Elementary rudiments of GRTensorII:

This is a brief introduction only. For more interesting examples consult the demonstrations page on the Web.
For more detailed information consult the help pages and Release Notes.

The following (optional) line allows the worksheet to be re-executed.
> restart:

We now load the GRTensorII package:
> grtw():

GRTensorII Version 1.79 (R6)
6 February 2001
Developed by Peter Musgrave, Denis Pollney and Kayll Lake
Copyright 1994-2001 by the authors.
Latest version available from: http://grtensor.phy.queensu.ca/

The current environment can be checked as follows:
> groptions();
grOptionAlterSize = false
grOptionCoordNames = true
grOptionDefaultSimp = 8
grOptionDisplayLimit = 5000
grOptionLLSC = true
grOptionMetricPath = `e:/Grtii(6)/Metrics`
grOptionqloadPath = (not assigned)
grOptionTermSize = 100
grOptionTrace = false
grOptionTimeStamp = true
grOptionVerbose = false
grOptionWindows = true

grOptionDefaultSimp values: 0=None, 1=simplify, 2=simplify[trig],
3=simplify[power] 4=simplify[hypergeom], 5=simplify[radical],
6=expand, 7=factor, 8=normal, 9=sort, 10=simplify[sqrt]
11=simplify[trigsin]

We now use makeg to create a file (schwarz) which is the Schwarzschild exterior metric in curvature coordinates. Here we choose to enter the line element directly:
> makeg(schwarz);

Makeg 2.0: GRTensor metric/basis entry utility
To quit makeg, type 'exit' at any prompt.

Do you wish to enter a 1) metric [g(dn,dn)],
2) line element [ds],
3) non-holonomic basis [e(1)....e(n)], or
4) NP tetrad [l,n,m,mbar]?

makeg>2;
The Schwarzschild exterior metric in curvature coordinates; 

The values you have entered are:

Coordinates = \[r, \theta, \phi, t\]

Metric:

\[
\begin{bmatrix}
1 & 0 & 0 & 0 \\
\frac{2m}{r} & 0 & 0 & 0 \\
0 & r^2 & 0 & 0 \\
0 & 0 & r^2 \sin^2(\theta) & 0 \\
0 & 0 & 0 & -1 + \frac{2m}{r}
\end{bmatrix}
\]

"The Schwarzschild exterior metric in curvature coordinates;"

You may choose to
0) Use the metric WITHOUT saving it,
1) Save the metric as it is,
2) Correct an element of the metric,
3) Re-enter the metric,
4) Add/change constraint equations,
5) Add a text description, or
6) Abandon this metric and return to Maple.

We choose not to load it so as to demonstrate the loading procedure.

We now load the metric we have created:

> qload(schwarz);
Coordinates

\[ x^a = [r, \theta, \phi, t] \]

Line element

\[ ds^2 = \frac{dr^2}{1 - \frac{2m}{r}} + r^2 \, d\theta^2 + r^2 \sin^2(\theta) \, d\phi^2 + \left( -1 + \frac{2m}{r} \right) \, dt^2 \]

The Schwarzschild exterior metric in curvature coordinates;

To see what objects are predefined, load the help library (?grtensor; for information on the help system, readlib(griihelp); to load the help library) and do ?grt_objects;. Here we calculate and display (grcalcd) the covariant Ricci tensor \( R(dn,dn) \) and Kretschmann scalar \( \text{RiemSq} \).

\[ \text{grcalcd}(R(dn,dn), \text{RiemSq}) \]

\[ \text{CPU Time} = .201 \]

For the schwarz spacetime:

Covariant Ricci

\[ R(dn, dn) \]

\[ R_{ab} = \text{All components are zero} \]

Full Contraction of Riemann

\[ K = 48 \frac{m^2}{r^6} \]

Normally you would not want to display an object without further simplification (see below).

Let us use the definition facility (grdef) to define the covariant Einstein tensor:

\[ \text{grdef(`E{a b}:=R{a b}-Ricciscalar\ast g{a b}/2`)}; \]

This object is already defined. The new definition has been ignored.

Notice that the attempt was ignored because the 2-tensor "E" is predefined. (It is the "electric part" of the Weyl tensor, described in ?grt_operators;). Let's use "En":

\[ \text{grdef(`En{a b}:=R{a b}-Ricciscalar\ast g{a b}/2`)}; \]

Created definition for En(dn,dn)

Now let us ask for the mixed components. The definition for these is created automatically:

\[ \text{grcalcd}(En(dn,up)) \]

\[ \text{CPU Time} = .070 \]

For the schwarz spacetime:

\[ En(dn,up) \]

\[ En_{a}{}^{b} = \text{All components are zero} \]
Spacetimes which are implicitly defined are handled by the application of constraints. Consider the Kruskal-Szekeres metric:

```plaintext
> qload(krn);
```

**Default spacetime = krn**

**For the krn spacetime:**

**Coordinates**

\( x^a = [u, v, \theta, \phi] \)

**Line element**

\[
\begin{align*}
    ds^2 = & -16 \frac{m^2 (2 m - r(u, v))}{uv r(u, v)} \frac{du}{r(u, v)} \frac{dv}{r(u, v)} + r(u, v)^2 \frac{d\theta^2}{r(u, v)} + r(u, v)^2 \frac{d\phi^2}{r(u, v)} \\
\text{Constraints} = & -2 \frac{m (2 m - r(u, v))}{r(u, v) u} \frac{\partial u}{u} - 2 \frac{m (2 m - r(u, v))}{r(u, v) v} \frac{\partial v}{v}
\end{align*}
\]

**Null form of Kruskal metric**

Since objects calculated for this metric must be simplified (at least by the application of constraints) we calculate some objects without displaying them (grcalc):

```plaintext
> grcalc(R(dn,dn),RiemSq);
```

**CPU Time = .101**

To simplify the objects we apply gralter. If no routine is given, a menu of options is given as shown below. Refer to the help on gralter. The argument _ is shorthand for the previous objects (in this case R(dn,dn) and RiemSq).

```plaintext
> gralter(_);
```

**Component simplification of a GRTensorII object:**

( use ?name for help on a particular simplification routine)

Choose which routine to apply:

0) none
1) simplify() try all simplification techniques
2) simplify[trig] apply trig simplification
3) simplify[power] simplify powers, exp and ln
4) simplify[hypergeom] simplify hypergeometric functions
5) simplify[radical] convert radicals, log, exp to canonical form
6) expand()
7) factor()
8) normal()
9) sort()
10) simplify[sqrt,symbolic] allows \( \sqrt{r^2} = r \)
11) simplify[trigsin] trig simp biased to sin
12) Apply constraint equations
13) Apply constraints repeatedly
14) other user specified routine

Number of routine to apply (followed by ;) >

```plaintext
gralter>13;
```

13

We now display the objects:
For the krn spacetime:

Covariant Ricci
\[ R(\text{dn}, \text{dn}) \]
\[ R_{a \ b} = \text{All components are zero} \]
Full Contraction of Riemann
\[ K = 48 \frac{m^2}{r(u, v)^6} \]

Let us now do a basis calculation. We use a covariant basis here:

> qload(schwb);

Default spacetime = schwb

For the schwb spacetime:

Coordinates
\[ x(\text{up}) \]
\[ x^a = [r, \theta, \phi, t] \]

Basis inner product
\[ \eta(\text{bup}, \text{bup}) \]
\[ \eta^{(a) (b)} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]

Basis (covariant components)
\[ \omega_1(\text{dn}) \]
\[ \omega_1_a = \left[ \frac{\sqrt{r}}{\sqrt{r-2m}}, 0, 0, 0 \right] \]
\[ \omega_2(\text{dn}) \]
\[ \omega_2_a = [0, r, 0, 0] \]
\[ \omega_3(\text{dn}) \]
\[ \omega_3_a = [0, 0, r \sin(\theta), 0] \]
\[ \omega_4(\text{dn}) \]
\[ \omega_4_a = \left[ 0, 0, 0, \frac{\sqrt{r-2m}}{\sqrt{r}} \right] \]

Schwarzschild basis

The basis components are distinguished by the prefix b. For example, let's calculate the mixed basis components of the Weyl tensor:

> grcalc(C(bdn, bdn, bup, bup));

Created definition for C(bdn, bdn, bup, bup)
Created definition for \( \text{rot}(bdn,bup,bdn) \)
Created a definition for \( e(bdn,dn,pdn) \)

\[ CPU \ Time = .200 \]

\[ > \ \text{grdisplay}(\_); \]

For the schwb spacetime:
\( C(bdn,bdn,bup,bup) \)

\[ C_{(1)(2)}^{(1)(2)} = \frac{m}{r^3} \]

\[ C_{(1)(3)}^{(1)(3)} = \frac{m}{r^3} \]

\[ C_{(1)(4)}^{(1)(4)} = -2 \frac{m}{r^3} \]

\[ C_{(2)(3)}^{(2)(3)} = -2 \frac{m}{r^3} \]

\[ C_{(2)(4)}^{(2)(4)} = \frac{m}{r^3} \]

\[ C_{(3)(4)}^{(3)(4)} = \frac{m}{r^3} \]

The Newman-Penrose formalism can be used from a contravariant or a covariant null tetrad.
We start with the contravariant tetrad:

\[ > \ \text{qload(npschw)}; \]

Default spacetime = npschw

For the npschw spacetime:

Coordinates
\( x(up) \)

\( x^a = [ r, \theta, \phi, t ] \)

Basis inner product

\( \eta(bup, bup) \)

\[ \eta^{(a)(b)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \]

Null tetrad (contravaraint components)

\( \text{NPI}(up) \)

\[ f^a = \begin{bmatrix} \frac{1}{2} \sqrt{\frac{2}{r}} & 0, 0, \frac{1}{2} \sqrt{\frac{2}{r - 2m}} \\ \sqrt{\frac{2}{r - 2m}} & \sqrt{\frac{2}{r - 2m}} & \sqrt{\frac{2}{r - 2m}} \end{bmatrix} \]
Contravariant NPtetrad for the Schwarzschild metric in curvature coordinates

Let's calculate the Ricci scalars:

```plaintext
> grcalcd(RicciSc);
'Basis/tetrad related object definitions'
'Last modified 23 January 2001'

CPU Time = .401

For the npschw spacetime:

Ricci Scalar, Phi00
Φ00 = 0

Ricci Scalar, Phi01
Φ01 = 0

Ricci Scalar, Phi02
Φ02 = 0

Ricci Scalar, Phi11
Φ11 = 0

Ricci Scalar, Phi12
Φ12 = 0

Ricci Scalar, Phi22
Φ22 = 0

NPLambda := Ricci Scalar/24
NPLambda = 0
```

From a covariant tetrad let us calculate the Petrov type:

```plaintext
> qload(npcschw);

Default spacetime = npcschw

For the npcschw spacetime:

Coordinates
\( x^{(up)} \)
\[ x^a = [r, \theta, \phi, t] \]

*Basis inner product*
\[ \eta(b^{up}, b^{up}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \]

**Null tetrad (covariant components)**

\( \text{NP}(dn) \)
\[ l_a = \begin{bmatrix} \frac{1}{2} & r^{\sqrt{2}} & \frac{r}{(r - 2m)} \sqrt{r - 2m} & \frac{1}{2} (r - 2m) \sqrt{2} \end{bmatrix} \]

\( \text{NPn}(dn) \)
\[ n_a = \begin{bmatrix} \frac{1}{2} & r^{\sqrt{2}} & \frac{r}{(r - 2m)} \sqrt{r - 2m} & \frac{1}{2} (r - 2m) \sqrt{2} \end{bmatrix} \]

\( \text{NPm}(dn) \)
\[ m_a = \begin{bmatrix} 0, -\frac{1}{2} r^{\sqrt{2}}, -\frac{1}{2} I r \sin(\theta) \sqrt{2}, 0 \end{bmatrix} \]

\( \text{NPmbar}(dn) \)
\[ m_{bar} = \begin{bmatrix} 0, -\frac{1}{2} r^{\sqrt{2}}, -\frac{1}{2} I r \sin(\theta) \sqrt{2}, 0 \end{bmatrix} \]

*Covariant NP tetrad for the Schwarzschild metric in curvature coordinates*

> `grcalcd(Petrov);`

**CPU Time** = .090

*For the npcschw spacetime:*

**Petrov Type**

*Petrov Type = D (or simpler)*

Since we were not careful to ensure that the Weyl scalars were in fully simplified form, we ask for a report:

> `PetrovReport();`

*The conclusion 'Petrov type = D (or simpler)'*

*for the npcschw metric*

*was based on the following results:*

Weyl scalar \( \Psi_0 = 0 \)
Weyl scalar \( \Psi_1 = 0 \)
Weyl scalar \( \Psi_2 \) could not be evaluated to zero.

\[
\begin{align*}
\text{Weyl scalar } \Psi_3 &= 0 \\
\text{Weyl scalar } \Psi_4 &= 0
\end{align*}
\]

---

Therefore the metric is Petrov D (or simpler).

The quantities that could not be evaluated to zero are:

\[
Weyl \text{ scalar } \Psi_2 = -\frac{m}{r^3}
\]

Notice that we have 5 spacetimes active in this session (schwarz, krn, schwb, npschw, npcschw). The current default is npcschw (the last one loaded). We can calculate an object for any one of the spacetimes (e.g. the Weyl invariant W1R for krn)

\[
> \text{grcalc(W1R[krn]);}
\]

Scalar invariant library.

\[
\text{CPU Time} = .190
\]

\[
> \text{gralter(_,13);} \\
\text{Component simplification of a GRTensorII object:}
\]

Applying routine `Apply constraints repeatedly` to object W1R

\[
\text{CPU Time} = .020
\]

\[
> \text{grdisplay(_);} \\
\text{For the krn spacetime:}
\]

CM invariant Re(W1)

\[
W1R = 6 \frac{m^2}{r(u, v)^6}
\]

Let us load another implicit form of the metric

\[
> \text{qload(israel);} \\
\text{Default spacetime} = \text{israel}
\]

For the israel spacetime:

Coordinates

\[
x^a = [u, w, \Theta, \Phi]
\]

Line element

\[
ds^2 = \frac{1}{2} \frac{w^2}{m \, r(u, w)} \, du^2 + 2 \, du \, dw + (u, w)^2 \, d\Theta^2 + (u, w)^2 \sin(\Theta)^2 \, d\Phi^2
\]

Constraints

\[
\begin{bmatrix} 
1 \frac{u \, w}{4} \\
(r(u, w) = 2 m + \frac{1}{m})
\end{bmatrix}
\]

Israel coordinates (Phys. Rev. 143,1016)
We define a differential invariant:

\[ \text{DiRiem} := R_{abcd} R_{cdab} R_{eaab} R_{ebcc} \]

Created a definition for \( R_{dn, dn, up, up, cdn} \)
Created a definition for \( R_{dn, dn, up, up, cdn} \)
Created definition for \( R_{dn, dn, up, up, cup} \)
Created definition for \( \text{DiRiem} \)

\[ \text{grcalc}(\text{DiRiem}); \]

\[
\text{CPU Time } = 0.250
\]

\[ \text{gralter}(\_ , 13); \]

Component simplification of a GRTensorII object:

Applying routine `Apply constraints repeatedly` to object \( \text{DiRiem} \)

\[
\text{CPU Time } = 0.020
\]

\[ \text{grdisplay}(\_); \]

\[
\text{For the israel spacetime:}
\]

\[ \text{DiRiem} = 47185920 \frac{m^{10} u w}{(8 m^2 + u w)^9} \]

Notice that the invariant vanishes at \( u=0 \) \((r(0,w)=2m)\), a property of this object that holds more generally.

Whereas we have interactively defined this invariant here, it is also predefined in the differential invariants library \text{dinvar} as \( \text{diRiem} \). Do \text{grlib} (\text{dinvar}); to load the library.

We can also change the default spacetime with \text{grmetric}. Let's go back to \text{schwarz} and CREATE a null tetrad:

\[ \text{grmetric(\text{schwarz});} \]

Default metric is now \text{schwarz}.

\[ \text{nptetrad([r,t]);} \]

The metric signature of the \text{schwarz} spacetime is +2.
In order to create an NP-tetrad, the signature of \( g(dn,dn) \) will be changed to −2.
Continue? \( 1=\text{yes} \) [default], other=\text{no} : \n
\[ \text{nptetrad}>1; \]

1

The situation here (in Schwarzschild) is unusually simple. The basis vectors would normally require further simplification for efficient use.