ctensor_calc_Einstein.wxm

An application document for Geometric Algebra using wxMaxima

Ref: The Survey, paragraph 4.1.2 for flat spacetime

The intrinsic functions for curved spacetime have been applied within the GAwxM environment,
without using the GAwxM extrinsic functions, in the hope that it might be useful someday

Use of ctensor functions christof() and uricci() to calculate the Ricci tensor, Ricci scalar
and hence the covariant Einstein tensor from a user defined covariant metric tensor

Initialization

```
(%i1) 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,e4}$
      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,e4]
```

```
(%i9) batchload("initialize_lsts")$
```

```
(%t9) lstblds=[[{e1},{e2},{e3},{e4}],[{e1,e2},{e1,e3},{e1,e4},{e2,e3},{e2,e4}
,{e3,e4}],[{e1,e2,e3},{e1,e2,e4},{e1,e3,e4},{e2,e3,e4}],[{e1,e2,e3,e4}]]
(%t10) allblds=[{e1},{e2},{e3},{e4},{e1,e2},{e1,e3},{e1,e4},{e2,e3},{e2,e4},{
e3,e4},{e1,e2,e3},{e1,e2,e4},{e1,e3,e4},{e2,e3,e4},{e1,e2,e3,e4}]
(%t11) invblds=[{e1},{e2},{e3},{e4},-{e1,e2},-{e1,e3},-{e1,e4},-{e2,e3},-{e2
,e4},-{e3,e4},-{e1,e2,e3},-{e1,e2,e4},-{e1,e3,e4},-{e2,e3,e4},{e1,e2,e3,e4}]
```

end of Initialization

set derivabbrev:false$

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

```
(%i13) ratprint:false$
```

The Survey, para.4.1.3

show the flat spacetime required for the imitation of G(1,3), where we have used the fourth
axis, e4, for the time axis and the intrinsic maxima imaginary, %i, for the space axes

```
(%i14) g1:%i*{e1}$
       g2:%i*{e2}$
       g3:%i*{e3}$
       g4:{e4}$
```

the spacetime coordinate vector

```
(%i18) xb:x*g1+y*g2+z*g3+t*g4;
```

```
(%o18) %i*{e3}*z+%i*{e2}*y+%i*{e1}*x+{e4}*t
```

The Survey, paragraph 4.1.2, suggests a spacetime split using the geometric product, here
that is xb&*g4; we could call this the split spacetime coordinate, splxb

```
(%i19) splxb:xb&*g4;
```

```
(%o19)/R/ {e3,e4}*%i*z+{e2,e4}*%i*y+{e1,e4}*%i*x+t
```

for the actual split into time and space

```
(%i20) realpart(splxb);
       imagpart(splxb);
```

```
(%o20) t
(%o21) {e3,e4}*z+{e2,e4}*y+{e1,e4}*x
```

in order to unsplit the split spacetime we postmultiply by the inverse of g4 (that also
occurs in the global list, invblds[] as {e4})

```
(%i22) invg4:g4/normod(g4)^2;
```

```
(%o22) {e4}
```

to recover the spacetime coordinate vector

```
(%i23) splxb&*invg4;
       is(equal(%,xb));
```

```
(%o23)/R/ %i*{e3}*z+%i*{e2}*y+%i*{e1}*x+{e4}*t
(%o24) true
```

for any 3D position vector, rb, in a flat spacetime

```
(%i25) rb:x*{e1}+y*{e2}+z*{e3}$
       normrb:normod(rb)$
       ldisplay(normrb)$
```

(%t27) $normrb=\sqrt{z^2+y^2+x^2}$

Start of tensor analysis for a nearly-flat spacetime

```
(%i28) load(ctensor)$
       init_ctensor()$
```

```
(%i30) ct_coords:[x,y,z,t];
       dim;
```

```
(%o30) [x,y,z,t]
(%o31) 4
```

```
(%i32) declare(a,real,b,real);
```

```
(%o32) done
```

GR seems to be a theory using differential equations to find a solution near
a particular point in space-time. With that in mind, and modern statements that
the universe seems to be very nearly flat, we can choose the covariant metric
tensor below for functions a(t) and b(t) such that 0 < a < 1 and 0 < b < 1

the covariant metric tensor

```
(%i33) gs:-(1+a(t))$
       gt:+(1-b(t))$
       lg:matrix([gs,0,0,0],[0,gs,0,0],[0,0,gs,0],[0,0,0,gt]);
```

(%o35) $$\begin{bmatrix} -a(t)-1 & 0 & 0 & 0 \\ 0 & -a(t)-1 & 0 & 0 \\ 0 & 0 & -a(t)-1 & 0 \\ 0 & 0 & 0 & 1-b(t) \end{bmatrix}$$

```
(%i36) matrixp(lg);
```

```
(%o36) true
```

the contravariant metric tensor

```
(%i37) ug:invert_by_lu(lg);
```

(%o37) $$\begin{bmatrix} \frac{1}{-a(t)-1} & 0 & 0 & 0 \\ 0 & \frac{1}{-a(t)-1} & 0 & 0 \\ 0 & 0 & \frac{1}{-a(t)-1} & 0 \\ 0 & 0 & 0 & \frac{1}{1-b(t)} \end{bmatrix}$$

```
(%i38) matrixp(ug);
```

```
(%o38) true
```

```
(%i39) derivabbrev:true$
```

the Christoffel tensor of the second kind

rational simplification

```
(%i40) ratchristof:true$
```

the third index is the contravariant index

```
(%i41) christof(false)$
```

```
(%i42) arrays;
```

```
(%o42) [nbases,lcs,mcs]
```

the mixed Ricci tensor, uric is derived from the covariant Ricci tensor, ric

```
(%i43) uricci(false)$
```

```
(%i44) arrays;
```

```
(%o44) [nbases,lcs,mcs,ric,uric]
```

lower the contravariant index of the mixed tensor, uric to form the covariant tensor, cric
in the form of a matrix, being more familiar that the array, ric from function ricci()

```
(%i45) cric:matrix([0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0])$
       for mg:1 thru dim do
       block(
       for ng:1 thru dim do
       block(
       csum:0,
       for kg:1 thru dim do
       block(
       csum:csum+lg[kg,ng]*uric[mg,kg]),
       cric[mg,ng]:csum))$
```

```
(%i47) matrixp(cric);
```

```
(%o47) true
```

```
(%i48) ldisplay(cric[1,1])$
       ldisplay(cric[4,4])$
```

(%t48) $$cric_{1,1}=\frac{\left(a(t)_t\right)*\left(b(t)_t\right)}{(2*b(t)-2)^2}-\frac{a(t)_{tt}}{2*b(t)-2}-\frac{\left(a(t)_t\right)^2}{(2*a(t)+2)*(2*b(t)-2)}$$

(%t49) $$cric_{4,4}=\frac{3*\left(a(t)_t\right)*\left(b(t)_t\right)}{(2*a(t)+2)*(2*b(t)-2)}-\frac{3*\left(a(t)_{tt}\right)}{2*a(t)+2}+\frac{3*\left(a(t)_t\right)^2}{(2*a(t)+2)^2}$$

calculate the Ricci scalar as the contraction of the covariant Ricci tensor, cric

```
(%i50) csum:0$
       for mg:1 thru dim do
       block(
       for ng:1 thru dim do
       block(
       csum:csum+ug[mg,ng]*cric[mg,ng]))$
       Rc:csum$
```

check against the intrinsic function for the Ricci scalar

```
(%i53) Ri:scurvature()$
```

```
(%i54) is(equal(Rc,Ri))$
       ldisplay(%)$
```

```
(%t55) %=true
```

use the calculated Ricci scalar, Rc to form R

```
(%i56) R:facsum(Rc,operator(diff))$
```

```
(%i57) ldisplay(R)$
```

(%t57) $$R=-\frac{3*\left(\left(a(t)_t\right)*\left(b(t)_t\right)-2*b(t)*\left(a(t)_{tt}\right)+2*\left(a(t)_{tt}\right)\right)}{2*(a(t)+1)*(b(t)-1)^2}$$

half the product of the covariant metric tensor and the calculated Ricci scalar

```
(%i58) hgR:(1/2)*R*lg$
```

```
(%i59) matrixp(hgR);
```

```
(%o59) true
```

```
(%i60) ldisplay(hgR[1,1])$
       ldisplay(hgR[4,4])$
```

(%t60) $$hgR_{1,1}=-\frac{3*(-a(t)-1)*\left(\left(a(t)_t\right)*\left(b(t)_t\right)-2*b(t)*\left(a(t)_{tt}\right)+2*\left(a(t)_{tt}\right)\right)}{4*(a(t)+1)*(b(t)-1)^2}$$

(%t61) $$hgR_{4,4}=-\frac{3*(1-b(t))*\left(\left(a(t)_t\right)*\left(b(t)_t\right)-2*b(t)*\left(a(t)_{tt}\right)+2*\left(a(t)_{tt}\right)\right)}{4*(a(t)+1)*(b(t)-1)^2}$$

form the calculated Einstein tensor as a matrix

```
(%i62) Gc:cric-hgR$
       matrixp(Gc);
```

```
(%o63) true
```

form the lhs of the 'spatial' field equation

```
(%i64) Gs:Gc[1,1]$
       ldisplay(Gs)$
```

(%t65) $$Gs=\frac{3*(-a(t)-1)*\left(\left(a(t)_t\right)*\left(b(t)_t\right)-2*b(t)*\left(a(t)_{tt}\right)+2*\left(a(t)_{tt}\right)\right)}{4*(a(t)+1)*(b(t)-1)^2}+\frac{\left(a(t)_t\right)*\left(b(t)_t\right)}{(2*b(t)-2)^2}-\frac{a(t)_{tt}}{2*b(t)-2}-\frac{\left(a(t)_t\right)^2}{(2*a(t)+2)*(2*b(t)-2)}$$

```
(%i66) Gss:facsum(Gs,operator(diff))$
       ldisplay(Gss)$
```

(%t67) $$Gss=-\left(2*a(t)*\left(a(t)_t\right)*\left(b(t)_t\right)+2*\left(a(t)_t\right)*\left(b(t)_t\right)-4*a(t)*b(t)*\left(a(t)_{tt}\right)-4*b(t)*\left(a(t)_{tt}\right)+4*a(t)*\left(a(t)_{tt}\right)+4*\left(a(t)_{tt}\right)+b(t)*\left(a(t)_t\right)^2-\left(a(t)_t\right)^2\right)/\left(4*(a(t)+1)*(b(t)-1)^2\right)$$

form the lhs of the 'temporal' field equation

```
(%i68) Gt:Gc[4,4]$
       ldisplay(Gt)$
```

(%t69) $$Gt=\frac{3*(1-b(t))*\left(\left(a(t)_t\right)*\left(b(t)_t\right)-2*b(t)*\left(a(t)_{tt}\right)+2*\left(a(t)_{tt}\right)\right)}{4*(a(t)+1)*(b(t)-1)^2}+\frac{3*\left(a(t)_t\right)*\left(b(t)_t\right)}{(2*a(t)+2)*(2*b(t)-2)}-\frac{3*\left(a(t)_{tt}\right)}{2*a(t)+2}+\frac{3*\left(a(t)_t\right)^2}{(2*a(t)+2)^2}$$

```
(%i70) Gtt:facsum(Gt,operator(diff))$
       ldisplay(Gtt)$
```

(%t71) $$Gtt=\frac{3*\left(a(t)_t\right)^2}{4*(a(t)+1)^2}$$