COMPUTER ALGEBRA IN RELATIVITY

1. SHEEP AND CLASSI

SHEEP and CLASSI are available via both ONID and COSINe (by first logging in to `shell.onid.oregonstate.edu` and `frontend.science.oregonstate.edu`, respectively). To start the LISP-based packages SHEEP and CLASSI, first define aliases (short names) for the needed commands by typing the command `source ~drayt/sheep/.shprc` for ONID and `source ~tevian/sheep/.shprc` for COSINe. (You can add this command to your `.cshrc` file if you like.) Now type `sheep` to start SHEEP, or `classi` to start CLASSI. To end either program, type `(quit)` .

> The above initialization files assume you are running `tcsh` . If they give error messages, you are probably running `bash` ; try replacing `.shprc` with `.shprc.sh` .

SHEEP and CLASSI were written specifically for relativity; one of their biggest advantages is that there is a very large library of source files containing various spacetime metrics.

a) SHEEP

SHEEP is designed to do tensor computations using components in a coordinate basis. It is very good at the sort of tensor manipulations needed to, say, compute the curvature tensor of a given metric, but very bad at doing algebra.

After starting SHEEP, it is useful to set some switches by typing `(pon ptevar)` , `(pon nozero)` , and `(on diagonal)` .

> The first two of these control how output is printed, for instance the second of these specifies that only nonzero tensor components should be printed. The last specifies that metrics are assumed to be diagonal. You can avoid typing these repeatedly by putting them in a file called `sheep.ini` in your home directory.

For example, try the following commands:

```
(dimension 2)
(vars h p)
(load cord)
(rpl gdd)        (Type first  r^2$  and then  r^2*sin(h)^2$  when prompted.)
(funs (r))
(wmake gamu)
(wmake ric)
(wmake rscl)
```

After setting the dimension (default is 4) and the names of the variables (optional; note the use of h for θ and p for ϕ), this loads the coordinate package `cord`. Tensor components are assigned with the "replace" command `rpl` , after which it is necessary to specify the functional dependence of r. (You could make r a function of θ by typing `(funs (r h))` .) Tensors are computed with the `make` command and printed to the screen with the `write` command, which can be combined as `wmake` ; all of these commands must be enclosed within parentheses and given an appropriate argument. A list of predefined tensors can be obtained with the command `(help tensors)` .

Here is a more complicated example, the Schwarzschild black hole, which illustrates the tricks which must be used to avoid polynomial division.

```
(vars t r h p)
(load cord)
(rpl f)        (Then type  1-2m/r$ .)
```

```
(rpl gdd)      (Then type  -f$ ,  1/f$ ,  r^2$  and  r^2*sin(h)^2$ .)
(funs (m) f)
(newsul ricsul)      (Then type  f$ ,  :f$ .)
(usesul ricsul ricc riemc)
(wmake rie)
(wmake ein)
```

These commands define the metric in terms of an explicitly given function $f = 1-2m/r$, but do the computation in terms of f and its symbolic derivatives, only substituting for f in the final expressions (This allows division by f to replace the much harder division by $1-2m/r$.) The Schwarzschild metric can also be loaded with the command

```
(load "schwar.crd")
```

b) CLASSI

CLASSI is a macro package written in SHEEP designed to do tensor computations using components in a (generalized) "orthonormal" frame, i.e. with respect to a basis of vector fields whose dot products are constant, typically 1's and 0's. (This framework includes bases in which one or more vectors are null.) The Riemann tensor has many fewer independent components in an orthonormal frame than in a coordinate basis! (One way to see this is to compare the quite small group of orthogonal transformations with the quite large group of arbitrary coordinate transformations.)

Since CLASSI is essentially a souped up version of SHEEP, it is again useful to set the above switches by typing `(pon ptevar)` , `(pon nozero)` , and `(on diagonal)` . These commands can be inserted instead into a file `classi.ini`.

For example, try the following commands:

```
(dimension 2)
(vars h p)
(rpl izud)      (Then type  r$ ,  0$ ,  0$  and  r*sin(h)$ .)
(cartesian iframe)
(funs (r))
(wmake ric)
(wmake rscl)
```

The tensor `izud` (don't ask...) contains the components of the (dual) orthonormal frame with respect to the coordinate dual basis $\{d\theta, d\phi\}$; there is no diagonal switch in this context. The next command indicates that this is a Euclidean orthonormal frame; a Lorentzian orthonormal frame would be indicated by `(lorentz iframe)` . Finally, note that the components of, say, the Ricci tensor are different from before — because the basis is different — but that a scalar, such as the Ricci scalar, is basis independent.

The main reason CLASSI was written was to solve the *equivalence problem* of determining when two metrics are really the same, i.e. are merely coordinate transformations of each other. For instance, CLASSI is able to determine whether a spacetime is spherically symmetric even when the metric is given in bizarre coordinates! For further information, see me.

The original problem tackled in the 1960's by the precursor of SHEEP and CLASSI was to calculate the Einstein tensor for an important spacetime with gravitational radiation, known as the Bondi metric. The original calculation by hand had taken 6 months; the computer took only a few minutes — and found 4 mistakes in the published computation. This program and computation formed the heart of the PhD thesis of none other than textbook author Ray d'Inverno! You can reproduce this computation — in a matter of seconds — by loading the Bondi metric into CLASSI with the command `(load "bondi.lor")` , and typing `(wmake ein)` .