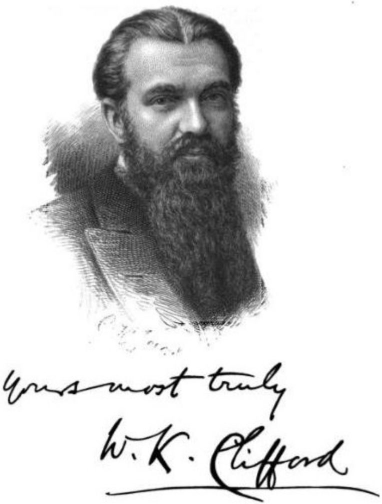
Clifford (Geometric) Algebra for Practitioners

Dimiter Prodanov
Laboratory of Neurotechnology, IICT–BAS, Sofia, Bulgaria
Geometry, Integrability and Quantization, Varna, 2026

Overview

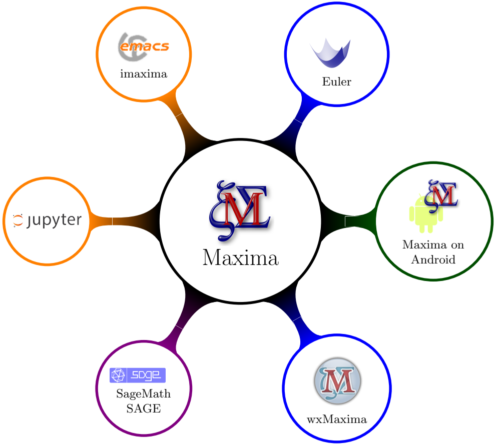


Funded by the European Union

Introduction to Maxima

Funded by the European Union

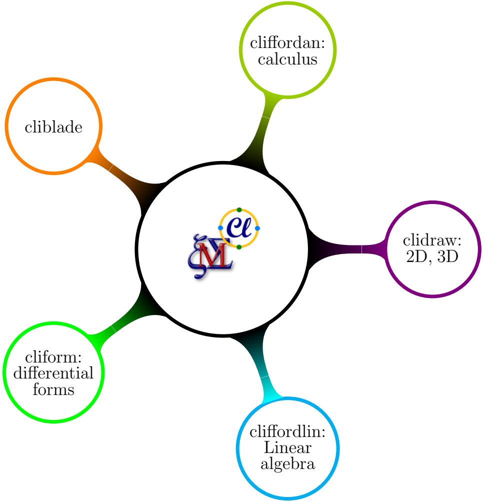
The Maxima ecosystem



- Maxima is widely used
- Maxima is mature: many tutorials, courses and textbooks use it
- Maxima is open source (GPL)
- open source allows for shorted development cycles

Funded by
the European Union

The `Clifford` package



- minimalist design
- mature: 2015 – 2026
- used in published math research
- open source: GPL license

Available from GitHub
`http://dprodanov.github.io/clifford/`

Funded by
the European Union

Clifford algebras

Funded by
the European Union

Geometric Algebra Fundamentals

The Euclidean Geometric Algebra $C\ell_{p,q}$

Generated by orthonormal basis vectors $\{e_1, e_2, \ldots, e_n\}$ with the associative geometric product:

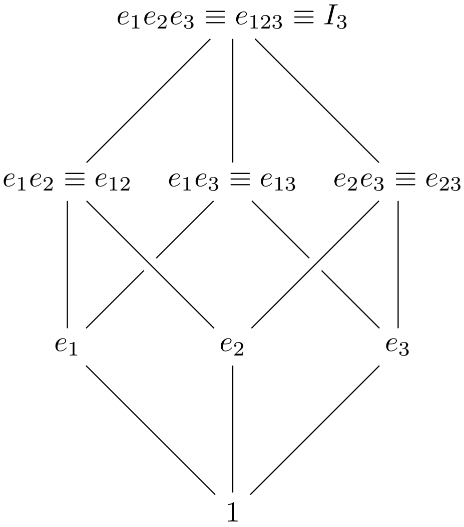
$$e_i e_i = \delta_{ij},\ i \leq p; \quad e_i e_i = -\delta_{ij},\ i > p$$

$$e_i e_j = -e_j e_i \ \ (i \neq j)$$

- **Product Decomposition**: $ab = a \cdot b + a \wedge b$ (Inner symmetric contraction + Outer antisymmetric wedge).
- **Multivector Structure**: Mixes scalars (grade 0), vectors (grade 1), bivectors (grade 2), and pseudoscalars (grade 3).

Funded by the European Union

Example: Euclidean Geometric Algebra $\mathbb{G}^3$



- 1 scalar; 3 vectors; 3 bivectors; 1 pseudoscalar $I_3 = e_{123}$ (commutes with everything)
- bivectors encode rotations

Funded by the European Union

Geometric Algebra Products

- scalar product

$$A * B := \langle AB \rangle_0$$

commutative: $A * B = B * A$

- outer product

$$a \wedge A_k := \frac{1}{2}\left(aA_k + (-1)^k A_k a\right)$$

mixed: $a \wedge A_k = (-)^k A_k \wedge a$

- inner product (Hestenes convention)

$$a \cdot b = a * b$$

$$a \cdot A_k := \frac{1}{2}\left(aA_k - (-1)^k A_k a\right)$$

mixed: $a \cdot A_k = (-)^{k-1} A_k \wedge a$

$$(a_1 a_2 \ldots a_r) \cdot (b_1 b_2 \ldots b_s) := \begin{cases} ((a_1 a_2 \ldots a_r) \cdot b_1) \cdot (b_2 \ldots b_s), & r \geq s \\ (a_1 a_2 \ldots a_{r-1}) \cdot (a_r \cdot (b_1 b_2 \ldots b_s)), & r < s \end{cases}$$

Funded by the European Union

Main functions in `Clifford`

- GRADE FUNCTIONS

`grade` (expr)	grade decomposition of expression
`scalarpart` (expr)	$< expr >_0$
`vectorpart` (expr)	$< expr >_1$
`grpart` (expr,k)	$< expr >_k$
`mvectorpart` (expr)	$< expr >_{2+}$
`bdecompose` (expr)	blade decomposition of expression

Grade projection operator : $\langle \ \rangle_k$

Grade decomposition : $M = \sum_{k=0}^{n} \langle M \rangle_k$

Funded by the European Union

Product Simplification Demo

Funded by
the European Union

Main functions in `Clifford`

- PRODUCT SIMPLIFICATION

`cliffsimpall` (expr)	full simplification of expressions
`dotsimp1` (ab)	canonic reordering of dot products
`dotsimpc` (ab)	simplification of dot products
`dotinvsimp` (ab)	simplification of inverses
`powsimp` (ab)	simplification of exponents

Funded by the European Union

Quaternions

```
load('clifford);
clifford(e,0,2);
mtable1([1, e[1],e[2], e[1] . e[2]]);
```

Quaternion multiplication table

$$\begin{pmatrix} 1 & e_1 & e_2 & e_1.e_2 \\ e_1 & -1 & e_1.e_2 & -e_2 \\ e_2 & -e_1.e_2 & -1 & e_1 \\ e_1.e_2 & e_2 & -e_1 & -1 \end{pmatrix}$$

(minimal manual formatting)

Funded by
the European Union

Outer product in $\mathbb{G}^3$

```
clifford(e, 3);
mtable2o();
```

Outer product multiplication table

$$\begin{pmatrix} 1 & e_1 & e_2 & e_3 & e_1 \cdot e_2 & e_1 \cdot e_3 & e_2 \cdot e_3 & e_1 \cdot e_2 \cdot e_3 \\ e_1 & 0 & e_1 \cdot e_2 & e_1 \cdot e_3 & 0 & 0 & e_1 \cdot e_2 \cdot e_3 & 0 \\ e_2 & -e_1 \cdot e_2 & 0 & e_2 \cdot e_3 & 0 & -e_1 \cdot e_2 \cdot e_3 & 0 & 0 \\ e_3 & -e_1 \cdot e_3 & -e_2 \cdot e_3 & 0 & e_1 \cdot e_2 \cdot e_3 & 0 & 0 & 0 \\ e_1 \cdot e_2 & 0 & 0 & e_1 \cdot e_2 \cdot e_3 & 0 & 0 & 0 & 0 \\ e_1 \cdot e_3 & 0 & -e_1 \cdot e_2 \cdot e_3 & 0 & 0 & 0 & 0 & 0 \\ e_2 \cdot e_3 & e_1 \cdot e_2 \cdot e_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ e_1 \cdot e_2 \cdot e_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Funded by the European Union

Inner product(s) in $\mathbb{G}^3$

Inner product types: `lc` – left contraction, `rc` – right contraction , `sym` – for symmetric

```
inprotype:lc;
mtable2i();
```

Left contraction multiplication table

$$\begin{pmatrix}
1 & e_1 & e_2 & e_3 & e_1.e_2 & e_1.e_3 & e_2.e_3 & e_1.e_2.e_3 \\
0 & 1 & 0 & 0 & e_2 & e_3 & 0 & e_2.e_3 \\
0 & 0 & 1 & 0 & -e_1 & 0 & e_3 & -e_1.e_3 \\
0 & 0 & 0 & 1 & 0 & -e_1 & -e_2 & e_1.e_2 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & -e_3 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & e_2 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & -e_1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}$$

Funded by the European Union

Main functions in `Clifford`

- INVOLUTIONS

`dotreverse` (ab)	Clifford reverse
`cinvolve` (expr)	Clifford involution
`dotconjugate` (expr)	Clifford conjugation
`invautom` (expr)	Inverse Blade Automorphism (IBA)

Vectors
Grade reversal :

$$(abcd)^{\sim} = dcba$$

involution :

$$(abcd)^{t} = (-a)(-b)(-c)(-d)(-a)$$

IBA :

$$(abcd)^{\sharp} = d^{\sharp}c^{\sharp}b^{\sharp}a^{\sharp}, \quad a^{\sharp} = \frac{a}{a*a}$$

Clifford conjugation : $\bar{A} = (A^{t})^{\sim}$

Funded by the European Union

Inverse in STA $C\ell_{1,3}^{+}$

Spinor

$$\Psi = a_1\mathbb{I} + a_2\gamma^{12} + a_3\gamma^{23} + a_4\gamma^{31} + a_5\gamma^{01} + a_6\gamma^{02} + a_7\gamma^{03} + a_8\gamma^{5}$$

Inverse

$$\Psi^{-1} = \frac{\widetilde{\Psi}(\Psi\widetilde{\Psi})^{\sharp}}{\Delta}, \quad \text{where} \quad \Delta = \Psi\widetilde{\Psi}(\Psi\widetilde{\Psi})^{\sharp}$$

```
clifford(e,1,3);
A:celem(c)$
C:Ae.dotreverse(Ae),expand, dotsimpc;
C:C,clifact;
(C)    c[1]^2-c[6]^2-c[7]^2-c[8]^2+c[9]^2+c[10]^2+c[11]^2-c[16]^2+2*(-(c[8]*c
    [9])+c[7]*c[10]-c[6]*c[11]+c[1]*c[16])*(e[1] . e[2] . e[3] . e[4])
DD:C.invautom(C),expand, dotsimpc,ratsimp$
F: dotreverse(Ae).invautom(C),expand, dotsimpc$
A2: F/DD$
Ae.A2,expand, dotsimpc$
%,ratsimp;
(%o13) 1
```

Representation $C\ell^{+}_{1,3}$

$$PP = \begin{bmatrix} \psi_1 - i\psi_9 & \psi_{10} - i\psi_{11} & \psi_8 - i\psi_{16} & \psi_6 - i\psi_7 \\ -\psi_{10} - i\psi_{11} & \psi_1 + i\psi_9 & \psi_6 + i\psi_7 & -\psi_8 - i\psi_{16} \\ \psi_8 - i\psi_{16} & \psi_6 - i\psi_7 & \psi_1 - i\psi_9 & \psi_{10} - i\psi_{11} \\ \psi_6 + i\psi_7 & -\psi_8 - i\psi_{16} & -\psi_{10} - i\psi_{11} & \psi_1 + i\psi_9 \end{bmatrix}$$

Extracting a the bivector component γ_{23}

```
 P11:diff(PP, psi[11]);
(P11)     matrix(
[0, -%i,     0,   0],
[-%i,    0,  0,   0],
[0,  0,   0,  -%i],
[0,  0,  -%i,     0]
)
(%i82)   K1:P11.PP,expand$
(%i83)   mat_trace(K1)/mat_trace(P11.P11);
(%o83)   psi[11]
```

Funded by the European Union

Numerical Demo

Faddeev-Le Verrer-Souriau algorithm

Theorem (Reduced-grade FVS algorithm)

Suppose that $A \in C\ell_{p,q}$ *is a multivector of maximal grade* $r \leq s$ *and* $A \subseteq \mathrm{span}[e_1, \ldots, e_s]$. *The Clifford inverse, if it exists, can be computed in* $k = 2^{\lceil s/2 \rceil}$ *steps as*

$$
\begin{array}{ll|ll}
m_1 &= A & c_1 = -kA * 1, & t_1 := -c_1 \\
m_2 &= Am_1 - t_1 & c_2 = -\frac{k}{2} A * m_1, & t_2 := -c_2 \\
\ldots & & \ldots & \\
m_k &= Am_{k-1} - t_k & c_k = -A * m_{k-1}, & t_k := -c_k
\end{array}
$$

until the step where $m_k = 0$ *so that*

$$A^{-1} = -m_{k-1}/c_k. \tag{1}$$

The inverse does not exist if $c_k = -\det A = 0$.
The (reduced) characteristic polynomial of A *is*

$$p_A(\lambda) = \lambda^k + c_1 \lambda^{k-1} + \ldots c_{k-1}\lambda + c_k. \tag{2}$$

Funded by the European Union

Inverse in $C\ell_{2,5}$

$A = 1 - 2e_{15} + 5e_{134}$

- $\mathrm{span}[A] = \{e_1, e_3, e_4, e_5\}$
- Reduced grade algorithm will run in $k = 2^{\lceil 4/2 \rceil} = 4$ steps:

$$
\begin{aligned}
&t_1 = -4, && m_1 = 1 + 5C - 2B, \quad B := e_{15}, \quad C := e_{134};\\
&t_2 = 48, && m_2 = -24 - 10C + 4B;\\
&t_3 = -88, && m_3 = 66 + 110C - 44B;\\
&t_4 = 484, && m_4 = 0
\end{aligned}
$$

- The characteristic polynomial factorizes

$$p_A(v) = 484 - 88v + 48v^2 - 4v^3 + v^4 = (22 - 2v + v^2)^2$$

$$A^{-1} = (1 - 5C + 2B)/22 \tag{3}$$

- The optimal formula is

$$detA = AA^{\sim} = 22 \Longrightarrow A^{-1} = A^{\sim}/22$$

Funded by the European Union

Inverse in $C\ell_{2,5}$

$A = 1 - 2e_{15} + 5e_{134}$

- $\mathrm{span}[A] = \{e_1, e_3, e_4, e_5\}$
- Reduced grade algorithm will run in $k = 2^{\lceil 4/2 \rceil} = 4$ steps:

$$
\begin{aligned}
t_1 &= -4, & m_1 &= 1 + 5C - 2B, \quad B := e_{15}, \quad C := e_{134}; \\
t_2 &= 48, & m_2 &= -24 - 10C + 4B; \\
t_3 &= -88, & m_3 &= 66 + 110C - 44B; \\
t_4 &= 484, & m_4 &= 0
\end{aligned}
$$

- The characteristic polynomial factorizes

$$p_A(v) = 484 - 88v + 48v^2 - 4v^3 + v^4 = (22 - 2v + v^2)^2$$

$$A^{-1} = (1 - 5C + 2B)/22 \tag{3}$$

- The optimal formula is

$$detA = AA^{\sim} = 22 \Longrightarrow A^{-1} = A^{\sim}/22$$

Funded by the European Union

Inverse in $C\ell_{2,5}$

$A = 1 - 2e_{15} + 5e_{134}$

- $\mathrm{span}[A] = \{e_1, e_3, e_4, e_5\}$
- Reduced grade algorithm will run in $k = 2^{\lceil 4/2 \rceil} = 4$ steps:

$$\begin{aligned}
t_1 &= -4, & m_1 &= 1 + 5C - 2B, \quad B := e_{15}, \quad C := e_{134}; \\
t_2 &= 48, & m_2 &= -24 - 10C + 4B; \\
t_3 &= -88, & m_3 &= 66 + 110C - 44B; \\
t_4 &= 484, & m_4 &= 0
\end{aligned}$$

- The characteristic polynomial factorizes

$$p_A(v) = 484 - 88v + 48v^2 - 4v^3 + v^4 = (22 - 2v + v^2)^2$$

$$A^{-1} = (1 - 5C + 2B)/22 \tag{3}$$

- The optimal formula is

$$det A = AA^{\sim} = 22 \Longrightarrow A^{-1} = A^{\sim}/22$$

Funded by the European Union

Inverse in $C\ell_{2,5}$

$A = 1 - 2e_{15} + 5e_{134}$

- $\mathrm{span}[A] = \{e_1, e_3, e_4, e_5\}$
- Reduced grade algorithm will run in $k = 2^{\lceil 4/2 \rceil} = 4$ steps:

$$\begin{aligned}
t_1 &= -4, & m_1 &= 1 + 5C - 2B, \quad B := e_{15}, \quad C := e_{134}; \\
t_2 &= 48, & m_2 &= -24 - 10C + 4B; \\
t_3 &= -88, & m_3 &= 66 + 110C - 44B; \\
t_4 &= 484, & m_4 &= 0
\end{aligned}$$

- The characteristic polynomial factorizes

$$p_A(v) = 484 - 88v + 48v^2 - 4v^3 + v^4 = (22 - 2v + v^2)^2$$

$$A^{-1} = (1 - 5C + 2B)/22 \tag{3}$$

- The optimal formula is

$$detA = AA^{\sim} = 22 \Longrightarrow A^{-1} = A^{\sim}/22$$

Funded by the European Union

Inverse in $C\ell_{5,2}$

$A = 1 - e_2 + I$
Reduced grade algorithm will run in $2^{\lceil 7/2 \rceil} = 2^4 = 16$ steps

$$
\begin{aligned}
t_1 &= -16, & m_1 &= 1 - e_2 + I; \\
t_2 &= 120, & m_2 &= -15 + 14e_2 - 14I + 2C, \quad C = e_{134567}; \\
t_3 &= -560, & m_3 &= 105 - 89e_2 + 93I - 26C; \\
t_4 &= 1836, & m_4 &= -459 + 340e_2 - 388I + 156C; \\
t_5 &= -4560, & m_5 &= 1425 - 881e_2 + 1145I - 572C; \\
t_6 &= 9064, & m_6 &= -3399 + 1682e_2 - 2562I + 1454C; \\
t_7 &= -14960, & m_7 &= 6545 - 2529e_2 + 4557I - 2790C; \\
t_8 &= 20886, & m_8 &= -10443 + 3096e_2 - 6648I + 4296C; \\
t_9 &= -24880, & m_9 &= 13995 - 3051e_2 + 8091I - 5448C; \\
t_{10} &= 25480, & m_{10} &= -15925 + 2386e_2 - 8242I + 5694C; \\
t_{11} &= -22416, & m_{11} &= 15411 - 1475e_2 + 7007I - 4934C; \\
t_{12} &= 16716, & m_{12} &= -12537 + 596e_2 - 4932I + 3548C; \\
t_{13} &= -10480, & m_{13} &= 8515 - 35e_2 + 2795I - 1980C; \\
t_{14} &= 5400, & m_{14} &= -4725 - 50e_2 - 1150I + 850C; \\
t_{15} &= -2000, & m_{15} &= 1875 + 125e_2 + 375I - 250C; \\
t_{16} &= 625, & m_{16} &= 0
\end{aligned}
$$

Funded by the European Union

Inverse in $C\ell_{5,2}$

$A = 1 - e_2 + I$

- Inverse

$$A^{-1} = (1 - e_2 - 3\,I + 2\,C)/5, \quad C = e_{134567} \tag{4}$$

- The characteristic polynomial factorizes:

$$p_A(v) = (5 - 4v + 6v^2 - 4v^3 + v^4)^4$$

- The optimal formula is

$$B = AA^t = 1 - 2I$$

$$\det A = BB^{\sim} = 5$$

$$A^{-1} = A^t (AA^t)^{\sim}/5$$

Funded by the European Union

Inverse in $C\ell_{5,2}$

$A = 1 - e_2 + I$

- Inverse
$$A^{-1} = (1 - e_2 - 3\,I + 2\,C)/5, \quad C = e_{134567} \tag{4}$$
- The characteristic polynomial factorizes:
$$p_A(v) = (5 - 4v + 6v^2 - 4v^3 + v^4)^4$$
- The optimal formula is
$$B = AA^t = 1 - 2I$$
$$\det A = BB^{\sim} = 5$$
$$A^{-1} = A^t (AA^t)^{\sim}/5$$

Funded by the European Union

Calculus Demo

Funded by the European Union

Main functions in `Clifford`an

`ctotdiff`(f, x)	total derivative w.r.t. multivector x
`ctotintdiff`(f, x)	inner total derivative w.r.t. x
`ctotextdiff`(f, x)	outer total derivative w.r.t. x
`vectdiff`(f, ee, k)	vector derivative of order k w.r.t. basis vector list *ee*
`mvectdiff`(f, x, k)	multivector derivative of order k w.r.t. multivector x
`parmvectdiff`(f, x, k)	partial multivector derivative of order k w.r.t. multivector x
`convderiv`(f, t, xx, [vs])	convective derivative w.r.t. multivector x
`coordsubst`(x, eqs)	substitutes coordinates in multivector x w.r.t. new variables in the list *eqs*
`clivolel`(x, eqs)	computes volume element of $Span\,\{x\}$ w.r.t. new variables in the list *eqs*

Generalized derivatives in Clifford algebras

Definition

Vector $r \in Span\{e_1, \dots, e_n\}$ and reciprocal frames $e^k = e_k^{-1}$ so that.

$$e_k * e^j = \delta_k^j$$

Directional derivative

projection along the unit norm (multi)vector v:

$$(v * \nabla)F(x) \equiv {}_v\nabla F(x) := \sum_j v_j \frac{\partial F}{\partial x_j}$$

Vector derivative

$$\nabla_r F(x) := \frac{dF}{dr} = \sum_J \underbrace{\frac{\partial x_j}{\partial r}}_{e^J} \frac{\partial F}{\partial x_j} = e^J \partial_J F(x)$$

Funded by the European Union

Potential problems in $\mathbb{G}^3$

Potential equation for the Green's function

$$r = e_a x^a = e^a x_a$$

$$\nabla_r G = \delta(r)$$

$$G(x, y, z) = \frac{e_1\, x + e_2\, y + e_3\, z}{\sqrt{(x^2 + y^2 + z^2)^3}}$$

Cartesian coordinates

```
/* Initialization */
load('clifford);
load ('cliffordan);

/* G(3) construction */
clifford(e,3);
r:cvect([x,y,z]);
G:r/sqrt(-cnorm(r))^3;
mvectdiff(G,r);


mvectdiff(-1/sqrt(-cnorm(r)),r);
mvectdiff(-1/sqrt(-cnorm(r)),r,2);
```

Funded by the European Union

Potential problems in $\mathbb{G}^3$

Cylindrical coordinates

```
cyl_eq:[x=rho*cos(phi), y=rho*sin(phi)];
  declare( [rho, phi], scalar);
  GG_c:coordsubst(G, cyl_eq),factor;
  rc:coordsubst(r, cyl_eq);
  mvectdiff(GG_c,rc);
  V:coordsubst(-1/sqrt(-cnorm(r)),cyl_eq);
  mvectdiff(V,rc);
```

$$GG_c = \frac{e_1 \rho \cos\phi + e_2 \rho \sin\phi + e_3 z}{\left(\rho^2 + z^2\right)^{\frac{3}{2}}}$$

$$V = \frac{e_1 \rho \cos\phi + e_2 \rho \sin\phi + e_3 z}{\left(\rho^2 + z^2\right)^{\frac{3}{2}}}$$

Funded by the European Union

Euler-Lagrange problems in $\mathbb{G}^3$

```
AA: celem(A,[t,x,y,z]);
dependsv(A,[t,x,y,z]);
r:cvect([x,y,z]);
F:mvectdiff(AA,t-r);
F:factorby(F, %elements);
L:lambda([x],1/2*scalarpart(cliffsimpall(x.x)))(F);
S:scalarpart(F);
V:vectorpart(F);
Q:grpart(F,2);
L-1/2*(S.S+V.V+Q.Q),cliffsimpall;
dA:mvectdiff(AA,r);
EuLagEq2(L, t+r,[AA,dA]);
```

D'Alembert's equation

$$\nabla_r \nabla_r \bullet A = 0$$

$$\left(-A_{t\,tt} + A_{t\,xx} + A_{t\,yy} + A_{t\,zz}\right) + e_1 \left(-A_{x\,tt} + A_{x\,xx} + A_{x\,yy} + A_{x\,zz}\right)$$
$$+ e_2 \left(-A_{y\,tt} + A_{y\,xx} + A_{y\,yy} + A_{y\,zz}\right) + e_3 \left(-A_{z\,tt} + A_{z\,xx} + A_{z\,yy} + A_{z\,zz}\right) = 0$$

Funded by
the European Union

References

A. Macdonald.
An elementary construction of the geometric algebra.
Advances in Applied Clifford Algebras, 12(1):1 – 6, 2002.

A. Macdonald.
A survey of geometric algebra and geometric calculus.
Advances in Applied Clifford Algebras, pages 1–39, 2016.

D. Prodanov and V. T. Toth.
Sparse representations of clifford and tensor algebras in Maxima.
Advances in Applied Clifford Algebras, 27(1), 661–683, 2017.
http://arxiv.org/pdf/1604.06967.pdf

D. Prodanov
Clifford Algebra Implementations in Maxima
Journal of Geometry and Symmetry in Physics, 43, 73–105, 2016.

Funded by the European Union

THANK YOU FOR THE ATTENTION!

The present work was funded by the European Union's Horizon Europe program under grant agreement VIBraTE 101086815.