GRTensorIII

GRTensorIII Release 1.10
For Maple 2016

G. Hypersurfaces, Junctions and Shells

Peter Musgrave
Denis Pollney
Kayll Lake

Jan 2017

Contents

Queen's University at Kingston, Ontario

GRTensor III supports the definition and evaluation of hyper-surfaces within a spacetime, and the junction of two spacetimes along hyper-surfaces. It facilitates the evaluation of the Darmois-Israel matching conditions and the determination of the hyper-surface evolution and properties of the shell (if one exists). Hyper-surfaces and junctions for timelike, spacelike and null shells are supported. This functionality was initially provided in the GRJunction package. It is now part of GRTensorIII. The commands from that package have been re-designed to allow for direct (non-interactive) definitions that allow worksheet recalculation in a natural way.

There are numerous text book treatments and review papers that describe the junction formalism in detail. This booklet does not attempt to cover this material. We present a personal choice of references to establish notation and object definitions. In most things we are guided by "A Relativist's Toolkit" by Eric Poisson [1]. Some of the issues of developing computer algorithms for hypersurface and junctions are described in the original GRJunction papers [2, 3].

GRTensorIII software is available with documentation and examples from https://github.com/grtensor/grtensor. Source code is available from https://github.com/grtensor/grtensor3src.

1 Hypersurfaces

A hyper-surface Σ in a spacetime M is a 3-dimensional sub-space of M. The co-ordinates of Σ are in general distinct from those in M. In practice it is common to use some of the same co-ordinates in both M and Σ. Here we follow the definitions and conventions in [1] and label the co-ordinates on Σ as y^a, using Roman indices on tensorial objects. The co-ordinates on M are x^α and Greek indices are used.

A hypersurface can be defined by either a set of relations of the form $x^\alpha = f(y^a)$ or by a scalar function $\Phi(x^\alpha)$ that is zero on Σ.

1.1 Timelike and Spacelike Surfaces

The basic vectors of the surface in M are defined by:

$$e^\alpha_a = \frac{\partial x^\alpha}{\partial y^a}$$

The intrinsic metric g_{ab} on Σ is defined by:

$$g_{ab} = g_{\alpha\beta} e^\alpha_a e^\beta_b$$

Here we deviate slightly from the nomenclature of [1] retaining the object `g(dn,dn)` for the intrinsic metric instead of h_{ab}.

The normal to the hypersurface can be specified explicitly or derived as the gradient of a scalar definition $\Phi(x^\alpha) = 0$ of the surface. GRTIII allows either approach. If a scalar surface definition is provided the normal is defined by:

$$n_\alpha = \frac{\epsilon \Phi_{,\alpha}}{|g^{\mu\nu} \Phi_{,\mu} \Phi_{,\nu}|}$$

 GRTensorIII development has been supported by the Natural Science and Engineering Research Council of Canada and the Advisory Research Committee of Queen's University.

where $\epsilon = 1$ for a timelike surface and -1 for a spacelike surface.

The extrinsic curvature `K(dn,dn)` of the surface is determined by:

$$K_{ab} = \nabla_\alpha n_\beta e^\alpha_a e^\beta_b = n_\alpha \left(\frac{\partial^2 x^\alpha}{\partial y^a \partial y^b} + \Gamma^\alpha_{\mu\nu} \frac{\partial x^\mu}{\partial y^a} \frac{\partial x^\nu}{\partial y^b} \right)$$

GRTensorIII uses the second definition since it is defined in terms that can be evaluated directly in Σ.

The contracted forms of the Gauss-Codazzi equations are provided. These are:

$$-2\epsilon G_{\mu\nu} n^\mu n^\nu = R + \epsilon \left(K^{ij} K_{ij} - K^2 \right)$$

and

$$G_{\mu\nu} e^\mu_a n^\nu = K^b_{a|b} - K_{,a}$$

These equations using $G_{\mu\nu}$ are identities and serve as useful validation check for the package implementation. Using them to examine the physics of a spacetime is done by using Einstein's equation and providing a specific form for $T_{\mu\nu}$ by using `grdef` to define `T(dn,dn)` and provide a phenomenology. (See the `hyper_frw_constraint` worksheet for an example that follows section 3.6.2 in [1]).

$$16\pi T_{\mu\nu} n^\mu n^\nu = R + \epsilon \left(K^{ij} K_{ij} - K^2 \right)$$

and

$$8\pi T_{\mu\nu} e^\mu_a n^\nu = K^b_{a|b} - K_{,a}$$

Note that these equations mix the contraction of objects in M defined in the co-ordinates of M into an equation defined on Σ. By default the calculation will not apply the equations restricting the objects in M to Σ. These relations are automatically added to M as constraints and they can be applied by using the `gralter` command with the `cons` argument.

The GRTensorIII objects relating to timelike/spacelike hypersurfaces are listed in Table 1 and 2.

Objects defined on M:

GRTensorIII name	**Common representation**
`g(dn,dn)`	g_{ab}
`K(dn,dn)`	K_{ab}
`Ksq`	$K_{ab}K^{ab}$
`trK`	K_a^a
`C1GeqnRHS`	$R + \epsilon\left(K^{ij}K_{ij} - K^2\right)$
`C1Geqn`	$-2\epsilon G_{\mu\nu}n^\mu n^\nu = R + \epsilon\left(K^{ij}K_{ij} - K^2\right)$
`C1Teqn`	$16\pi T_{\mu\nu}n^\mu n^\nu = R + \epsilon\left(K^{ij}K_{ij} - K^2\right)$
`C2GeqnRHS(dn)`	$K^b_{a\vert b} - K_{,a}$
`C2Geqn(dn)`	$G_{\mu\nu}e^\mu_a n^\nu = K^b_{a\vert b} - K_{,a}$
`C2Teqn(dn)`	$8\pi T_{\mu\nu}e^\mu_a n^\nu = K^b_{a\vert b} - K_{,a}$

Table 1: GRTensorIII objects defined on Σ for timelike/spacelike surfaces

GRTensorIII name	**Common representation**
`Gnn`	$G^{\mu\nu}n_\mu n_\nu$
`Gxn(dn)`	$G_{\mu\nu}\frac{\partial x^\mu}{\partial y^a}n^\nu$
`Tnn`	$T^{\mu\nu}n_\mu n_\nu$
`Txn(dn)`	$T_{\mu\nu}\frac{\partial x^\mu}{\partial y^a}n^\nu$
`n(dn)`	n_α, normal to the surface
`xform(up)`	$x^\alpha(y^a)$, definition of the surface

Table 2: GRTensorIII objects defined on M for timelike/spacelike surfaces

1.2 Null Surfaces

The null surfaces in GRTensorIII follow the presentation in Section 3.11 of [1]. This adaptation of the Barrabes-Israel forumulation [4] was first presented in [5]. The null case is distinct because the normal vector k^μ is also tangent to the surface. The coordinates on the surface are (λ, A, B) and the tangent vectors are:

$$k^\mu = \left(\frac{\partial x^\mu}{\partial \lambda}\right)_{\theta^A}, e^\mu_A = \left(\frac{\partial x^\mu}{\partial \theta^a}\right)_\lambda$$

The surface metric itself is degenerate and described by a two metric:

$$\sigma_{AB} = g_{\mu\nu} e^\mu_A e^\nu_B$$

And this two tensor acts as a metric on Σ

To compute transverse curvature a vector N_α orthogonal to Σ is required. It must satisfy:

$$N^\alpha N_\alpha = 0, N_\alpha k^\alpha = 0, N_\alpha e^\alpha_A = 0$$

The transverse curvature is then

$$C_{ab} = -N_\alpha e^\alpha_{a;b} e^\beta_b$$

GRTensorIII name	**Common representation**
`sigma(dn,dn)`	σ_{ab}
`C(dn,dn)`	C_{ab}

Table 3: GRTensorIII objects defined on Σ for null surfaces

GRTensorIII name	**Common representation**
`k(up)`	k^α
`eA(up)`	$e^\alpha_A, A = y^2$
`eB(up)`	$e^\alpha_B, B = y^3$
`kdotk`	$k^\alpha k_\alpha$
`kdotN`	$k^\alpha N_\alpha$
`NdotN`	$N^\alpha N_\alpha$

Table 4: GRTensorIII objects defined on M for null surfaces

1.3 The `hypersurf` command

A hypersurface is defined in GRTensorIII with the `hypersurf` command. This command takes a series of `param=value` arguments to specify the surface. For example (putting each parameter on it's own line for readability):

```
hypersurf(shellOut,
type = timelike,
coord = [theta, phi, tau],
xform = [r = R(tau), theta = theta, phi = phi, t = T(tau)],
ndn = [diff(T(tau), tau), 0, 0, -(diff(R(tau), tau))],
);
```

The parameters supported by the package are: Note that the corder of the coordinates in

Parameter	**Value**	Example
`type`	type of surface	`timelike`, `null`
`coord`	y^a as list	`[tau,theta,phi]`
`xform`	$x^\alpha(y^a)$ as list	`[t=T(tau),theta=theta,phi=phi,r=R(tau)]`
`ndn`	n_α as list	`[1,0,0,0]`
`nup`	n^α as list	`[1,0,0,0]`
`Ndn`	N_α as list	`[1,0,0,0]`
`Nup`	N^α as list	`[1,0,0,0]`
`surf`	$\Phi(x^\alpha)$	`r-R(tau)`

Table 5: `hypersurf` command parameters

the `xform` parameter must match the co-odinates in M as must the components of any normal or lapse vector that is specified.

In the case of a null surface the co-ordinates must be in the order (λ, A, B) using the notation of [1].

1.4 Restricting components to Σ

The formalization of hypersurface definition in a computer algebra system such as Maple highlights the fact that it is common in the literature to be inconsistent in the application of the surface definition where doing so complicates the expressions or makes them less intuitive.

For example, a null surface metric may be:

$$\sigma_{AB} d\theta^A d\theta^B = \lambda^2 \left(d\theta^2 + sin^2\theta d\phi^2\right)$$

but the extrinsic curvature is written as

$$C_{AB} = \frac{1}{2r}\sigma_{AB}$$

where the surface definition includes $r = -\lambda$.

In some cases the direct substitution of the $x^\alpha(y^a)$ is desired, but not in all cases. In addition there are cases where a direct substitution will cause issues with Maple. Consider a term:

```
diff(f(r), r)
```

in the case where $r = -lambda$. Evaluating this in Maple:

```
eval(subs(r = -lambda, diff(f(r), r)));
Error, invalid input: diff received -lambda, which is not valid for its 2nd argument
```

Maple expects the argument to a `diff` function to be a name, not an expression.

GRTensorIII will restrict components in M to Σ as follows:

1. Determine a list of direct substitutions: the $x^\alpha(y^a)$ where the RHS is one of the y^a
2. Map all derivatives that involve a non-direct substitution to the Maple `Diff` function (the inactive form of the `diff` function).
3. Freeze all the inactive `Diff` functions so that a substitution of the $x^\alpha(y^a)$ will not be applied
4. Substitute the $x^\alpha(y^a)$
5. Un-freeze the `Diff` functions

In some cases the end result will have inactive `Diff` functions that can be "reactivated". This can be done using Maple's convert command using `grmap`. For example in the `hyper_frw_constraint` worksheet

```
grmap(K(dn,up), convert, 'x', diff);
```

is used to convert the inactive `Diff` to `diff`.

The $x^\alpha(y^a)$ relations specified by the `xform` parameter are automatically added to the constraints for the metric. They can be inspected using `grconstraint` and applied via e.g.

```
gralter(K(dn,up), cons);
```

1.5 Working with more than one metric

GRTensorIII allows multiple metrics in a session. Here we highlight several ways of calculating and display objects in a session with more than one metric.

To change the default metric, use the `grmetric` command (see `?grmetric`).

To calculate or display an object in a non-default metric, use the metric name in the argument list, or place the metric name in square braces. Eg.

```
grcalc(schw, R(dn,dn));
grcalc(R[schw](dn,dn));
```

2 Junction Conditions

Joining two spacetimes $M^{\pm}$ requires we work with two surfaces $\Sigma^{\pm}$. The hypersurfaces are identified and then the discontinuity in the intrinsic properties of the hypersurfaces are evaluated. i.e. $[g_{ab}]$ and $[K_{ab}]$, where $[x]$ indicates the value on Σ^{+} minus the value on Σ^{-} (or $M^{\pm}$ restricted to Σ in some cases).

For non-null shells, the surface stress-energy tensor `S3(dn,dn)` is defined as:

$$S_{ab} = -\frac{\epsilon}{8\pi}\left([K_{ab}] - [K]\, g_{ab}\right)$$

with $\epsilon = n^{\alpha} n_{\alpha}$.

The command `join` is used in GRTensorIII to make this identification.

```
join(houtside, hinside);
```

where `houtside` and `hinside` are hypersurfaces defined using the `hypersurface` command. This command links the $M^{\pm}$ and $\Sigma^{\pm}$ by defining the `join` object for each metric as the name of the metric it is identified with. For example the join of `houtside` is `hinside`. With `houtside` as the default metric the discontinuity in K_{ab} can be determined using the `Jump` operator:

```
grcalc(Jump[K(dn,dn)]);
```

It is often the case that careful simplification of K_{ab} in each hypersurface metric prior to evaluating the jump is worthwhile.

The objects defined for junctions are:

GRTensorIII name	**Common representation**	Surface Type
`S3(dn,dn)`	S_{ab}	timelike/spacelike
`j_null`	$j^{A} = \frac{1}{8\pi}\sigma^{AB}\,[C_{\lambda B}]$	null
`mu_null`	$\mu = -\frac{1}{8\pi}\sigma^{AB}\,[C_{AB}]$	null
`p_null`	$p = -\frac{1}{8\pi}\,[C_{\lambda\lambda}]$	null

Table 6: GRTensorIII objects defined on Σ for junctions

3 Examples

The directory `worksheet/junctions` contains demonstrations of the use of the package.

In all but one case, the results from the reference are easily reproduced. The exception is the derivation of the equation of motion for a collapsing dust shell in a Schwarzschild background. In this case the resulting differential equations are not very accesible. The grouping of terms K_{ab} into intermediate functions in [1], 3.9 allows a very transparent progression to the solution. This is where human skill is required to augment the computer assisted calculations!

Worksheet	**Description**	Source
`hyper_frw_constraint`	Hypersurface in FRW	[1], 3.6.2
`os_dust_collapse`	Oppenheimer-Snyder junction FRW to Schw	[1], 3.8
`null_accreting`	Accreting Kerr BH to order a	[1], 3.11.7
`null_accreting_k`	Accreting Kerr BH to order a	[1], 3.11.7
	(check k^α N^α)	
`null_cosmo`	Null junction cosmology phase transitions	[1], 3.11.8
`null_shell_implosion`	Null shell implosion	[1], 3.11.6
`null_shells`	Spherical null shells	[5], 2nd application
`schw_shell`	Spherical dust shell collapse	[1], 3.9
`spinning_shell`	Spherical shell as Kerr source (order a)	[1], 3.10

Table 7: Example Worksheets for Hypersurfaces and Junctions

References

[1] Eric Poisson. *A Relativist's Toolkit*. Cambridge University Press, Cambridge, 2004.

[2] Peter Musgrave. Junctions and thin-shells in general relativity using computer algebra I: The Darmois-Israel formalism. *Class. Quantum Grav.*, 13:1885–1900, 1996.

[3] Peter Musgrave. Junctions and thin-shells in general relativity using computer algebra II: the null formalism. *Class. Quantum Grav.*, 14:1285–94, 1996.

[4] C. Barrabes and W. Israel. Thin shells in general relativity and cosmology: the lightlike limit. *Phys. Rev. D*, 43:1129–42, 1991.

[5] Eric Poisson. A reformulation of the Barrabes-Israel null-shell formalism. 2002. https://arxiv.org/abs/gr-qc/0207101v1.