section 6.1 pour les détails).
- développer un programme d'exemple d'utilisation de votre bibliothèque.
- écrire un rapport d'une dizaine de pages.
- rendre votre projet sous forme d'archive `nom1_nom2.tgz` compressant un répertoire du même nom contenant votre programme ainsi que votre rapport au format pdf.

6.1 Bibliothèque

La bibliothèque doit satisfaire les conditions suivantes :

- être écrit en `C++`, utiliser un `makefile` générée par `cmake`, et ne plus afficher de warning lors de la compilation en compilant avec `-Wall`.
- fonctionner au moins sous Linux sur les machines de l'université.
- vous pouvez tout à fait utiliser `eigen` (http ://eigen.tuxfamily.org), en faisant de l'héritage par exemple.

6.1.1 Prototypes

Votre bibliothèque doit comprendre :
- les scalaires : `gca::GCA_scalar` et `gca::GCA_antiquadvector`.
- les vecteurs : `gca::GCA_vector` et `gca::GCA_antitrivector`.
- les bivecteurs : `gca::GCA_bivector` et `gca::GCA_antibivector`.
- les trivecteurs : `gca::GCA_trivector` et `gca::GCA_antivector`.
- les quadvecteurs : `gca::GCA_quadvector` et `gca::GCA_antiscalar`.

Lors du developpement, il est recommandé (sans obligation) de vous aider de la bibliothèque `eigen`, notament avec de l'héritage.

Il sera nécessaire d'identifier la base de chacune des compoasantes d'un k-balde (cet ordre est arbitratire), par exemple de dire que la 3ème composante d'un bivecteur est supporté par $\mathbf{e_{13}}$. Une possibilité consiste à associer à chaque composante un `uint` dont le $i$ème bit est mis à 1 si le $i$ème vecteur de base de dimension un est une de ses composantes. Il y a d'autres alternatives très performantes.

6.1.2 Méthodes

Chacun d'eux devra implémenter
- un constructeur par défaut.
- un constructeur par recopie.
- un constructeur du type `gca::GCA_xxx maVariable(mesParametres, ...)`.
- un opérateur `=`
- un opérateur wedge `^`
- un convertisseur en base duale noté `~`
- un opérateur `<<` pour les `std::cout`.
- un opérateur `<<` pour l'initialisation.

Vous pouvez tout à fait rajouter d'autres méthodes en y faisant référence dans votre rapport si elles présentent un intéret particulier.

6.1.3 Exemple

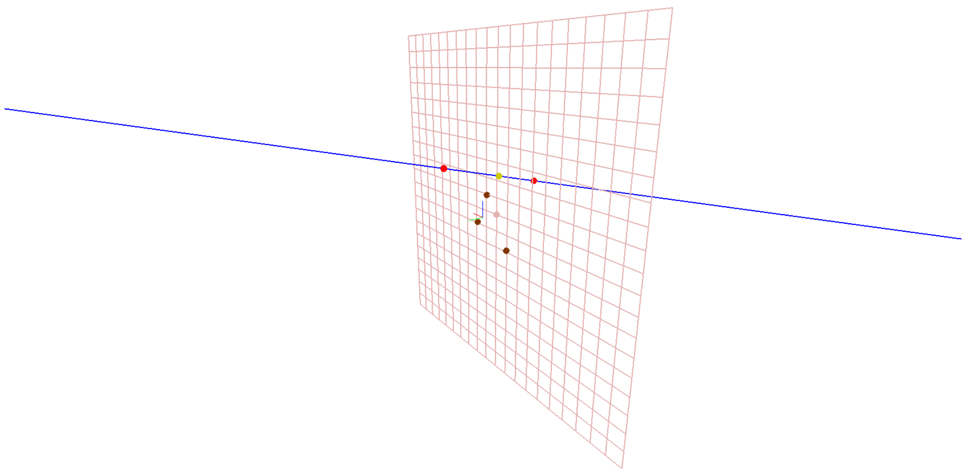
En définitive, votre bibliothèque devra absolument être compatible avec les notations suivantes :

```
#include <iostream>
#include "grassmannCayley.hpp"

...

void plop(){

  gca::GCA_vector a(1.0,2.0,3.0,1.0);
  gca::GCA_vector b;
  b << -1.0, -3.0,  2.0, 1.0;

  gca::GCA_bivector l = a^b; // l is a Plucker line
  std::cout << "l : " << l << std::endl;

  gca::GCA_vector x1(2.0,-1.0,-1.0,1.0);
  gca::GCA_vector x2(1.0,-1.0,1.0,1.0);
  gca::GCA_vector x3(-1.0,-1.0,-2.0,1.0);
  gca::GCA_trivector d = x1^x2^x3; // d is a plane

  std::cout << "intersection : " << ~d^~l << std::endl;

}

...
```

6.2 Un code d'utilisation

Vous devez faire un code d'exemple d'utilisation de votre bibliothèque. Par ailleurs, vous serez noté sur un exemple standard, il faut par conséquent qu'il soit possible de remplacer votre code par celui de l'enseignant (cf. section 6.1.3).

Vous pouvez tout à fait rajouter un module optionnel de visualisation 3D en openGL.



6.3 Rapport

Vous fournirez en version électronique un rapport de 10 pages maximum et 5 pages minimum. **Consacrez suffisamment de temps au rapport car il représente une bonne partie de la note finale**. Essayez de respecter au mieux les directives suivantes :

- **Votre rapport doit commencer par une page listant vos travaux avec les indications suivantes** :
  - éléments demandés et codés qui fonctionnent.
  - éléments demandés et codés qui ne fonctionnent pas.
  - éléments demandés mais pas codés.
  - éléments non demandés (options) et codés qui fonctionnent.
  - éléments non demandés mais pas codés ou qui ne fonctionnent pas.

- Ne perdez pas de temps à réexpliquer le sujet du projet, l'enseignant le connaît déjà, faites seulement un bref résumé de quelques lignes. De manière plus générale, ne détaillez pas des méthodes déjà expliquées dans l'énoncé à moins que vous les ayez modifiées.
- Un rapport sert surtout à montrer comment vous avez fait face aux problèmes (d'ordre algorithmique/mathématique). Certains problèmes sont connus (on en parle dans l'énoncé), d'autres sont imprévus. Montrez que vous les avez remarqués et compris. Donnez la liste des solutions à ce problème et indiquez votre choix. Justifiez votre choix (vous avez le droit de dire que c'est la méthode la plus facile à coder).

– Il ne doit figurer aucune ligne de code dans votre rapport. Un rapport n'est pas un listing de votre programme où vous détaillez chaque fonction. Vous devez par contre détailler vos structures de données et mettre du pseudocode pour expliquer vos choix algorithmiques.
  Il est autorisé d'utiliser des “raccourcis” tels que “`initialiser T à 0`” plutôt que de détailler la boucle faisant la même chose.
– n'hésitez pas à mettre des images dans votre rapport pour illustrer vos propos et vos résultats.

7 Pour finir

Vous pouvez laisser libre cours à votre imagination, toute amélioration sera la bienvenue. Vous trouverez quelques informations complémentaires à l'adresse :

`http://www-igm.univ-mlv.fr/~vnozick/`

Bon courage.