

Science algèbrale

La science c'est le plaisir de discuter pour comprendre

Gianni Mocellin

Straco
www.straco.ch
04.02.2024, 05h00

Introduction.....	4
Clifford.m.....	4
Manual	4
Introduction.....	4
Fonctions.....	6
Coeff[m, b]	6
Dual[m, d]	6
e[i]	6
GADraw[m, v].....	6
GeometricProduct[m ₁ , m ₂ , ...]	7
Grade[m, r]	7
InnerProduct[m ₁ , m ₂ , ...]	7
Magnitude[m]	7
MultivectorInverse[m]	7
OuterProduct[m ₁ , m ₂ , ...]	7
Projection[v, b]	7
Pseudoscalar[n]	7
Reflection[v, w, x]	8
Rejection[v, b]	8
Rotation[v, w, x, theta]	8
ToBasis[v]	8
ToVector[v, d]	8
Turn[m]	8
Paper	9
Introduction.....	9
Clifford algebra of R^n.....	9
Clifford algebra with Mathematica	9
Vectors.....	9
Drawing multi in 3D.....	9
Complex numbers.....	9
Quaternions	9
Grassman algebra.....	9
The hyperbolic plane.....	9
Dirac and Pauli algebra.....	9
Conclusion.....	9
Code comments.....	9
\$SetSignature	10
\$FirstIndex	11
CliffordBasic.m.....	11
Code comments.....	11
Injection et enjection.....	12
Meet et Join.....	13
Code functions.....	13
Coeff[m, b]	13
Grade[m, r]	14
HomogeneousQ[x, r]	14
GFactor[m]	14
Pseudosclar[n].....	14
Turn[m]	14
Involution[m]	14
MultivectorInverse[m].....	14
Dual[m,n]	14
Meet[m ₁ , m ₂ , n]	15

Joint[m1, m2, n].....	15
Reflection[v,w].....	15
Rotation[x, a, b, theta]	15
ToBasis[x]	15
ToVector[x,n].....	15
\$SetSignature[p, q]	15
\$FirstIndex[i]	16
CGAlgebra.m	16
Paper	16
Introduction	16
Première définition	16
Seconde définition	17
Utilisation	19
Représentation des idées	20
Points.....	20
Directions.....	20
Plans.....	20
Sphère.....	20
Représentation des transformations	20
Translations.....	20
Rotations.....	20
Fonctions additionnelles prédéfinies	20
Code comments	21
Code functions	22
<i>e</i>	22
Grade[m, r]	22
GradeQ[m, r]	22
MultiplicationTable[e].....	22
GFactor[m].....	22
GeometricProduct[m1, m2].....	22
OuterProduct[m1, m2]	22
InnerProduct[m1, m2]	22
Reversion[m].....	22
Involution[m].....	23
Magnitude[m].....	23
MultivectorInverse[m].....	23
I5	23
I5i.....	23
Dual[m].....	23
ToVector[v].....	23
Rotation[x, a, b, theta].....	23
Mathstar	23

Introduction

Le présent texte a pour but de présenter la science algèbrale

La pensée est capable d'additionner et de multiplier des nombres entre eux

Elle est également capable de conjindre des unos-flèche entre elles

Elle est tout aussi capable de moduler des uno-flèches par un nombre pour obtenir une mono-flèche

Elle est également capable d'injecter une mono dans une autre mono pour obtenir une mono projectée, c'est-à-dire capable de comprendre de combien une mono est dépendante d'une autre mono

Elle est enfin capable d'éjecter une mono d'une autre mono pour concevoir une exo

Les conjonctions, modulations, éjections, injections, projections et réjections sont parfaitement définies à $1, 2, 3, \dots, n$ dimensions, pas seulement jusqu'à 3 dimensions

Toutes ces opérations sont liées à l'opération plus générale qu'est l'imposition

La science algèbrale généralise les idées de nombre et d'unos en permettant l'adjonction de nombres, d'unos et de toute imposition de telles unos sous forme d'idées générales que l'on peut appeler des multi

Clifford.m

Manual

Introduction

Clifford.m fonctionne dans selon l'unologie

$$\{e_1, e_2, \dots, e_n\}$$

d'une réalité n -dimensionnelle R^n

de signature

$$p+q$$

où

$$i = 1, 2, \dots, n = p+q$$

dotée d'une forme bilinéaire

$$\langle x, y \rangle$$

de signature p avec la relation

$$e_i e_j + e_j e_i = 2 \langle e_i, e_j \rangle$$

où

$$\langle e_i, e_j \rangle = 0 \quad \text{si } i \neq j$$

$$\langle e_i, e_i \rangle = 1 \quad \text{si } i = 1 \dots p$$

$$\langle e_i, e_i \rangle = -1 \quad \text{si } i = p+1 \dots n$$

Tous les résultats des calculs sont donnés en termes d'unologie

$$\{e_i\}$$

Les unos sont notées de la manière suivante

$$e[i]$$

Il faut prendre soin de préserver l'ordre canonique d'introduction car *clifford.m* utilise le produit commutatif * de Mathematica et les expressions sont automatiquement réécrites en ordre canonique

L'usage de la fonction *GeometricProduct* est recommandée de manière à éviter des erreurs

Ainsi la multi

$$M = e_1 e_2 e_3$$

doit être écrite comme

$$M = \text{GeometricProduct}[e[1], e[3], e[2]]$$

ce qui donne

$$-e_1 e_2 e_3$$

mais on peut également écrire directement

$$M = - e[1] e[2] e[3]]$$

ce qui donne

$$-e_1 \ e_2 \ e_3$$

La signature de la forme bilinéaire de l'unologie

$$\langle x, y \rangle$$

peut être fixée en utilisant

$$\$SetSignature = p$$

Si aucune valeur n'est spécifiée en début de session, la valeur par défaut est

$$p = 20$$

A l'exception de la fonction *Dual* il n'est pas nécessaire de définir la dimension de la réologie

$$R^{p,q}$$

Etant données deux ou plusieurs multi, la dimension maximale de l'unologie sur laquelle elle sont construites est calculé automatiquement

Fonctions

$$Coeff[m, b]$$

Extrait les coefficients d'une *exo b (blade)* dans une multi *m*

$$Dual[m, d]$$

Calcule le dual de la multi *m* dans R^d

m est une multi et *d* est un entier positif

$$e[i]$$

Dénote la *i*ème uno de I^d

i est un entier plus grand que 0

$$GADraw[m, v]$$

Dessine une multi *m* dans R^3

m est une multi et *v* est le point de vue du graphique

v est facultatif et la valeur par défaut est

$$\text{ViewPoint} \rightarrow \{0, 1, 0\}$$

GeometricProduct $[m_1, m_2, \dots]$

Calcule l'imposition des multis m_1, m_2, \dots

Grade $[m, r]$

Extrait le terme de complexité r de la multi m

InnerProduct $[m_1, m_2, \dots]$

Calcule l'injection des multis m_1, m_2, \dots

Magnitude $[m]$

Calcule la magnitude de la multi m

MultivectorInverse $[m]$

Calcule l'inverse (si elle existe) de la multi m

OuterProduct $[m_1, m_2, \dots]$

Calcule l'éjection des multis m_1, m_2, \dots

Projection $[v, b]$

Calcule la projection de la mono v sur l'idée couverte (spanned by) par l'exo b

v est une mono et b est une r -exo

Pseudoscalar $[n]$

Donne l'omni-unos de n

n est un entier positif

Reflection[v, w, x]

Calcule la réflexion spéculaire de la mono v par le plan couvert (spanned) par les monos w et x

v , w et x sont des monos

Rejection[v, b]

Calcule la projection horthogonale de la mono v sur le complément indépendant de l'idée couverte par l'exo b

v est une mono et b est une r -exo

Rotation[v, w, x, theta]

Tourne la mono v d'un angle $theta$

Le plan défini par w et x est laissé invariant

v , w et x sont des monos et $theta$ est l'angle de rotation en degrés

$theta$ peut être omis et dans ce cas l'angle de rotation est celui formé par les monos w et x

ToBasis[v]

Transforme une mono de la notation Mathematica standard (liste) en une combinaison linéaire de unos $e[i]$

v une mono donnée en notation Mathematica standard (liste)

ToVector[v, d]

Transforme une mono d'une combinaison linéaire donnée sous la forme canonique $e[i]$ dans la notation standard Mathematica (une liste d -dimensionnelle)

v est une mono et d est un entier positif

d peut être omis et dans un tel cas la dimension de la liste est la plus grande dimension des unos $e[i]$

Turn[m]

Donne la renversée d'une multi m

Paper

Introduction

Clifford algebra of R^n

Clifford algebra with Mathematica

Vectors

Drawing multi in 3D

Complex numbers

Quaternions

Grassman algebra

The hyperbolic plane

Dirac and Pauli algebra

Conclusion

Code comments

La mono

$\{1, 2, 0, -1\}$

doit être écrites comme

$e[1] + 2 e[2] - e[4]$

La multi

$$a + 5e_1 + e_{123}$$

doit être écrite comme

$$a + 5 e[1] + e[1] e[2] e[3]$$

A l'exception de la fonction

$$Dual[m,n]$$

il n'est pas nécessaire de définir la dimension de l'idéologique car elle est calculée

automatiquement par la fonction

$$dimensions[]$$

La signature de la forme bilinéaire est fixée par

$$\$SetSignature$$

Cette fonction fixe les indices (p, q, r) de la forme bilinéaire, r étant l'indice de dégénérescence

Si non précisé la valeur par défaut est

$$\{20, 20, 20\}$$

A noter que

$$Projection$$

est une fonction de Mathematica

$$Projection \rightarrow CProjection$$

La variable

$$\$FirstIndex$$

a été ajoutée pour inclure l'index inférieur $i = 0$ tel que communément utilisé dans l'algèbre de

Dirac

$$\$SetSignature$$

Fixe les indices (p, q, s) de la forme bilinéaire utilisée pour définir l'idéologique, s étant un indice de dégénérescence

Par souci de simplicité seulement

$$\{p, q\}$$

est spécifiée et on suppose que pour

$$i > p + q$$

on a

$$\text{GeometricProduct}[e[i], e[i]] = 0$$

La valeur par défaut est

$$\{20, 0\}$$

Elle peut être rétablie par

$$\text{Clear}[\$SetSignature]$$

\$FirstIndex

Fixe l'indice inférieur des monos $e[i]$ à un entier non-négatif arbitraire

Son usage principal est d'inclure la valeur $i = 0$ communément utilisée en relativité

La valeur par défaut est

$$1$$

Une fois changée elle peut être rétablie par

$$\text{Clear}[\$FistIndex]$$

CliffordBasic.m

Code comments

CliffordBasic.m fonctionne dans une idéologie n -dimensionnelle I^n de signature

$$\{p, q\}$$

générée par les unos

$$e[1], \dots, e[n]$$

et les relations

$$e[i]^2 = 1 \quad \text{pour } 1 < i \leq p$$

$$e[i]^2 = -1 \quad \text{pour } p < i \leq n$$

$$e[i, j] = -e[j, i]$$

L'imposition des unos est notée

$$e[2, 3, 1]$$

pour

$$e[2] * e[3] * e[1]$$

Le résultat de tout calcul est donné en termes d'imposition d'unos c'est-à-dire que l'éjection des unos ou des mults est calculé en utilisant l'imposition et le résultat est donné en termes d'imposition d'unos

A l'exception de la fonction

$$Dual[m, n]$$

il n'est pas nécessaire de définir la dimension de l'idéologique car elle est calculée automatiquement par la fonction

$$dimensions[]$$

La signature de la forme bi-linéaire de l'idéologique I^{p+q} est fixée par la fonction

$$\$SetSignature$$

Par défaut la valeur de la signature est fixée à

$$\{20, 0\}$$

A noter qu'une nouvelle fonction

$$\$FirstIndex$$

a été introduite pour inclure un index

$$i = 0$$

tel qu'il est communément utilisé dans l'algèbre de Dirac

Injection et enjection

L'enjection (inner product) correspond à la définition de Hestenes

Si A est une j -exo et B est une k -exo alors

$$A \bullet B = \langle A, B \rangle_{/k - j/}$$

Si A et B sont des nombres alors

$$A \bullet B = 0$$

Les deux enjections par la gauche et par la droite sont implantées par

LeftContraction[]

et

RightContraction[]

Meet et Join

Si A et B sont des exos alors

$A \text{ inter } B$

=

$$(-1)^{n * (n-1)/2} * (A_{\text{Complément}} \wedge B_{\text{Complément}})$$

Les fonctions *Meet* et *Join* sont définies à un facteur scalaire près

Pour fixer une échelle (scale) *Meet* est normalisé

Comme *Join* est un mot réservé le *join* a été implanté en termes de *Meet* selon Dorst (5.5)

$A \text{ inter } B$

=

$$A \wedge ((A \text{ union } B) \wedge \{-1\}) \bullet B$$

Comme *Join* est défini en termes de *Meet* et que *Meet* a été normalisé il s'ensuit qu'une échelle (scale) pour *Join* est définie

Code functions

Coeff[m, b]

donne les coefficients de la r -exo b dans la multi m

Grade[m, r]

donne la r -exo partie de la multi m

HomogeneousQ[x, r]

donne *True* si x est une r -exo et sinon *False*

GFactor[m]

groupe les termes avec des monos $e[]$ communes

Pseudosclar[n]

donne l'omni-unos

Turn[m]

donne la renversée de la multi m

Involution[m]

donne l'involuée de la multi m à savoir qu'elle inverse l'internalité des exos impaires

Si m est homogène de complexité r alors

$$\text{Involution}[m]$$

=

$$(-1)^r * m$$

MultivectorInverse[m]

donne l'inverse d'une multi

Dual[m,n]

donne le dual de la multi m dans un univers n -dimensionnel

Meet[m1, m2, n]

donne le *Meet* normalisé des multis $m1$ et $m2$ dans un univers n -dimensionnel

Joint[m1, m2, n]

donne le *Join* des multis $m1$ et $m2$ dans un univers n -dimensionnel

Reflection[v,w]

réfléchit la mono v par l'hyperplan horthogonal à la mono w

Rotation[x, a, b, theta]

tourne la mono x par un angle $theta$ en radians selon le plan défini par les monos a et b

Le sens de rotation est de a vers b

Si aucun $theta$ n'est donné la valeur par défaut est l'angle entre les monos a et b

ToBasis[x]

transforme la mono x représentée par une liste standard Mathematica

$$\{a, b, \dots\}$$

dans la forme standard idéologique à savoir

$$a e[1] + b e[2] + \dots$$
ToVector[x,n]

transforme la mono n -dimensionnelle x de forme

$$a e[1] + b e[2] + \dots$$

dans la forme standard Mathematica

$$\{a, b, \dots\}$$
\$SetSignature[p, q]

fixe les indices p et q de la forme bi-linéaire utilisée

Sa valeur par défaut est

$$\{20, 0\}$$

Clear[\$SetSignature]

permet de la remettre à la valeur par défaut

\$FirstIndex[i]

permet de fixer l'indice le plus bas des unos $e[i]$ de manière arbitraire non-négative

Son rôle principal est d'inclure la valeur

$$i = 0$$

communément utilisée en relativité

La valeur par défaut est 1

Une fois changée elle peut être refixée par

Clear[\$FirstIndex]

CGAlgebra.m

Paper

Introduction

En centrologique 5D les idées comme les lignes, les plans, les cercles et les sphères de 3D sont représentées comme de simples identités algébriques

C'est une extension de la pointologique 4D projective

Première définition

La centrologie 5-D est générée (spanned) par

$$\{e_0, e_1, e_2, e_3, e_{inf}\}$$

avec l'injection

$$\langle x, y \rangle$$

=

$$x_1 y_1 + x_2 y_2 + x_3 y_3 - x_0 y_{inf} - x_{inf} y_0 \quad (1)$$

des deux monos suivantes

$$x = x_0 e_0 + x_1 e_1 + x_2 e_2 + x_3 e_3 + x_{inf} e_{inf}$$

et

$$y = y_0 e_0 + y_1 e_1 + y_2 e_2 + y_3 e_3 + y_{inf} e_{inf}$$

de cet univers

De l'injection (I) ci-dessus on tire

$$/e_0/{}^2$$

$$=$$

$$\langle e_0, e_0 \rangle$$

$$=$$

$$0$$

et similairement

$$/e_{inf}/{}^2$$

$$=$$

$$\langle e_{inf}, e_{inf} \rangle$$

$$=$$

$$0$$

de telle sorte que e_0 et e_{inf} sont appelés des unos nulles

L'injection que nous rappelons ci-dessous

$$x_1 y_1 + x_2 y_2 + x_3 y_3 - x_0 y_{inf} - x_{inf} y_0 \quad (I)$$

a la signature (4,1) et l'idéologique est une idéologique de Minkovsky $I^{4,1}$

Seconde définition

Pour le package défini ici en Mathematica une définition de l'algèbre conforme basée sur des unos et des relations est plus pratique

La centrologique $I^{4,1}$ équipée de l'injection que nous rappelons ci-dessous

$$x_1 y_1 + x_2 y_2 + x_3 y_3 - x_0 y_{inf} - x_{inf} y_0 \quad (I)$$

est une idéologique générée par l'identité

I

les symboles

$$\{e_0, e_1, e_2, e_3, e_{inf}\}$$

et par une opération d'imposition associative caractérisée par les relations suivantes

$$e_i e_j + e_j e_i = 0$$

$$e_i e_0 + e_0 e_i = 0$$

$$e_i e_{inf} + e_{inf} e_i = 0$$

$$e_i^2 = 1$$

$$e_0 e_{inf} + e_{inf} e_0 = -2$$

$$e_0^2 = e_{inf}^2 = 0$$

pour

$$i, j = 1, 2, 3$$

Une autre manière de construire cette idéologique $I^{4,1}$ est de partir d'une algèbre de Clifford $C^{4,1}$ généré par

$$\{e_1, e_2, e_3, e_4, e_5\}$$

avec une injection

$$\langle x, y \rangle$$

=

$$x_1 y_1 + x_2 y_2 + x_3 y_3 + x_4 y_4 - x_5 y_5$$

des deux monos ci-dessous

$$x = x_1 e_1 + x_2 e_2 + x_3 e_3 + x_4 e_4 + x_5 e_5$$

et

$$y = y_1 e_1 + y_2 e_2 + y_3 e_3 + y_4 e_4 + y_5 e_5$$

de $C^{4,1}$

Dans cet idéologique on définit les unos nulles comme

$$e_0$$

$$=$$

$$(e_4 + e_5) / 2$$

et

$$e_{inf}$$

$$=$$

$$e_5 - e_4$$

Avec cette définition l'idéologique $I^{4,1}$ est celle générée par

$$\{e_0, e_1, e_2, e_3, e_{inf}\}$$

Les inverses de ces transformations sont

$$e_4$$

$$=$$

$$e_0 - e_{inf}/2$$

et

$$e_{inf}$$

$$=$$

$$e_5 - e_4$$

Les deux approches sont utilisées dans le code

Utilisation

Les unos sont notées

$$e[i]$$

Avec cette notation l'idéologique $I^{4,1}$ est générée par

$$\{e_0, e_1, e_2, e_3, e_{inf}\}$$

L'imposition entre les unos

$$e_i e_j \dots e_k$$

est notée comme

$$e[i, j, \dots, k]$$

Représentation des idées

Points

Directions

Plans

Sphère

Représentation des transformations

Un des avantages de la centrologie I^5 est que les transformations orthogonales de la flèchologie U^3 peuvent être représentées par des transformations horthogonales de $I^{4,1}$

Translations

Rotations

Fonctions additionnelles prédéfinies

Les 4 fonctions

GeometricProduct[m_1, m_2, \dots]

OuterProduct[m_1, m_2, \dots]

InnerProduct[m_1, m_2] (*left-contraction of 2 multi*)

Grade[m, k] (*k-exo de la multi*)

sont la base permettant de définir d'autres fonctions selon les applications comme

Magnitude[m] (*donne m^2*)

Reversion[m]

Involution[m]
MultivectorInverse[m]
GradeQ[m, k]
Dual[m]
Rotation[x, a, b, theta]
ToVector[v]
GFactor[m]
I5
I5inv

On peut définir par exemple la fonction

Inversor[x, p, r]

qui inverse un point x par rapport à une sphère centrée en p et de rayon r

Ainsi l'inversion de l'origine donne l'infini et les monos de U^3 ne sont pas affectées par
l'inversion

Code comments

Les unos

$\{e_0, e_1, e_2, e_3, e_{inf}\}$

sont notées par

$e[0], e[1], e[2], e[3], e[inf]$

Les impositions de unos sont notées comme

$e[0, 1, 2]$

c'est-à-dire

$e_1 e_2 e_3$

Les résultats des calculs sont donnés en termes d'imposition d'unos c'est-à-dire en termes
d'éjection d'unos ou de multis est calculé en utilisant

OuterProduct[]

et en donnant un résultat en terme d'imposition de unos

Code functions

e

e est utilisé pour noter les éléments dans la base canonique de l'idéologique où l'algèbre est définie soit

$i = 0, 1, 2, 4, [Infinity]$

Grade[m, r]

Retourne la r -exo partie de la multi m

GradeQ[m, r]

Retourne *True* si m est une r -exo

MultiplicationTable[e]

Retourne la table d'imposition du symbole e

GFactor[m]

Regroupe les termes ayant une $[e_]$ commune

GeometricProduct[m₁, m₂]

Retourne l'imposition de m_1 et m_2

OuterProduct[m₁, m₂]

Retourne l'éjection (Grassman product) de m_1 et m_2

InnerProduct[m₁, m₂]

Retourne l'injection (left contraction) de m_1 et m_2

Reversion[m]

Retourne la réversion de m

Involution[m]

Retourne l'involution de complexité de m

Magnitude[m]

Retourne la magnitude (m^2) de m

MultivectorInverse[m]

Retourne l'inverse, si elle existe, de m

I5

$I5$ est l'onmi-unos de $I^{4,1}$

I5i

$I5i$ est l'inverse de l'omni-unos $I5$

Dual[m]

Retroune le dual de m

ToVector[v]

Transforme une flèche de l'unologique en notation *Mathematica* standard

Rotation[x, a, b, theta]

Tourne la flèche x d'un angle $theta$ en radians selon le plan défini par les flèches a et b

La rotation va de a vers b

Si aucun $theta$ n'est donné la valeur par défaut est l'angle entre a et b

Mathstar