VAGC_grad_div_curl_Equation5.3.wxm

A short development document for Geometric Algebra with wxMaxima
just to test some calculus functions within the GAwxM environment,
contains...
Initialization
Loading of functions (intrinsic and GA specific)
Pseudoscalar definition (specifies the space dimension) and
Calculation of the inverse pseudoscalar used to generate the dual of a multivector
Enumeration of the standard basis for the specified dimension

Equation 5.3, page 60, grad F = div F + curl F, only for the vector operator

Initialization

```
(%i29) ext:["wxm"]$
       file_type_maxima:append(ext,file_type_maxima)$
       batchload("initialize_fns")$
```

the pseudoscalar and its inverse
the lowest useable dimension pseudoscalar should be {e1,e2} i.e. Plen = 2
e.g. for four dimensions edit Pseudos:{e1,e2,e3}$ to Pseudos:{e1,e2,e3,e4}$

```
(%i1)  Pseudos:{e1,e2,e3}$
       Pvar:listofvars(Pseudos)$
       Plen:length(Pvar)$
       I:Pseudos$
       ni:(Plen-1)*Plen/2$
       Ii:(-1)^ni*I$
       kill(ni)$
       ldisplay(Pvar)$

  (%t8) Pvar=[e1,e2,e3]

(%i9)  batchload("initialize_lsts")$

  (%t9) lstblds=[[{e1},{e2},{e3}],[{e1,e2},{e1,e3},{e2,e3}],[{e1,e2,e3}]]
 (%t10) allblds=[{e1},{e2},{e3},{e1,e2},{e1,e3},{e2,e3},{e1,e2,e3}]
 (%t11) invblds=[{e1},{e2},{e3},-{e1,e2},-{e1,e3},-{e2,e3},-{e1,e2,e3}]
```

end of Initialization

set derivabbrev:false$

```
(%i12) derivabbrev:false$
```

the identity is only true for the vector differential operator (ref. LAGA Theorem 6.21)

```
(%i13) Xlst:[x1,x2,x3,0,0,0,0]$

(%i14) eJ:allblds$
```

form the coordinate vector, X, from the list of coefficients

```
(%i15) lenlst:2^Plen-1$
       X:0$
       for j:1 thru lenlst do
       block(X:X+Xlst[j]*eJ[j])$

(%i18) ldisplay(X)$

 (%t18) X={e3}*x3+{e2}*x2+{e1}*x1

(%i19) Fstr:"F"$

(%i20) gradF:mvgrad(Fstr,Xlst)$
       ldisplay(gradF)$

 (%t21) gradF={e3}&* (d/(d*x3)*F)+{e2}&* (d/(d*x2)*F)+{e1}&* (d/(d*x1)*F)

(%i22) divF:mvdiv(Fstr,Xlst)$
       ldisplay(divF)$

 (%t23) divF={e3}&. (d/(d*x3)*F)+{e2}&. (d/(d*x2)*F)+{e1}&. (d/(d*x1)*F)

(%i24) curlF:mvcurl(Fstr,Xlst)$
       ldisplay(curlF)$

 (%t25) curlF={e3}&^ (d/(d*x3)*F)+{e2}&^ (d/(d*x2)*F)+{e1}&^ (d/(d*x1)*F)
```

form a multivector function of X to contain higher grades than the coordinate vector

```
(%i26) X&*{e1,e2}+X&^{e2}+X&^{e3}$
       F:facsum(%,allblds)$
       ldisplay(F)$

 (%t28) F=-{e2,e3}*(x3-x2)+{e1,e2,e3}*x3-{e1}*x2+{e2}*x1+{e1,e3}*x1+{e1,e2}
*x1

(%i29) ev(gradF,diff)$
       lhs:facsum(%,allblds)$
       ldisplay(lhs)$

 (%t31) lhs=2*{e3}+2*{e2}+3*{e1,e2}
```

form the rhs in two different ways

```
(%i32) divF+curlF$
       ev(%,diff)$
       rhs:facsum(%,allblds)$
       ldisplay(rhs)$

 (%t35) rhs=2*{e3}+2*{e2}+3*{e1,e2}

(%i36) dF:ev(divF,diff)$
       cF:ev(curlF,diff)$
       dF+cF$
       rhs:facsum(%,allblds)$
       ldisplay(rhs)$

 (%t40) rhs=2*{e3}+2*{e2}+3*{e1,e2}

(%i41) is(equal(lhs,rhs));

(%o41) true
```