This notebook:

- displays the help for the GA30 module, and its functions
- has a number of test cases.
- some manual tests that require visual verification.

These tests can also be used to see examples of how to use the GA functions and operators defined by the GA30 package.

In[48]:=
```
(*<< altcomplex`;*)
<< GA30`;

? GA30
(*?grade*)
? Scalar
? Vector
? Bivector
? Trivector
? gradeQ
? scalarQ
? vectorQ
? bivectorQ
? trivectorQ
? bladeQ
? gradeAnyQ
? notGradeQ
? GradeSelection
? ScalarSelection
? VectorSelection
? BivectorSelection
? TrivectorSelection
? ScalarValue
? ScalarProduct
? grad
? div
? curl
On[Assert]

ClearAll[e0, e1, e2, e3, e23, e31, e12, e32, e13,
  e21, e123, m01, m02, m03, m12, m23, m012, m013, m023, m123];
e0 = Scalar[1];
e1 = Vector[1, 1] ;
e2 = Vector[1, 2];
e3 = Vector[1, 3];
```

```
e23 = Bivector[1, 2, 3] ;
e31 = Bivector[1, 3, 1];
e12 = Bivector[1, 1, 2];
e32 = -e23 ;
e13 = -e31 ;
e21 = -e12 ;
e123 = Trivector[1];
m01 = e0 + e1;
m02 = e0 + e23;
m03 = e0 + e123;
m12 = e1 + e23;
m13 = e1 + e123;
m23 = e23 + e123;
m012 = e0 + e1 + e23;
m013 = e0 + e1 + e123;
m023 = e0 + e23 + e123;
m123 = e1 + e23 + e123;
```

GA30: An implementation of Euclidean (CL(3,0)) Geometric Algebra.

Pauli matrices are used to represent the algebraic elements. This provides an efficient and compact representation of the entire algebraic space.

Internally, a multivector is represented by a pair (grade, pauli–representation). The grade portion will be obliterated when adding objects that have different grade, or multiplying vectors or bivectors. When it is available, certain operations can be optimized. Comparison ignores the cached grade if it exists.

Elements of the algebra can be constructed with one of

```
  Scalar[ v ]
  Vector[ v, n ]
  Bivector[ v, n, m ]
  Trivector[ v ]
```

Example:

```
  m = Scalar[ Sin[ x ] ] + Vector[ Log[ z ], 3 ] + Trivector[ 7 ] ;
  m // StandardForm
```

> 7 e[123] + e[3] Log[z] + Sin[x]

A few operators are provided:

```
  ==        Compare two multivectors, ignoring the cached grade if any.
  m1 + m2
  m1 – m2
  – m
  st * vb    Scalars and trivectors can multiply vectors and bivectors in any order
  vb1 ** vb1 Vectors and bivectors when multiplied have
```

to use the NonCommutativeMultiply operator, but any grade object may also.
m1 . m2 Dot product. The functional form Dot[m1, m2] may also be used.
m1 ^ m2 Wedgeproduct. Enter with m1 [Esc]^[Esc] m2. The functional form Wedge[m1, m2]
<m> Scalar selection. Enter with [Esc]<[Esc] m [Esc]>[Esc
]. The functional form ScalarValue[m] may also be used. This returns the numeric
(or expression) value of the scalar grade of the multivector, and not a grade[ ] object.
<m1,m2> Scalar product. Enter with [Esc]<[Esc] m1,m2 [Esc]>[Esc]. The functional
form ScalarProduct[m1, m2] may also be used. This returns the numeric (or
expression) value of the scalar product of the multivectors, and not a grade[ ] object.

Functions provided:

– GradeSelection
– ScalarSelection
– VectorSelection
– BivectorSelection
– TrivectorSelection
– ScalarValue, < m >
– ScalarProduct, < m1, m2 >

The following built–in methods are overridden:

– TraditionalForm
– DisplayForm
– StandardForm

Internal functions:

– scalarQ
– vectorQ
– bivectorQ
– trivectorQ
– bladeQ
– gradeAnyQ
– notGradeQ

TODO:

1) How to get better formatted output by default without using one of TraditionalForm, DisplayForm, StandardForm ?

2) Can a package have options (i.e. to define the name
of the e[ ] operator used in StandardForm that represents a basis vector).

3) proper packaging stuff: private for internals.

Scalar[v] constructs a scalar grade quantity with value v.

Vector[v, n], where n = {1,2,3} constructs a vector grade quantity with value v in direction n.

Bivector[v, n1, n2], where n1,n2 = {1,2,3} constructs a bivector grade quantity with value v in the plane n1,n2.

Trivector[v] constructs a trivector (pseudoscalar) grade quantity scaled by v.

gradeQ[m, n] tests if the multivector m is of grade
n. n = –1 is used internally to represent values of more than one grade.

scalarQ[m] tests if the multivector m is of grade 0 (scalar)

vectorQ[m] tests if the multivector m is of grade 1 (vector)

bivectorQ[m] tests if the multivector m is of grade 2 (bivector)

trivectorQ[m] tests if the multivector m is of grade 3 (trivector)

bladeQ[m] tests if the multivector is of a single grade.

gradeAnyQ[ ]. predicate pattern match for grade[_]

notGradeQ[ ]. predicate pattern match for !grade[ ]

GradeSelection[m, k] selects the grade k elements from the multivector m. The selected
result is represented internally as a grade[ ] type (so scalar selection is not just a number).

ScalarSelection[m] selects the grade 0 (scalar) elements from the multivector m. The
selected result is represented internally as a grade[ ] type (not just a number or an expression).

VectorSelection[m] selects the grade 1 (vector) elements from
the multivector m. The selected result is represented internally as a grade[ ] type.

BivectorSelection[m] selects the grade 2 (bivector) elements from
the multivector m. The selected result is represented internally as a grade[ ] type.

TrivectorSelection[m] selects the grade 3 (trivector) element from the multivector m if it exists.
The selected result is represented internally as a grade[ ] type (not just an number or expression).

ScalarValue[m]. Same as AngleBracket[m], aka [Esc]<[Esc] m1 [Esc]>[Esc].

ScalarProduct[ ]. Same as AngleBracket[m1, m2], aka [Esc]<[Esc] m1, m2 [Esc]>[Esc].

grad[m,{x,y,z}] computes the vector product of the
gradient with multivector m with respect to cartesian coordinates x,y,z..

div[m,{x,y,z}] of a grade k+1 blade m, computes < ∇ m
>_k, where the gradient is evaluated with respect to cartesian coordinates x,y,z.

Given a grade (k−1) blade m, curl[m, {x, y, z}] = < ∇ m
>_k, where the gradient is evaluated with respect to cartesian coordinates x,y,z.

Predicate tests (automatic)

```
{Assert[bladeQ[#]]} & /@ {e0, e1, e2, e3, e23, e31, e12, e123};
{Assert[! bladeQ[#]]} & /@ {m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[gradeAnyQ[#]]} & /@ {e1, e2, e3, e23, e31,
    e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[! gradeAnyQ[#]]} & /@ {1, Sin[x], Exp[I theta]};
{Assert[! notGradeQ[#]]} & /@ {e0, e1, e2, e3, e23, e31,
    e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[notGradeQ[#]]} & /@ {1, Sin[x], Exp[I theta]};
{Assert[gradeQ[#, 0]], Assert[scalarQ[#]]} & /@ {e0};
{Assert[! gradeQ[#, 0]], Assert[! scalarQ[#]]} & /@ {e1, e2, e3, e23,
    e31, e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[gradeQ[#, 1]], Assert[vectorQ[#]]} & /@ {e1, e2, e3};
{Assert[! gradeQ[#, 1]], Assert[! vectorQ[#]]} & /@
   {e0, e23, e31, e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[gradeQ[#, 2]], Assert[bivectorQ[#]]} & /@ {e23, e31, e12};
{Assert[! gradeQ[#, 2]], Assert[! bivectorQ[#]]} & /@
   {e0, e1, e2, e3, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[gradeQ[#, 3]], Assert[trivectorQ[#]]} & /@ {e123};
{Assert[! gradeQ[#, 3]], Assert[! trivectorQ[#]]} & /@
   {e0, e1, e2, e3, e23, e31, e12, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[gradeQ[#, -1]]} & /@ {m01, m02, m03, m12, m13, m23, m012, m013, m023, m123};
{Assert[! gradeQ[#, -1]]} & /@ {e0, e1, e2, e3, e23, e31, e12, e123};

(*Grade selection tests.*)

{Assert[GradeSelection[#, 0] == e0], Assert[ScalarSelection[#] == e0]} & /@
   {e0, m01, m02, m03, m013, m012, m023};
{Assert[GradeSelection[#, 0] == 0], Assert[ScalarSelection[#] == 0]} & /@
   {e1, e2, e3, e23, e31, e12, e123, m12, m13, m23, m123};

{Assert[GradeSelection[#, 1] == e1], Assert[VectorSelection[#] == e1]} & /@
   {e1, m01, m12, m13, m012, m013, m123};
{Assert[GradeSelection[#, 1] == 0], Assert[VectorSelection[#] == 0]} & /@
   {e0, e23, e31, e12, e123, m02, m03, m23, m023};
```

```
{Assert[GradeSelection[#, 2] == e23], Assert[BivectorSelection[#] == e23]} & /@
  {e23, m02, m12, m23, m023, m123, m012};
{Assert[GradeSelection[#, 2] == 0], Assert[BivectorSelection[#] == 0]} & /@
  {e0, e1, e2, e3, e123, m01, m03, m13, m013};

{Assert[GradeSelection[#, 3] == e123], Assert[TrivectorSelection[#] == e123]} & /@
  {e123, m03, m13, m23, m013, m023, m123};
{Assert[GradeSelection[#, 3] == 0], Assert[TrivectorSelection[#] == 0]} & /@
  {e0, e1, e2, e3, e23, e31, e12, m01, m02, m12, m012};

(*Minus tests*)
Assert[-e0 == Scalar[-1]];
Assert[-e1 == Vector[-1, 1]];
Assert[-e2 == Vector[-1, 2]];
Assert[-e3 == Vector[-1, 3]];
Assert[-e23 == Bivector[-1, 2, 3]];
Assert[-e31 == Bivector[-1, 3, 1]];
Assert[-e12 == Bivector[-1, 1, 2]];
Assert[-e123 == Trivector[-1]];
Assert[-m01 == -e0 - e1];
Assert[-m02 == -e0 - e23];
Assert[-m03 == -e0 - e123];
Assert[-m12 == -e1 - e23];
Assert[-m13 == -e1 - e123];
Assert[-m23 == -e23 - e123];
Assert[-m012 == -e0 - e1 - e23];
Assert[-m013 == -e0 - e1 - e123];
Assert[-m023 == -e0 - e23 - e123];
Assert[-m123 == -e1 - e23 - e123];

(* Scalar/Pseudoscalar multiplication tests*)

{Assert[(#[[1]]) (#[[2]]) == #[[3]]], Assert[(#[[2]]) (#[[1]]) == #[[3]]]} & /@
  {{2, e0, Scalar[2]}, {2, e1, Vector[2, 1]}, {2, e2, Vector[2, 2]},
   {2, e3, Vector[2, 3]}, {2, e23, Bivector[2, 2, 3]}, {2, e31, Bivector[2, 3, 1]},
   {2, e12, Bivector[2, 1, 2]}, {2, e123, Trivector[2]},
   {2, m01, 2 e0 + 2 e1}, {2, m02, 2 e0 + 2 e23}, {2, m03, 2 e0 + 2 e123},
   {2, m12, 2 e1 + 2 e23}, {2, m13, 2 e1 + 2 e123}, {2, m23, 2 e23 + 2 e123},
   {2, m012, 2 e0 + 2 e1 + 2 e23}, {2, m013, 2 e0 + 2 e1 + 2 e123},
   {2, m023, 2 e0 + 2 e23 + 2 e123}, {2, m123, 2 e1 + 2 e23 + 2 e123}} ;

{
    Assert[(#[[1]]) (#[[2]]) == #[[3]]],
    Assert[(#[[2]]) (#[[1]]) == #[[3]]],
```

```
     Assert[(#[[1]]) ** (#[[2]]) == #[[3]]],
     Assert[(#[[2]]) ** (#[[1]]) == #[[3]]]
    } & /@ {
    {e0, e0, Scalar[1]}, {e0, e1, Vector[1, 1]}, {e0, e2, Vector[1, 2]},
    {e0, e3, Vector[1, 3]}, {e0, e23, Bivector[1, 2, 3]}, {e0, e31, Bivector[1, 3, 1]},
    {e0, e12, Bivector[1, 1, 2]}, {e0, e123, Trivector[1]}, {e0, m01, e0 + e1},
    {e0, m02, e0 + e23}, {e0, m03, e0 + e123}, {e0, m12, e1 + e23}, {e0, m13, e1 + e123},
    {e0, m23, e23 + e123}, {e0, m012, e0 + e1 + e23}, {e0, m013, e0 + e1 + e123},
    {e0, m023, e0 + e23 + e123}, {e0, m123, e1 + e23 + e123}, {e123, e0, Trivector[1]},
    {e123, e1, Bivector[1, 2, 3]}, {e123, e2, Bivector[1, 3, 1]},
    {e123, e3, Bivector[1, 1, 2]}, {e123, e23, Vector[-1, 1]},
    {e123, e31, Vector[-1, 2]}, {e123, e12, Vector[-1, 3]}, {e123, e123, Scalar[-1]},
    {e123, m01, e123 e0 + e123 e1}, {e123, m02, e123 e0 + e123 e23},
    {e123, m03, e123 e0 + e123 e123}, {e123, m12, e123 e1 + e123 e23},
    {e123, m13, e123 e1 + e123 e123}, {e123, m23, e123 e23 + e123 e123},
    {e123, m012, e123 e0 + e123 e1 + e123 e23}, {e123, m013, e123 e0 + e123 e1 + e123 e123},
    {e123, m023, e123 e0 + e123 e23 + e123 e123},
    {e123, m123, e123 e1 + e123 e23 + e123 e123}};

(*Tests for (non-commutitive) multiplication, dot and wedge.*)
ClearAll[ mbasis, ptable, dtable, wtable, stable ] ;
mbasis = {e1, e2, e3, e23, e31, e12, e123};

ptable = (*e1,e2,e3,e23,e31,e12,e123*)
  (*e1*){{e0, e12, e13, e123, -e3, e2, e23},
   (*e2*){e21, e0, e23, e3, e123, -e1, e31},
   (*e3*){e31, e32, e0, -e2, e1, e123, e12},
   (*e23*){e123, -e3, e2, -e0, e21, e31, -e1},
   (*e31*){e3, e123, -e1, e12, -e0, e32, -e2},
   (*e12*){-e2, e1, e123, e13, e23, -e0, -e3},
   (*e123*){e23, e31, e12, -e1, -e2, -e3, -e0}};

dtable = (*e1,e2,e3,e23,e31,e12,e123*)
  (*e1*){{e0, 0, 0, 0, -e3, e2, e23},
   (*e2*){0, e0, 0, e3, 0, -e1, e31},
   (*e3*){0, 0, e0, -e2, e1, 0, e12},
   (*e23*){0, -e3, e2, -e0, 0, 0, -e1},
   (*e31*){e3, 0, -e1, 0, -e0, 0, -e2},
   (*e12*){-e2, e1, 0, 0, 0, -e0, -e3},
   (*e123*){e23, e31, e12, -e1, -e2, -e3, -e0}};

wtable = (*e1,e2,e3,e23,e31,e12,e123*)
  (*e1*){{0, e12, e13, e123, 0, 0, 0},
   (*e2*){e21, 0, e23, 0, e123, 0, 0},
```

```
    (*e3*){e31, e32, 0, 0, 0, e123, 0},
    (*e23*){e123, 0, 0, 0, 0, 0, 0},
    (*e31*){0, e123, 0, 0, 0, 0, 0},
    (*e12*){0, 0, e123, 0, 0, 0, 0},
    (*e123*){0, 0, 0, 0, 0, 0, 0}};

stable = (*e1,e2,e3,e23,e31,e12,e123*)
   (*e1*){{1, 0, 0, 0, 0, 0, 0},
    (*e2*){0, 1, 0, 0, 0, 0, 0},
    (*e3*){0, 0, 1, 0, 0, 0, 0},
    (*e23*){0, 0, 0, -1, 0, 0, 0},
    (*e31*){0, 0, 0, 0, -1, 0, 0},
    (*e12*){0, 0, 0, 0, 0, -1, 0},
    (*e123*){0, 0, 0, 0, 0, 0, -1}};


Table[
  {
   Assert[ mbasis[[i]] ** mbasis[[j]] == ptable[[i, j]]],
   Assert[ NonCommutativeMultiply[ mbasis[[i]], mbasis[[j]]] == ptable[[i, j]]]
  },
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] ;

Table[
  {
   Assert[ mbasis[[i]] . mbasis[[j]] == dtable[[i, j]]],
   Assert[ Dot[mbasis[[i]], mbasis[[j]]] == dtable[[i, j]]]
  },
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] ;

Table[
  {
   Assert[ mbasis[[i]] ∧ mbasis[[j]] == wtable[[i, j]]],
   Assert[ Wedge[mbasis[[i]], mbasis[[j]]] == wtable[[i, j]]]
  },
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] ;

Table[
  {
   Assert[ 〈mbasis[[i]] , mbasis[[j]]〉 == stable[[i, j]]],
   Assert[ ScalarProduct[mbasis[[i]] , mbasis[[j]]] == stable[[i, j]]]
  },
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] ;
```

```
Table[
  {
   Assert[⟨mbasis[[i]] ** mbasis[[j]]⟩ == stable[[i, j]]],
   Assert[ScalarValue[(mbasis[[i]] ** mbasis[[j]])] == stable[[i, j]]]
  },
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}];
```

Manual tests, showing the results of various products in traditional form.

```
ClearAll[x, y]

Row[{"(",
     (#[[1]]) // TraditionalForm,
     ")(",
     (#[[2]]) // TraditionalForm,
     ") = ",
     ((#[[1]]) ** (#[[2]])) // TraditionalForm
    }] &/@ {
   {e2, e2},
   {e2, e21},
   {e2 - 5 e21, e2},
   {e2, e2 + 3 e21} ,
   {e2 + Tan[y] e21, e2},
   {Cos[y] e2, e2 + Sin[x] e21}
  } // Column

Table[
  Row[{
    mbasis[[i]] // TraditionalForm,
    " ",
    mbasis[[j]] // TraditionalForm,
    " = ",
    (mbasis[[i]] ** mbasis[[j]]) // TraditionalForm
   }],
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] // Grid

Table[
  Row[{
    mbasis[[i]] // TraditionalForm,
    "·",
    mbasis[[j]] // TraditionalForm,
```

```
    " = ",
    (mbasis[[i]].mbasis[[j]]) // TraditionalForm
   }],
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] // Grid

Table[
  Row[{
    mbasis[[i]] // TraditionalForm,
    "∧",
    mbasis[[j]] // TraditionalForm,
    " = ",
    (mbasis[[i]] ∧ mbasis[[j]]) // TraditionalForm
   }],
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] // Grid

Table[
  Row[{
    "<",
    mbasis[[i]] // TraditionalForm,
    " ",
    mbasis[[j]] // TraditionalForm,
    ">",
    " = ",
    (〈mbasis[[i]], mbasis[[j]]〉) // TraditionalForm
   }],
  {i, 1, mbasis // Length}, {j, 1, mbasis // Length}] // Grid


(*XXForm tests (manual verification) *)
Column[(# // TraditionalForm) & /@ {e0, e1, e2, e3, e23, e31,
   e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123}]

Column[(# // StandardForm) & /@ {e0, e1, e2, e3, e23, e31,
   e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123}]

Column[(# // DisplayForm) & /@ {e0, e1, e2, e3, e23, e31,
   e12, e123, m01, m02, m03, m12, m13, m23, m012, m013, m023, m123}]
```

$(\mathbf{e}_2)(\mathbf{e}_2) = 1$

$(\mathbf{e}_2)(-\mathbf{e}_{12}) = \mathbf{e}_1$

$(5\,\mathbf{e}_{12} + \mathbf{e}_2)(\mathbf{e}_2) = 5\,\mathbf{e}_1 + 1$

$(\mathbf{e}_2)(\mathbf{e}_2 - 3\,\mathbf{e}_{12}) = 3\,\mathbf{e}_1 + 1$

$(\mathbf{e}_2 - \mathbf{e}_{12}\tan(y))(\mathbf{e}_2) = 1 - \mathbf{e}_1\tan(y)$

$(\mathbf{e}_2\cos(y))(\mathbf{e}_2 - \mathbf{e}_{12}\sin(x)) = \mathbf{e}_1\sin(x)\cos(y) + \cos(y)$

$\mathbf{e}_1\ \mathbf{e}_1 = 1$	$\mathbf{e}_1\ \mathbf{e}_2 = \mathbf{e}_{12}$	$\mathbf{e}_1\ \mathbf{e}_3 = -\mathbf{e}_{31}$	$\mathbf{e}_1\ \mathbf{e}_{23} = \mathbf{e}_{123}$	$\mathbf{e}_1\ \mathbf{e}_{31} = -\mathbf{e}_3$	$\mathbf{e}_1\ \mathbf{e}_{12} = \mathbf{e}_2$	$\mathbf{e}_1\ \mathbf{e}_{123} = \mathbf{e}_{23}$
$\mathbf{e}_2\ \mathbf{e}_1 = -\mathbf{e}_{12}$	$\mathbf{e}_2\ \mathbf{e}_2 = 1$	$\mathbf{e}_2\ \mathbf{e}_3 = \mathbf{e}_{23}$	$\mathbf{e}_2\ \mathbf{e}_{23} = \mathbf{e}_3$	$\mathbf{e}_2\ \mathbf{e}_{31} = \mathbf{e}_{123}$	$\mathbf{e}_2\ \mathbf{e}_{12} = -\mathbf{e}_1$	$\mathbf{e}_2\ \mathbf{e}_{123} = \mathbf{e}_{31}$
$\mathbf{e}_3\ \mathbf{e}_1 = \mathbf{e}_{31}$	$\mathbf{e}_3\ \mathbf{e}_2 = -\mathbf{e}_{23}$	$\mathbf{e}_3\ \mathbf{e}_3 = 1$	$\mathbf{e}_3\ \mathbf{e}_{23} = -\mathbf{e}_2$	$\mathbf{e}_3\ \mathbf{e}_{31} = \mathbf{e}_1$	$\mathbf{e}_3\ \mathbf{e}_{12} = \mathbf{e}_{123}$	$\mathbf{e}_3\ \mathbf{e}_{123} = \mathbf{e}_{12}$
$\mathbf{e}_{23}\ \mathbf{e}_1 = \mathbf{e}_{123}$	$\mathbf{e}_{23}\ \mathbf{e}_2 = -\mathbf{e}_3$	$\mathbf{e}_{23}\ \mathbf{e}_3 = \mathbf{e}_2$	$\mathbf{e}_{23}\ \mathbf{e}_{23} = -1$	$\mathbf{e}_{23}\ \mathbf{e}_{31} = -\mathbf{e}_{12}$	$\mathbf{e}_{23}\ \mathbf{e}_{12} = \mathbf{e}_{31}$	$\mathbf{e}_{23}\ \mathbf{e}_{123} = -\mathbf{e}_1$
$\mathbf{e}_{31}\ \mathbf{e}_1 = \mathbf{e}_3$	$\mathbf{e}_{31}\ \mathbf{e}_2 = \mathbf{e}_{123}$	$\mathbf{e}_{31}\ \mathbf{e}_3 = -\mathbf{e}_1$	$\mathbf{e}_{31}\ \mathbf{e}_{23} = \mathbf{e}_{12}$	$\mathbf{e}_{31}\ \mathbf{e}_{31} = -1$	$\mathbf{e}_{31}\ \mathbf{e}_{12} = -\mathbf{e}_{23}$	$\mathbf{e}_{31}\ \mathbf{e}_{123} = -\mathbf{e}_2$
$\mathbf{e}_{12}\ \mathbf{e}_1 = -\mathbf{e}_2$	$\mathbf{e}_{12}\ \mathbf{e}_2 = \mathbf{e}_1$	$\mathbf{e}_{12}\ \mathbf{e}_3 = \mathbf{e}_{123}$	$\mathbf{e}_{12}\ \mathbf{e}_{23} = -\mathbf{e}_{31}$	$\mathbf{e}_{12}\ \mathbf{e}_{31} = \mathbf{e}_{23}$	$\mathbf{e}_{12}\ \mathbf{e}_{12} = -1$	$\mathbf{e}_{12}\ \mathbf{e}_{123} = -\mathbf{e}_3$
$\mathbf{e}_{123}\ \mathbf{e}_1 = \mathbf{e}_{23}$	$\mathbf{e}_{123}\ \mathbf{e}_2 = \mathbf{e}_{31}$	$\mathbf{e}_{123}\ \mathbf{e}_3 = \mathbf{e}_{12}$	$\mathbf{e}_{123}\ \mathbf{e}_{23} = -\mathbf{e}_1$	$\mathbf{e}_{123}\ \mathbf{e}_{31} = -\mathbf{e}_2$	$\mathbf{e}_{123}\ \mathbf{e}_{12} = -\mathbf{e}_3$	$\mathbf{e}_{123}\ \mathbf{e}_{123} = -1$

$\mathbf{e}_1 \cdot \mathbf{e}_1 = 1$	$\mathbf{e}_1 \cdot \mathbf{e}_2 = 0$	$\mathbf{e}_1 \cdot \mathbf{e}_3 = 0$	$\mathbf{e}_1 \cdot \mathbf{e}_{23} = 0$	$\mathbf{e}_1 \cdot \mathbf{e}_{31} = -\mathbf{e}_3$	$\mathbf{e}_1 \cdot \mathbf{e}_{12} = \mathbf{e}_2$	$\mathbf{e}_1 \cdot \mathbf{e}_{123} = \mathbf{e}_{23}$
$\mathbf{e}_2 \cdot \mathbf{e}_1 = 0$	$\mathbf{e}_2 \cdot \mathbf{e}_2 = 1$	$\mathbf{e}_2 \cdot \mathbf{e}_3 = 0$	$\mathbf{e}_2 \cdot \mathbf{e}_{23} = \mathbf{e}_3$	$\mathbf{e}_2 \cdot \mathbf{e}_{31} = 0$	$\mathbf{e}_2 \cdot \mathbf{e}_{12} = -\mathbf{e}_1$	$\mathbf{e}_2 \cdot \mathbf{e}_{123} = \mathbf{e}_{31}$
$\mathbf{e}_3 \cdot \mathbf{e}_1 = 0$	$\mathbf{e}_3 \cdot \mathbf{e}_2 = 0$	$\mathbf{e}_3 \cdot \mathbf{e}_3 = 1$	$\mathbf{e}_3 \cdot \mathbf{e}_{23} = -\mathbf{e}_2$	$\mathbf{e}_3 \cdot \mathbf{e}_{31} = \mathbf{e}_1$	$\mathbf{e}_3 \cdot \mathbf{e}_{12} = 0$	$\mathbf{e}_3 \cdot \mathbf{e}_{123} = \mathbf{e}_{12}$
$\mathbf{e}_{23} \cdot \mathbf{e}_1 = 0$	$\mathbf{e}_{23} \cdot \mathbf{e}_2 = -\mathbf{e}_3$	$\mathbf{e}_{23} \cdot \mathbf{e}_3 = \mathbf{e}_2$	$\mathbf{e}_{23} \cdot \mathbf{e}_{23} = -1$	$\mathbf{e}_{23} \cdot \mathbf{e}_{31} = 0$	$\mathbf{e}_{23} \cdot \mathbf{e}_{12} = 0$	$\mathbf{e}_{23} \cdot \mathbf{e}_{123} = -\mathbf{e}_1$
$\mathbf{e}_{31} \cdot \mathbf{e}_1 = \mathbf{e}_3$	$\mathbf{e}_{31} \cdot \mathbf{e}_2 = 0$	$\mathbf{e}_{31} \cdot \mathbf{e}_3 = -\mathbf{e}_1$	$\mathbf{e}_{31} \cdot \mathbf{e}_{23} = 0$	$\mathbf{e}_{31} \cdot \mathbf{e}_{31} = -1$	$\mathbf{e}_{31} \cdot \mathbf{e}_{12} = 0$	$\mathbf{e}_{31} \cdot \mathbf{e}_{123} = -\mathbf{e}_2$
$\mathbf{e}_{12} \cdot \mathbf{e}_1 = -\mathbf{e}_2$	$\mathbf{e}_{12} \cdot \mathbf{e}_2 = \mathbf{e}_1$	$\mathbf{e}_{12} \cdot \mathbf{e}_3 = 0$	$\mathbf{e}_{12} \cdot \mathbf{e}_{23} = 0$	$\mathbf{e}_{12} \cdot \mathbf{e}_{31} = 0$	$\mathbf{e}_{12} \cdot \mathbf{e}_{12} = -1$	$\mathbf{e}_{12} \cdot \mathbf{e}_{123} = -\mathbf{e}_3$
$\mathbf{e}_{123} \cdot \mathbf{e}_1 = \mathbf{e}_{23}$	$\mathbf{e}_{123} \cdot \mathbf{e}_2 = \mathbf{e}_{31}$	$\mathbf{e}_{123} \cdot \mathbf{e}_3 = \mathbf{e}_{12}$	$\mathbf{e}_{123} \cdot \mathbf{e}_{23} = -\mathbf{e}_1$	$\mathbf{e}_{123} \cdot \mathbf{e}_{31} = -\mathbf{e}_2$	$\mathbf{e}_{123} \cdot \mathbf{e}_{12} = -\mathbf{e}_3$	$\mathbf{e}_{123} \cdot \mathbf{e}_{123} = -1$

$\mathbf{e}_1\wedge\mathbf{e}_1 = 0$	$\mathbf{e}_1\wedge\mathbf{e}_2 = \mathbf{e}_{12}$	$\mathbf{e}_1\wedge\mathbf{e}_3 = -\mathbf{e}_{31}$	$\mathbf{e}_1\wedge\mathbf{e}_{23} = \mathbf{e}_{123}$	$\mathbf{e}_1\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_1\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_1\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_2\wedge\mathbf{e}_1 = -\mathbf{e}_{12}$	$\mathbf{e}_2\wedge\mathbf{e}_2 = 0$	$\mathbf{e}_2\wedge\mathbf{e}_3 = \mathbf{e}_{23}$	$\mathbf{e}_2\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_2\wedge\mathbf{e}_{31} = \mathbf{e}_{123}$	$\mathbf{e}_2\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_2\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_3\wedge\mathbf{e}_1 = \mathbf{e}_{31}$	$\mathbf{e}_3\wedge\mathbf{e}_2 = -\mathbf{e}_{23}$	$\mathbf{e}_3\wedge\mathbf{e}_3 = 0$	$\mathbf{e}_3\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_3\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_3\wedge\mathbf{e}_{12} = \mathbf{e}_{123}$	$\mathbf{e}_3\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_{23}\wedge\mathbf{e}_1 = \mathbf{e}_{123}$	$\mathbf{e}_{23}\wedge\mathbf{e}_2 = 0$	$\mathbf{e}_{23}\wedge\mathbf{e}_3 = 0$	$\mathbf{e}_{23}\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_{23}\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_{23}\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_{23}\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_{31}\wedge\mathbf{e}_1 = 0$	$\mathbf{e}_{31}\wedge\mathbf{e}_2 = \mathbf{e}_{123}$	$\mathbf{e}_{31}\wedge\mathbf{e}_3 = 0$	$\mathbf{e}_{31}\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_{31}\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_{31}\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_{31}\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_{12}\wedge\mathbf{e}_1 = 0$	$\mathbf{e}_{12}\wedge\mathbf{e}_2 = 0$	$\mathbf{e}_{12}\wedge\mathbf{e}_3 = \mathbf{e}_{123}$	$\mathbf{e}_{12}\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_{12}\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_{12}\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_{12}\wedge\mathbf{e}_{123} = 0$
$\mathbf{e}_{123}\wedge\mathbf{e}_1 = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_2 = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_3 = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_{23} = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_{31} = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_{12} = 0$	$\mathbf{e}_{123}\wedge\mathbf{e}_{123} = 0$

$\langle\mathbf{e}_1\ \mathbf{e}_1\rangle = 1$	$\langle\mathbf{e}_1\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_1\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_1\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_1\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_1\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_1\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_2\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_2\ \mathbf{e}_2\rangle = 1$	$\langle\mathbf{e}_2\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_2\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_2\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_2\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_2\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_3\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_3\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_3\ \mathbf{e}_3\rangle = 1$	$\langle\mathbf{e}_3\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_3\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_3\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_3\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_{23}\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_{23}\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_{23}\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_{23}\ \mathbf{e}_{23}\rangle = -1$	$\langle\mathbf{e}_{23}\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_{23}\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_{23}\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_{31}\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_{31}\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_{31}\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_{31}\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_{31}\ \mathbf{e}_{31}\rangle = -1$	$\langle\mathbf{e}_{31}\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_{31}\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_{12}\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_{12}\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_{12}\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_{12}\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_{12}\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_{12}\ \mathbf{e}_{12}\rangle = -1$	$\langle\mathbf{e}_{12}\ \mathbf{e}_{123}\rangle = 0$
$\langle\mathbf{e}_{123}\ \mathbf{e}_1\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_2\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_3\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_{23}\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_{31}\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_{12}\rangle = 0$	$\langle\mathbf{e}_{123}\ \mathbf{e}_{123}\rangle = -1$

1
$\mathbf{e}_1$
$\mathbf{e}_2$
$\mathbf{e}_3$
$\mathbf{e}_{23}$
$\mathbf{e}_{31}$
$\mathbf{e}_{12}$
$\mathbf{e}_{123}$
$\mathbf{e}_1 + 1$
$\mathbf{e}_{23} + 1$
$\mathbf{e}_{123} + 1$
$\mathbf{e}_{23} + \mathbf{e}_1$
$\mathbf{e}_{123} + \mathbf{e}_1$
$\mathbf{e}_{123} + \mathbf{e}_{23}$
$\mathbf{e}_{23} + \mathbf{e}_1 + 1$
$\mathbf{e}_{123} + \mathbf{e}_1 + 1$
$\mathbf{e}_{123} + \mathbf{e}_{23} + 1$
$\mathbf{e}_{123} + \mathbf{e}_{23} + \mathbf{e}_1$

```
1
e[1]
e[2]
e[3]
e[2] e[3]
e[1] e[3]
e[1] e[2]
e[1] e[2] e[3]
1 + e[1]
1 + e[2] e[3]
1 + e[1] e[2] e[3]
e[1] + e[2] e[3]
e[1] + e[1] e[2] e[3]
e[2] e[3] + e[1] e[2] e[3]
1 + e[1] + e[2] e[3]
1 + e[1] + e[1] e[2] e[3]
1 + e[2] e[3] + e[1] e[2] e[3]
e[1] + e[2] e[3] + e[1] e[2] e[3]
```

1
$\mathbf{e}_1$
$\mathbf{e}_2$
$\mathbf{e}_3$
$\mathbf{e}_{23}$
$\mathbf{e}_{31}$
$\mathbf{e}_{12}$
$\mathbf{e}_{123}$
$1 + \mathbf{e}_1$
$1 + \mathbf{e}_{23}$
$1 + \mathbf{e}_{123}$
$\mathbf{e}_1 + \mathbf{e}_{23}$
$\mathbf{e}_1 + \mathbf{e}_{123}$
$\mathbf{e}_{123} + \mathbf{e}_{23}$
$1 + \mathbf{e}_1 + \mathbf{e}_{23}$
$1 + \mathbf{e}_1 + \mathbf{e}_{123}$
$1 + \mathbf{e}_{123} + \mathbf{e}_{23}$
$\mathbf{e}_1 + \mathbf{e}_{123} + \mathbf{e}_{23}$

```
TODO : test multivector products : dot, wedge, **
```

```
(* manual test, or just the dot product *)
ClearAll[m1, m2]
m1 = Scalar[1] + Vector[1, 2] + Bivector[1, 2, 3] + Trivector[1];
m2 = 2 Scalar[1] - Vector[1, 2] + 3 Bivector[1, 3, 1] - Trivector[1];

m1.m2 // TraditionalForm
```

(Manual) tests for grad, div, and curl.

```
In[151]:= ClearAll[s, v, b, t,
  grads,
  gradv, curlv, divv, vcurlv,
  gradb, curlb, divb,
  gradt, curlt, divt,
  x, y, z, f, g, h]


s := Scalar[g[x, y, z]];
grads = grad[s, {x, y, z}];

v := Vector[f[x, y, z], 1] + Vector[g[x, y, z], 2] + Vector[h[x, y, z], 3];
b := Trivector[1] v ;
t := Trivector[1] s ;

gradv := grad[v, {x, y, z}] ;
divv := div[v, {x, y, z}] ;
```

```
curlv := curl[v, {x, y, z}] ;
vcurlv := vcurl[v, {x, y, z}] ;

gradb := grad[b, {x, y, z}] ;
divb := div[b, {x, y, z}] ;
curlb := curl[b, {x, y, z}] ;

gradt := grad[t, {x, y, z}];
divt := div[t, {x, y, z}];
curlt := curl[t, {x, y, z}];


({# // First, " = ", (# // Last) // DisplayForm} &/@
   {{"s", s},
    {"∇ s", grads},
    {"v", v},
    {"∇ v", gradv},
    {"∇ · v", divv},
    {"∇ ∧ v", curlv},
    {"∇ × v", vcurlv},
    {"b", b},
    {"∇ b", gradb},
    {"∇ · b", divb},
    {"∇ ∧ b", curlb},
    {"t", t},
    {"∇ t", gradt},
    {"∇ · t", divt},
    {"∇ ∧ t", curlt}
   }) // Grid
```

Out[167]=

$$
\begin{array}{lcl}
s & = & g[x,y,z] \\
\nabla\ s & = & \mathbf{e}_3\, g^{(0,0,1)}[x,y,z] + \mathbf{e}_2\, g^{(0,1,0)}[x,y,z] + \mathbf{e}_1\, g^{(1,0,0)}[x,y,z] \\
v & = & f[x,y,z]\,\mathbf{e}_1 + g[x,y,z]\,\mathbf{e}_2 + h[x,y,z]\,\mathbf{e}_3 \\
\nabla\ v & = & h^{(0,0,1)}[x,y,z] + g^{(0,1,0)}[x,y,z] + \mathbf{e}_{23}\left(-g^{(0,0,1)}[x,y,z] + h^{(0,1,0)}[x,y,z]\right) + \\
 & & f^{(1,0,0)}[x,y,z] + \mathbf{e}_{12}\left(-f^{(0,1,0)}[x,y,z] + g^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_{31}\left(f^{(0,0,1)}[x,y,z] - h^{(1,0,0)}[x,y,z]\right) \\
\nabla\cdot v & = & h^{(0,0,1)}[x,y,z] + g^{(0,1,0)}[x,y,z] + f^{(1,0,0)}[x,y,z] \\
\nabla\wedge v & = & \mathbf{e}_{23}\left(-g^{(0,0,1)}[x,y,z] + h^{(0,1,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_{12}\left(-f^{(0,1,0)}[x,y,z] + g^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_{31}\left(f^{(0,0,1)}[x,y,z] - h^{(1,0,0)}[x,y,z]\right) \\
\nabla\times v & = & \mathbf{e}_1\left(-g^{(0,0,1)}[x,y,z] + h^{(0,1,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_3\left(-f^{(0,1,0)}[x,y,z] + g^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_2\left(f^{(0,0,1)}[x,y,z] - h^{(1,0,0)}[x,y,z]\right) \\
b & = & h[x,y,z]\,\mathbf{e}_{12} + f[x,y,z]\,\mathbf{e}_{23} + g[x,y,z]\,\mathbf{e}_{31} \\
\nabla\ b & = & \mathbf{e}_1\left(g^{(0,0,1)}[x,y,z] - h^{(0,1,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_{123}\left(h^{(0,0,1)}[x,y,z] + g^{(0,1,0)}[x,y,z] + f^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_3\left(f^{(0,1,0)}[x,y,z] - g^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_2\left(-f^{(0,0,1)}[x,y,z] + h^{(1,0,0)}[x,y,z]\right) \\
\nabla\cdot b & = & \mathbf{e}_1\left(g^{(0,0,1)}[x,y,z] - h^{(0,1,0)}[x,y,z]\right) + \mathbf{e}_3\left(f^{(0,1,0)}[x,y,z] - g^{(1,0,0)}[x,y,z]\right) + \\
 & & \mathbf{e}_2\left(-f^{(0,0,1)}[x,y,z] + h^{(1,0,0)}[x,y,z]\right) \\
\nabla\wedge b & = & \mathbf{e}_{123}\left(h^{(0,0,1)}[x,y,z] + g^{(0,1,0)}[x,y,z] + f^{(1,0,0)}[x,y,z]\right) \\
t & = & g[x,y,z]\,\mathbf{e}_{123} \\
\nabla\ t & = & \mathbf{e}_{12}\, g^{(0,0,1)}[x,y,z] + \mathbf{e}_{31}\, g^{(0,1,0)}[x,y,z] + \mathbf{e}_{23}\, g^{(1,0,0)}[x,y,z] \\
\nabla\cdot t & = & \mathbf{e}_{12}\, g^{(0,0,1)}[x,y,z] + \mathbf{e}_{31}\, g^{(0,1,0)}[x,y,z] + \mathbf{e}_{23}\, g^{(1,0,0)}[x,y,z] \\
\nabla\wedge t & = & 0
\end{array}
$$