University of Southampton

Algebraic and numerical techniques in general relativity:

The classification of spacetimes via the Cartan-Karlhede method, and Cauchy-Characteristic matching for numerically generated spacetimes

by

Denis Pollney

Submitted for the degree of Doctor of Philosophy

Faculty of Mathematical Studies

February 2000

Abstract

This thesis concerns two distinct areas of research (i) the development of a practical set of methods for the classification of spacetimes in general relativity, and (ii) the numerical solution of the vacuum Einstein equations on null hypersurfaces.

The first part examines the Cartan-Karlhede method for determining a unique classification of algebraically distinct spacetimes. A key aspect of this method is the establishment of a set of 'standard forms' for symmetric spinors. A set of standard forms is developed, with an emphasis on the need for consistency, and reduction of computational complexity. The Cartan-Karlhede method has been incorporated into a set of programs for the computer algebra system Maple and form a general set of tools for the use of relativists. Certain inherent difficulties with the Cartan-Karlhede methods, such as the need to determine the roots of high order polynomials, are identified, along with potential alternative methods for handling these difficulties.

The second part of the thesis details the development of a Cauchy-characteristic matching (CCM) code in axisymmetry. In the CCM technique, a spacetime is evolved on two separate grids with information passed between the two. The advantage of this method is the ability to use well-developed Cauchy codes in the interior (where characteristics would tend to develop caustics) and a characteristic region which extends to null infinity, alleviating the need for artificial boundary conditions. The code which is being developed passes information in both directions across the boundary. A condition at null infinity ensures that Bondi-type slicing of the spacetime is maintained, with the advantage that the mass and news functions are thus easily identifiable.

A full description of the evolution systems in both the Cauchy and characteristic regions is given. The interface is implemented along a single $r = constant$ surface to avoid difficulties arising from interpolation between the grids over a region. Difficulties arise in coordinating the evolution in each region so that the required information is provided at the boundary when it is needed, however for the given system of equations they can be surmounted in a manner consistent with the overall evolution scheme.

Acknowledgements

The subject matter in covered by this thesis is broad in its scope, ranging from 2-spinor calculus, to applied numerical techniques, to asymptotic expansions of gravitational wave equations. I do not claim to be an expert in any of these fields, in fact I sometimes see myself in the role of 'jack of all trades, master of none.' But in putting this work together, I have been grateful to be able to call on the varied expertise of the many others who have been so helpful whenever I have asked for advice.

In particular, I would like to thank Jim Skea, of the University of Rio de Janeiro, for his help in planning and implementing the set of classification algorithms that make up the first part of this thesis.

I'd also like to thank Kayll Lake, my previous supervisor at Queen's University, for giving me the opportunity to get involved in the GRTensor project. As much as I complain, fixing Maple bugs is often a welcome distraction from numerical cookery in Fortran.

The Cauchy code developed for the second half of this thesis is largely the work of Mark Dubal, for whom I am very grateful for the effort he devoted to developing this complicated and crucial piece of the CCM puzzle.

I would like to thank Chris Clarke for his patient and precise work on the interface on both the theoretical and coding sides, and for his persistence with it when many other interesting problems called for his attention.

The Southampton relativity grad students during my time here, John Goodwin, Paul Lambert, and Paolo Matteucci, Robert Sjödin, Uli Sperhake, and Rhiannon Williams have been a great source of friendship and advice during my time here, and a special thanks to the Jonathan Wilson for keeping the group (or at least our machines) up and running.

James Vickers has contributed his theoretical knowledge and intuition to every aspect of this project, and I wish I were able to code new ideas as fast as he thinks them up.

And finally, I'd like to thank my supervisor, Ray d'Inverno, who has headed the Southampton CCM effort and been an invaluble source of support, encouragement, ideas and optimism at times when these things were very hard to come by, and especially for giving me the opportunity to come to Southampton in the first place. In a scientific world that can often seem quite specialised, the breadth of experience that I've been exposed to over my time in Southampton is something that I'm sure will benefit me for the rest of my career and for which I am very grateful.

Denis Pollney
Potsdam, February 2000.

Contents

Part I

The classification of spacetimes in general relativity

1

Introduction: The Cartan-Karlhede method

When Einstein first presented his geometrical theory of the gravitational field, general relativity, he was dismayed to think that the theory would find little practical use. The reason for his pessimism was the complicated non-linear nature of his field equations, a property which he thought would render exact solution virtually impossible. Within a year of its publication, however, a most significant solution had already been discovered, namely the spherically symmetric 'black hole' of Schwarzschild. Since then, hundreds of exact solutions have been found, some more and some less physically significant, and the study of their properties has become an industry in itself.

The theory of general relativity is a geometrical theory in that its solutions describe surfaces of four dimensional spacetime. The conventional way of representing solutions is via a line-element or a set of basis vectors, and crucial to this form of representation is the assignment of labels to points on the surface, which amounts to the choice of coordinates.

Early on it was recognised that the freedom to choose coordinates could be both a benefit and a complication. On the one hand, coordinates can be fixed in such a way as to take advantage of special symmetries of certain configurations of Einstein's equations. In this way, the equations can be simplified dramatically and often this is the most powerful tool available in finding exact solutions.

On the other hand, the choice of a particular set of coordinates can lead to properties which only seem to cloud the analysis by introducing effects which are not at all physical but rather due to some limitation in the method used for labelling points on the spacetime. A significant example of this is the coordinate singularity at $r = 2m$ of the Schwarzschild solution. This was long thought to be a problem with the underlying

theory, and only later shown to be simply a result of the fact that Schwarzschild's choice of coordinates were not appropriate for describing the entire solution.

Another problem arises when two solutions are found by different means, but possessing the same symmetries, and in different coordinates. The question arises, what are the relations between two such solutions, and in particular, is it possible that they could be equivalent solutions simply expressed in different coordinates? One way to answer this question is to look for a coordinate transformation relating the two solutions. If one can be found, then it is clear that the solutions are equivalent. However, if a coordinate transformation can not be found, there is the problem of proving whether or not one exists.

As might be expected, such a proof turns out to be a non-trivial exercise, at least in practice. The theory behind such proofs has a long history, beginning with Christoffel (1869), who first proposed a comparison of curvature tensor components and their derivatives. The most significant step forward was provided by Cartan (1946), who was able to show that in 4 dimensions the required components are those of the Riemann tensor and its first ten derivatives. With these in hand, one has in principal, all of the necessary information. The equivalence problem then reduces to finding a set of coordinate relations between these components calculated for the two spacetimes.

The great problem with Cartan's method comes in its implementation, since even for algebraically simple spacetimes it can involve the computation of enormous amounts of tensor components. A number of attempts have been made to improve on its practical implementation, notably by Brans (1965) and Karlhede (1980). Brans' contribution was to suggest that the $SL(2,\mathbb{C})$ freedom inherent in the coordinates (a result of our freedom to change the velocity and orientation of the coordinate frame) can be removed by fixing the tensor components to a 'canonical' frame which is determined by the algebraic symmetries of the solution. Karlhede formalised this idea and improved upon it by suggesting that the frame should be fixed using the low order derivatives of the Riemann tensor, in this way reducing the number of derivatives which eventually need to be calculated. In this way, the number of theoretically necessary derivatives was reduced from 10 to 7 and the total number of components which needed to be calculated was reduced from some 27 962 020 to 436 900, a significant improvement, though still a daunting task.

The final theoretical improvement to the basic algorithm was provided by MacCallum and Åman (1986), who used identities satisfied by the Riemann tensor and its derivatives to arrive at a minimal set of independent components which need to be calculated. For

Karlhede's method, they arrived at an upper bound for the number of components as 3156, which finally brings the problem into the realm of tractability, at least by modern computer algebra systems. Specific algebraic types of solution have also been examined, for instance by Collins et al. (1990), and Collins (1991). Together these papers show that in fact it is only a very rare situation in which the full number of derivatives of the Riemann tensor need to be calculated. In practice it seems that one rarely needs to go past the third derivative before Karlhede's method is completed. (The first example to require fourth order classification was discovered by Koutras (1992); see also Skea (1997a).) Even so, the computations required to calculate the individual tensor components are often fairly labour intensive. Brans (1965) suggested that computer algebra systems might help in this regard. Åman (1986) was the first to develop such a system, CLASSI, which he based on Karlhede's algorithm.

CLASSI has been used with great success in fully classifying a large number of exact solutions. However it suffers from some shortcomings. The most serious of these is that it is unable to perform certain algebraic operations, such as polynomial division, which are extremely useful in tensor calculations. It is also weak in performing certain simplifications, such as factorisation and applying trigonometric identities. In some cases such simplifications can be crucial to the calculation of tensor components, for when they are not applied the individual expressions can expand to a size which will fill the memory of the computer on which it is running. The lack of advanced simplification facilities results from the fact that CLASSI is implemented as an addition to the program SHEEP, a computer algebra system which specialises in tensor computation but has not been optimised for general algebraic calculations. Since SHEEP has been implemented as a stand-alone package using the LISP programming language, it must contain its own routines for carrying out polynomial simplifications, in general a non-trivial task which would require many times more effort to implement than the actual tensor calculations themselves.

To some extent, SHEEP (and hence CLASSI) are able to avoid these problems through the use of powerful facilities for substitution of expressions. By specifying the substitutions correctly, the user is effectively able to instruct the computer which simplifications to apply when, often leading to a very quick and compact result. The problem with this method, however, is that it often requires a great deal of fine-tuning before the result is obtained. The user must step through the calculation and recognise where substitutions might be helpful, then code these substitutions into an input file. It is not always obvious if such substitutions will lead to an optimal form of the solution.

Many modern commercial computer algebra systems, which did not exist at the time of CLASSI's development, have built in facilities for carrying out simplifications of large polynomials. The algorithms are often automatic, requiring little intervention on the part of the user, and great efforts have been made to see that the algorithms are in some way optimised. It would be useful, then, to be able to use such a general system as an underlying tool for a tensor computation package, as this frees the programmer of the tensor package from needing to implement all of the basic algebraic operations. A large number of independent tensor packages have been written for the major computer algebra systems Maple, Mathematica, Macsyma and Reduce.[1] Notably, the package GRTensor, (see Pollney et al. (1996)) provides extensive tensor calculation tools for relativists working within the Maple computer algebra system.

This thesis details an implementation of the Cartan-Karlhede method within Maple, developed by the author in collaboration with Jim Skea (UERJ, Brazil) and Ray d'Inverno (Southampton). The study of the applicability of standard forms carried out in the next chapter, the algorithms presented, and the computer algebra software designed for the required operations on spinor components are the original work of the author. The procedures described aim at creating an efficient, powerful, and usable set of computational tools for the manipulation of 2-spinor components in relativity.

1.1 The spinor formalism

To answer the question of how we can decide whether two spacetimes are equivalent, we must first ask what information do we have regarding a spacetime, and how can it be used. Generally, this information comes in the form of curvature tensor components, whether specified in a particular coordinate system via a metric, or in terms of a given set of basis vectors (frame). The difficulty arises from the fact that the individual components of the curvature tensors vary under changes of coordinates as well as rotations of the underlying frame. A pair of spacetimes will be called *equivalent* if a coordinate transformation exists which maps any tensor component calculated in the one spacetime onto the corresponding tensor component in the other.

In fact, from this definition, the solution to the equivalence problem is already apparent. Namely, we examine the action of an arbitrary coordinate transformation on some number of tensor components in the one spacetime, and set these equal to the

[1] An incomplete listing of publicly available tensor packages can be found at `http://astro.queensu.ca/~grtensor/`.

components of the second spacetime to solve for the coordinate transformation.

The question then arises as to how many, and in particular which, tensor components will be sufficient to guarantee the consistency of such a solution, if it exists. In the following sections, the general solution, due to Cartan, to this problem is presented. Important improvements arising from specialising the frame were later introduced by Karlhede, and finally MacCallum and Åman were able to determine a necessary and sufficient set of tensor components required to establish equivalence.

The method of Cartan-Karlhede requires only that the components of the curvature be represented in terms of a frame with fixed metric components. We will find, however, that certain symmetries among components of the curvature tensor are most easily seen when they are expressed in a particular frame formalism, namely the 2-spinor approach of Penrose.

Spin transformations

Consider standard Minkowski space, $\mathbb{M}$, with coordinates (t, x, y, z) and metric η_{ab} given by

$$\eta_{ab} = \eta^{ab} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}, \tag{1.1}$$

so that vectors u^a and v^a in $\mathbb{M}$ have their inner product given by

$$u^a v^b \eta_{ab} = u_t v_t - u_x v_x - u_y v_y - u_z v_z. \tag{1.2}$$

The *Lorentz norm* is defined by taking the inner product of a vector with itself, $u^a u^b \eta_{ab}$. A transformation which preserves the Lorentz norm is called a *Lorentz transformation.*

The form of the Lorentz norm divides vectors into three distinct classes, *timelike*, *spacelike*, and *null*, based on whether their inner product is, respectively, positive, negative or zero. Thus, a null vector k^a is a vector at P satisfying

$$k^a = (k_t, k_x, k_y, k_z), \tag{1.3}$$

$$(k_t)^2 - (k_x)^2 - (k_y)^2 - (k_z)^2 = 0. \tag{1.4}$$

We will focus our attention on a particular class of null vectors, the *null directions*, defined to be those null vectors whose spacelike parts have unit norm, that is, those for

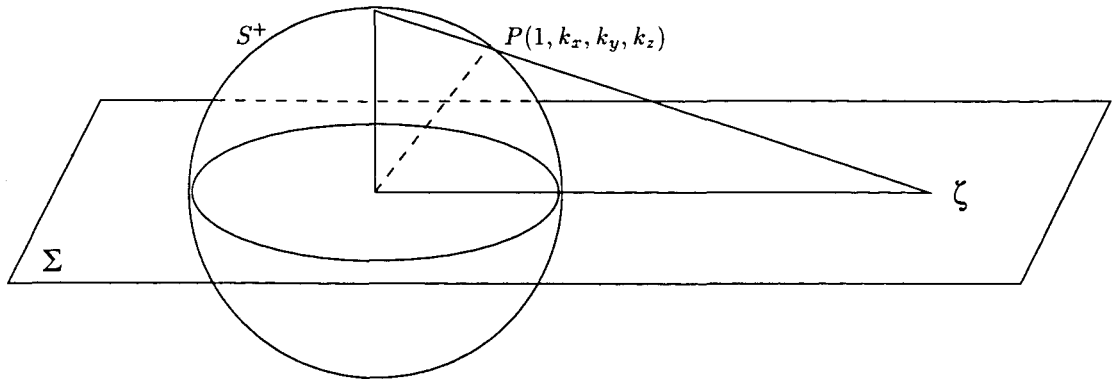


Figure 1.1: Stereographic projection from S^+ to the Argand plane.

which

$$(k_x)^2 + (k_y)^2 + (k_z)^2 = 1. \tag{1.5}$$

We can distinguish two classes of such vectors, those for which k_4 is $+1$ or -1, said to exist on either the future null cone, S^+, or the past null cone, S^-. Note that each of these classes of null directions can be parametrized by points on the unit sphere, for instance,

$$k_x = \sin\theta^+ \cos\phi^+, \qquad k_y = \sin\theta^+ \sin\phi^+, \qquad k_z = \cos\theta^+ \tag{1.6}$$

and as such can be brought into one-to-one correspondence with points ζ of the Argand plane (with ∞ attached) via the standard Riemann stereographic projection,

$$\zeta = e^{i\phi^+} \cot\frac{\theta^+}{2} = \frac{k_x + ik_y}{1 - k_z}. \tag{1.7}$$

(see Fig. 1.1).

The final refinement to this picture which we will make is that rather than use a single complex number ζ, we choose to map each point of S^+onto a complex pair (ξ, η), where

$$\zeta = \xi/\eta. \tag{1.8}$$

This seemingly extra complication allows us to define the notion of a *spin transformation*, namely the map

$$\xi \mapsto \alpha\xi + \beta\eta, \tag{1.9}$$

$$\eta \mapsto \gamma\xi + \delta\eta, \tag{1.10}$$

where α, β, γ, and δ, are complex scalars satisfying

$$\alpha\delta - \beta\gamma = 1. \tag{1.11}$$

This transformation can be represented in terms of a *spin matrix*, $\boldsymbol{A}$,

$$\begin{pmatrix}\xi\\ \eta\end{pmatrix} \mapsto \boldsymbol{A}\begin{pmatrix}\xi\\ \eta\end{pmatrix}, \tag{1.12}$$

where

$$\boldsymbol{A} := \begin{pmatrix}\alpha & \beta\\ \gamma & \delta\end{pmatrix}, \qquad \det \boldsymbol{A} = 1. \tag{1.13}$$

Significantly, the spin matrices form a group, the special linear group of 2×2 complex matrices, referred to as $SL(2,\mathbb{C})$. In fact, it is possible to prove the following result:

Theorem 1 *(PR 1.2.27) Every spin transformation corresponds to a unique restricted Lorentz transformation. Conversely every restricted Lorentz transformation corresponds to precisely two spin transformation, one being the negative of the other.*

That two spin transformations correspond to a single Lorentz transformation is a result of the fact that both $\boldsymbol{A}$ and $-\boldsymbol{A}$ have the same effect on ζ. A proof of the given result can be found in Penrose and Rindler (1984).

To summarise, we have defined a correspondence between null directions at P and the complex pair (ξ, η). The correspondence is such that Lorentz transformations of the former can be mapped on to spin transformations of the latter.

Spinors and spin bases

The complex pair (η, ξ) representing a particular null direction can be used to define a *spin vector* via

$$\boldsymbol{\kappa} := (\kappa^0, \kappa^1), \tag{1.14}$$

$$\kappa^0 := \xi, \qquad \kappa^1 := \eta. \tag{1.15}$$

For the space of spin vectors, the operations of scalar multiplication and addition are defined as usual, as well as a symplectic inner product,

$$\lambda(\kappa^0, \kappa^1) = (\lambda\kappa^0, \lambda\kappa^1), \tag{1.16}$$

$$(\kappa^0, \kappa^1) + (\omega^0, \omega^1) = (\kappa^0 + \omega^0, \kappa^1 + \omega^1), \tag{1.17}$$

$$< \boldsymbol{\kappa}, \boldsymbol{\omega} > = \kappa^0\omega^1 - \kappa^1\omega^0, \tag{1.18}$$

for spin vectors $\boldsymbol{\kappa}$ and $\boldsymbol{\omega}$, and complex scalar λ. It can be shown that these operations are invariant under spin transformations.

We can introduce indices to the notation by writing

$$\kappa^A = (\kappa^0, \kappa^1), \qquad \omega^A = (\omega^0, \omega^1), \tag{1.19}$$

and defining an antisymmetric object, ε_{AB}, to play the role of the inner product,

$$< \boldsymbol{\kappa}, \boldsymbol{\omega} >= \varepsilon_{AB}\kappa^A\omega^B, \tag{1.20}$$

with

$$\varepsilon_{AB} = -\varepsilon_{BA}, \tag{1.21}$$

and summation over the repeated indices assumed.

The ε_{AB} object can be used to define a mapping to the dual κ_B of the spin vector κ^B via

$$\kappa_B := \kappa^A\varepsilon_{AB}. \tag{1.22}$$

The components of κ_B are related to those of κ^A via

$$\kappa_0 = -\kappa^1, \qquad \kappa_1 = \kappa^0. \tag{1.23}$$

A pair of spin vectors (o^A, ι^A) are called a *spin basis* if their inner product satisfies

$$\varepsilon_{AB}o^A\iota^B = 1. \tag{1.24}$$

An arbitrary spin vector can be represented in terms of the spin basis by

$$\kappa^A = \kappa^0 o^A + \kappa^1\iota^A, \tag{1.25}$$

where $\kappa^0 := -\iota_A\kappa^A$ and $\kappa^1 := o_A\kappa^A$, and the inner product itself can be written

$$\varepsilon_{AB} = o_A\iota_B - \iota_A o_B, \qquad \varepsilon^{AB} = o^A\iota^B - \iota^A o^B. \tag{1.26}$$

A final technical issue arises when we consider the operation of complex conjugation on spin vectors, necessary if we wish to recover the real valued components of null vectors from the complex valued spin vectors. Unfortunately, if we are to maintain Lorentz covariance, then the conjugates of spin vectors can not be considered to be in the same spin space as the original vectors. As a result, we introduce a new spin space corresponding to the conjugate space of spin vectors κ^A, and delineate its members by placing a prime on its index labels. Thus if we denote complex conjugation by a bar, we write

$$\overline{\kappa^A} = \bar{\kappa}^{A'}. \tag{1.27}$$

By transforming a null vector k^a to a point on the complex plane, $\zeta = \xi/\eta$, via the stereographic projection described above, it is possible to show that the components of any null vector are given by the following formula:

$$\begin{pmatrix} k_0 & k_1 \\ k_2 & k_3 \end{pmatrix} = \begin{pmatrix} \xi\bar{\xi} & \xi\bar{\eta} \\ \bar{\xi}\eta & \eta\bar{\eta} \end{pmatrix} = \begin{pmatrix} \xi \\ \eta \end{pmatrix} (\bar{\xi} \quad \bar{\eta}). \tag{1.28}$$

Remembering the definition of the spin vector $\kappa^A = (\kappa^0, \kappa^1) = (\xi, \eta)$ suggests that we can make the following correspondence between null directions and spin vectors:

$$k^a \longleftrightarrow \kappa^A \bar{\kappa}^{A'}, \tag{1.29}$$

where the indices are identified via $(0,1,2,3) \leftrightarrow (00', 01', 10', 11')$. Given a spin basis (o^A, ι^A) we can use this relationship to establish a set of four independent null vectors,

$$l^a := o^A \bar{o}^{A'}, \quad m^a := o^A \bar{\iota}^{A'}, \quad \bar{m}^a := \iota^A \bar{o}^{A'}, \quad n^a := \iota^A \bar{\iota}^{A'}, \tag{1.30}$$

which satisfy the requirements of a Newman-Penrose null tetrad. Namely,

$$l^a n_a = 1, \qquad m^a \bar{m}^a = -1, \tag{1.31}$$

and all other inner products are zero.

Thus, any vector k^a whose components are expressed in the given null frame as

$$k^a = k_0 l^a + k_1 m^a + k_2 \bar{m}^a + k_3 n^a, \tag{1.32}$$

has a natural expression in terms of a 2-spinor formed by replacing the null basis vectors with the corresponding spin basis combinations,

$$k^a \longleftrightarrow \kappa^{AA'} = k_0 o^A \bar{o}^{A'} + k_1 o^A \bar{\iota}^{A'} + k_2 \iota^A \bar{o}^{A'} + k_3 \iota^A \bar{\iota}^{A'}, \tag{1.33}$$

For the given spin basis, the identification can be carried out explicitly using the *Infeld-van der Waerden symbols*,

$$\begin{aligned} g_0{}^{AB'} &:= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = g_{AB'}{}^0, \qquad g_1{}^{AB'} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = g_{AB'}{}^1, \\ g_2{}^{AB'} &:= \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & i \\ -i & 0 \end{pmatrix} = g_{AB'}{}^2, \qquad g_3{}^{AB'} := \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = g_{AB'}{}^3, \end{aligned} \tag{1.34}$$

so that

$$k^a = g_{AA'}{}^a \kappa^{AA'}. \tag{1.35}$$

In fact, we will use these symbols more generally to define the correspondence between tensors of n indices $a, b, c, \ldots$, with spinors of $2n$ indices $A, A', B, B', C, C', \ldots$.

Curvature

We can now proceed to define the notion of covariant differentiation in spinor terms. This is a more complicated procedure than for a standard coordinate basis because of the inability to express the Lie derivative as a spinor. However, it can be shown (Penrose and Rindler (1984)) that a derivative operator $\nabla_{AA'}$ exists uniquely as a map $\nabla_{AA'} : \chi_{\ldots} \mapsto \nabla_{AA'}\chi_{\ldots}$ when it is subjected to the following requirements:

i. $\nabla_{AA'}(\theta + \phi) = \nabla_{AA'}\theta + \nabla_{AA'}\phi$,

ii. $\nabla_{AA'}(\theta\phi) = \phi\nabla_{AA'}\theta + \theta\nabla_{AA'}\phi$,

iii. $\phi = \nabla_{AA'}\theta$ implies $\bar{\phi} = \nabla_{AA'}\bar{\theta}$,

iv. $\nabla_{AA'}\varepsilon_{BC} = \nabla_{AA'}\varepsilon^{BC} = 0$,

v. $\nabla_{AA'}$ commutes with any index substitution not involving A or A',

vi. $\nabla_{AA'}\nabla_{BB'}f = \nabla_{BB'}\nabla_{AA'}f$ for all scalars f,

vii. for any derivation D acting on spinor fields there exists a spinor $\xi^{AA'}$ such that $D\phi = \xi^{AA'}\nabla_{AA'}\phi$ for all ϕ.

With this definition, we can construct a set of spinor *Ricci rotation coefficients*, in analogy with the usual procedure for bases, via

$$\Gamma_{CC'AB} = \varepsilon_{AA}\varepsilon_{C}{}^{C}\varepsilon_{C'}{}^{C'}\nabla_{CC'}\varepsilon_{B}{}^{A}, \tag{1.36}$$

where components are understood to be in terms of the spin basis $\varepsilon_{A}{}^{A} = (o^A, \iota^A)$.

We next proceed to construct the spinor form of the curvature tensors. First note that in terms of the null tetrad $e_{a}{}^{a} = (l^a, m^a, \bar{m}^a, n^a)$, we have the following definition of the *Riemann curvature tensor*,

$$R_{abcd} := 2e_{aa}\nabla_{[c}\nabla_{d]}e^{a}{}_{b}. \tag{1.37}$$

where ∇_a is the normal coordinate covariant derivative. The spinor analogy of this definition is

$$R_{ABCDA'B'C'D'} = 2\varepsilon_{BA}\varepsilon_{B'A'}\nabla_{[CC'}\nabla_{DD']}(\varepsilon^{B}{}_{B}\varepsilon^{B'}{}_{B'}) \tag{1.38a}$$

$$= 2\varepsilon_{BA}\varepsilon_{B'A'}\varepsilon^{B'}{}_{B'}\nabla_{[CC'}\nabla_{DD']}\varepsilon^{B}{}_{B} + \text{c.c.}, \tag{1.38b}$$

$$= 2\varepsilon_{BA}\varepsilon_{A'B'}\nabla_{[CC'}\nabla_{DD']}\varepsilon^{B}{}_{B} + \text{c.c.}, \tag{1.38c}$$

where 'c.c.' denotes the complex conjugates of the listed terms. By defining the operator

$$\Box_{AB} = \varepsilon^{A'B'}\nabla_{A'(A}\nabla_{B)B'}, \tag{1.39}$$

and noting that

$$2\nabla_{[a}\nabla_{b]} = \varepsilon_{AA'}\Box_{BB'} + \varepsilon_{BB'}\Box_{AA'}, \tag{1.40}$$

(see Stewart (1990)) we can decompose the curvature spinor as follows. First define the scalar Λ, and the *Weyl spinor*, and *Ricci spinor*, respectively, as

$$\Lambda := \frac{1}{6}\varepsilon_{BA}\Box^{AB}\varepsilon^{B}{}_{B}, \tag{1.41}$$

$$\Psi_{ABCD} := \varepsilon_{BA}\Box_{C(D}\varepsilon^{B}{}_{B)} = \Psi_{(ABCD)}, \tag{1.42}$$

$$\Phi_{ABA'B'} := \varepsilon_{BA}\Box_{A'B'}\varepsilon^{B}{}_{B} = \Phi_{(AB)(A'B')}. \tag{1.43}$$

Then the curvature spinor can be written in terms of these quantities as

$$R_{ABCDA'B'C'D'} = \varepsilon_{A'B'}\varepsilon_{C'D'}(\Psi_{ABCD} - 2\Lambda\varepsilon_{(A(C}\varepsilon_{D)B)}) + \varepsilon_{A'B'}\varepsilon_{CD}\Phi_{ABC'D'} + \text{c.c.}. \tag{1.44}$$

We note finally that the curvature spinors are related to their tensorial counterparts as follows:

$$R = 24\Lambda, \tag{1.45}$$

$$S_{ab} = -2\Phi_{ABA'B'}, \tag{1.46}$$

$$C_{abcd} = \Psi_{ABCD}\varepsilon_{A'B'}\varepsilon_{C'D'} + \Psi_{A'B'C'D'}\varepsilon_{AB}\varepsilon_{CD}, \tag{1.47}$$

$$R_{abcd} = R_{ABCDA'B'C'D'}, \tag{1.48}$$

where the Infeld-van der Waerden symbols are implicitly used to relate quantities across the equals signs, and the curvature tensors are defined as

$$C_{abcd} := R_{abcd} - 2S_{[a[c}g_{d]b]} - \frac{1}{6}Rg_{[a[c}g_{d]b]}, \tag{1.49}$$

$$S_{ab} := R_{ab} - \frac{1}{4}R, \tag{1.50}$$

$$R_{ab} := g^{cd}R_{cabd}, \tag{1.51}$$

$$R := g^{ab}R_{ab}. \tag{1.52}$$

Some properties of spinors at a point

The following theorems relating to the properties of spinors will prove useful for the analysis of the following chapters. Full proofs and discussion can be found in Penrose and Rindler (1984) under the listed theorem numbers.

Theorem 2 *(PR 2.5.56) The condition $\alpha_A\beta^B = 0$ at a point is necessary and sufficient for α_A and β_B to be scalar multiples of each other at that point.*

Theorem 3 *(PR 3.3.54) Any spinor $\phi_{AB\ldots N}{}^{A'B'\ldots N'}$ is the sum of the symmetric spinor $\phi_{(AB\ldots N)}{}^{(A'B'\ldots N')}$ and of outer products of ε's with symmetric spinors of lower valence.*

Theorem 4 *(PR 3.5.18) If $\phi_{AB\ldots L} = \phi_{(AB\ldots L)} \neq 0$ then*

$$\phi_{AB\ldots N} = \alpha_{(A}\beta_B \ldots \lambda_{N)} \tag{1.53}$$

for some spin vectors $\alpha_A, \beta_B, \lambda_N$, called the principal spinors *of $\phi_{AB\ldots N}$. This decomposition is unique up to proportionality or reordering of the factors.*

Further, since any spin vector can be expanded as a sum of the spin basis vectors,

$$\alpha_A = \alpha_1 o_A + \alpha_2 \iota_A, \tag{1.54}$$

we see that it is possible to expand any unprimed symmetric spinor in terms of the spin basis as

$$\begin{aligned}\phi_{AB\ldots N} =& \phi_0 o_A o_B \ldots o_N - n\phi_1 (\iota_A o_B \ldots o)_N + \ldots + \binom{n}{2}\phi_2 (\iota_A \iota_B \ldots o)_N + \ldots \\ &\ldots + \phi_n (\iota_A \iota_B \ldots \iota)_N,\end{aligned} \tag{1.55}$$

with $\phi_1, \phi_2, \ldots, \phi_n$, complex scalars dependent on the principle spinors. (The factors $(-1)^k$ and $\binom{n}{k}$ in front of each term have been introduced to allow us to write

$$\phi_k := \phi_{AB\ldots N} o^A o^B \ldots \iota^N, \tag{1.56}$$

where $\phi_{AB\ldots N}$ is contracted with k ι^As and $n-k$ o^As.)

A method of determining the principle spinors is suggested by Theorem 2. Namely, consider the contraction of $\phi_{AB\ldots N}$ with the spinor $\zeta^A = (1, z)$, with complex valued z:

$$\phi_{AB\ldots N}\zeta^A\zeta^B \ldots \zeta^N = \alpha_{(A}\zeta^A\beta_B\zeta^B \ldots \lambda_{N)}\zeta^N. \tag{1.57}$$

This polynomial in z only equals zero when at least one of the individual index contractions, eg. $\alpha_A\zeta^A$, equals zero. By Theorem 2, this indicates that ζ^A is proportional to the particular principle spinor with which it has been contracted. Thus, the principle spinors can be found by determining the roots of the nth order polynomial

$$\phi_0 + \phi_1 z + \phi_2 z^2 + \ldots + \phi_n z^n = 0, \tag{1.58}$$

where the coefficients $\phi_0 \ldots \phi_n$ are exactly those of the spin basis expansion, (1.55).

The Newman-Penrose formalism

Later, we will often find it useful to represent spinor components in terms of the notation established by Newman and Penrose (1962), referred to as the NP formalism. As a final bit of notation, we list the relationships between the spinor quantities and their NP counterparts.

The NP quantities are based upon the representation of the spacetime in terms of a tetrad composed of four null vectors, $(l^a, m^a, \bar{m}^a, n^a)$ which satisfies the inner product relationships

$$g_{ab}l^a n^a = 1 \qquad g_{ab}m^a \bar{m}^a = -1, \tag{1.59}$$

with all other inner products being zero. As mentioned above, exactly such a tetrad can be formed in terms of the spin basis $\varepsilon_A{}^A = (o^A, \iota^A)$, namely, by defining

$$l^a := o^A \bar{o}^{A'}, \quad m^a := o^A \bar{\iota}^{A'}, \quad \bar{m}^a := \iota^A \bar{o}^{A'}, \quad n^a := \iota^A \bar{\iota}^{A'}, \tag{1.60}$$

$$l_a := o_A \bar{o}_{A'}, \quad m_a := o_A \bar{\iota}_{A'}, \quad \bar{m}_a := \iota_A \bar{o}_{A'}, \quad n_a := \iota_A \bar{\iota}_{A'}. \tag{1.61}$$

Derivatives along these directions are denoted by the symbols

$$D := l^a \nabla_a, \quad \delta := m^a \nabla_a, \quad \bar{\delta} := \bar{m}^a \nabla_a, \quad \Delta := n^a \nabla_a, \tag{1.62}$$

corresponding respectively to the $00', 01', 10', \text{and} 11'$ components of the spinor covariant derivative operator $\nabla_{AA'}$.

The *NP spin coefficients* are a set of twelve scalars representing the components of the Ricci rotation coefficients. They are defined in Table 1.1.

1.2 The equivalence problem

According to the above prescription, we describe a spacetime $\mathscr{M}$ by a coordinate patch x^μ on which is defined a spin basis (o^A, ι^A), which can be subjected to $SL(2, \mathbb{C})$ rotations, ϵ, or discrete transformations of the basis, m. Two spacetimes are said to be *equivalent* if and only if there is a correspondence

$$\tilde{x}^\mu = \tilde{x}^\mu(x^\nu), \tag{1.63}$$

$$\tilde{\epsilon} = \tilde{\epsilon}(\epsilon), \tag{1.64}$$

$$\tilde{m} = \tilde{m}(m), \tag{1.65}$$

NP coefficients

κ	ϵ	π
ρ	α	λ
σ	β	μ
τ	γ	ν

GHP coefficients

κ	$-\gamma'$	$-\tau'$
ρ	$-\beta'$	$-\sigma'$
σ	β	$-\rho'$
τ	γ	$-\kappa'$

spin basis components

$o^A Do_A$	$\iota^A Do_A$	$\iota^A D\iota_A$
$o^A \bar{\delta} o_A$	$\iota^A \bar{\delta} o_A$	$\iota^A \bar{\delta} \iota_A$
$o^A \delta o_A$	$\iota^A \delta o_A$	$\iota^A \delta \iota_A$
$o^A \Delta o_A$	$\iota^A \Delta o_A$	$\iota^A \Delta \iota_A$

NP frame components

$m^a Dl_a$	$\frac{1}{2}(n^a Dl_a + m^a D\bar{m}_a)$	$\bar{m}^a Dm_a$
$m^a \bar{\delta} l_a$	$\frac{1}{2}(n^a \bar{\delta} l_a + m^a \bar{\delta} \bar{m}_a)$	$\bar{m}^a \bar{\delta} m_a$
$m^a \delta l_a$	$\frac{1}{2}(n^a \delta l_a + m^a \delta \bar{m}_a)$	$\bar{m}^a \delta m_a$
$m^a \Delta l_a$	$\frac{1}{2}(n^a \Delta l_a + m^a \Delta \bar{m}_a)$	$\bar{m}^a \Delta m_a$

Rotation coefficients

$\Gamma_{00'00}$	$\Gamma_{00'01}$	$\Gamma_{00'11}$
$\Gamma_{10'00}$	$\Gamma_{10'01}$	$\Gamma_{10'11}$
$\Gamma_{01'00}$	$\Gamma_{01'01}$	$\Gamma_{01'11}$
$\Gamma_{11'00}$	$\Gamma_{11'01}$	$\Gamma_{11'11}$

Table 1.1: The Newman-Penrose spin coefficients are listed in the upper left-hand box. Their definitions in various notations are given by the corresponding entries of the subsequent boxes. Though not described in the text, the GHP versions of the spin coefficients are listed for cross reference with work using the formalism of Geroch et. al. (1973).

which gives

$$\tilde{o}^A(\tilde{x}^\mu, \tilde{\epsilon}, \tilde{m}) = o^A(x^\mu, \epsilon, m), \tag{1.66}$$

$$\tilde{\iota}^A(\tilde{x}^\mu, \tilde{\epsilon}, \tilde{m}) = \iota^A(x^\mu, \epsilon, m). \tag{1.67}$$

The early work on the equivalence problem was due to Christoffel (Christoffel (1869), Thomas (1934)), who showed that, for metrics without symmetry, the equivalence of a pair of metrics could be decided by computing the Riemann tensor and a finite number of its derivatives in coordinate bases, and for four dimensional spaces, at most the 20th derivatives would be required.

By specifying the space using a frame with constant metric components, Cartan (1946) was able to improve this technique significantly, showing that the maximum number of differentiations to be carried out is 10. Translated into the language of spinors, Cartan's equivalence theorem can be stated as follows:

Theorem 5 *Two regions $\mathscr{M}$ and $\tilde{\mathscr{M}}$ of two n-dimensional Riemannian manifolds are*

locally equivalent if and only if the set of equations

$$\tilde{\Psi}_{ABCD} = \Psi_{ABCD} \tag{1.68a}$$

$$\tilde{\nabla}_{EE'}\tilde{\Psi}_{ABCD} = \nabla_{EE'}\Psi_{ABCD} \tag{1.68b}$$

$$\dots$$

$$\tilde{\nabla}_{E_1E_1'}\dots\tilde{\nabla}_{E_{p+1}E_{p+1}'}\tilde{\Psi}_{ABCD} = \nabla_{E_1E_1'}\dots\nabla_{E_{p+1}E_{p+1}'}\Psi_{ABCD} \tag{1.68c}$$

and

$$\tilde{\Phi}_{ABA'B'} = \Phi_{ABA'B'} \tag{1.69a}$$

$$\tilde{\nabla}_{EE'}\tilde{\Phi}_{ABA'B'} = \nabla_{EE'}\Phi_{ABA'B'} \tag{1.69b}$$

$$\dots$$

$$\tilde{\nabla}_{E_1E_1'}\dots\tilde{\nabla}_{E_{p+1}E_{p+1}'}\tilde{\Phi}_{ABA'B'} = \nabla_{E_1E_1'}\dots\nabla_{E_{p+1}E_{p+1}'}\Phi_{ABA'B'}, \tag{1.69c}$$

and

$$\tilde{\Lambda} = \Lambda \tag{1.70a}$$

$$\tilde{\nabla}_{EE'}\tilde{\Lambda} = \nabla_{EE'}\Lambda \tag{1.70b}$$

$$\dots$$

$$\tilde{\nabla}_{E_1E_1'}\dots\tilde{\nabla}_{E_{p+1}E_{p+1}'}\tilde{\Lambda} = \nabla_{E_1E_1'}\dots\nabla_{E_{p+1}E_{p+1}'}\Lambda, \tag{1.70c}$$

are compatible as equations in the coordinates x^μ and $\tilde{x}^\mu$, and the frame rotations ϵ and $\tilde{\epsilon}$, and where $p+1$ is the first derivative which is functionally dependent on the lower order derivatives of the curvature spinors, $p+1 \leq n(n+1)/2$.

For convenience, we define a number of sets $R^i, i = 0, \dots, p+1$ to contain the components of the ith derivatives of the curvature spinors. The term *independence* used in the above theorem, corresponds to the requirement that the rank of the Jacobian constructed from the components of $S = R^0 \cup \dots \cup R^p$ is equal to the rank of the Jacobian of $S \cup R^{p+1}$. It can be shown that $p+1 \leq n$, the number of coordinates. For a proof of the equivalence theorem, the reader is referred to Cartan (1946) or Karlhede (1979).

Cartan's method provided an important reduction in the calculational effort in comparison with Christoffel's original coordinate based approach. However, if the equivalence theorem is followed explicitly, the components to be calculated still numbers 27 926 020. A further reduction is obtained when one recognises that not all of these

components are independent, due to the Ricci and Bianchi identities and their differential concomitants. A minimal set of components (numbering 8690 in total) was provided by MacCallum and Åman (1986), and we quote their result here. At the qth derivative of the curvature, the independent spinor components are given by the following:

- The totally symmetrized q-th derivatives of Λ.
- The totally symmetrized q-th derivatives of Ψ_{ABCD}.
- The totally symmetrized q-th derivatives of $\Phi_{ABA'B'}$.
- For $q \geq 1$, the totally symmetrized $(q-1)$-th derivatives of $\Xi_{ABCA'} := \nabla^{D}{}_{A'}\Psi_{ABCD}$ (which is one side of the Bianchi identities).
- For $q \geq 2$, the d'Alembertian $\nabla^{AA'}\nabla_{AA'}$ applied to all of the quantities calculated for order $q-2$.

The term 'totally symmetrized' here refers to a symmetrization over each of its primed and unprimed indices. In fact, every spinor which is calculated in the classification procedure is symmetric in its indices.

As a notational convenience (which we will also make use of throughout the rest of this report), we follow the common practice of labelling individual components of such spinors using two subscripts which count the number of ι^As and $\bar{\iota}^A$s which have been contracted with the spinor to arrive at the component. Thus,

$$\Psi_{3_{0'}} := \Psi_{ABCD} o^A \iota^B \iota^C \iota^D, \tag{1.71}$$

$$\Phi_{1_{1'}} := \Phi_{ABA'B'} o^A \iota^B \bar{o}^{A'} \bar{\iota}^{B'}, \tag{1.72}$$

$$(\nabla\Phi)_{2_{1'}} := \nabla_{(CC'}\Phi_{ABA'B')} o^A \iota^B \iota^C \bar{o}^{A'} \bar{o}^{B'} \bar{\iota}^{C'}. \tag{1.73}$$

Note that since all of the spinors above are symmetric under interchange of indices, the notation assigns a unique label to each component.

The total number of primed and unprimed indices of a particular spinor is called its *valence*. Thus Ψ_{ABCD}, $\Phi_{ABA'B'}$, and $\nabla_{EE'}\Psi_{ABCD}$ have valences of $(4,0)$, $(2,2)$ and $(5,1)$ respectively.

1.3 The Cartan-Karlhede Method

A great deal of the effort expended in carrying out Cartan's method for classifying spacetimes arises from the need to determine the six degrees of $SL(2,\mathbb{C})$ rotational freedom

of the frame. Realizing this, Karlhede (1980) suggested that the computational effort might be reduced if for a given spacetime the frame could be fixed in some unambiguous way, effectively reducing the dimension of the frame bundle. In practice, the frame is fixed using the components of the curvature spinors, by requiring that they take on some canonical, or standard, form.

Karlhede's method proceeds as follows:

1. Calculate the curvature spinor components, $R^0 = \{\Psi_{ABCD}, \Phi_{ABA'B'}, \Lambda\}$.
2. Determine the isotropy group, H, under which the components are invariant.
3. Fix the frame up to H by requiring that the curvature components take on a canonical form.
4. Calculate the first derivatives, R^1, of the curvature spinors.
5. If H is non-empty, then determine if it can be reduced by fixing a canonical form for the derivatives of the Riemann tensor.
6. Continue the procedure to the higher derivatives until the frame is maximally fixed and the required number of free components are found.

Using this method, it can be shown that for a general spacetime, at most the 7th covariant derivative of the curvature tensor need be calculated, or a total of 3156 independent terms. In fact, one rarely needs to go past the third covariant derivative (Koutras (1992), Skea (1997a), Skea (1997c)). For Petrov Type N metrics, the upper bound is known to be the sixth derivative (Collins (1991), Machado Ramos and Vickers (1996)), while for vacuum Type D metrics, a theoretical upper bound of three derivatives has been determined (Collins et al. (1990), Collins et al. (1991)), though it may be possible to reduce this in light of the fact that all of the metrics in Kinnersley's classes have required at most two derivatives for full classification (Kinnersley (1969)).

Finally, we point out the crucial dependence of Karlhede's method on the ability to define a consistent set of standard forms for the curvature spinors and to calculate the transformations to the standard form. In fact, this makes up the bulk of the effort in carrying out the method, as the emphasis in the next chapter makes clear.

2

Standard forms for symmetric spinors

The equivalence of a pair of spacetimes is most easily established when the components of the relevant curvature spinors have been expressed in some standard (or *canonical*) form for which the frame rotational freedom has been restricted as far as possible. In fact, the specification of a consistent set of canonical forms and the procedures for transforming spinors into the appropriate form, makes up the most involved task in carrying out the Cartan-Karlhede method.

Recall that $SL(2,\mathbb{C})$ transformations of spinors can be represented as 2×2 complex valued matrices such that

$$\begin{pmatrix} o^A \\ \iota^A \end{pmatrix} \mapsto \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \begin{pmatrix} o^A \\ \iota^A \end{pmatrix} \quad \text{where} \quad \alpha\delta - \beta\gamma = 1. \tag{2.1}$$

Under such transformations, components of an arbitrary symmetric spinor, χ, are modified by a polynomial of its components, $\chi_{ab'}$, and the transformation coefficients α, β, γ and δ,

$$\chi_{ab'} \mapsto P_{ab'}(\chi, \alpha, \beta, \gamma, \delta) \tag{2.2}$$

By making use of these transformations, restrictions can be placed on the individual components of a spinor in order that the six parameter $SL(2,\mathbb{C})$ freedom can be removed as much as possible. For instance, by requiring that the $\chi_{ab'}$ component be zero, we restrict two parameters of the frame freedom by requiring that

$$P_{ab'}(\chi, \alpha, \beta, \gamma, \delta) = 0. \tag{2.3}$$

Such restrictions are applied to as many components as necessary, until the $SL(2,\mathbb{C})$ freedom has been entirely removed or the spinor components are invariant under any remaining freedom.

When the restrictions are applied in such a way that the spinor components are consistently transformed to the same form, independent of their initial configuration, then we say that the spinor is in a *standard form.* The spin basis (or equivalently null frame) under which the spinor takes on its standard form is called the *standard frame.*

Two practical problems arise when trying to solve for a frame rotation which will bring a spinor into a standard form. The first is that it is not guaranteed, for a particular standard form, that such a solution exists. In particular, the spinor components may themselves satisfy algebraic interrelationships which make prohibit a consistent solution for α, β, γ, δ, for a given set of component restrictions, $P_{ab'}(\chi, \alpha, \beta, \gamma, \delta)$. These special situations must be isolated and alternate standard forms which guarantee a solution defined. That is, there will be alternate standard forms depending on the algebraic type of the spinor in question.

A further difficulty arises when one tries to solve polynomials of the form (2.3) for the spin transformation components. In general, these polynomials will be of high order in the components, which themselves form a six parameter family, and obtaining a solution, even if it exists, will be a non-trivial task. As a result, in practice it is common first to consider the transformation of spinor components under specific subgroups of $SL(2,\mathbb{C})$. For instance, it is possible to show that any member of $SL(2,\mathbb{C})$ can be represented as a product of a *spin*, *boost*, or *null rotation* about either o^A or ι^A, where these terms are defined by the following matrices:

$$\underset{\textit{spin}}{\begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix}} \qquad \underset{\textit{boost}}{\begin{pmatrix} \lambda & 0 \\ 0 & 1/\lambda \end{pmatrix}} \qquad \underset{\substack{\textit{null rotation} \\ \textit{about } \iota^A}}{\begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix}} \qquad \underset{\substack{\textit{null rotation} \\ \textit{about } o^A}}{\begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix}} \tag{2.4}$$

where θ and λ are real, and α and β are complex valued parameters. This corresponds directly to the analogous reduction of SO(3) rotations in terms of Euler angles.

We can examine the effect of these transformations on a given spinor, and use this to define a standard form for each type. This procedure requires some care, however, because if the chosen subgroups of $SL(2,\mathbb{C})$ do not commute, then the final form of the spinor components after each of the transformations has been applied will be dependent on the original orientation of the given frame.

For the particular case of the equivalence problem in general relativity, however, both of these problems can be overcome. Note that the first step of the Cartan-Karlhede procedure involves the reduction of the 0th order (Weyl and Ricci) spinors to a standard form. In fact, well established invariant procedures have long existed for doing just

that, namely the Petrov classification of the Weyl spinor, and the Segré classification of the Ricci spinor. These classifications force the components of the named spinors into particular form depending on their algebraic type. Each of these types is invariant under only limited subgroups of $SL(2,\mathbb{C})$. Thus for the remaining spinors (ie. the derivatives of the Weyl and Ricci spinors, and the Ricci scalar), we only have to specify the standard forms under these particular subgroups.

This chapter establishes the procedures necessary for the reduction of the 0th order Weyl and Ricci spinors to a standard form. Assuming that such a reduction has been carried out, we show that the remaining degrees of freedom in the frame can be systematically removed using the components of the derivatives of these tensors in a way such that the final forms of the spinor components are independent of the initial frame orientation. Some problems with the given methods are discussed, as well as some possible modifications of the standard problem in order to avoid them.

2.1 The Weyl spinor

The first spinor to be considered in fixing the frame of a general spacetime is the Weyl spinor. Its classification into six distinct types according to the multiplicities of its principal null directions was first carried out by Petrov (1954) (cf. Penrose and Rindler (1986)). The classification is based on the decomposition of the Weyl spinor into its principle spinors, as described by Theorem 4,

$$\Psi_{ABCD} = \alpha_{(A}\beta_B\gamma_C\delta_{D)}. \tag{2.5}$$

The different *Petrov types* of Ψ_{ABCD} are given by the coincidences of these principle spinors, specified as follows:

$$I: \quad \Psi_{ABCD} = \alpha_{(A}\beta_B\gamma_C\delta_{D)}, \tag{2.6}$$

$$II: \quad \Psi_{ABCD} = \alpha_{(A}\alpha_B\gamma_C\delta_{D)}, \tag{2.7}$$

$$D: \quad \Psi_{ABCD} = \alpha_{(A}\alpha_B\beta_C\beta_{D)}, \tag{2.8}$$

$$III: \quad \Psi_{ABCD} = \alpha_{(A}\alpha_B\alpha_C\delta_{D)}, \tag{2.9}$$

$$N: \quad \Psi_{ABCD} = \alpha_{(A}\alpha_B\alpha_C\alpha_{D)}, \tag{2.10}$$

$$0: \quad \Psi_{ABCD} = 0, \tag{2.11}$$

where it is assumed that α_A, β_A, γ_A, and δ_A are all non-proportional and non-vanishing.

Determination of the Petrov type

Referring to the methods outlined in Section 1.1, the principle null directions can be found by determining the roots of the quartic equation

$$\Psi_4 z^4 + \Psi_3 z^3 + \Psi_2 z^2 + \Psi_1 z + \Psi_0 = 0. \tag{2.12}$$

An algorithm for solving this problem was first presented by d'Inverno and Russell-Clark (1971), based on calculating the invariants

$$I := \Psi_{ABCD}\Psi^{ABCD}, \tag{2.13}$$

$$J := \Psi_{ABCD}\Psi^{CD}{}_{EF}\Psi^{ABEF}, \tag{2.14}$$

and making note of the identities

$$I = J = 0 \quad \text{for Petrov types } III \text{ and } N, \tag{2.15}$$

$$I^3 = 27J^2 \quad \text{for Petrov types } II \text{ and } D. \tag{2.16}$$

These are combined with the identities

$$K = 0 \quad \text{for Petrov Types } D \text{ and } N, \tag{2.17}$$

$$L = 0 \quad \text{for Petrov Type } N, \tag{2.18}$$

$$N = 0 \quad \text{for Petrov Type } D, \tag{2.19}$$

where

$$K := \Psi_1 {\Psi_4}^2 - 3\Psi_4\Psi_3\Psi_2 + 2{\Psi_3}^3, \tag{2.20}$$

$$L := \Psi_2\Psi_4 - {\Psi_3}^3, \tag{2.21}$$

$$N := 12L^2 - {\Psi_4}^2 I, \tag{2.22}$$

to form a set of conditions which lead to an unambiguous determination of the Petrov type.

A number of improvements to this algorithm have since arisen (Fitch (1971), Hon (1975), Åman et al. (1991), and Letniowski and McLenaghan (1988)), focusing on refinements of the given conditions and the practicality of leaving the higher order calculations until as late as possible. In Section 3.4 we describe software which has been written to calculate the Petrov type in Maple. The algorithm used is essentially the version of Letniowski and McLenaghan (1988) and is described in full in Appendix A.

Standard forms for the Weyl spinor

As discussed in the introduction to this chapter, the specification of a standard form for Ψ_{ABCD} can be carried out using spin rotations to fix some subset of its 5 complex valued components $\Psi_0 \ldots \Psi_4$. This is carried out by aligning the spin basis relative to the principle null directions of the Weyl spinor. It is useful to consider each Petrov type individually.

Type N: Using a null rotation about ι^A, we can fix o^A to be a multiple of the single principle spinor. From the expansion (1.55), it is clear that in the new frame the components $\tilde{\Psi}_a$ of the Weyl spinor are all equal to zero except for $\tilde{\Psi}_4$. A combination of spin and boost can be used to fix the value of $\tilde{\Psi}_4$ to 1. The resulting components are

$$\tilde{\Psi}_0 = \tilde{\Psi}_1 = \tilde{\Psi}_2 = \tilde{\Psi}_3 = 0, \quad \tilde{\Psi}_4 = 1, \tag{2.23}$$

which are invariant under the action of null rotations about o^A.

Type III: A null rotation about ι_A can be used to fix o_A to lie in the direction of the repeated principle spinor. A second null rotation, this time about the new $\tilde{o}_A$, brings ι_A in line with the second principle null spinor. A contraction with the Weyl tensor in the new basis reveals that all of the components are zero except for $\tilde{\Psi}_3$. Using the remaining spin and boost freedom we can fix the value of this component to 1. Thus,

$$\tilde{\Psi}_0 = \tilde{\Psi}_1 = \tilde{\Psi}_2 = \tilde{\Psi}_4 = 0, \quad \tilde{\Psi}_3 = 1, \tag{2.24}$$

Note that $\tilde{\Psi}_3$ is not invariant under any of the chosen subgroups of $SL(2,\mathbb{C})$, and by requiring these values for the Weyl components the spin basis is entirely fixed.

Type D: The two null rotations are used successively to bring spin basis vectors into line with the two repeated principle spinors. As a result, the only non-zero component is $\tilde{\Psi}_2$, which is invariant under spins and boosts.

$$\tilde{\Psi}_0 = \tilde{\Psi}_1 = \tilde{\Psi}_3 = \tilde{\Psi}_4 = 0, \quad \tilde{\Psi}_2 \neq 0, \tag{2.25}$$

Type II: A null rotation about ι_A fixes o_A to in the direction of the repeated principle spinor, setting the $\tilde{\Psi}_0$ and $\tilde{\Psi}_1$ components equal to zero in the new frame. If we examine the effect of a null rotation about o_A on the resulting components, we find

that

$$\tilde{\Psi}_2 \mapsto \tilde{\Psi}_2, \tag{2.26a}$$

$$\tilde{\Psi}_3 \mapsto \tilde{\Psi}_3 + 3\alpha\tilde{\Psi}_2, \tag{2.26b}$$

$$\tilde{\Psi}_4 \mapsto \tilde{\Psi}_4 + 4\alpha\tilde{\Psi}_3 + 6\alpha^2\tilde{\Psi}_2. \tag{2.26c}$$

Thus by solving for α using the second of these formulas,

$$\tilde{\Psi}_3 + 3\alpha\tilde{\Psi}_2 = 0, \tag{2.27}$$

we can use the null rotation to align ι_A in such a way that $\tilde{\Psi}_3$ is reduced to zero. A spin and boost is then used to fix the value of $\tilde{\Psi}_4$ to 1, resulting in the standard form

$$\tilde{\Psi}_0 = \tilde{\Psi}_1 = \tilde{\Psi}_3 = 0, \quad \tilde{\Psi}_2 \neq 0, \quad \tilde{\Psi}_4 = 1. \tag{2.28}$$

Since $\tilde{\Psi}_3$ is not invariant under any of the specified subgroups of $SL(2,\mathbb{C})$, the frame is entirely fixed by this standard form.

Type I: A null rotation about o_A with coefficient β given by the solution to the equation

$$\Psi_3\beta^3 + 3\Psi_2\beta^2 + 3\Psi_1\beta + \Psi_0 = 0, \tag{2.29}$$

sets $\tilde{\Psi}_1$ equal to zero. A further null rotation about ι_A with parameter α given by

$$\tilde{\Psi}_0\alpha^3 + 3\tilde{\Psi}_2\alpha + \tilde{\Psi}_3 = 0, \tag{2.30}$$

aligns the basis in such a way that $\tilde{\Psi}_3$ is also zero. The remaining non-zero components transform under spins and boosts as

$$\tilde{\Psi}_0 \mapsto \lambda^4 e^{4i\theta}\tilde{\Psi}_0, \tag{2.31a}$$

$$\tilde{\Psi}_2 \mapsto \tilde{\Psi}_2, \tag{2.31b}$$

$$\tilde{\Psi}_4 \mapsto \lambda^{-4} e^{-4i\theta}\tilde{\Psi}_4, \tag{2.31c}$$

By choosing λ and θ, we are able to set $\tilde{\Psi}_0$ equal to $\tilde{\Psi}_4$. The resulting components are

$$\tilde{\Psi}_1 = \tilde{\Psi}_3 = 0, \quad \tilde{\Psi}_2 \neq 0, \quad \tilde{\Psi}_3 = \tilde{\Psi}_4 \neq 0, \tag{2.32}$$

which together are not invariant under the subgroups of $SL(2,\mathbb{C})$, and so with the components of the Weyl spinor in this form the spin basis is entirely fixed.

Petrov	Weyl components					Remaining
Type	Ψ_0	Ψ_1	Ψ_2	Ψ_3	Ψ_4	Isotropy
I	X	0	Y	0	$\pm X$	-
II	0	0	X	0	1	-
III	0	0	0	1	0	-
D	0	0	X	0	0	spin and boost
N	0	0	0	0	1	2d null
O	0	0	0	0	0	$SL(2,\mathbb{C})$

Table 2.1: Canonical components of the Weyl spinor. Entries X and Y represent independent functions of the coordinates. The isotropy groups are described in the text.

The results of the operations carried out above are summarised in Table 2.1.

The choice of canonical frames specified above does not constitute the only possible choice. In particular, for Type I it may be the case that other forms might be preferred. An alternate standard form would be to set $\tilde{\Psi}_0 = \tilde{\Psi}_4 = 0$ and $\tilde{\Psi}_1 = \tilde{\Psi}_3$. This entails a disadvantage in that the transformation coefficient is, in general, the solution of a quartic equation in the Ψ_4s. However, in some practical situations, a frame may be given in a form for which the Weyl spinor components naturally fall into this latter form. In this case, as we will see below, the equations which must be solved in order to transform the spacetime into the alternate form can be prohibitively difficult.

A more trivial variation of the Type I standard form is to use the spin/boost transformation to set $\tilde{\Psi}_4 = 1$, leaving $\tilde{\Psi}_0$ free. Note that in reducing a Type I spacetime with components

$$\{\Psi_0, \Psi_1, \Psi_2, \Psi_3, \Psi_4\} = \{\Psi_0, 0, \Psi_2, 0, \Psi_4\} \tag{2.33}$$

to its canonical form, a spin/boost transformation is applied, with the parameters

$$\lambda e^{i\theta} = (\Psi_0/\Psi_4)^{1/4}, \tag{2.34}$$

in order to transform the components to the canonical form of Table 2.1. The transformed components become

$$(\Psi_0\Psi_4)^{1/2}, \quad 0, \quad \Psi_2, \quad 0, (\Psi_0\Psi_4)^{1/2}. \tag{2.35}$$

On the other hand, if the canonical form

$$\{\Psi_0, \Psi_1, \Psi_2, \Psi_3, \Psi_4\} = \{1, 0, Y, 0, Z\} \tag{2.36}$$

is chosen, the transformed components will take the form

$$1, \quad 0, \quad \Psi_2, \quad 0, \quad \Psi_0\Psi_4. \tag{2.37}$$

Thus without any computational cost, we've avoided the need to introduce a radical into the expressions for the components. From the standpoint of computer implementation of the procedure, this can be an important difference, as computer algebra systems often run into difficulty with the choice of the correct root in radical expressions. A further advantage will arise when it comes time to calculate the derivatives of the spinor components. Since the derivative of the constant $\tilde{\Psi}_4$ is zero, it is possible that the alternate canonical form will lead to fewer non-zero components in later calculations. The avoidance of the radical can also be an important simplification to the calculation of the derivatives of the Weyl spinor.

Note that although a frame in standard form leads to simplified expressions for its curvature tensor components, it is not always the case that the components of the frame itself will be in their simplest possible form. An alternate standard form may result in simpler expressions for the $l^a, n^a, m^a, \bar{m}^a$ basis vectors. For example, CLASSI currently allows an alternate standard form for Type I spacetimes, namely allowing $\Psi_0 = \pm\Psi_4$. In two practical situations that have arisen (the classification of the cylindrical van Stockum metric (Kramer et al. (1980)) and the Tariq-Tupper spacetime (Tariq and Tupper (1975))), a simplified standard frame is found by setting $\Psi_0 = -\Psi_4$. Similarly, in the Types II and N cases, one might also expect that simpler frames might arise if Ψ_4 is alternately allowed to equal -1, and similarly for Ψ_3 in the Type III case, though no practical examples are known in which this is the case.

In fact, however, by examining the transformation of, for instance, Ψ_0 and Ψ_4,

$$\Psi_0 \mapsto e^{4i\theta}\Psi_0, \qquad \Psi_4 \mapsto e^{-4i\theta}\Psi_4, \tag{2.38}$$

we see that the difference between the two standard forms $\Psi_0 = \pm\Psi_4$ is simply a constant rotation by an angle $\theta = \pi/8$. Further, while alternate standard frames are acceptable when one is trying to determine the independent components specifying the spacetime, if these components are to be compared with those of another spacetime in order to determine a possible coordinate transformation, it is necessary that the two share the same standard form. For these reasons, the current version of the classification software holds to the canonical forms specified in Table 2.1, carrying out the extra $\pi/8$ rotation if necessary.

Transformations to standard form

In Appendix B, the transformations required to bring a general Weyl spinor into the standard forms listed in Table 2.1 are listed. The transformations are given in the form of $SL(2,\mathbb{C})$spin matrices, and the particular transformation to be applied is determined by

i. the Petrov type of the Weyl spinor,

ii. the initial form of the Weyl spinor components.

Here 'initial form' refers to which of the components are zero or non-zero in the initial frame. Thus, for example, if a particular spacetime is determined to be of Type D, and its components are in the form

$$\Psi_0 = \Psi_1 = 0, \quad \Psi_2 \neq \Psi_3 \neq \Psi_4 \neq 0, \tag{2.39}$$

then a transformation to the standard form for Type D is given by

$$\begin{pmatrix} o_A \\ \iota_A \end{pmatrix} \mapsto \begin{pmatrix} f_1 & 0 \\ f_2 & 1/f_1 \end{pmatrix} \begin{pmatrix} o_A \\ \iota_A \end{pmatrix} \tag{2.40}$$

with

$$f_1 = \Big(-\frac{2}{3}\frac{\Psi_3{}^2}{\Psi_2} + \Psi_4 \Big)^{1/4}, \qquad f_2 = -\frac{1}{3}\frac{\Psi_3}{f_1 \Psi_2}. \tag{2.41}$$

Two points are of note. Firstly, explicit transformation functions have not been specified for certain initial configurations of Type I and II spacetime, namely those for which the initial components are in the form

Type	Ψ_0	Ψ_1	Ψ_2	Ψ_3	Ψ_4
I	$\neq 0$	0	0	$\neq 0$	$\neq 0$
I,II	$\neq 0$	0	$\neq 0$	$\neq 0$	$\neq 0$
I,II	$\neq 0$	$\neq 0$	0	$\neq 0$	$\neq 0$
I,II	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$	$\neq 0$

For these cases, the determination of the components of the spin transformation involves the solution of a quartic equation, and thus have not been specified. To date, these types are not handled by any software programs for classification, eg. CLASSI.

Secondly, although the transformations might be determinable, their coefficients may themselves turn out to be quite complicated, to the extent that their application may be limited.

Finally, the spin matrices that bring the Weyl components into canonical form are not unique in the sense that their coefficients are often the solution of some quadratic, cubic, or quartic equation, of which one root must be selected. Selection of alternate roots will lead to the same values for the Weyl coefficients, but they specify different rotations of the frame. Though, by definition, they each result in the same Weyl spinor components, other spinors may differ in their component values depending on which root is chosen. This issue has not arisen in practical situations and as yet it is not clear how it can best be handled.

These problems are fundamental to the Karlhede method and must be solved in order for it to be applicable to truly general situations.

2.2 The Ricci spinor

The classification of the Ricci spinor is a somewhat more involved matter than that of the Weyl spinor as a consequence of the fact that it possesses both primed and unprimed indices. The most widely applied classification scheme was presented by Segré (1884), distinguishing 15 different types of Ricci spinor.

From the standpoint of the Cartan-Karlhede method, a full Segré classification is only required in the case in which the spacetime is conformally flat ($\Psi_{ABCD} = 0$). In this case, the frame retains its full Lorentz freedom, and must be fixed as far as possible by transforming the Ricci components into a canonical form. A suggested set of canonical components is given in Table 2.2, which uses the existing CLASSI forms as a reference.

For the non-trivial Weyl types, it will generally be impossible to bring the Ricci tensor into its canonical form, since there will not be enough freedom left in the frame. In such cases the Segré classification can be foregone, and the methods of the next section applied to find a canonical form for the Ricci tensor by treating it as a general symmetric spinor. In practice, however, since the Segré classification can yield useful physical insight, it is carried out even though the transformation to the components given in Table 2.2 is not applied.

A practical method of carrying out the classification on the computer was presented by Joly and MacCallum (1990) and improved by Seixas (1991). It should be noted that while the algorithm is sufficient to distinguish most Segré types, there are cases in

Segré Type	Common Name		Ricci Components $\Phi_{00'}$	$\Phi_{01'}$	$\Phi_{02'}$	$\Phi_{11'}$	$\Phi_{12'}$	$\Phi_{22'}$	Remaining Isotropy
$[111,1]$	general	g	X	0	$Y \in \mathbb{R}$	Z	0	X	none
$[11,Z\bar{Z}]$		z	$-X$	0	$Y \in \mathbb{R}$	Z	0	X	none
$[11,2]$	PP II	2	0	0	$X \in \mathbb{R}$	Y	0	± 1	none
$[1,3]$	PP III	3	0	$X \notin \mathbb{R}$	$2Y$	Y	0	0	none
$[1(1,2)]$	null	n	0	0	$2X$	X	0	± 1	1d null
$[(1,3)]$	coincident	4	0	0	0	0	1	0	1d null
$[(11),Z\bar{Z}]$	complex	c	$-X$	0	0	Y	0	X	spin
$[(11)2]$	halfway	h	0	0	0	X	0	± 1	spin
$[11(1,1)]$	boost	b	0	0	$X \in \mathbb{R}$	Y	0	0	boost
$[(11)1,1]$	spin	s	X	0	0	Y	0	X	spin
$[(11)(1,1)]$	non-null electromagnetic	e	0	0	0	X	0	0	spin/boost
$[(11,2)]$	pure radiation	r	0	0	0	0	0	± 1	2-d null, spin
$[1(11,1)]$	tachyon fluid	t	$-2X$	0	0	X	0	$-2X$	$SO(2,1)$
$[(111),1]$	perfect fluid	p	$2X$	0	0	X	0	$2X$	$SO(3)$
$[(111,1)]$	vacuum	0	0	0	0	0	0	0	$SL(2,\mathbb{C})$

Table 2.2: Canonical forms for the Ricci spinor. Entries X, Y and Z represent functions of the coordinates. Isotropy groups are described in the text. (Based on Skea (1996))

which the distinguishing factor between one and another case is the sign of one of the canonical components. Since computer algebra systems cannot generally make such an evaluation for symbolic quantities, these cases can not be distinguished algorithmically. These difficulties are discussed in Skea (1996).

2.3 Standard forms for general spinors

To this point, we have specified standard forms for the Weyl and Ricci spinors. By requiring that the frame be such that the components of these tensors satisfy the given standard forms, the rotational freedom is reduced to a subgroup of $SL(2,\mathbb{C})$. In fact, for most algebraic types of spacetime, the frame is immediately fixed by the form of the Weyl and Ricci spinor and it is only for the particular cases listed in Table 2.3 that there is any rotational freedom left in the frame. For such cases, it is necessary to examine the derivatives of the curvature spinors in order to determine if the frame can be further fixed. Thus, in order to complete the process of frame fixing, what is required is a method of determining the invariance groups of spinor components, and a set of consistent standard forms for general spinors of valence (m,n).

Weyl types	Ricci types	Freedom
0	p, 0	$SO(3)$
0	t, 0	$SO(2,1)$
N, 0	r, 0	2d null
N, 0	r, 4, n, 0	1d null
D, 0	e, b, 0	boost
D, 0	r, e, s, h, c, 0	spin
I, D, 0	p, t, e, b, z, g, 0	$o - \iota$ interchange

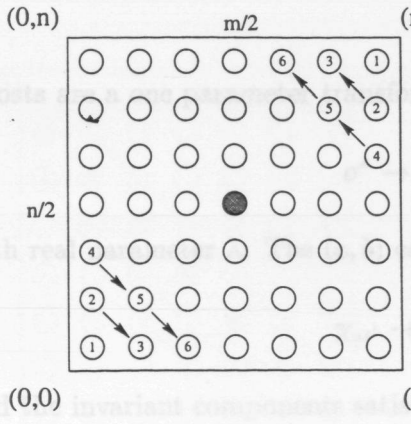
Table 2.3: Subgroups of $SL(2,\mathbb{C})$ under which the Weyl and Ricci components may be invariant. In such cases, canonical forms for the higher derivatives of the curvature spinors must be specified.

The general procedure begins by noting which components are invariant under the given subgroup. Once these are known, a method can be specified for stepping through the components to determine a unique component which does vary under the spin transformation. An algebraic condition is then placed on the component and the frame transformed to ensure that the condition is satisfied, removing the group freedom. It is necessary that the procedure for locating this component be unambiguous in the sense that although many of the components may vary under the given transformation, the procedure must always locate and fix the same component of a particular spinor in a given spacetime. This can usually be done by specifying one component as a starting point, and a procedure for stepping through the remaining spinor components.

The following sections consider each subgroup of $SL(2,\mathbb{C})$ in turn, detailing how the invariance can be detected in a general spinor, and, given that the spinor is not invariant, how the group is used to fix the spinor into a standard form specific to that rotation group.

$o - \iota$ *interchange*

Once the Weyl tensor has been classified, this is the only discrete transformation under which a spacetime may be invariant. For a spinor χ of valence (m,n), interchange of o^A and ι^A has the effect of swapping the $\chi_{ab'}$ component with the $\chi_{(m-a)(n-b)'}$ component. The test for $o - \iota$ invariance consists of checking each such pair in turn. If one of the pair is zero and the other non-zero, then we choose to be biased towards the upper diagonal, so that o^A and ι^A are selected to make the component with higher indices non-zero.



Test sequence:
for $a = 0 \ldots (m+n)/2$ *do*
  for $b = \min(a,n) \ldots 0$ *do*
    $C_1 = \chi_{(a-b)b'}$,
    $C_2 = \chi_{(m-a+b)(n-b)'}$
    if $C_1 = 0$ *and* $C_2 \neq 0$ *then*
      standard form = False
    else if $C_1 \neq 0$ *and* $C_2 = 0$ *then*
      standard form = True
    else
      Continue
    end if
  end do
end do

Figure 2.1: The standard form for the $\iota - o$ interchange. A search for non-invariant components begins with the $(0,0)$ and (m,n) components and proceeds along diagonals, comparing cross diagonal pairs. If the component in the lower diagonal is non-zero, the spinor is in canonical form (*True*). If the component in the upper diagonal is non-zero, the spinor is not in canonical form (*False*) and an $o - \iota$ swap transformation can be applied to bring it into canonical form. If both components are non-zero, the form is indeterminate and the test continues to the next pair.

The test begins with a comparison of the $(0,0)$ and (m,n) components, and proceeds by stepping first the unprimed, then the primed indices until the first $o - \iota$ asymmetry is detected.

For example, if χ is a spinor of valence $(4,4)$, first the pair $\chi_{00'}$ and $\chi_{44'}$ are checked, then $\chi_{01'}$ and $\chi_{43'}$, etc., until all the test for $o-\iota$ symmetry fails, or all of the components have been checked. If, for instance, the next component to be checked, $\chi_{20'}$, is non-zero while its 'partner', $\chi_{24'}$, is zero, then the $o-\iota$ symmetry is lost. In this case, the canonical preference for non-zero components in the upper indices means that o and ι would be swapped to bring the spinors into canonical form.

Note that the given test does not necessarily determine 'invariance', for if a given pair of components are both non-zero but not equal, then the spinor is clearly not invariant, but this will not be detected. However, in such cases it is ambiguous as to which component should preferably sit in the upper diagonal. That is, it is difficult to specify a consistent standard form for components which differ in this way. From an algorithmic point of view, the best that can be done is to keep track of the fact that this freedom has not yet been fixed, as it is not a difficult matter to transform the spacetime under this discrete symmetry later on, if need be.

Boosts

Boosts are a one parameter transformation under which the spin basis transforms as

$$o^A \to \lambda o^a, \qquad \iota^A \to \lambda^{-1}\iota^A, \tag{2.42}$$

with real parameter λ. The (a, b) component of a symmetric spinor transforms as

$$\chi_{ab'} \to \lambda^{a-(m-a)+b-(n-b)}\chi_{ab'}, \tag{2.43}$$

and the invariant components satisfy

$$a + b = \frac{1}{2}(m + n). \tag{2.44}$$

The boost transformation can be used to fix the magnitude of individual spinor components. A useful standard form for boosts is suggested when we note that in practical situations, it is often the case that a spinor is presented in a frame such that the magnitude of cross-diagonal components is equal. By requiring that this be true for some given cross-diagonal pair $\chi_{ab'}$ and $\chi_{(m-b)(n-a)'}$ it is possible to solve for the transformation parameter λ via

$$\lambda^{a-(m-a)+b-(n-b)}|\chi_{ab'}| = \lambda^{(m-b)-b+(n-a)-a}|\chi_{(m-b)(n-a)'}|, \tag{2.45}$$

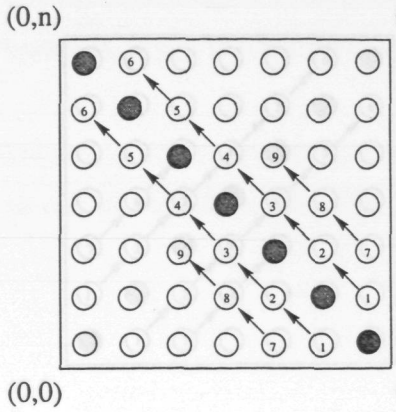
or,

$$\lambda^{4(a+b)-2(m+n)} = \frac{|\chi_{(m-b)(n-a)'}|}{|\chi_{ab'}|} \tag{2.46}$$

If the two components already have equal magnitude, as is often the case, then the spinor is already in standard form, $\lambda = 1$, and no transformation is necessary.

For spinors which are not automatically in standard form, it is significant that the exponent for λ has its maximum value for the $(0, 0)$ and (m, n) components. Thus by requiring that this pair have equal magnitude, it is likely that the solution for λ will be a high order radical. A better choice would be a component pair closer to the invariant diagonal. For instance, along the first diagonal the components are transformed as a quadratic in λ, while the order increases by a factor of two with each step away from the invariant diagonal.

Resulting from these considerations, the standard form compares pairs of components across the invariant diagonal starting with components closest to the invariant diagonal. The first pair in which at least one of the members is non-zero is used to fix the canonical form by requiring that the pair have equal magnitude. If one of the pair is zero, then the requirement is that the other have magnitude one.



Test sequence:
for $a = (m+n-1)/2 \ldots 0$ *do*
  $c = \max(0, n-a)$
  for $b = 0 \ldots n-c$ *do*
    $C_1 = \chi_{(a-b)b'}$,
    $C_2 = \chi_{(m+n-a-b-c)(b+c)'}$
    if $C_1 \neq 0$ *or* $C_2 \neq 0$ *then*
      if $|C_1| = |C_2|$ *or*
        $(C_1 C_2 = 0$ *and* $|C_1 + C_2| = 1)$ *then*
        standard form = True
      else
        standard form = False
      end if
    else
      Continue
    end if
  end do
end do

Figure 2.2: The standard form for boost transformations. Components are checked along the $a+b = constant$ diagonals, moving outwards from the invariant diagonal. The standard form requires that the first non-zero component pair have equal magnitude, or in the case that one is zero then the other must have a magnitude of one. Invariant components are marked in grey in the figure.

Spins

Spins are a one parameter group representing the transformations

$$o^A \to e^{i\theta} o^A, \qquad \iota^A \to e^{-i\theta} \iota^A. \tag{2.47}$$

Under such a transformation, the $\chi_{ab'}$ component of a spinor are changed by a phase factor

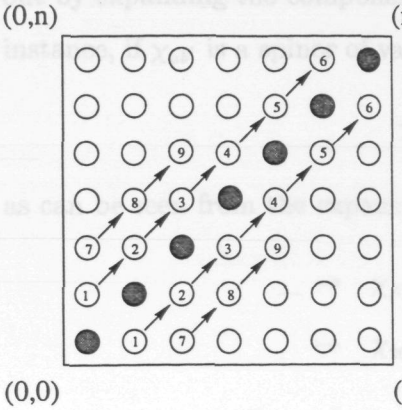
$$\chi_{ab'} \to e^{i(a-(m-a)-b+(n-b))\theta} \chi_{ab'} \tag{2.48}$$

where (m, n) is the valence of χ. The components which are invariant under (2.48) are the ones for which (a, b) satisfy

$$a - b = \frac{1}{2}(m-n) \tag{2.49}$$

(for hermitian spinors, these are the components on the diagonal).

The spin freedom can be fixed by specifying the phase of some non-zero, non-invariant component. In order to determine which component to fix, note that in this case it is not as important to reduce the exponent of the transformation, as it only affects the multiple of some angle θ which is applied. In the example displayed in Figure 2.3, the first diagonal (components 1 to 6) are transformed by an angle 2θ, while those in the corners ($\chi_{60'}$ and $\chi_{06'}$) are transformed by an angle 12θ. Effectively, there should be no



Test sequence:
for $a = (m-n)/2+1 \ldots m$ *do*
  $c = \min(n, m-a)$
  for $b = 0 \ldots c$ *to*
    $C_1 = \chi_{(a+b)b'}$,
    $C_2 = \chi_{(m-c-a+b)(n-c+b)'}$
    if $C_1 \neq 0$ *or* $C_2 \neq 0$ *then*
      if phase(C_1) = *phase*(C_2) *or*
        $(C_1C_2{=}0$ *and* $C_1 + C_2 \in \mathbb{R})$ *then*
        standard form = True
      else
        standard form = False
      end if
    else
      Continue
    end if
  end do
end do

Figure 2.3: The standard form for spin transformations. Components are checked along the $a - b = constant$ diagonals, moving outwards from the invariant diagonal. The standard form requires that the first non-zero component pair have equal phase, or in the case that one is zero, the other must be real. Invariant components are marked in grey in the figure.

extra difficulty in solving for the exponent of the latter. However, for consistency with the boost case, we choose to step first along diagonals nearest to the invariant diagonal, thus ensuring that the multiplier is as small as possible when solving for θ.

The search is carried out by examining pairs of components across the diagonal until one or the other of a pair is non-zero. The canonical form requires that both members of the pair have equal phase (if the spinor is hermitian, this implies that the components be real and equal). If one of the pair is zero, the other is required to be real.

Another potential standard form would be, for instance, to require that one of the pair be pure real. It is not clear that this brings any particular advantage over the chosen form. Further, in practical situations it is often the case that a spinor is presented in a frame in which the cross diagonal components are already of equal phase, and thus already in the suggested canonical form.

Null rotations

Null rotational freedom forms a slightly more complicated class than the groups studied so far. Rather than simply multiplying each component by a factor raised to some exponent, the image of the transformation is a polynomial in the non-zero coefficients.

Consider first the null rotation about the o^A spinor,

$$o^A \to o^A, \qquad \iota^A \to \iota^A + \alpha o^A, \qquad \alpha \in \mathbb{C}. \tag{2.50}$$

The transformation of a particular spinor component $\chi_{ab'}$ can most easily be worked out by expanding the component in terms of terms of the basis spinors, o^A and ι^A. For instance, if $\chi_{ab'}$ is a spinor of valence $(2,2)$, then the $(2,0)$ component can be written as

$$\chi_{20'} = \chi_{ABA'B'}\iota^A\iota^B\bar{o}^{A'}\bar{o}^{B'}, \tag{2.51a}$$

as can be seen from the expansion (1.55) Under (2.50) this component transforms to

$$\rightarrow \quad \chi_{ABA'B'}(\iota^A + \alpha o^A)(\iota^B + \alpha o^B)\bar{o}^{A'}\bar{o}^{B'} \tag{2.51b}$$

$$\rightarrow \quad \chi_{20'} + 2\alpha\chi_{10'} + \alpha^2\chi_{00'}. \tag{2.51c}$$

For a symmetric spinor of valence (m,n), the general formula for the transformed (a,b) component is given by

$$\chi_{ab'} \rightarrow \sum_{r=0}^{a}\sum_{s=0}^{b}\binom{a}{r}\binom{b}{s}\alpha^r\bar{\alpha}^s\chi_{(a-r)(b-s)'}. \tag{2.52}$$

For the opposite form of null rotation ($o^A \rightarrow o^A + \beta\iota^A, \iota^A \rightarrow \iota^A$) the analogous formula is found by replacing a with $m-a$ and b with $n-b$.

To define the standard form appropriately, we must determine which of the spinor components are invariant under the null rotations and how the remaining components can be manipulated. Consider first the expansion of the transformed coefficient with the highest index values, namely $\chi_{mn'}$,

$$\begin{aligned}\chi_{mn'} \rightarrow\ & \chi_{mn'} + m\alpha\chi_{(m-1)n'} + n\bar{\alpha}\chi_{m(m-1)'} + \frac{1}{2}m(m-1)\alpha^2\chi_{(m-2)n'} \\ & + mn\alpha\bar{\alpha}\chi_{(m-1)(n-1)'} + \frac{1}{2}n(n-1)\bar{\alpha}^2\chi_{m(n-2)'} + \ldots\end{aligned} \tag{2.53}$$

Since this is a polynomial in each of the coefficients of χ, it is clear that if any of the $\chi_{ab'}$ are non-zero, then the value of $\chi_{mn'}$ will be altered by the null rotation. Thus the condition for $\chi_{mn'}$ to be invariant under (2.50) is that it be the only non-zero component. In this case the spinor itself, $\chi_{ABA'B'}$, will clearly be invariant.

There are other situations, however, when the spinor is invariant, namely under certain 1-parameter subgroups of the null rotations. Consider first the case in which α is pure real and examine the resulting transformation of the $\chi_{mn'}$ component:

$$\begin{aligned}\chi_{mn'} \rightarrow\ & \chi_{mn'} + \alpha\left(m\chi_{(m-1)n'} + n\chi_{m(n-1)'}\right) \\ & + \frac{1}{2}\alpha^2\left(m(m-1)\chi_{(m-2)n'} + 2mn\chi_{(m-1)(n-1)'} + n(n-1)\chi_{m(n-2)'}\right] + \\ & + \frac{1}{6}\alpha^3\left[m(m-1)(m-2)\chi_{(m-3)n'} + 3m(m-1)n\chi_{(m-2)(n-1)'} + \right. \\ & \left. +3mn(n-1)\chi_{(m-1)(n-2)'} + +n(n-1)(n-2)\chi_{m(n-3)'}\right] + \ldots \\ & \ldots + \alpha^{m+n}\chi_{00'}.\end{aligned} \tag{2.54}$$

Since α can be chosen arbitrarily, each of the terms must equal zero if $\chi_{mn'}$ is to be invariant. We arrive at the conditions

$$m\chi_{(m-1)n'} + n\chi_{m(n-1)'} = 0, \tag{2.55a}$$

$$m(m-1)\chi_{(m-2)n'} + 2mn\chi_{(m-1)(n-1)'} + n(n-1)\chi_{m(n-2)'} = 0, \tag{2.55b}$$

$$m(m-1)(m-2)\chi_{(m-3)n'} + 3m(m-1)n\chi_{(m-2)(n-1)'} + \\ +3mn(n-1)\chi_{(m-1)(n-2)'} + n(n-1)(n-2)\chi_{m(n-3)'} = 0, \tag{2.55c}$$

$$\vdots$$

$$\chi_{00'} = 0. \tag{2.55d}$$

The transformations of the other components of χ give further conditions. Notably, from $\chi_{(m-1)n'}$, we find

$$(m-1)\chi_{(m-2)n'} + n\chi_{(m-1)(n-1)'} = 0, \tag{2.56a}$$

$$(m-1)(m-2)\chi_{(m-3)n'} + 2(m-1)n\chi_{(m-2)(n-1)'} + n(n-1)\chi_{(m-1)(n-2)'} = 0, \tag{2.56b}$$

$$\vdots$$

From the transformation of $\chi_{m(n-1)'}$,

$$m\chi_{(m-1)(n-1)'} + (n-1)\chi_{m(n-2)'} = 0, \tag{2.57a}$$

$$m(m-1)\chi_{(m-2)(n-1)'} + 2m(n-1)\chi_{(m-1)(n-2)'} + (n-1)(n-2)\chi_{m(n-3)'} = 0, \tag{2.57b}$$

$$\vdots$$

And from the first order in α terms of the transformations of $\chi_{(m-2)n'}$, $\chi_{(m-1)(n-q1)'}$ and $\chi_{m(n-2)'}$, we find

$$(m-2)\chi_{(m-3)n'} + n\chi_{(m-2)(n-1)'} = 0, \tag{2.58a}$$

$$(m-1)\chi_{(m-2)(n-1)'} + (n-1)\chi_{(m-1)(n-2)'} = 0, \tag{2.58b}$$

$$m\chi_{(m-1)(n-2)'} + (n-2)\chi_{m(n-3)'} = 0. \tag{2.58c}$$

Equation (2.55a) give the only condition on the pair of components $\{\chi_{(m-1)n'}, \chi_{m(n-1)'}\}$ while equations (2.55b),(2.56a),(2.57a) are three compatible restrictions on the components $\{\chi_{(m-2)n'}, \chi_{(m-1)(n-1)'}, \chi_{m(n-2)'}\}$. However, equations (2.55c),(2.56b),(2.57b),(2.58a),(2.58b), (2.58c) are a set of six conditions on the four components $\chi_{(m-3)n'}$, $\chi_{(m-2)(n-1)'}$, $\chi_{(m-1)(n-2)'}$, $\chi_{m(n-3)'}$. These components must therefore be zero.

We arrive at the following result: A symmetric spinor χ of valence (m, n) is invariant under null rotations of type (2.50) with real parameter α if and only if the following

conditions are satisfied:

$$\chi_{(m-1)(n-1)'} = -\frac{n-1}{m}\chi_{m(n-2)'} = -\frac{m-1}{n}\chi_{(m-2)n'}, \tag{2.59a}$$

$$m\chi_{(m-1)n'} = -n\chi_{m(n-1)'}, \tag{2.59b}$$

$$\chi_{ab'} = 0 \quad \text{otherwise.} \tag{2.59c}$$

Corresponding conditions exist for the case where α is allowed to be any other 1-parameter subgroup of (2.50). For instance, in the case that α is pure imaginary, we have:

$$\chi_{(m-1)(n-1)'} = \frac{n-1}{m}\chi_{m(n-2)'} = \frac{m-1}{n}\chi_{(m-2)n'}, \tag{2.60a}$$

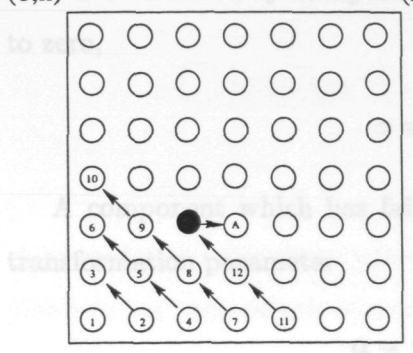
$$m\chi_{(m-1)n'} = n\chi_{m(n-1)'}, \tag{2.60b}$$

$$\chi_{ab'} = 0 \quad \text{otherwise.} \tag{2.60c}$$

The current code is able to detect whether the components are invariant under the general 1-parameter null rotations, however canonical forms are only defined for the case for which the parameter is purely real or purely imaginary.

The problem of finding an appropriate standard form is more involved in the case of null rotations than for the previously considered isotropy groups. One possibility is simply to use the null rotation to set the value of the $\chi_{mn'}$ component, so that in the case of 2-dim null isotropy, $\chi_{mn'}$ would be set to zero, while in the 1-dim case, either its real or complex parts would be eliminated. The advantage of this choice of standard form is that it is easily recognisable by examining a single component of the tensor in question. The major drawback, however, is that if the spinor is not already in its standard form, then a value of α must be determined to bring it into its standard form by setting equation (2.54) to zero. Of all of the components of $\chi_{ABA'B'}$, the transformation of the (m,n) component involves the highest order polynomial in α, and is generally the most complicated, every other non-zero component contributing. From a practical standpoint, then, it may prove impossible to determine an appropriate α to bring the spinor into its standard form.

Another potential standard form involves searching for the non-invariant component with the lowest order transformation function. To do this, the components of the spinor are stepped through beginning with the $(0,0)$ component and proceeding along diagonals as depicted in Figure 2.3, until the first non-zero component is reached. The diagonal in which this component appears is invariant since each term in its transformation function involves a spinor component 'above and to the left,' all of which are known to be zero.




```
Test sequence:
for a = 0...m + n − 1do
  c = max(0, a − m)
  for b = 0...min(a, n) − c do
    if χ_{(a−b−c)(b+c)'} ≠ 0 then
      for ã = a...m + n − 1 do
        c̃ = max(0, ã − m)
        for b̃ = 0...min(ã, n) − c̃ do
          if Re-invariance test fails and
            Re(χ_{ãb̃'}) = 0 then
            R=True
          else
            R=False
          end if
          if Im-invariance test fails and
            Im(χ_{ãb̃'}) = 0 then
            I=True
          else
            I=False
          end if
          if R and I are assigned then
            re-standard form = R
            im-standard form = I
          else
            Continue
          end if
        end do
      end do
    end if
  end do
end do
```

Figure 2.4: The standard form for null rotations about o^A. Components are searched along the diagonals beginning with $\chi_{00'}$. When the first non-zero component is found (B in the diagram), the next diagonal is checked for its invariance properties under null rotations, beginning with the first non-invariant component A. The spinor is in canonical form if the first component which fails the test for invariance under real parameters α (equation (2.61a)) has a zero real part, and the first component which fails the test for invariance under imaginary parameters (equation (2.61a)) has a zero imaginary part. In this case the value *True* is returned. Components continue to be checked until both invariance types have been established.

The next diagonal is the first non-invariant diagonal. The transformations of components in this diagonal are linear functions of the rotation parameter α. For each component $\chi_{ab'}$ along this diagonal, the tests

$$a\chi_{(a-1)b'} = b\chi_{a(b-1)'}, \quad \text{and} \tag{2.61a}$$

$$a\chi_{(a-1)b'} = -b\chi_{a(b-1)'}, \tag{2.61b}$$

are carried out. If the former fails, then the component varies under the real part of α. If the latter fails, then the component varies under the imaginary part of α. If neither fails, then the next component in the diagonal is checked and the process continued until a non-invariant component is found. For the first component $\chi_{ab'}$ which fails the imaginary

invariance test (2.61b), the standard form is determined by fixing the imaginary part of α so that the imaginary part of $\chi_{ab'}$ is transformed to zero,

$$\alpha = \frac{\chi_{ab'}\bar{\chi}_{a(b-1)'} - \bar{\chi}_{ab'}\chi_{a(b-1)'}}{2b|\chi_{a(b-1)'}|^2}. \tag{2.62}$$

For the first component $\chi_{ab'}$ which fails the real invariance test, (2.61a), the standard form is determined by fixing the real part of α so that the real part of $\chi_{ab'}$ is transformed to zero,

$$\alpha = \frac{\chi_{ab'}\bar{\chi}_{a(b-1)'} + \bar{\chi}_{ab'}\chi_{a(b-1)'}}{2b|\chi_{a(b-1)'}|^2}. \tag{2.63}$$

A component which has failed both invariance tests can be set to zero using the transformation parameter

$$\alpha = \frac{-a\chi_{ab'}\bar{\chi}_{(a-1)b'} + b\bar{\chi}_{ab'}\chi_{a(b-1)'}}{a^2|\chi_{(a-1)b)'}|^2 + b^2|\chi_{a(b-1))'}|^2}. \tag{2.64}$$

We have seen, then, that for any of the possible isometry groups which remain once the Weyl spinor has been fixed, a canonical form for general symmetric spinors can be defined which fixes the components of the spinor relative to the isometry group. To implement the Cartan-Karlhede method, then, there remains to choose an appropriate set of spinors, and an order for which the isometry tests are to be applied to produce a unambiguous set of components which can be compared between different spacetimes. A practical procedure for carrying out this comparison is provided in the next chapter.

3

Classification in practice

A procedure for classifying general spacetimes using the components of their spinor components has been described in the previous chapter. Unfortunately, the technical effort involved in carrying out this procedure can be enormous, even for very simple classes of spacetimes, due to the need to take multiple derivatives and perform transformations on a large number of spinor components. This type of operation, however, is ideally suited to computer algebra systems, which are able to carry out the necessary multiplications and expansions with great speed and accuracy.

The first to suggest the use of a computer in solving the equivalence problem was Brans (1965), who developed a modified form of Cartan's method involving fixing the frame using the highest derivatives of the curvature tensors. With the introduction of Karlhede's method, the first serious effort at using computer algebra to solve the equivalence problem was carried out in Stockholm, with the major development work being done by Jan Åman (Åman and Karlhede (1980), Åman (1986)). Over the years a number of other authors have contributed to produce the package known as CLASSI. CLASSI is a Lisp-based system, derived from the previously existing computer algebra system SHEEP. As such, it has a number of beneficial features, including speed, compactness and, after a long development cycle in which it has been quite thoroughly tested, accuracy. Its practical use has been demonstrated by its authors by the compilation of a large database of fully classified exact solutions Skea (1997b).

In compiling this database, however, certain deficiencies in the software became clear. First of all, the SHEEP code which carries out the computer algebra was developed in the early days of computer algebra by researchers working in relativity. As such, a number of modern features which one has come to expect in more powerful systems are not present within SHEEP, such as the ability to carry out polynomial divisions and

factorisations. This limits CLASSI's ability to carry out certain simplifications which would greatly ease the classification of some spacetimes. In practice, this has meant that a great deal of work must be done by the user in configuring the input so that the resulting expressions can be handled. This can be especially difficult for users who are not proficient in Lisp or the subtleties of the SHEEP system. The use of the software thus entails a steep learning curve which has limited its widespread acceptance.

Further, the techniques used by the CLASSI software have not been fully elucidated in any publication or user manual. This includes specification of the standard forms which it uses for the classifications of high order spinors. This has made results from the software difficult to verify. In fact, the canonical forms can not be determined without reading the source code which performs the transformations. This has also influenced its reliability. For some time the electromagnetic Type N solutions given by Eq. 32.61 of Kramer et al. (1980) could not be correctly put into a standard form. The source of the problem was difficult to determine, since the intended standard form was not known. It was only through a re-examination of how a standard form for null rotations might be defined that it was determined that an error existed in CLASSI's specification of a standard form for that particular isometry.

For these reasons, it was decided that the classification methods should be revisited, with the aim of producing an updated package. This work was begun in 1997 as a collaboration between Ray d'Inverno and Jim Skea (two of the original developers of the SHEEP and CLASSI systems), and Kayll Lake and the author, who have, with Peter Musgrave, developed an alternate tensor calculation within Maple, called GRTensor. It was hoped that by re-examining the various aspects of the classification methods, a better understanding of how they might best be carried out could be obtained. In addition, the methods could be fully elucidated and explained, and particular sticking points highlighted, and possibly overcome. The result of this work is the spinor computer algebra package, which works within Maple in conjunction with GRTensor, and which will be described in this chapter.

3.1 The choice of platform

One of the main objectives of the current work is to place the classification methods in the context of a powerful modern computer algebra system.

A great deal of progress has been made in the field of computer algebra since the appearance of the first SHEEP systems in the 1970s. Modern systems possess well

developed, fast, efficient, and reliable algorithms for performing the computationally intensive algebraic tasks required by the classification methods. The development of these algorithms is a highly technical science, and one for which a specialised knowledge is required. The effort involved in developing such a system from scratch is enormous. It is thus a great advantage if one can make use of a system which already exists.

Further, modern computer algebra systems generally are supplied with advanced user interfaces, making use of displayed PostScript for the output of equations. They also possess various tools for the formatting of output, allowing equations to be converted to a variety of formats (eg. LaTeX) for text processing, or C or Fortran code for implementation in numerical codes.

Among such systems are the packages Axiom, Macsyma, Maple, Mathematica, MuPad, and Reduce. Each of these packages possesses similar features and capabilities. Determining which of these is the most appropriate based on purely technical grounds is a highly controversial business. Attempts have been made at providing independent test suites of problems to test the capabilities of computer algebra systems. However, the results can usually be questioned on the grounds that they emphasise particular types of problems, or that the solutions may have simply not been implemented correctly in a given system. In fact, because each of the named systems provide their own command language, it can be argued that any problem that can be solved in one is likely to be soluble in another by writing a program corresponding to the given algorithm. The distinguishing aspect between the different systems then becomes its 'ease of use', a somewhat less quantifiable property.

There are a number of other non-technical aspects that must be considered. The most important of these is availability. Each of the systems above is a commercial package, and as such must be purchased (the exception being the relatively new system, MuPad, which is available free of charge to academic institutions). The extent to which a system is used will depend on whether a researcher will have access to the software through an institutional license, or whether they must purchase the software individually themselves (as is often the case with students). In the latter case, it is important that the chosen system have fully functional editions available at a low price.

In the original development of the GRTensor system, it was decided that of the systems available at the time, MapleV formed the most appropriate platform. It is a powerful system with a simple user interface and interpreted programming language. In tests of simplification of large polynomials (crucial for work with tensors), it significantly outperformed the concurrent editions of Mathematica and Macsyma. It provides well

developed interfaces for the output of data to LaTeX and other programming languages. Further, Maple has a widespread acceptance in academic faculties. It is common for maths and physics faculties to possess site-wide licences, so that researchers and students are able to make use of the software at no personal expense. The added benefit of this wide acceptance is that that code written for Maple on a particular machine can be run on any other Maple system, so in this sense the programs become platform-independent. These aspects singled out Maple as a quite sensible choice for development of a tensor algebra system.

There are, however, certain drawbacks in writing code for a single computer algebra system, and in particular Maple. The greatest of these is that there is no control of the development of the underlying platform. Maple continues to be developed, and with each new release software written for it must be updated and re-tested. Although later versions of Maple attempt to remain consistent in their programming language with earlier versions, this has never worked with absolute accuracy, and with each new Maple release from Release 2 (in which the original versions of GRTensor were written) to Release 6 (the current Maple release which is expected to complete beta testing in March 2000) changes to the GRTensor code have been required. More disturbing, later editions have Maple have not always proved to be faster or more efficient in performing individual calculations, in some cases quite the contrary.

Maple itself also has some inconvenient properties in its handling of algebraic expressions. The largest of these is that the ordering of expressions is not determined lexically, but rather is dependent on the memory addressing of individual quantities. As such, calculations carried out in subsequent Maple sessions can seem quite different in appearance. Code which performs substitutions based on the ordering of expressions will also not work reliably. Radicals can also prove to be a problem, in particular the identification of expressions such as

$$\sqrt{x-y}, \qquad i\sqrt{y-x}. \tag{3.1}$$

In terms where both forms of such components arise, it can require a great deal of effort to convince Maple to perform simplifications which are obvious to the user.

It must be realized, however, that each of the packages in question possess similar disadvantages arising from their individual behaviour, and that for the algebraic operations in question, Maple has generally been found to outperform its rivals.

3.2 GRTensorII

The GRTensor package was originally developed in 1991 by Peter Musgrave and Kayll Lake, as a means of calculating curvature tensors and scalar polynomial invariants in general relativity. Although it proved to be both fast and powerful in calculating the objects for which it was designed, its data structures and interface left little room for expansion of its capabilities. As such, in collaboration with the author, in 1993 development switched to a more ambitious package, called GRTensorII. The new package included the ability to calculate tensors in any index configuration, and allowed users to define their own tensors using a simple interface. Further, support for tetrad formalisms, and in particular the Newman-Penrose formalism, was added.

Basic commands

The design philosophy of GRTensor[1] can be embodied in the following points:

i. No quantity should need to be computed twice. Index symmetries are used whenever possible. Intermediate objects (such as the Christoffel symbols in the calculation of the Riemann tensor) are stored and re-used when required.

ii. The user should have control over each step of the calculation. Rather than calculating the Riemann tensor in a single step, the user is able to first calculate and examine the intermediate objects, to which simplifications and substitutions can be applied.

The importance of the second point can not be overstated. In fact, using GRTensor it has been possible to show that the most significant aspect in improving the speed of a tensor calculation are the simplifications that are applied at each stage, independent of the particular summation algorithm or formalism in which the components are calculated Pollney et al. (1996).

The user interface of GRTensor is built around three principle commands,

grcalc calculates the components of tensors;

grdisplay displays the components;

gralter applies simplification routines to tensors.

[1]The package GRTensor is understood to refer to GRTensorII, as the original MapleV.2 package is defunct.

Tensor names, with their index configurations, are supplied as arguments to these commands. For example, the Ricci tensor can be referenced as

$$\texttt{R(dn,dn)} \quad (= R_{ab}), \qquad \texttt{R(up,dn)} \quad (= R^{a}{}_{b}), \qquad \texttt{R(bup,bdn)} \quad (= R^{\mathtt{a}}{}_{\mathtt{b}}),$$

where the last example uses the labels `bup` and `bdn` to reference the components of the Ricci tensor in terms of a basis rather than metric coordinates.

Thus, to calculate the Ricci tensor for a given spacetime, apply trigonometric simplification, and then display the resulting components, the following set of commands can be used:

```
grcalc ( R(dn,dn) );
gralter ( R(dn,dn), trig );
grdisplay ( R(dn,dn) );
```

Additional commands exist to carry out further manipulations of the output. The `grcomponent()`command can be used to assign tensor components to a Maple variable. The `grmap()`command can be used to apply any Maple function to each component of a tensor.

Input of spacetimes

The primary input to GRTensor is a file specifying the spacetime geometry either in the form of a metric, g_{ab}, or frame, $e_{\mathtt{a}}{}^{b}$, with basis vectors satisfying some constant valued inner product,[2] $\eta_{\mathtt{ab}} := e_{\mathtt{a}}{}^{a} e_{\mathtt{b}a}$. An example of an input file for the Tolman dust solution is given in Figure (3.1). The coordinates are specified by the variables `x1_` to `x4_`, and the covariant metric tensor components, g_{ab}, are contained in the variables `g11_` to `g44_`. Note that only the non-zero, upper-diagonal components are stored. The optional variable `sig_` can be used to specify the signature of the metric. The `constraint_` variable exists if there are additional conditions which are satisfied by the metric functions which the user might wish to supply at some time later in the calculation. Finally, the `Info_` variable can be used to store some descriptive information regarding the spacetime. The particular example can be stored in ASCII form as , for instance, `dust1.mpl`. The command `qload(dust1)` loads the file into GRTensor and assigns the individual tensor components to their respective internal data structures.

[2] In fact, certain fundamental tensors, such as the Riemann and Ricci tensors and their polynomial invariants, can be calculated with a non-constant basis inner product. The usefulness of such definitions was suggested to the authors by C. W. Misner (personal communication) in relation to some calculations

```
Ndim_ := 4:
x1_ := r:
x2_ := theta:
x3_ := phi:
x4_ := t:
sig_ := 2:
g11_ := diff ( R(r,t),r)^2/(1+f(r) ):
g22_ := R(r,t)^2:
g33_ := R(r,t)^2*sin(theta)^2:
g44_ := -1:
constraint_ :=[ diff(diff(R(r,t),r),t) = (2*diff(m(r),r)/R(r,t)
                - 2*m(r)*diff(R(r,t),r)/R(r,t)^2
                + diff(f(r),r))/(2*sqrt(2*m(r)/R(r,t)+f(r))),
                diff(R(r,t),t) = sqrt(2*m(r)/R(r,t)+f(r)),
                diff(diff(R(r,t),t),t) = -m(r)/R(r,t)^2,
                diff(diff(diff(R(r,t),t),r),t) = -diff(m(r),r)/R(r,t)^2 +
                2*m(r)*diff(R(r,t),r)/R(r,t)^3
              ]:
Info_:= 'The Tolman dust solution (Proc. Nat. Acad. Sci. 20, 169,1934)':
```

Figure 3.1: The metric file `dust1.mpl` from the standard metric library.

Tensor definitions within GRTensor

The tensor definitions themselves are stored in the `grG_ObjDef` data structure. Individual tensors are referenced using the tensor name, so that `grG_ObjDef[G(dn,dn)]` holds the definition for the covariant components of the Einstein tensor. A tensor definition consists of a number of fields giving information as to how a tensor is to be calculated, displayed, and referenced. The `grC_depends` field contains a set of tensors on which the given tensor depends. Thus, for the Einstein tensor, the line

```
grG_ObjDef[G(dn,dn)][grC_depends] := {g(dn,dn), R(dn,dn), Ricciscalar}:
```

indicates that in order to calculate `G(dn,dn)`, the objects `g(dn,dn)`, `R(dn,dn)`, and `Ricciscalar` must first be known. When a request to calculate a given tensor is made using `grcalc()`, a list of such dependencies is constructed and each tensor in the list calculated in turn.

The actual component calculation is carried out by a pair of functions. The field `grC_symFn` points to a function which specifies which of the tensor components are to be calculated. It takes the form of a number of loops through the independent components, with cross assignments for components which can be identified by the index symmetry. For instance, for a symmetric two index tensor, `grC_symFn` would specify a loop through the components in the upper diagonal and cross-references to the components in the lower diagonal. For each component in the loop, the `grC_symFn` calls a function which is pointed to by the variable `grC_calcFn`. This is the function which actually specifies

for mixmaster spacetimes.

```
grG_ObjDef[G(dn,dn)][grC_header] := 'Covariant Einstein':
grG_ObjDef[G(dn,dn)][grC_rootStr] := 'G ':
grG_ObjDef[G(dn,dn)][grC_root] := Gdndn_:
grG_ObjDef[G(dn,dn)][grC_indexList] := [dn,dn]:
grG_ObjDef[G(dn,dn)][grC_calcFn] := grF_calc_Gdndn:
grG_ObjDef[G(dn,dn)][grC_calcFnParms] := []:
grG_ObjDef[G(dn,dn)][grC_symmetry] := grF_sym_sym2:
grG_ObjDef[G(dn,dn)][grC_depends] := g(dn,dn), R(dn,dn), Ricciscalar:
grF_calc_Gdndn := proc (object, index)
    s := grG_Rdndn[gname,a1_,a2_]
        - 1/2*(grGgdndn_[gname,a1_,a2_] * grG_scalarR_[gname]):
end:
```

Figure 3.2: The GRTensor object definition for the covariant Einstein tensor. The `grC_calcFn` field names the function used to calculate each component. The symmetry function, `grF_sym_sym2`, is the name of a generic function used to calculate the components of symmetric tensors with two indices.

the formula for the calculation of each tensor component in terms of the components of tensors specified in the dependency list.

The final important field of the object definition structure, `grC_root`, specifies the name of the array in which the final components are stored. In general, the name is constructed from the name of the tensor with its index configuration. In the case of the covariant Einstein tensor, `G(dn,dn)`, the results will be placed in the array `grG_Gdndn_`. An additional complicaion comes from the fact that we may wish to calculate the components of the Einstein tensor for various spacetimes in a given session. By convention, to distinguish these the first field of the component array will be the metric name. The remaining fields are indexed numerically and contain the individual components. Thus, `grG_Gdndn_[dust1,1,2]` stores the G_{12} component of the Einstein tensor for the `dust1` spacetime, while `grG_Gdndn_[schw,3,3]` stores the G_{33} component for the `schw` spacetime.

A full example of definition of the Einstein tensor within GRTensor is given in Figure (3.2). Note that although the index configuration is specified as part of the definition, only a single definition for each object is required. Tensor definitions for the alternate index configurations of a given tensor are generated automatically when they are required. That is, if the $G^a{}_b$ components of the Einstein tensor are to be calculated, then an internal definition of `G(up,dn)` is created, its calculation function corresponding to the contraction of `G(dn,dn)` with the contravariant metric tensor.

Tensor definitions such as the one described above can be created much more easily using the function `grdef()`. This command creates both a calculation function and an optimised symmetry function from a parsed string. For instance, the tensor definition displayed in Figure (3.2) could have been created automatically using the command

```
grdef ( 'G{ (a b) } := R{ a b} - (1/2) * g{a b}* Ricciscalar' );
```

(Note that the indices on the left-hand side are enclosed in round brackets, indicating the symmetry of the Einstein tensor). A full description of the usage of `grdef()`is beyond the scope of this thesis. However it should be noted that in principle, almost every object defined within the GRTensor standard libraries could have been defined using a `grdef()`command. This would entail a number of benefits, such as reduced size of source code and increased readability, the only disadvantage being the slight increase in calculation time the first time a tensor is referenced due to the need to construct a tensor definition from the `grdef()`string. At the current time, however, within the GRTensor source code objects continue to have explicit definitions in the form of Figure (3.2). Because the standard library has existed for some time and is extremely well tested, it is thought to be more prudent to retain the original definitions for the time being.

Object libraries

As a final point, we note that a certain amount of modularity can be achieved by grouping together related tensor definitions into separate Maple libraries. This has the advantage that only certain definitions need to be loaded at any time, corresponding the the parts of the package which is being used. For instance, almost the entire set of standard objects defined for tetrads and the NP-formalism exist in the external `basislib.m` library, which is loaded automatically when the user attempts to `grcalc()`one of the objects in question. If calculations are done exclusively in metric coordinates, the NP definitions are never loaded. Similar add-on libraries exist for scalar polynomial invariants and some vector field definitions, as well as a large library for calculations involving the junctions between spacetimes Musgrave and Lake (1996), Musgrave and Lake (1997). The spinor object definitions and tools for carrying out the Cartan-Karlhede method have similarly been coded into an external library for GRTensor, called `spinor.m`.

3.3 Spinor tools

The main requirements of the Cartan-Karlhede method are a set of tools for calculation of the minimal set of spinors determined by MacCallum and Åman (1986), and their transformation under $SL(2,\mathbb{C})$ rotations.

Although the required objects are spinors, and thus defined for a spin basis,the relation between spin bases and null tetrads (see Eq. (1.30)) allows all of the calculations to be carried out within the context of the NP formalism which is already defined within GRTensor. The spacetime specified in the form of an NP tetrad is required as input.

Derivative operators and symmetric spinors

New objects within the `spinor` package are the set of symmetric spinors listed by MacCallum and Åman (1986). Although calculations are done within the NP formalism, it is convenient notationally to retain the notion of the objects as symmetric spinors. Thus objects can be referenced using the 2-index notation described in Section 1.2.

The fundamental objects, the Weyl spinor, Ricci spinor, and Ricci scalar, are already defined within the NP formalism, and thus are calculable from the frame within GRTensor. Within the `spinor` package, they are referenced as `WeylSp`, `RicciSp`, and `Lambda`. Thus the command

```
grcalc ( WeylSp );
```

would calculate the covariant Weyl spinor components. Individual components are displayed and accessed using the 2-index notation. Thus the output of the above command would be displayed using

```
grdisplay ( WeylSp );
```

Weyl spinor

for the schw spacetime

$$\Psi_{20} = \frac{2m}{r^3}$$

where Ψ_{20} corresponds to the usual NP component Ψ_2. The 2-index notation carries over to the use of the `grcomponent()`command. Thus the $\Psi_{20'}$ component can be accessed using the command

```
grcomponent ( WeylSp, [2,0] );
```

Note that because the index pair are not conventional spinor indices, there are some restrictions placed on the operations allowed with symmetric spinors. Thus the operation of raising indices, or contracting indices within `grdef()`, are not defined for the symmetric spinors given by the `spinor` package.

The higher order objects correspond to different forms of the covariant derivative of the curvature spinors. Three derivative operators are defined to perform these calculations, each of which taking a valence (m,n) symmetric spinor, $\chi_{AB'} = \chi_{A_1\ldots A_m B'_1\ldots B'_n}$ as their argument.

`SymD[`χ`]` calculates the valence $(m+1,n+1)$ spinor corresponding to its symmetrized covariant derivative,

$$\nabla\chi_{AB'} = \nabla_{(A_{m+1}B'_{n+1}}\chi_{A_1\ldots A_m B'_1\ldots B'_n)}. \tag{3.2}$$

`Dal[`χ`]` calculates the valence (m,n) spinor which is the d'Alembertian applied to χ:

$$\nabla_{CD'}\nabla^{CD'}\chi_{AB'} = \nabla_{CD'}\nabla^{CD'}\chi_{(A_1\ldots A_m B'_1\ldots B'_n)}. \tag{3.3}$$

`Curl[`χ`]` calculates the valence $(m-1,n+1)$ spinor corresponding to the symmetrized covariant derivative with one index contracted,

$$\nabla^{A}{}_{B'}\chi_{AC'} = \nabla^{A_m}{}_{(B'_{n+1}}\chi_{A_1\ldots A_m B'_1\ldots B'_n)}. \tag{3.4}$$

Internally, the operators are expressed in terms of the NP formalism so that the components of the new spinor are calculated in terms of the components of the old spinor, the spin coefficients, and their derivatives. In particular, the (a,b) component of the symmetrized covariant derivative, ∇, applied to a spinor of valence (m,n) is given by

$$\begin{aligned}
(\nabla\chi)_{ab'} =& c_1 D\chi_{ab'} + c_4\Delta\chi_{(a-1)(b-1)'} + c_3\delta\chi_{a(b-1)'} + c_2\bar{\delta}\chi_{(a-1)b'} \\
&+ c_1((m-a-1)\kappa\chi_{(a+1)b'} + (2a-m+1)\epsilon\chi_{ab'} - a\pi\chi_{(a-1)b'} \\
&+ (n-b-1)\bar{\kappa}\chi_{a(b+1)'} + (2b-n+1)\bar{\epsilon}\chi_{ab'} - b\bar{\pi}\chi_{a(b-1)'}) \\
&+ c_2((m-a)\rho\chi_{ab'} + (2a-m-1)\alpha\chi_{(a-1)b'} - (a-1)\lambda\chi_{(a-2)b'} \\
&+ (n-b-1)\bar{\sigma}\chi_{(a-1)(b+1)'} + (2b-n+1)\bar{\beta}\chi_{(a-1)b'} - b\bar{\mu}\chi_{(a-1)(b-1)'}) \\
&+ c_3((m-a-1)\sigma\chi_{(a+1)(b-1)'} + (2a-m+1)\beta\chi_{a(b-1)'} - a\mu\chi_{(a-1)(b-1)'} \\
&+ (n-b)\bar{\rho}\chi_{ab'} + (2b-n-1)\bar{\alpha}\chi_{a(b-1)'} - (b-1)\bar{\lambda}\chi_{a(b-2)'}) \\
&+ c_4((m-a)\tau\chi_{a(b-1)'} + (2a-m-1)\gamma\chi_{(a-1)(b-1)'} - (a-1)\nu\chi_{(a-2)(b-1)'} \\
&+ (n-b)\bar{\tau}\chi_{(a-1)b'} + (2b-n-1)\bar{\gamma}\chi_{(a-1)(b-1)'} - (b-1)\bar{\nu}\chi_{(a-1)(b-2)'})
\end{aligned} \tag{3.5}$$

where

$$c_1 = \frac{(m-a)(n-b)}{mn}, \qquad c_2 = \frac{a(n-b)}{mn}, \qquad c_3 = \frac{(m-a)b}{mn}, \qquad c_4 = \frac{ab}{mn}. \tag{3.6}$$

The components of the d'Alembertian, $\Box$, of a symmetric spinor are given by

$$\begin{aligned}
(\Box\chi)_{ab'} =& 2(\Delta D - \delta\bar{\delta})\chi_{ab'} + (2(\epsilon+\bar{\epsilon}-\rho) + c_3\epsilon + c_3'\bar{\epsilon})\Delta\chi_{ab'} \\
&+ (-2\bar{\pi} - c_3\alpha - c_3'\bar{\beta})\delta\chi_{ab'} \\
&+ (2(\alpha-\bar{\beta}-\pi) - c_3\beta - c_3'\bar{\alpha})\bar{\delta}\chi_{ab'} + (2\mu + c_3\gamma + c_3'\bar{\gamma})\chi_{ab'} \\
&+ (c_3(\Delta\epsilon - \bar{\delta}\alpha - \delta\beta + D\gamma) + c_3'(\Delta\bar{\epsilon} - \delta\bar{\alpha} - \bar{\delta}\bar{\beta} + D\bar{\gamma}))\chi_{ab'} \\
&- c_2(\kappa\Delta\chi_{(a+1)b'} - \rho\bar{\delta}\chi_{(a+1)b'} - \sigma\delta\chi_{(a+1)b'} + \tau D\chi_{(a+1)b'}) \\
&- c_2(\Delta\kappa - \bar{\delta}\rho - \delta\sigma + D\tau)\chi_{(a+1)b'} \\
&- c_2'(\bar{\kappa}\Delta\chi_{a(b+1)'} - \bar{\rho}\delta\chi_{a(b+1)'} - \bar{\sigma}\bar{\delta}\chi_{a(b+1)'} + \bar{\tau}D\chi_{a(b+1)'}) \\
&- c_2'(\Delta\bar{\kappa} - \delta\bar{\rho} - \bar{\delta}\bar{\sigma} + D\bar{\tau})\chi_{a(b+1)'} \\
&+ c_1(\pi\Delta\chi_{(a-1)b'} - \lambda\bar{\delta}\chi_{(a-1)b'} - \mu\delta\chi_{(a-1)b'} + \nu D\chi_{(a-1)b'}) \\
&+ c_1(\Delta\pi - \bar{\delta}\lambda - \delta\mu + D\nu)\chi_{(a-1)b'} \\
&+ c_1'(\bar{\pi}\Delta\chi_{a(b-1)'} - \bar{\lambda}\delta\chi_{a(b-1)'} - \bar{\mu}\delta\chi_{a(b-1)'} + \bar{\nu}D\chi_{a(b-1)'}) \\
&+ c_1'(\Delta\bar{\pi} - \delta\bar{\lambda} - \bar{\delta}\bar{\mu} + D\bar{\nu})\chi_{a(b-1)'},
\end{aligned} \tag{3.7}$$

where

$$c_1 = a/m, \qquad c_2 = \frac{m-a}{m}, \qquad c_3 = \frac{m-2a}{m}, \tag{3.8a}$$

$$c_1' = b/n, \qquad c_2' = \frac{n-b}{n}, \qquad c_3' = \frac{n-2b}{n}. \tag{3.8b}$$

And finally, the 'curl' operator, $\hat{\nabla}$, applied to an arbitrary symmetric spinor has components of the form

$$\begin{aligned}
(\hat{\nabla}\chi)_{ab'} =& c_1(D\chi_{(a+1)b'} - \bar{\delta}\chi_{ab'}) - c_2(\Delta\chi_{a(b-1)'} - \delta\chi_{(a+1)(b-1)'}) \\
&+ c_1[(m-a)\kappa\chi_{(a+2)b'} + (2a-m+1)\epsilon\chi_{(a+1)b'} - (a+1)\pi\chi_{ab'} \\
&+ (n-b-1)\bar{\kappa}\chi_{(a+1)(b+1)'} + (2b-n+1)\bar{\epsilon}\chi_{(a+1)b'} \\
&- b\bar{\pi}\chi_{(a+1)(b-1)'} - (m-a+1)\rho\chi_{(a+1)b'} \\
&- (2a-m-1)\alpha\chi_{ab'} + a\lambda\chi_{(a-1)b'} - (n-b-1)\bar{\sigma}\chi_{a(b+1)'} \\
&- (2b-n+1)\bar{\beta}\chi_{ab'} + b\bar{\mu}\chi_{a(b-1)'}] \\
&+ c_2[(m-a)\sigma\chi_{(a+2)(b-1)'} + (2a-m+1)\beta\chi_{(a+1)(b-1)'} - (a+1)\mu\chi_{a(b-1)'} \\
&+ (n-b)\bar{\rho}\chi_{(a+1)b'} + (2b-n-1)\bar{\alpha}\chi_{(a+1)(b-1)'} - (b-1)\bar{\lambda}\chi_{(a+1)(b-2)'} \\
&- (m-a+1)\tau\chi_{(a+1)(b-1)'} - (2a-m-1)\gamma\chi_{a(b-1)'} + a\nu\chi_{(a-1)(b-1)'} \\
&- (n-b)\bar{\tau}\chi_{ab'} - (2b-n-1)\bar{\gamma}\chi_{a(b-1)'} + (b-1)\bar{\nu}\chi_{a(b-2)'}],
\end{aligned} \tag{3.9}$$

where

$$c_1 = \frac{n-b}{n}, \qquad c_2 = \frac{b}{n}. \tag{3.10}$$

(In fact, the equivalence method only requires that this operator be applied once, in order to determine the spinor $\Xi_{ABCD'}$.)

Each of the spinors required by the equivalence method have been given specific definitions in the spinor package. The names by which they are referenced are listed in Table 3.3. Note that the new object names do not follow the GRTensor convention of listing the index configuration of each spinor. Since the 2-index convention for listing spinor components is followed, it is implicit that each of the objects has two indices, both of which are necessarily covariant. Thus the components of the object `DalD2Psi` can be calculated using the command

```
grcalc ( DalD2Psi );
```

and the $(\Box\nabla^2\Psi)_{23'}$ component can be accessed using

```
grcomponent ( DalD2Psi, [2,3] );
```

Frame rotations

The `spinor` package also provides a set of routines for determining the components of the spinor under $SL(2,\mathbb{C})$ rotations of the spin basis,

$$\begin{pmatrix} o^A \\ \iota^A \end{pmatrix} \to \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} o^A \\ \iota^A \end{pmatrix}, \quad ad - bc = 1. \tag{3.11}$$

An $SL(2,\mathbb{C})$ transformation can be applied to any of the spinors listed above using the command `applydytr()`, which in addition to a list of spinors to be transformed, accepts the following arguments:

dytr: A 2x2 array containing the components of the spin transformation to be applied.

oldname: The name of the spacetime for which the spinors are to be transformed. If this argument is not specified, the current spacetime is assumed (ie. the spacetime for which the most recent calculations have been carried out).

newname: A name to assign to the rotated frame. If this argument is not specified, then a unique new name is created automatically by appending an integer to the name of the current default spacetime.

Name	Definition	Name	Definition
WeylSp	Ψ_{ABCD}	RicciSp	$\Phi_{ABA'B'}$
DPsi	$\nabla_{(AA'}\Psi_{BCDE)}$	DPhi	$\nabla_{(AA'}\Phi_{BCB'C')}$
D2Psi	$\nabla_{(AA'}\nabla_{BB'}\Psi_{CDEF)}$	D2Phi	$\nabla_{(AA'}\nabla_{BB'}\Phi_{CDC'D')}$
DnPsi	$\nabla_{(A_1A'_1}\dots\nabla_{A_nA'_n}\Psi_{BCDE)}$	DnPhi	$\nabla_{(A_1A'_1}\dots\nabla_{A_nA'_n}\Phi_{BCB'C')}$
	$n = 1\dots7$		$n = 1\dots7$
DalPsi	$\Box\Psi_{ABCD}$	DalPhi	$\Box\Phi_{ABA'B'}$
Dal$n$Psi	$\Box^n\Psi_{ABCD}$	Dal$n$Phi	$\Box^n\Phi_{ABA'B'}$
	$n = 1\dots3$		$n = 1\dots3$
DalDnPsi	$\Box\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Psi_{ABCD}$	DalDnPhi	$\Box\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Phi_{ABA'B'}$
	$n = 1\dots5$		$n = 1\dots5$
Dal2DnPsi	$\Box^2\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Psi_{ABCD}$	Dal2DnPhi	$\Box^2\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Phi_{ABA'B'}$
	$n = 1\dots3$		$n = 1\dots3$
Dal3DPsi	$\Box^3\nabla_{(AA'}\Psi_{BCDE)}$	Dal3DPhi	$\Box^3\nabla_{(AA'}\Phi_{BCB'C')}$

Name	Definition	Name	Definition
Lambda	Ricci scalar	Xi	$\nabla^A{}_{(B'}\Psi_{ABCD)}$
DLambda	$\nabla_{AA'}\Lambda$	DXi	$\nabla_{(AA'}\Xi_{BCDB')}$
D2Lambda	$\nabla_{(AA'}\nabla_{BB')}\Lambda$	D2Xi	$\nabla_{(AA'}\nabla_{BB'}\Xi_{CDEC')}$
DalLambda	$\Box\Lambda$	DalXi	$\Box\Xi_{ABCA'}$
DnLambda	$\nabla_{(A_1A'_1}\dots\nabla_{A_nA'_n)}\Lambda$	DnXi	$\nabla_{(A_1A'_1}\dots\nabla_{A_nA'_n}\Xi_{BCDB')}$
	$n = 1\dots7$		$n = 1\dots6$
Dal$n$Lambda	$\Box^n\Lambda$	Dal$n$Xi	$\Box^n\Xi_{ABCA'}$
	$n = 1\dots3$		$n = 1\dots3$
DalDnLambda	$\Box\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Lambda$	DalDnXi	$\Box\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Xi_{ABCA'}$
	$n = 1\dots5$		$n = 1\dots4$
Dal2DnLambda	$\Box^2\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Lambda$	Dal2DnXi	$\Box^2\nabla_{A_1A'_1}\dots\nabla_{A_nA'_n}\Xi_{ABCA'}$
	$n = 1\dots3$		$n = 1\dots2$
Dal3DLambda	$\Box^3\nabla_{AA'}\Lambda$		

Table 3.1: Symmetric spinors defined by the spinor library.

rotateTetrad: Specifies whether the tetrad vectors and spin coefficients are to be rotated in addition to the listed spinors. This argument can take the values `auto` (the frame vectors and spin coefficients are automatically rotated and assigned values in the new frame), `false` (only the listed spinors are assigned in the new frame), or `prompt` (the user is prompted as to whether the frame should be rotated [default]).

overwrite: If a spinor has already been assigned in the rotated frame (whose name is specified by `newname`), the user is prompted as to whether the components should be overwritten. If `overwrite` is set to `auto`, the components are automatically overwritten, whereas if it is set to `false` the components will not be overwritten.

Thus, for example, if the spinor `DPsi` has been calculated for the `schw` spacetime, an arbitrary null rotation can be applied using the commands:

```
> T := matrix (2, 2, [1,0,lambda,1]);
```

$$T = \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix}$$

```
> applydytr (DPsi, dytr=T, newname=schw2, rotateTetrad=auto);
```

This command will apply the transformation `T` to the components of `DPsi`. The new components are assigned to the object `DPsi` in the `schw2` spacetime. Note that within GRTensor, the components of each calculable object is associated with the name of a spacetime in order that, for instance, the Riemann tensor can be calculated in multiple spacetimes without having to overwrite its components. In order to avoid ambiguity about which frame the components for a given spinor have been calculated, the `spinor` package regards a rotated frame to be a different spacetime (at least in the GRTensor sense) from the original. Just as two separate metric files specify the Schwarzschild spacetime in Bondi coordinates and Eddington-Finkelstein coordinates, a pair of frames which differ by a $SL(2,\mathbb{C})$ rotation are also considered to be 'different'. In order to switch between frames, the `grmetric()`command is used, as in standard GRTensor. Thus, in the above example, the user would then type

```
> grmetric (schw2);
> grdisplay (DPsi);
```

in order to view the rotated components in the new (rotated) frame. Further calculations would proceed in this frame unless the command '`grmetric(schw)`' is used to return the user to the original frame, where the original spinor components are still stored.

The final argument in the example above specifies that the rotation is also to be applied automatically to the frame and the spin coefficients of the `schw` spacetime. If this argument were not specified, the user would be prompted as to whether this additional operation is to be performed. In general the user would wish to answer 'yes', since these objects will be required in the new frame if further spinor calculations are to be carried out.

Note that in order for `applydytr` to work, it must have a specification of how each of the given objects transform under spin transformations. For symmetric spinors, a single function is used to calculated the new components. Performing an expansion of the spinor in terms of the spin basis, Eq. (1.55), it is not difficult to see that under the given transformation a spinor component $\chi_{ij'}$ transforms as a polynomial in a, b, c, and d, and the spinor components,

$$\sum_{p=0}^{m-i}\sum_{q=0}^{i}\sum_{r=0}^{n-j}\sum_{s=0}^{j}\binom{m-i}{p}\binom{i}{q}\binom{n-j}{r}\binom{j}{s}a^p b^{m-i-p}c^q d^{i-q}\bar{a}^r\bar{b}^{n-j-r}\bar{c}^s\bar{d}^{j-s}\chi_{(m-p-q)(n-r-s)'}, \tag{3.12}$$

where the spinor has valence (m,n).

Only objects for which a rotation function has been assigned can be rotated using the `applydytr()` command. At the present time, these are the frame, the NP spin coefficients, and the symmetric spinors listed in Table 3.3.

For any of the spinors listed in Table 3.3, as well as for the frame vectors $(l^a, n^a, m^a, \bar{m}^a)$, the `applydytr()` command evaluates the new components using this polynomial.

The transformations of the spin coefficients, however, are somewhat more difficult, as they involve derivatives of the transformation parameters. A lengthy calculation results in the following formulas for the transformation of the spin coefficients under arbitrary $SL(2,\mathbb{C})$ rotations:

$$\begin{aligned}\kappa \to &- ab\bar{a}Da - b^2\bar{b}\Delta a - ab\bar{b}\delta a - b^2\bar{a}\bar{\delta}a + a^2\bar{a}Db + a^2\bar{b}\delta b\\ &+ ab\bar{a}\bar{\delta}b + ab\bar{b}\Delta b - a^3\bar{a}\kappa - 2a^2\bar{a}b\epsilon\\ &- a^3\bar{b}\sigma - 2a^2\bar{b}b\beta - a^2b\bar{a}\rho - 2ab^2\bar{a}\alpha - a^2b\bar{b}\tau - 2ab^2\bar{b}\gamma\\ &- ab^2\bar{a}\pi - ab^2\bar{b}\mu - b^3\bar{a}\lambda - b^3\bar{b}\nu\end{aligned} \tag{3.13a}$$

$$\begin{aligned}\rho \to & - bc\bar{a}Da - bd\bar{b}\Delta a - bc\bar{b}\delta a - bd\bar{a}\bar{\delta}a + ac\bar{a}Db + ad\bar{b}\Delta b \\ & + ac\bar{b}\delta b + ad\bar{a}\bar{\delta}b - a^2c\bar{a}\kappa - 2ac\bar{a}b\epsilon \\ & - a^2c\bar{b}\sigma - 2ac\bar{b}b\beta - a^2d\bar{a}\rho - 2ad\bar{a}b\alpha - a^2d\bar{b}\tau - 2ad\bar{b}b\gamma \\ & - b^2c\bar{a}\pi - b^2c\bar{b}\mu - b^2d\bar{a}\lambda - b^2d\bar{b}\nu \end{aligned} \tag{3.13b}$$

$$\begin{aligned}\sigma \to & - abc\bar{c}Da - b^2d\bar{d}\Delta a - abd\bar{d}\delta a - b^2c\bar{c}\bar{\delta}a + a^2c\bar{c}Db + abd\bar{d}\Delta \\ & + a^2d\bar{d}\delta b + abc\bar{c}\bar{\delta}b - a^3c\bar{c}\kappa - 2a^2c\bar{c}b\epsilon \\ & - a^3d\bar{d}\sigma - 2a^2d\bar{d}b\beta - a^2bc\bar{c}\rho - 2ab^2c\bar{c}\alpha - a^2bd\bar{d}\tau - 2ab^2d\bar{d}\gamma \\ & - ab^2c\bar{c}\pi - ab^2d\bar{d}mx - b^3c\bar{c}\lambda - b^3d\bar{d}\nu \end{aligned} \tag{3.13c}$$

$$\begin{aligned}\tau \to & - bcc\bar{c}Da - bdd\bar{d}\Delta a - bcd\bar{d}\delta a - bdc\bar{c}\bar{\delta}a + acc\bar{c}Db + add\bar{d}\Delta b \\ & + acd\bar{d}\delta b + adc\bar{c}\bar{\delta}b - a^2cc\bar{c}\kappa - 2acc\bar{c}b\epsilon \\ & - a^2cd\bar{d}\sigma - 2acd\bar{d}b\beta - a^2dc\bar{c}\rho - 2adc\bar{c}b\alpha - a^2dd\bar{d}\tau - 2add\bar{d}b\gamma \\ & - b^2cc\bar{c}\pi - b^2cd\bar{d}\mu - b^2dc\bar{c}\lambda - b^2dd\bar{d}\nu \end{aligned} \tag{3.13d}$$

$$\begin{aligned}\epsilon \to & - da\bar{a}Da - bd\bar{b}\Delta a - da\bar{b}\delta a - bd\bar{a}\bar{\delta}a + ac\bar{a}Db + cb\bar{b}\Delta b \\ & + ac\bar{b}\delta b + cb\bar{a}\bar{\delta}b - a^2c\bar{a}\kappa - ac\bar{a}b\epsilon - a^2c\bar{b}\sigma - ac\bar{b}b\beta \\ & - ab\bar{a}c\rho - b^2\bar{a}c\alpha - ab\bar{b}c\tau - b^2\bar{b}c\gamma - ab\bar{a}d\pi - a^2\bar{a}d\epsilon \\ & - ab\bar{b}d\mu - a^2\bar{b}d\beta - b^2d\bar{a}\lambda - ad\bar{a}b\alpha - b^2d\bar{b}\nu - ad\bar{b}b\gamma \end{aligned} \tag{3.13e}$$

$$\begin{aligned}\alpha \to & - cd\bar{a}Da - d^2\bar{b}\Delta a - cd\bar{b}\delta a - d^2\bar{a}\bar{\delta}a + c^2\bar{a}Db + cd\bar{b}\Delta b \\ & + c^2\bar{b}\delta b + cd\bar{a}\bar{\delta}b - ac^2\bar{a}\kappa - bc^2\bar{a}\epsilon - ac^2\bar{b}\sigma - bc^2\bar{b}\beta \\ & - ad\bar{a}c\rho - bd\bar{a}c\alpha - ad\bar{b}c\tau - bd\bar{b}c\gamma - bc\bar{a}d\pi - ac\bar{a}d\epsilon - bc\bar{b}d\mu - ac\bar{b}d\beta \\ & - bd^2\bar{a}\lambda - ad^2\bar{a}\alpha - bd^2\bar{b}\nu - ad^2\bar{b}\gamma \end{aligned} \tag{3.13f}$$

$$\begin{aligned}\beta \to & - da\bar{c}Da - bd\bar{d}\Delta a - da\bar{d}\delta a - bd\bar{c}\bar{\delta}a + ac\bar{c}Db + cb\bar{d}\Delta b \\ & + acd\bar{\delta}b + cb\bar{c}\bar{\delta}b - a^2c\bar{c}\kappa - ac\bar{c}b\epsilon - a^2cd\bar{\sigma} - acd\bar{b}\beta \\ & - ab\bar{c}c\rho - b^2\bar{c}c\alpha - ab\bar{d}c\tau - b^2\bar{d}c\gamma - ab\bar{c}d\pi - a^2\bar{c}d\epsilon \\ & - abd\bar{d}\mu - a^2\bar{d}d\beta - b^2d\bar{c}\lambda - ad\bar{c}b\alpha - b^2dd\bar{\nu} - add\bar{b}\gamma \end{aligned} \tag{3.13g}$$

$$\begin{aligned}\gamma \to & - cd\bar{c}Da - d^2\bar{d}\Delta a - cd\bar{d}\delta a - d^2\bar{c}\bar{\delta}a + c^2\bar{c}Db + cd\bar{d}\Delta b \\ & + c^2\bar{d}\delta b + cd\bar{c}\bar{\delta}b - ac^2\bar{c}\kappa - bc^2\bar{c}\epsilon - ac^2\bar{d}\sigma - bc^2\bar{d}\beta \\ & - ad\bar{c}c\rho - bd\bar{c}c\alpha - add\bar{c}\tau - bdd\bar{c}\gamma - bc\bar{c}d\pi - ac\bar{c}d\epsilon - bc\bar{d}d\mu - acd\bar{d}\beta \\ & - bd^2\bar{c}\lambda - ad^2\bar{c}\alpha - bd^2\bar{d}\nu - ad^2\bar{d}\gamma \end{aligned} \tag{3.13h}$$

$$\begin{aligned}\pi \to & - da\bar{a}Dc - bd\bar{b}\Delta c - da\bar{b}\delta c - bd\bar{a}\bar{\delta}c + ac\bar{a}Dd + cb\bar{b}\Delta d \\ & + ac\bar{b}\delta d + cb\bar{a}\bar{\delta}d - ac^2\bar{a}\kappa - 2ac\bar{a}d\epsilon \\ & - ac^2\bar{b}\sigma - 2ac\bar{b}d\beta - c^2b\bar{a}\rho - 2bd\bar{a}c\alpha - c^2b\bar{b}\tau - 2bd\bar{b}c\gamma \\ & - d^2a\bar{a}\pi - d^2a\bar{b}\mu - bd^2\bar{a}\lambda - bd^2\bar{b}\nu \end{aligned} \tag{3.13i}$$

$$\begin{aligned}\lambda \to & - cd\bar{a}Dc - d^2\bar{b}\Delta c - cd\bar{b}\delta c - d^2\bar{a}\bar{\delta}c + c^2\bar{a}Dd + cd\bar{b}\Delta d \\ & + c^2\bar{b}\delta d + cd\bar{a}\bar{\delta}d - c^3\bar{a}\kappa - 2c^2\bar{a}d\epsilon \\ & - c^3\bar{b}\sigma - 2c^2\bar{b}d\beta - c^2d\bar{a}\rho - 2cd^2\bar{a}\alpha - c^2d\bar{b}\tau - 2cd^2\bar{b}\gamma \\ & - cd^2\bar{a}\pi - cd^2\bar{b}\mu - d^3\bar{a} - d^3\bar{b}\nu \end{aligned} \tag{3.13j}$$

$$\begin{aligned}\mu \to & - da\bar{c}Dc - bd\bar{d}\Delta c - da\bar{d}\delta c - bd\bar{c}\bar{\delta}c + ac\bar{c}Dd + cb\bar{d}\Delta d \\ & + ac\bar{d}\delta d + cb\bar{c}\bar{\delta}d - ac^2\bar{c}\kappa - 2ac\bar{c}d\epsilon \\ & - ac^2\bar{d}\sigma - 2ac\bar{d}d\beta - c^2b\bar{c}\rho - 2bd\bar{c}c\alpha - c^2b\bar{d}\tau - 2bd\bar{d}c\gamma \\ & - d^2a\bar{c}\pi - d^2a\bar{d}\mu - bd^2\bar{c}\lambda - bd^2\bar{d}\nu \end{aligned} \tag{3.13k}$$

$$\begin{aligned}\nu \to & - cd\bar{c}Dc - d^2\bar{d}\Delta c - cd\bar{d}\delta c - d^2\bar{c}\bar{\delta}c + c^2\bar{c}Dd + cd\bar{d}\Delta \\ & + c^2\bar{d}\delta d + cd\bar{c}\bar{\delta}d - c^3\bar{c}\kappa - 2c^2\bar{c}d\epsilon \\ & - c^3\bar{d}\sigma - 2c^2\bar{d}d\beta - c^2d\bar{c}\rho - 2cd^2\bar{c}\alpha - c^2d\bar{d}\tau - 2cd^2\bar{d}\gamma \\ & - cd^2\bar{c}\pi - cd^2\bar{d}\mu - d^3\bar{c}\lambda - d^3\bar{d}\nu \end{aligned} \tag{3.13l}$$

3.4 Classification tools

To this point, tools have been described for calculating the components of all of the independent symmetric spinors listed in MacCallum and Åman (1986), and applying general dyad transformations to these objects. In order to carry out a full classification of the given spacetime, the following additional tasks, specific to the equivalence problem, must be accomplished:

- The Petrov and Segré types of the spacetime must be determined.
- The spinors must be checked to determine whether they are in a standard form.
- The isometry group of the frame must be determined.
- If a spinor is not in standard form, a spin transformation must be determined which will bring it into its standard form.
- Independent functions among the spinor components must located.

Petrov type

The Petrov type is determined from the Weyl spinor using the command

```
grcalc (ptype):
```

The calculation of `ptype` proceeds as described in Appendix A, and the result is stored as one of `0`, `1`, `2`, `3`, `d`, `n`. Note that in many cases, the determination of the Petrov type involves the calculation of a high order polynomial in the spin coefficients which must be equated to zero. In general, this is well done within Maple, and by implicitly applying `expand()` to the results, a correct evaluation (zero or non-zero) is almost guaranteed. However, it is not difficult to construct situations (especially involving radicals, as described in Section 3.1) for which a zero-function can not be automatically simplified to zero. Thus, it is necessary that the user be able to check the intermediate calculations to see that everything has proceeded correctly, and if not, to manipulate the troublesome expressions.

To facilitate this, as the `ptype` calculation proceeds the path taken (a number from 0 to 31 corresponding to an entry of the table in Appendix A) is stored in the variable `grG_Ppath_`, while the intermediate expressions on which decisions have been based, are stored in the global variable `grG_Pscalars`. This variable is a set whose elements correspond to the individual scalars which must be calculated along the given path. Thus, if the Weyl spinor corresponds to Case 15 of the Petrov algorithm, then `grG_Ppath_` will equal 15, while the `grG_Pscalars_` variable will contain entries of the form $I = \ldots$, $F1 = \ldots$, $F2 = \ldots$, $D = \ldots$.

The user can manipulate the entries of `grG_Pscalars_` in order to ensure that all of the zeros have been found, then run `grcalc (ptype)` once again. This time, the program will recognise that the required (and hopefully fully simplified) scalars have already been stored in the `grG_Pscalars_` variable, and will prompt the user as to whether these variables should be used as opposed to recalculating them from scratch. By this procedure, the user is given complete control over the intermediate decision making that takes place within the Petrov type algorithm. Fortunately, this level of control seems only to be required in rare situations.

Isotropy testing

The next two closely related tasks are greatly simplified once the Petrov and Segré types have been established. From Table 2.1, for instance, we see that a simple test of its

components determines whether the Weyl spinor is in standard form for a given Petrov type, and given a non-zero Petrov type, only a limited subgroup of the rotational freedom remains in the frame.

The program `isotest()` is a general tool which can be used to determine whether any given spinor is in the standard form for a particular isotropy group. Two arguments are specified, the first being the spinor which is to be checked (any of those listed in Table 3.3), and the second is the subgroup of $SL(2,\mathbb{C})$under which the frame is still free to rotate. These are specified by the following strings: `lnswap`, `Boost`, `Spin`, `Null2D`, `Null1D`, `NullR`, `NullI`. The `lnswap` freedom refers to a swap of the o^A and ι^A spin basis vectors, which in a frame has the effect of swapping the l^a and n^a and the m^a and $\bar{m}^a$ vectors. The `Null2D` isometry refers to freedom to null rotations about o^A with a complex rotation parameter λ. If the null rotational freedom only holds for a real parameter, the isotropy group can be specified as `NullR`, or for a strictly imaginary parameter, `NullI`. The `Null1D` group refers to rotations by a complex-valued parameter for which a restriction exists relating the real and imaginary parts. (Note that although this form of isotropy can be detected, in fact no corresponding standard form has been defined.)

The `isotest()` function returns the Maple `NULL`-value if the spinor is judged to be in a standard form for the given isotropy group, as specified in Chapter 2. If it is not in standard form, a `list` is returned, specifying both the isotropy group which has been lost and the particular component which deviates from the standard.

Automatic generation of dyad transformations

Given that a spinor is not in the standard form for a particular $SL(2,\mathbb{C})$subgroup, a spin transformation must be applied to its components. The function `dytrgen()` takes as arguments the name of the non-standard spinor, and the isotropy group which is to be used. The components of the spinor are then used to calculate the entries in a spin matrix which, when applied to the spinor, will bring it into the standard from. The 2x2 array which is returned by the `dytrgen()` function can be used as input to the `applydytr()` function, described above.

The `dytrgen()` function can be applied to any of the spinors listed in Table 3.3. Note, however, that in order to determine the appropriate standard form for a general spinor, it is necessary that the $SL(2,\mathbb{C})$freedom already have been reduced to one of the listed subgroups. This is only possible once the Weyl and Ricci spinors have already been

classified, and special algorithms for the Weyl spinor are required in order to determine the spin transformations which bring it into one of the forms listed in Table 2.1.

Given a particular configuration of zero/non-zero Weyl components, as listed in Appendix A, it is generally not difficult to determine expressions for the components of a spin transformation as functions of the Ψ_is. These are listed in Appendix B. However, in certain complicated cases for which there is a large number on non-zero Weyl components, it is extremely difficult to determine a general formula for the spin transformation matrix. In order to arrive at the correct spin transformation, a general rotation is applied to the coefficients of the Weyl spinor, and the restrictions specified by the canonical frame (eg. that certain components be zero) are used to solve for the undetermined rotation parameters. Generally, this involves solving for the roots of a quartic in the Weyl coefficients. Though for special cases, the problem can be simplified and the roots are not difficult to find, the general solution is not known. The current set of algorithms is not able to determine a standard form for Type I and Type II Weyl spinors arising from paths 19, 23, 27, and 31 of the Petrov algorithm described in Appendix A. Further, in certain cases (notably 9, 11, and 15), although the formulas for the transformation coefficients can be found, they are complicated and may be impractical to apply. Note, however, that this is not a problem which can be solved through better algorithm design or more powerful simplification tools. It is a result of the fact that the transformation coefficients are the roots of a quartic. As such, this is a fundamental limitation of the Cartan-Karlhede method.

A final tool is required to scan the list of spinor components and extract the potentially independent components. The function `find_indep_cmpts()` exists to perform this task. The real and imaginary parts of each spinor component are extracted. From these, the Jacobian determinant is used to determine the set of linearly independent functions. The results are returned as a set whose entries specify: the spinor name, the independent component number, whether it was the real or imaginary part, the actual value of the component, and its derivatives in terms of the coordinates.

Automatic classification

The functions described in the previous paragraphs provide all of the functionality required to carry out the Cartan-Karlhede method. As a step-by-step process, however, the method is still quite complicated to carry out. As such, the `spinor` package provides a `classify()` function which will automatically carry out the steps required to

fully classify a spacetime. The function takes no arguments, and is assumed to work on the current default spacetime which has been loaded into GRTensor as a null frame.

The `classify()` function proceeds stepwise, first determining the Petrov type of the given spacetime (calculating the Weyl in the process, if necessary). Using the result of the Petrov type calculation, the Weyl tensor is checked to determine whether or not it is in canonical form, and if not, `dytrgen()` determines a spin transformation which is passed to `applydytr()` in order to rotate the frame (and spin coefficients) to their Weyl standard form. The new frame assuming the spacetime is not conformally flat, is known to be invariant under a restricted subgroup of $SL(2,\mathbb{C})$. This information is stored in the global variable `grG_Isotropy_`. The independent components of the Weyl spinor in the standard frame are located using `find_indep_cmpts()`, and this information stored in the global variable `grG_indep_fns_`. Note that at any time later this information can be examined by the user in order to verify that the procedure has been carried out correctly.

Once the Weyl-standard frame has been established, the Ricci spinor is calculated and brought into canonical form by applying another dyad transformation if necessary. Once again, the remaining isotropy group at this stage is recorded, and independent functions isolated. The procedure carries on by checking the Ricci scalar for independence. At this point, 'Level 0' is completed and the program prints out some status messages and prompts the user as to whether they would like to proceed on to 'Level 1', namely the first derivatives of the curvature spinors.

At this point it is possible for the user to consider whether the information that has been displayed to this point seems correct. It is often a good idea to examine the components of the spinors and spin coefficients in the current frame. If they are of an extremely complicated form, the user can attempt to simplify them by applying Maple routines through `gralter()`.

Once the procedure has been checked to this level, the user can re-initiate the classification process by using the `classify()` command once again. The program will use the information which it has already determined for the given frame, and continue from where it left off to calculate the first derivatives of the curvature spinors. These calculated, the user is again given an opportunity to halt the process in order to check the results before proceeding on to 'Level 2', the second derivatives.

The program halts when it determines that all of the information required for a complete classification has been found. That is, if after two consecutive levels no more independent functions have been found, and the isotropy group under which the frame

is invariant has not changed, the program declares that the spacetime has been fully classified.

Note that although the `classify()` command attempts to carry out the classification automatically, the potential for user intervention is critical for all but the simplest cases. In general, a frame which is not already in standard form will require the application of spin transformations which result in complicated forms for the spinor components and spin coefficients. Unless simplifications are applied to these objects, further calculations will tend to bog down (see Pollney et al. (1996)) and a correct determination of the independent components and standard form for subsequently calculated spinors will become less likely. That said, it can also be noted that frames in which spacetimes are expressed, often take advantage of algebraic symmetries, and as such it often happens that they are presented in near-standard form. In such cases, the classification may be able to proceed without intervention.

An example of the use of the `classify()` function in an examination of the Edgar-Ludwig conformally flat metrics is given in Appendix C.

3.5 Additional tools: Complex quantities in Maple/GRTensor

The implementation of complex valued quantities within Maple possesses some inherent difficulties which inconvenience spinor calculations.

The first of these arises from the specification of which variables are to be considered complex and which are to be considered real. An inconvenience arises from the fact that Maple assumes that all symbolic quantities are complex. This contradicts the common practice in specifying spacetimes, where coordinates and functions contained in spinor components, are generally chosen to be real. It is possible to give a variable within Maple the property of being real using the `assume()` facility. Thus, in order for complex conjugation to work as expected in the context of spinors, it would be necessary to apply `assume()` to each of the coordinates and metric functions at the time that the spacetime is loaded.

A more serious problem with complex valued functions, however, comes from Maple's implementation of complex conjugation. The conjugate of a value is represented by applying the function `conjugate()` to the value. Thus, for a complex valued function x, we find

```
> conjugate (x);
```

$$\bar{x}$$

The problem arises when one wishes to use the conjugate within other functions, such as `diff()`, which can not take functions as arguments. For instance, given a function $f(z,\bar{z})$ of complex coordinate z, it should be possible to take the derivative

$$\frac{\partial f(z,\bar{z})}{\partial \bar{z}}. \tag{3.14}$$

Within Maple, this is expressed as

```
> diff ( f(z,conjugate(z)), conjugate(z) );
```

However, this statement returns an error, because Maple does not allow a function to be used as the variable of differentiation.

To avoid these difficulties, GRTensor and the `spinor` package, make use of an alternate specification implemented through the use of the conjugation function `conj` rather than the standard Maple `conjugate` function. The `conj` function has two distinctions. First of all, it assumes that all variables and functions are real, unless they are listed in the set `complexSet_`. Secondly, the complex conjugate of a variable is represented by appending the string '`bar`' to the end of its name. Consider the following example:

```
> complexSet_ := {y,f};
> conj (x); conj (y); conj ( f(y,ybar) ); conj (ybar);
```

$$x, \qquad ybar, \qquad fbar(y,ybar), \qquad y$$

Since the conjugate has been replaced by a name, it can be used in any other function which accepts names as arguments, that is, '`y`' has the same status as '`ybar`'. Thus, commands such as `diff` yield the expected results:

```
> g(y,ybar) := y^2 + conj(y)^2;
```

$$g(y,ybar) := y^2 + ybar^2$$

```
> diff ( y^2 - ybar^2, ybar );
```

$$2ybar$$

Within the context of GRTensor, the complex valued functions and coordinates for a given spacetime can be specified apon loading of the spacetime by adding the expression

```
complex_ := {complex variables};
```

to the metric file.

3.6 Outstanding problems

Through use of the computer algebra tools described in this chapter, the Cartan-Karlhede method can be implemented in order to obtain a unique algebraic classification of a large variety of spacetimes. To regard the method as an 'algorithm' for classification would be incorrect, however, as the definition of an algorithm implies that a solution will be found for any set of initial data, and this is not the case for the Cartan-Karlhede method.

The greatest difficulties arise from the fact that the fixing of a canonical frame for the Weyl spinor, the first stage of the process, can involve solving for the roots of a quartic equation. If this can be accomplished for a non-zero Weyl spinor, then the methods described above will lead to a classification. The methods of fixing the frames under spins, boosts, and null rotations, have been specified in such a way as to ensure that the roots specifying the rotation coefficients can always be found.

As discussed in Section 3.4, however, there are examples of Weyl component configurations for which the components specifying the dyad transformation to standard form can not be determined. In addition to this, for some of the cases for which an expression for the dyad transformation exists, the expression turns out to be complicated enough that it is possible that it will be of little practical use, for components transformed under the specified transformation will not yield to simplifications. We emphasise that the problem is not one that increased computational power is likely to solve. It is a result of the complicated nature of the roots of general quartic equations which must be solved in order to carry out the Cartan-Karlhede method successfully.

A possible way of avoiding this problem is to expand the allowed set of standard forms. Certain initial configurations of Weyl tensor components can be transformed more naturally into standard forms other than the ones specified. For instance, the frame for the Petrov Type I cylindrical Van Stockum spacetime van Stockum (1937) (see also Kramer et al. (1980), page 222), is specified in the CLASSI database Skea (1997b) as

$$l_a = [1/\sqrt{2}, 0, e^{-a^2\rho^2/2}/\sqrt{2}, a\rho^2/\sqrt{2}], \tag{3.15a}$$

$$n_a = [1/\sqrt{2}, 0, -e^{-a^2\rho^2/2}/\sqrt{2}, a\rho^2/\sqrt{2}], \tag{3.15b}$$

$$m_a = [0, e^{-a^2\rho^2/2}/\sqrt{2}, 0, i\rho/\sqrt{2}], \tag{3.15c}$$

where the coordinates are (t, ρ, z, ϕ), and a is a real constant. For this frame, the Weyl

components take the simple form

$$\Psi_1 = \frac{1}{2}ia^3e^{a^2\rho^2}, \qquad \Psi_2 = \frac{1}{3}a^2e^{a^2\rho^2}, \qquad \Psi_3 = -\frac{1}{2}ia^3e^{a^2\rho^2}. \tag{3.16}$$

In order to bring the Weyl spinor into the standard form specified for Type I spacetimes (ie. $\Psi_0 = \Psi_4$, $\Psi_1 = \Psi_3 = 0$), a transformation of the form

$$T = \begin{pmatrix} \frac{1}{2}(1-i)A^{1/8} & \frac{1}{2}(1+i)A^{1/8} \\ -\frac{1}{2}(1-i)A^{-1/8} & \frac{1}{2}(1+i)A^{-1/8} \end{pmatrix}, \qquad \text{where } A = \frac{1+2a\rho}{1-2a\rho}, \tag{3.17}$$

is required. Under this rotation, the frame vectors take the form

$$l_a = [\frac{\sqrt{2}}{4}\frac{\sqrt{-1+2a\rho}+\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}}, \frac{e^{-a^2\rho^2/2}}{\sqrt{2}}, \frac{\sqrt{2}}{4}e^{-a^2\rho^2/2}\frac{\sqrt{-1+2a\rho}-\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}},$$
$$\frac{\sqrt{2}}{4}a\rho^2\frac{\sqrt{-1+2a\rho}+\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}}], \tag{3.18a}$$

$$n_a = [\frac{\sqrt{2}}{4}\frac{\sqrt{-1+2a\rho}+\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}}, -\frac{e^{-a^2\rho^2/2}}{\sqrt{2}}, \frac{\sqrt{2}}{4}e^{-a^2\rho^2/2}\frac{\sqrt{-1+2a\rho}-\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}},$$
$$\frac{\sqrt{2}}{4}a\rho^2\frac{\sqrt{-1+2a\rho}+\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}}], \tag{3.18b}$$

$$m_a = [-\frac{\sqrt{2}}{4}i\frac{\sqrt{-1+2a\rho}-\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}}, 0, -\frac{\sqrt{2}}{4}ie^{-a^2\rho^2/2}\frac{\sqrt{-1+2a\rho}+\sqrt{-1-2a\rho}}{(1-4a^2\rho^2)^{1/4}},$$
$$-\frac{\sqrt{2}}{4}\frac{2(1-4a^2\rho^2)^{1/4}+ia\rho(\sqrt{-1+2a\rho}-\sqrt{-1-2a\rho})}{(1-4a^2\rho^2)^{1/4}}]. \tag{3.18c}$$

and the Weyl components are

$$\Psi_0 = \frac{1}{2}a^2(1-4a^2\rho^2)^{1/2}e^{a^2\rho^2}, \qquad \Psi_2 = \frac{1}{3}a^2e^{a^2\rho^2}, \qquad \Psi_4 = \frac{1}{2}a^2(1-4a^2\rho^2)^{1/2}e^{a^2\rho^2}. \tag{3.19}$$

Unfortunately, when this spacetime is rotated into the standard frame, its frame components take on a much more complicated form than in the original frame. This is not a fatal problem, however it can lead to computational difficulties later on when higher order derivatives of the curvature spinors need to be evaluated.

The current implementation of the CLASSI software avoids this problem by allowing a variety of 'alternate' standard forms for Type I metrics. Of the nineteen Type I metrics listed in the CLASSI database, three have been allowed to assume an alternate Type I standard form. These are listed in Table 3.6. Of these cases, both the van Stockum and Newman-Tambourino spacetimes require moderately complicated transformations of the form listed above in order to bring them into canonical form, while the Tariq-Tupper spacetime requires a simple spin with coefficient $(-1)^{1/8}$ in order to fix $\Psi_0 = \Psi_4$, thus introducing little computational expense.

Spacetime	Ψ_0	Ψ_1	Ψ_2	Ψ_3	Ψ_4
`CylindricalVanStockum`	0	X	Y	$-X$	0
`NewmanTambourinoLimit`	0	X	Y	X	0
`TariqTupper`	X	0	Y	0	$-X$

Table 3.2: Spacetimes for which CLASSI accepts alternate standard forms.

While the tolerance of alternate standard forms can reduce the computational effort in determining a set of independent components for the curvature spinors, it is difficult to reconcile with the Cartan-Karlhede method, which is based upon a strict adherence to a given set of standard forms. The final stage of the determination of the equivalence of two spacetimes requires that the individual spinor components be compared one against the other. If the two spacetimes have been expressed in different standard frames, this can not be done directly, and a mapping is required from the components of one to those of the other. This mapping is given by the transformation between the two alternate standard forms. For this reason, at the present time the `spinor` package imposes a strict adherence to the standard forms specified in Table 2.1.

A final point of note is that even when a unique set of standard forms for the curvature spinors has been fixed, it is not necessarily true that the frames which satisfy these forms are unique. For instance, in the above example the transformation to the standard form involve an 8th root of a factor in the matrix coefficients. Depending on which root is chosen, there are eight potential frames which will result in the given standard form, corresponding to rotations of the m^a vector by an angle of $\pi/4$. This can lead to difficulties in the final stage of the procedure when components between two spacetimes are compared, for if the two frames differ, a consistent set of equations may not result. This said, it should also be pointed out that in practice the potential discrepancy has not arisen.

All of the problems described above are a direct result of the difficulty in defining an unambiguous standard frame for a given spacetime. It is worth considering, then, whether alternate methods might exist in which rigid adherence to a standard frame is a less stringent requirement. As mentioned, the CLASSI program already accepts alternate standard frames for Petrov Type I spacetimes, though a formalism for how these are to be used in solving the equivalence problem has not been documented.

One such method to use is the original procedure described by Cartan, in which the frame is left completely arbitrary with the penalty of having to extract the rotational information from higher order derivatives of the curvature spinors. Although this requires the calculation of many more spinor components, the ability to choose the frame arbitrarily means that one can always choose a frame in which the curvature components take on a relatively simple form. It is not impossible that the resulting reduction in computational effort will more than compensate for the need to calculate higher order derivatives. As evidence, note that the time required to calculate an object even as simple as the Riemann tensor can vary by many orders of magnitude depending on the input and simplification strategy used (see the discussion in Pollney et al. (1996)). If the standard frame for a given spacetime does not correspond to an optimal frame for computation, it is possible that the procedure will get bogged down in calculation of even the low order derivatives, while in the computationally optimal frame, the high order derivatives are calculated with ease.

In fact, a procedure embodying the benefits of both methods might be found in a compromise between the fixed and free frame methods. For example, the independent spinor components can be expressed in the form of GHP style quantities with given spin and boost weights. These two degrees of freedom can be removed from the frame quite easily, while the remaining freedom could be solved for using higher spinor derivatives.

Finally, we make note of the information available from an alternate source, the scalar polynomial (sp) invariants formed from index contractions of the curvature spinors. In many senses, they are ideal for the purpose of determining coordinate transformations between spacetimes, for the information which they give is completely independent of the frame in which the spacetime is expressed. Traditionally their use has been avoided, however, because of the inherent ambiguities in the information which they provide. Crucially, it has often been pointed out that for plane waves all of the sp-invariants are zero, making such spacetimes indistinguishable from flat space. This fact can be attributed to the fact that the Lorentz rotation group $SL(2,\mathbb{C})$ is non-compact, and flat space arises as the limit of an infinite boost of the plane wave spacetimes (Schmidt (1994)), a point enforced by an example of a class of Type N spacetimes which are distinguishable from flat space but not from each other using sp-invariants (Schmidt (1998)). Thus it seems that sp-invariants can at best distinguish spacetimes up to their Geroch limits (Geroch (1977)).

A further difficulty arises from the fact that for a general spacetime, a necessary

and sufficient set of sp-invariants to determine whether a consistent set of coordinate transformations might exist has not been determined. This is closely related to the problem of finding an independent set of sp-invariants, which has occupied a number of researchers over the years (see Penrose and Rindler (1986), and Carminati and McLenaghan (1991) and the references therein). Even at the level of the 0th order curvature spinors (the Weyl and Ricci spinors) a sufficient independent set for general spacetimes has proved elusive. However, for specific spacetimes the problem can be solved using the techniques of Carminati and McLenaghan (1991) of specifying relations between the spinor components and evaluating invariants constructed from index contractions until an independent set satisfying the expected number of degrees of freedom is found. This process might even yield to a degree of mechanisation, as the construction of invariants is simply a matter of permutations of index contractions, and software for carrying out this task has been developed by the author to find all of the identities among invariants up to degree 5 in the curvature spinors (Pollney (1996)).

Despite the difficulties, the ease of calculation of the sp-invariants make them an attractive alternative to the Cartan-Karlhede components. For Type I spacetimes, the functional information in the Weyl tensor is found through the calculation of the two scalars

$$I := \Psi_{ABCD}\Psi^{ABCD}, \qquad J := \Psi^{AB}{}_{CD}\Psi^{CD}{}_{EF}\Psi^{EF}{}_{AB}, \tag{3.20}$$

which can be used to obtain the same information as one gets from the Ψ_0 and Ψ_2 components of the standard frame. Importantly, although the calculation of I and J involves respectively quadratic and cubic polynomials in the Weyl components, one avoids the need to solve a quartic equation in order to determine transformation coefficients for the standard frame. Thus, in cases where the Cartan-Karlhede method can make no progress, information can still be extracted from the sp-invariants. A formalised method making use of this information might form a useful alternative for cases to which it proves difficult to apply the Cartan-Karlhede method.

A number of difficulties in applying the Cartan-Karlhede method have been identified. Generally they arise from the trade-off between the number of computations required when the frame is left free, as in Cartan's original description, against the difficulties involved with fixing the frame into a generally applicable standard form. While the latter method can be computationally the most efficient in terms of the number of components to be calculated, there are cases where, in contrast to Cartan's

Petrov Type:	0	N	D	III	II	I	Total
# of examples:	42	27	114	8	3	19	213

Table 3.3: The number of examples contained in the spacetime database at the time of writing, grouped by Petrov type.

method, it can not be carried out, due to the fact that the transformation coefficients arise as solutions to high order polynomials. That said, it should be noted that in practice the Cartan-Karlhede method has been used to construct a database of some 213 spacetimes, with the only departures from the method arising from the three spacetimes discussed above. Table 3.6 gives the breakdown for the number of spacetimes included in the database for each Petrov type. The grouping of the spacetimes which have been included reflect the fact that the study of exact solutions has concentrated on the algebraically special spacetimes. In this practical sense, a tool which efficiently classifies algebraically special metrics, as the Cartan-Karlhede method does, is appropriate. As the study of spacetimes moves on to the more general situations of Type I metrics, however, the need for alternative tools may become more apparent.

An important omission in the current state of the methods, however, is the inclusion of a Segré classification and dyad transformations to bring the Ricci spinor into a standard form. This work will be necessary for a fully functional classification package, for at the moment conformally flat spacetimes can not be handled reliably. That said, it can be noted that in a number of cases a reliable classification has resulted for conformally flat spacetimes simply by fixing the frame by placing the Ricci spinor in one of the general standard forms for spinors described in Chapter 2. This is possible for spacetimes which are input using frames which are already 'close' to their final standard form. However, because the standard forms are specified only in terms of reduced subgroups of $SL(2,\mathbb{C})$, these methods will not generally be successful without a first stage (Segré classification) to reduce the frame freedom to one of these subgroups. A number of prescriptions for classifying the Ricci spinor exist in the literature (eg. Penrose and Rindler (1986), Cormack and Hall (1979), Joly and MacCallum (1990)). Work to develop a Segré classification code compatible with the `spinor` package is currently being carried out in collaboration with Jim Skea of UERJ, Brazil.

Greater concerns for the success of any classification via the Cartan-Karlhede method

arise from the problems discussed at the end of Chapter 3. The need to solve a quartic equation as part of the method provides insurmountable difficulties for the classification of certain types of initial data. Alternative methods, either formalising some more general notion of 'standard' frames (for example, using alternate standard frames depending on the form of the original frame), or making use of alternate sources of algebraic information, such as the curvature invariants, are paths worthy of further exploration.

Finally, we note the great potential for application of the spinor tools. The CLASSI program has already been used by Skea (1997b) and co-workers to create a database of more than two hundred spacetimes, available online via the Internet. This database has the potential to serve as a valuable resource to the community of relativity researchers in the form of a consistently updated version of the well-used 'Exact Solutions' book Kramer et al. (1980). A continuation and expansion of this effort would provide a useful test of the new software and methods which have been developed.

Part II

Cauchy-characteristic matching in axial symmetry

4

Numerical techniques for the solution of Einstein's equations

To this point, this thesis has concerned itself with the algebraic study of exact solutions of Einstein's field equations. Indeed, the study of exact solutions has to this day provided virtually all of our current understanding of the properties and dynamics of gravitational fields. Unfortunately, these results have been derived only for very specialised situations. The complexity of the field equations demand that if any analytical progress is to be made, simplifying assumptions must be applied. Typically, restrictions on the geometry (such as spacelike homogeneity in cosmology, or spherical symmetry in the study of isolated bodies) are introduced in order to reduce the equations to a tractable system. Since these symmetries are expected to be good approximations to realistic physical configurations, they have been used to good effect to obtain an understanding of the features and effects that might be expected to arise from a given distribution of matter.

The physical relevance of these results, however, is generally limited to the symmetries of the spacetime in question. In recent years increased effort has gone into the construction of instruments capable of directly measuring gravitational effects. As such, the interest in more 'realistic' (or at least more generic) models approximating some potentially measurable physical system has intensified. At the same time, computer technology has advanced to the point where large-scale computations involving equations of the complexity of the full Einstein field equations can reasonably be carried out. As a result, the past two decades have seen a shift in focus among relativists towards numerical studies of the gravitational field. Such studies are now for the first time yielding insights into models which represent important physical process such as the inspiral of matter onto a neutron star or black hole, and neutron star and black hole collisions. Further, numerical computations have also been used in spacetimes

with a specialised symmetry to obtain results which had previously been unknown to analytical studies, notably the discovery of critical phenomena in spherical symmetry by Choptuik (1993).

A framework for studying the field equations numerically was laid out in the early 60's by Arnowitt et al. (1962). In their formalism, the spacetime is foliated by spacelike 3-surfaces whose metric components are evolved in the direction of a timelike vector. As such, it is often referred to as a 3+1 technique, or, because the initial spacelike surface can be thought of as a Cauchy surface (on which all of the data required by subsequent evolution has been specified), a Cauchy evolution.[1] The 3+1 methods provide a useful system of evolution equations for general relativistic systems. The data and coordinates are specified on spacelike slices, and thus have somewhat physical interpretations, and the equations can be set up in such a way as to provide a future evolution which tends to avoid both physical and coordinate singularities. As such, they have become the commonly applied method of performing numerical evolutions of isolated sources in general relativity. Problems still exist in specifying physically relevant initial data which satisfy the constraints, the choice of appropriate slicing and shift conditions, and the excision of trapped surfaces, and solutions to these difficulties are the topic of active study in the relativity community.

A particular problem associated with 3+1 evolutions arises exactly from the feature which makes these methods so intuitively attractive: the spacelike nature of the evolved surfaces. Because the computation is intended to be carried out within the physical memory of a computer, the computational grid is neccessarily finite. This problem is traditionally avoided through use of a grid which is large in comparison with the scale of the physical phenomena being studied so that boundary effects are minimised. Further, asymptotic expansions of the relevant fields can be used to induce artificial boundary conditions which have small error provided that they are applied at a large distance from the source.

Working against the need to use a grid which extends to large radius, however, is the fact that the grid must also be fine enough to resolve the detailed dynamics in the area of the source. Either the physical spacing of the grid points must be reduced over the whole grid, or an adaptive grid must be introduced which has increased resolution in the region of interest. The latter method is the subject of active research, but has proven

[1] In fact, because numerical evolutions necessarily have a finite grid size (ie. an edge), the term 'Cauchy' is used somewhat loosely.

difficult to implement in generic situations. Reducing the grid spacing over the whole grid, however, has the effect of moving the boundaries closer to the relevant physical region, increasing the error introduced by approximate boundary conditions.

A further problem arises from the fact that the 3+1 evolution neccessarily involves the choice of a time direction. As a result, the grid variables on the spacelike slices are not covariantly defined and will have values depending on the chosen slicing. Covariant physical quantities, such as the emitted gravitational radiation, can only be determined by examining the asymptotic forms of the grid variables at large radius where invariant definitions exist. As the boundaries of the grid are moved inwards, these approximations become correspondingly less accurate.

Alternative evolution methods have been considered for numerical relativity in an attempt to overcome the weaknesses of the standard 3+1 approach. In particular, the study of characteristic methods by Bondi and co-workers in the early 1960s, and the related compactification of asymptotically flat spacetimes by Penrose, suggest methods by which the problems of the finite boundary can be removed completely.

In order to study gravitational radiation, Bondi et al. (1962) developed a set of coordinates based on a foliation by characteristic surfaces whose generators are null geodesics. The field equations for the corresponding metric fall into a heirarchical system for which a well defined integration procedure can be defined, given appropriate initial data. Further, in the given coordinates the asymptotic behaviour of the metric fields is well defined provided that the spacetime satisfies some notion of asymptotic flatness and that particular components of the field can be unambiguously identified with the gravitational mass and two gravitational wave polarisation modes.

Penrose (1963) demonstrated that for such spacetimes, a new set of unphysical coordinates could be specified for which the infinite extensions of the time and space directions could be placed at a finite coordinate distance. These methods have been used to great effect in the study of the asymptotic behaviour of spacetimes, as well as providing an intuitive way of representing important features of spacetime structure, the familiar Penrose spacetime diagrams. Though originally developed for the purpose of analytic studies, the technique of compactification is potentially of great use for numerical work, since it corresponds to mapping the infinite extent of a spacetime onto a finite region, and thus the need for artificial boundaries is removed.

Characteristic methods, however, do bring with them their own set of difficulties, most noticeably in the form of caustics of the null surface generators. Since the coor-

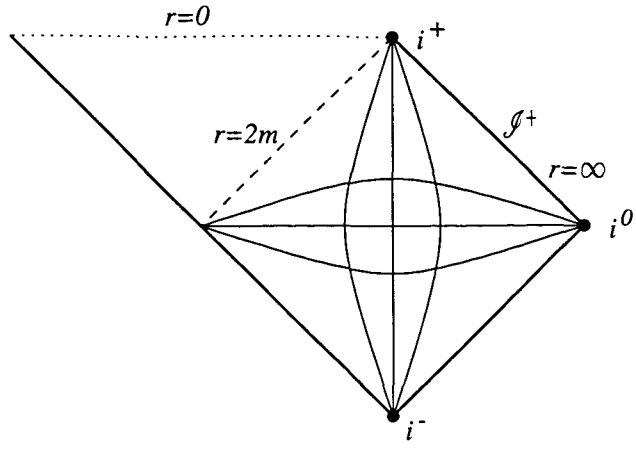


Figure 4.1: An example of coordinate compactification, the Schwarzschild spacetime in Kruskal coordinates. See Hawking and Ellis (1972) for a description.

dinate system is constructed from null geodesics, coordinate singularities result when the geodesics are focussed. This will happen in the presence of an arbitrarily small lens if it is located far enough from the radiation source. Thus, caustics are a generic feature of spacetimes possessing large-scale inhomogeneities (see the discussion in Winicour (1999)).

The problem is particularly acute in relativity, where the singularities occur in the coordinate systems themselves, rather than simply physical fields on a fixed backround as in other theories. A great deal of effort has been expended on attempting to integrate the equations past such points (see Stewart and Friedrich (1982), Friedrich and Stewart (1983)), though only point caustics in axially symmetric spacetimes have been successfully handled in this way numerically (Gómez et al. (1994)) at great computational expense. It is typical, then, to restrict characteristic evolutions to settings in which caustics can be avoided.

Cauchy-characteristic matching (CCM) is an attempt to take advantage of the strengths of both the 3+1 and characteristic evolution schemes. The spacetime is partitioned into regions consisting of an interior which is described using a 3+1 foliation, and an exterior described by a compactified null hypersurface foliation extending to null infinity. The boundary between the two is a (usually timelike) surface across which information is passed at each evolution step. Thus the characteristic region can be seen to provide accurate outer boundary conditions for the 3+1 region, which in turn provides inner boundary conditions for the characteristic region. Importantly from a motiva-

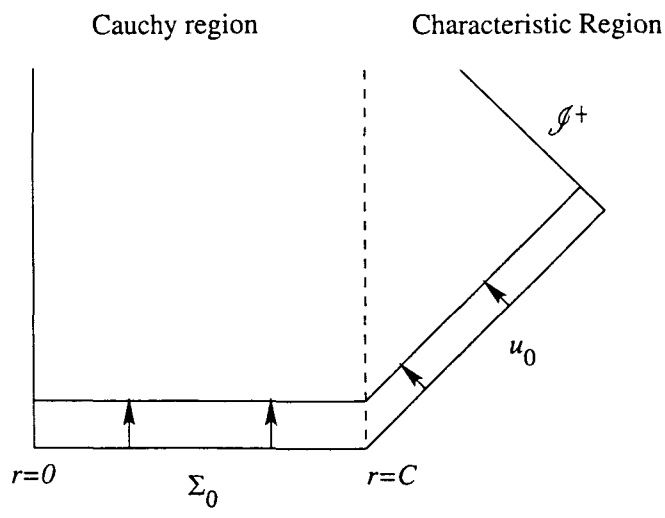


Figure 4.2: Basic setup for the CCM problem. The interior region (from $r = 0$ to some boundary $r = C$) is described by a 3+1 evolution, while the exterior, extending to $\mathscr{I}^+$, is evolved using characteristic methods.

tional standpoint, it was shown by Bishop et al. (1996) that the ratio of computation required by CCM compared to waveform extraction goes to zero as the target error in the calculation is decreased.

A great deal of work has gone into proving the viability of such methods since their introduction by Bishop (1992). Early studies passing information across an interface for a wave equation in curved space was achieved by Clarke and d'Inverno (1994). A relativistic code which passed information across a boundary in the cylindrical symmetric vacuum was investigated in Clarke et al. (1995) and Dubal et al. (1995), using an interface 'region' for which characteristic grid values were interpolated onto the Cauchy grid in order to obtain appropriate values for derivatives of metric variables on the Cauchy boundary. Recently, the cylindrical CCM problem has been revisited at Southampton in a study of the dynamics of time dependent cosmic strings (Sjödin et al. (2000)).

Cauchy-characteristic matching has also been the subject of detailed study by the Pittsburgh Relativity Group. Initial feasability tests evolved an Einstein-Klein-Gordon system in spherical symmetry (Gómez et al. (1996)). This work was later used to determine whether a characteristic code might form a useful inner boundary condition for black hole spacetimes as a potential replacement for apparent horizon excision Gómez et al. (1997). The first attempts at matching in full 3D involved a scalar wave evolution on flat spacetime Bishop et al. (1996). In this case, the interior Cauchy region used Cartesion coordinates, and interpolations were performed at the interface

to transform grid variables to the exterior radial coordinates and back. In each of these tests it was found that a characteristic exterior improved the performance over the corresponding waveform extraction to a given accuracy. A more comprehensive review of progress in the development of characteristic methods in numerical relativity, and in particular CCM methods, can be found in Winicour (1998).

The current work at Southampton focuses on constructing a Cauchy characteristic matching code in axial symmetry. A theoretical basis for studying the problem in axial symmetry was constructed by d'Inverno and Vickers (1996) and d'Inverno and Vickers (1997). This thesis expands on these foundations by providing a practical matching between a Bondi-type characteristic evolution on the exterior, and an interior Cauchy evolution based on the axisymmetric code of Stark and Piran (1987). Importantly, the Southampton approach matches numerical gridpoints in the interior to points on the exterior at the interface so that the need to interpolate data to points not on one or the other grid is minimised. Both the metric variables and their derivatives are passed between grids as required by means of the standard coordinate transformations along the $t = constant$ slices of the $r = constant$ world tube representing the interface. A consistent evolution scheme has been developed so that the boundary information required on either the interior or exterior grids of a timeslice can always be obtained from data known on the other grid.

The following sections give a general overview of the standard descriptions of spacetime foliation for both the Cauchy and characteristic regions. The treatment of the Cauchy region is the standard ADM evolution of data given by the metric and extrinsic curvature. The characteristic formulation follows that originally outlined by Bondi and Sachs. The description deviates from the original treatment in the asympototic behaviour of the metric functions. Fixing coordinates on a sphere of finite radius necessitates an important generalisation in the radial fall-off near null infinity. Specific details of the Southampton implementations of the codes are discussed in the next chapter.

4.1 Cauchy methods: Spacetime as a 3D foliation

The most commonly applied approach to solving the gravitational field equations numerically is to formulate them as an initial value problem with data specified on spacelike hypersurfaces. The first such formulation was carried out by Arnowitt et al. (1962), who developed a natural foliation of the spacetime into slices representing the 3-dimensional

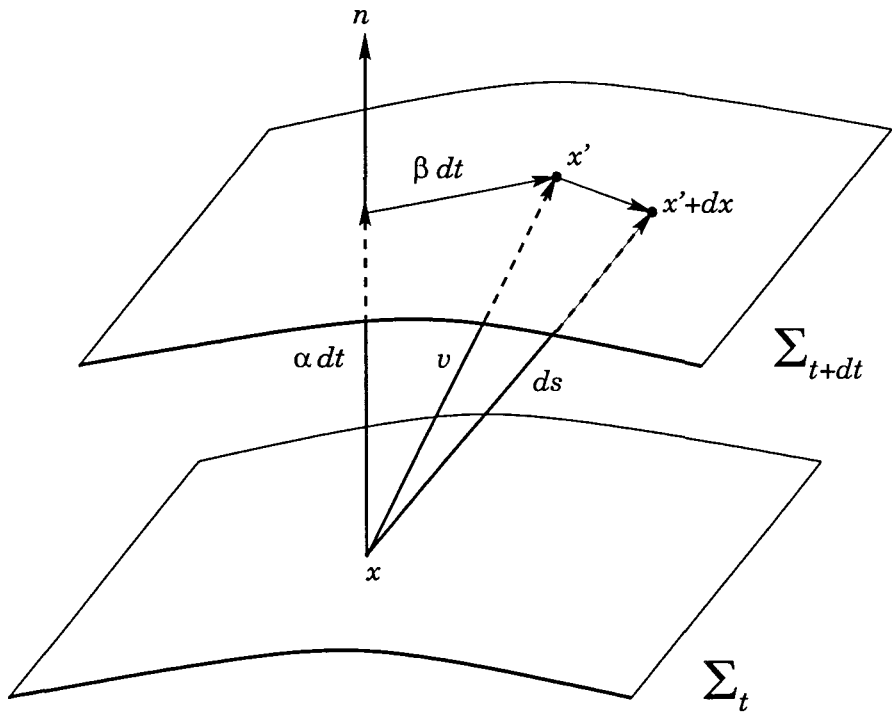


Figure 4.3: A pair of nearby spacelike slices separated by a proper time $d\tau = \alpha dt$. The grid point x is transported from Σ_t to Σ_{t+dt} along the vector v whereas Eulerian observers follow n. On Σ_{t+dt}, the displacement between the two are given by the vector $\beta^i dt$.

space at one instant in time, and specified propagation equations which, given initial data at one such instant, allow one to determine the corresponding data at any later time. This technique has been used in a number of studies, both numerical as well as analytical, where the formulation of Einstein's equations as an initial value (Cauchy) problem is required. The following section outlines the basic quantities defined by the formalism and establishes notation to be used in later sections. A comprehensive description of the formalism can be found in York (1979).

Initial data

The foliation is constructed as a family of spacelike hypersurfaces with normal vector field n^a satisfying $n^a n_a = 1$ and whose integral curves are parametrised by a variable t. A vector v^a represents the difference between a coordinate point on a single slice and the corresponding point on an adjacent slice. The 4-vector v^a can be decomposed in terms of the normal as

$$v^a = \alpha n^a + \beta^a, \tag{4.1}$$

where α is a scalar (called the *lapse*) and β is a 3-vector (the *shift*) chosen to be orthogonal to the surface normal,

$$\beta^a n_a = 0. \tag{4.2}$$

If Eulerian observers are regarded as moving along integral curves defined by n^a, then the shift can be thought of as the velocity of the coordinate system relative to these observers.

A metric is induced on a slice, Σ_t by projecting the 4-metric of the spacetime onto the slice defined by the normal, via the operator

$$P_b{}^a = P^a{}_b = \delta^a{}_b + n^a n_b, \tag{4.3}$$

so that

$$h_{ab} = P_a{}^c P_b{}^d g_{cd} = g_{ab} + n_a n_b. \tag{4.4}$$

We are free to choose a set of local coordinates $x^a = (t, x^\mu)$ which is adapted to the foliation and v^a such that the relevant quantities take the form

$$v^a = (1,0,0,0), \quad \beta^a = (0,\beta^1,\beta^2,\beta^3), \quad h^{ab} = \delta^a{}_\mu \delta^b{}_\nu h^{\mu\nu}. \tag{4.5}$$

In these coordinates, and defining $h_{\mu\nu}h^{\nu\sigma} = \delta^\sigma{}_\mu$, the four dimensional line element takes its canonical 3+1 form,

$$ds^2 = -\alpha^2 dt^2 + h_{\mu\nu}(dx^\mu + \beta^\mu dt)(dx^\nu + \beta^\nu dt). \tag{4.6}$$

The spatial 3-metric, $h_{\mu\nu}$ is sufficient to describe the intrinsic geometry of a given spacelike hypersurface. A complete description of the geometry, however, also requires a specification of how these surfaces are embedded in the 4-dimensional spacetime. This information is encoded in the *extrinsic curvature tensor* defined by

$$K_{ab} = -\frac{1}{2}\mathcal{L}_n h_{ab}, \tag{4.7}$$

where $\mathcal{L}_n$ is the Lie derivative with respect to n^a. The symmetric tensor K_{ab} is conjugate to h_{ab} and can be thought of as 'velocity' of the 3-metric on a slice. Note also that, like h_{ab}, the extrinsic curvature resides entirely within a given slice in the sense that

$$K_{ab}n^b = 0. \tag{4.8}$$

That is, it does not depend on any information which is not on the t = *constant* slice.

Field equations

Einstein's equations express a relationship between the curvature of the spacetime to the distribution of matter. In rewriting the equations in a 3+1 form, it is required, therefore, to have an expression for the curvature in terms of the data known on a slice.

The curvature tensor on a given 3-dimensional slice is defined by the commutator of the covariant derivative acting on a spacelike vector in the slice,

$$ {}^{(3)}R^{a}{}_{bcd}k^{a} = {}^{(3)}\nabla_{d}{}^{(3)}\nabla_{c}k^{a} - {}^{(3)}\nabla_{c}{}^{(3)}\nabla_{d}k^{a}. \tag{4.9} $$

where ${}^{(3)}\nabla$ is the covariant derivative in terms of the connection formed from the 3-metric on the slice, h_{ab}.

An expansion of the expressions for the second covariant derivative of k^{a} (see, for example, Hawking and Ellis (1973)), shows that ${}^{(3)}R_{abcd}$ is related to the curvature of the 4-space via Gauss' equation,

$$ {}^{(3)}R^{a}{}_{bcd} = R^{e}{}_{fgh}P^{a}{}_{e}P^{f}{}_{b}P^{g}{}_{c}P^{h}{}_{d} + K^{a}{}_{c}K_{bd} - K^{a}{}_{d}K_{bc}. \tag{4.10} $$

And the covariant derivatives of the extrinsic curvature are related to the 4-curvature via Codazzi's equation,

$$ R^{a}{}_{bcd}n^{d} = {}^{(3)}\nabla_{b}K^{a}{}_{c} - {}^{(3)}\nabla_{c}K^{a}{}_{b}. \tag{4.11} $$

The stress-energy tensor is decomposed in the 3+1 formalism into the quantities

$$ \rho = T_{ab}n^{a}n^{b}, \tag{4.12} $$

$$ j^{a} = -P^{a}{}_{c}T^{c}{}_{b}n^{b}, \tag{4.13} $$

$$ S_{ab} = P^{c}{}_{a}P^{d}{}_{b}T_{cd}, \tag{4.14} $$

representing respectively the *energy density*, *3-momentum*, and *stress tensor* of the matter fields as seen by the Eulerian observers.

The Einstein equations can be written in terms of their projections onto the spacelike slice and its normal as follows,

$$ G_{ab}n^{a}n^{b} = \frac{1}{2}({}^{(3)}R + (\mathrm{tr}K)^{2} - K_{ab}K^{ab}) = 2\pi\rho, \tag{4.15} $$

$$ G_{cb}P^{ac}n^{b} = {}^{(3)}\nabla_{b}(K^{ab} - h^{ab}\mathrm{tr}K) = 8\pi j^{a}, \tag{4.16} $$

$$ G_{cd}P^{c}{}_{a}P^{d}{}_{b} = S_{ab}. \tag{4.17} $$

where the Gauss-Codazzi equations have been used to simplify (4.15) and (4.16). Equation (4.17) can be solved for the time derivative of the extrinsic curvature to give

$$\begin{aligned}\mathcal{L}_n K_{ab} = \alpha(R_{ab} - 2K_{ac}K^c{}_b + K_{ab}\mathrm{tr}K) - 8\pi(S_{ab} - \frac{1}{2}h_{ab}\mathrm{tr}S) \\ - 4\pi\rho h_{ab} - {}^{(3)}\nabla_{ab}\alpha.\end{aligned} \tag{4.18}$$

With this expression and the definition of the extrinsic curvature via Eq. (4.7), time derivatives of both K_{ab} and h_{ab} are completely specified. In this sense, $\{h_{ab}, K_{ab}\}$ can be thought of as appropriate data to characterise the entire geometry of the spacetime; if they are known on a single slice then Eqs. (4.17) can be used to evolve them to a later time.

Equations 4.18 represent four of the ten Einstein equations. Equations (4.15) and (4.16), remain unused and would seem to overdetermine the system. Note that neither contain time derivatives, and thus depend only on data on a given slice. They are thus called the *constraint equations* and must be satisfied by any set data set $\{h_{ab}, K_{ab}\}$ that is consistent with the Einstein equations. In fact, the Bianchi identities can be used to show that if the constraint equations are satisfied on a single slice, then they will be satisfied at any later time Wald (1984).

The constraint equations are not explicitly required by any evolution scheme, but can be useful in numerical implementations, as they are sometimes easier to solve for a given piece of data than a corresponding evolution equation, or can be used to simplify the form of the evolution equations. Further, they can be used to provide a useful check of the accuracy of a numerical solution by comparing the data determined from the constraints with the evolved data. Evolution schemes which enforce the constraint equations at each timestep are known as *constrained evolutions.* As they usually take the form of elliptic equations, however, the constraint equations can be computationally expensive to solve, and thus the accuracy gained from a fully constrained evolution tends not to compensate for the increased time required. Further, procedures for enforcing the constraint equations are not well developed.

As a final note, it should be realised that the ADM system of equations for the metric and extrinsic curvature are by no means the only possible evolution system that can be used. In fact, the stability properties of the ADM equations remain obscure because of their complicated form. A great deal of effort has gone into the reformulation of the Einstein equations as a hyperbolic system to which well-known stability theorems can be applied (Reula (1998), Friedrich and Rendall (2000)). This reformulation usually comes at the expense of introducing additional metric variables as data, with the reward that

the resulting increased number of equations are of a simpler form and can be more easily converted to stable finite difference equations. In recent years a number of promising numerical results have emerged using such symmetric hyperbolic systems, suggesting that the comparitively 'brute-force' calculations of ADM may eventually have to yield to more subtle mathematics for the stable solution of some problems (for instance Friedrich (1981)).

4.2 Characteristic methods: The Bondi-Sachs coordinate system

The previous section described a method of slicing a 4-dimensional spacetime into a set of spacelike 3-surfaces, providing an intuitive picture of spacetime taken as a series of snapshots, each describing an instant of time as seen by some observer. Although it seems natural to divide the spacetime into space and time in this way, for many applications alternative formulations have proven quite useful in interpreting the equations and the behaviour of the physical variables. In particular, early studies of the nature of gravitational radiation focussed heavily on the analogies between the gravitational field equations and the well studied wave equations of electrodynamics (Pirani (1965)). As in electromagnetic theory, the existence of gravitational wavefronts is related to the ability to construct non-analalytic solutions of the field equations. Such discontinuities occur along *characteristic surfaces*, which can be shown to be the null surfaces in a general spacetime.

The characteristic formulation of the Einstein field equations became the subject of active study in the early 60s with the seminal works of Bondi et al. (1962) and Sachs (1962). They fixed an appropriate set of coordinates along the null geodesics of the spacetime. This relationship between the coordinates and the geometry simplified both the field equations as well as the interpretation of the gravitational degrees of freedom by means of exact asymptotic analysis.

Although the Bondi system has been treated in numerous places in the literature, a clear statement of the restrictions to the metric and coordinates involved is crucial for the specification of the interface with the 3+1 interior required by CCM. The following section unifies material which can be found elsewhere, but establishes and emphasises the points which will be relevant to the development of the Southampton CCM code, in particular with regards to the asymptotic falloff of the metric functions.

The Bondi-Sachs coordinate system is constructed by first supposing that there exists

a function $u(x)$ defined on the spacetime whose level surfaces are null,

$$g^{ab}\partial_a u \partial_b u = 0. \tag{4.19}$$

This involves no loss of generality, since it can be shown that if the field equations are to admit characteristic surfaces (ie. if wave-like solutions exist) then these surfaces are null and conversely that null surfaces are characteristic Pirani (1965).

Define k^a to be the normal to the $u = constant$ level surfaces. Then the k^a are also null

$$k^a = g^{ab}\partial_a u \implies k_a k^a = 0, \qquad k^b \nabla_b k^a = 0. \tag{4.20}$$

The vector field k^a determines a congruence of null geodesics which is both normal to the surfaces $u = constant$ but which also lies within these very surfaces.

Through any point, the family of ingoing and outgoing null geodesics each have spherical topology, and as such, can be labelled using polar-type coordinates θ and ϕ. Assume that this has been done on some appropriate S^2 within the spacetime. If they are chosen so that

$$k^a \partial_a \theta = 0, \qquad \text{and} \qquad k^a \partial_a \phi = 0 \tag{4.21}$$

then θ and ϕ are constant along each null ray.

Finally, points along the geodesics are parametrised by a scalar function r which can be written as a function of u, θ and ϕ whose only restriction is that Jacobian be non-zero. Among other things, this demands that the expansion of the congruence be non-zero

$$\rho = \frac{1}{2}\nabla_a k^a \neq 0, \tag{4.22}$$

excluding plane or cylindrical waves from the coordinate construction.

Since r is the only coordinate which varies along the null geodesics, tangent vectors to the null congruence have the form

$$l^a = (0, 1, 0, 0) = \delta^a{}_1, \tag{4.23}$$

in terms of the coordinates $(x^0, x^1, x^2, x^3) = (u, r, \theta, \phi)$. These are parallel to the k^a defining the congruence so that

$$k^a = \omega l^a = \omega \delta^a{}_1 \tag{4.24}$$

for some factor $\omega(u,\theta,\phi)$. These properties result in certain coordinate conditions on the metric. Namely, by construction we have

$$g^{a0} = l^a = \omega\delta^a{}_1. \tag{4.25}$$

Thus a metric based on the given coordinates satisfies

$$g^{00} = g^{02} = g^{03} = 0. \tag{4.26}$$

The final coordinate condition which can be applied is the choice of parametrisation of r, corresponding to a choice of ω. Bondi chose r to be the *luminosity distance*, defined by fixing the determinant of the g_{AB}

$$\det(g_{AB}) = \begin{vmatrix} g_{\theta\theta} & g_{\theta\phi} \\ g_{\phi\theta} & g_{\phi\phi} \end{vmatrix} = r^4 \sin^2\theta. \tag{4.27}$$

The result of applying this condition is that 2-spheres of constant u and r have a proper area $4\pi r^2$.

The four conditions (4.26) and (4.27) fix the metric as far as it is possible using the coordinates. The result is a metric which (following Bondi) we write in the form

$$ds^2 = -(Ve^{2\beta}/r)du^2 - 2e^{2\beta}dudr + r^2 h_{AB}(dx^A - U^A du)(dx^B - U^B du), \tag{4.28}$$

where

$$h_{AB} = \begin{pmatrix} e^{2\gamma}\cosh 2\delta & \sinh 2\delta \\ \sinh 2\delta & e^{-2\gamma}\cosh 2\delta \end{pmatrix}. \tag{4.29}$$

The six metric functions $\{\gamma, \delta, \beta, U^2, U^3, V\}$ correspond to the six degrees of freedom remaining in the metric once the coordinate conditions have been applied, and have been chosen in such a way as to provide ready geometric and physical interpretations. Specifically, the variables δ and γ define the conformal geometry of the spacelike surfaces of constant radius. They encode the two radiative degrees of freedom, as will be seen from the discussion of the Bondi news functions in Section 4.2, below (d'Inverno and Stachel (1978). An explicit calculation of Eq. (4.22) yields

$$\rho = e^{-2\beta}/r, \tag{4.30}$$

lending β an interpretation in terms of the expansion of the null congruence. The field V is the analog of the Newtonian potential and, as will be seen from its asymptotic expansion, encodes the mass aspect of the system. In fact, for the Schwarzschild spacetime,

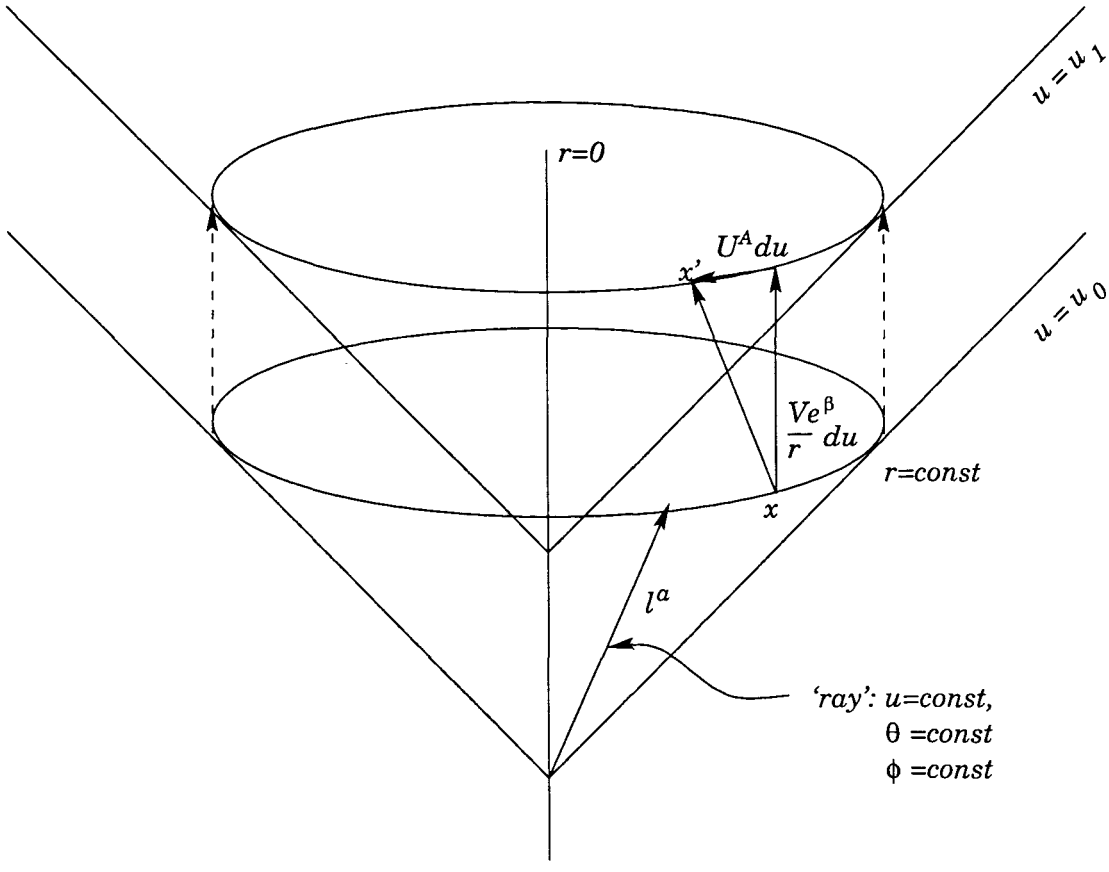


Figure 4.4: The Bondi-Sachs coordinate system fixes the coordinates to null rays of the spacetime, parametrised by the luminosity parameter r. The metric functions V and U^A act analogously to the lapse and shift in transporting coordinates from one $r = constant$ slice to another.

$V = M$, the constant parameter describing the mass of the black hole. A geometric interpretation arises from a consideration of the line element on an $r = constant$ surface,

$$ds^2 = -(Ve^{2\beta}/r)du^2 + r^2 h_{ab}(dx^A - U^A du)(dx^B - U^B du). \tag{4.31}$$

A comparison with the 3+1 line element, Eq. (4.6), suggests that $Ve^{2\beta}/r$ acts as a lapse function, $-U^A$ as a shift, and $r^2 h_{AB}$ as the metric on 2-surfaces of constant u which foliate the $r = constant$ world-tube.

The metric is specialised to axisymmetry by simply restricting the dependence of the metric variables on the coordinates so that

$$\frac{\partial \delta}{\partial \phi} = \frac{\partial \gamma}{\partial \phi} = \frac{\partial \beta}{\partial \phi} = \frac{\partial U}{\partial \phi} = \frac{\partial W}{\partial \phi} = \frac{\partial V}{\partial \phi} = 0, \tag{4.32}$$

and the further restriction $\gamma = \delta$ can be used to ensure that the ϕ is a hypersurface orthogonal Killing vector.

It is appropriate to make one more important point regarding the choice of the θ and ϕ coordinates. In the original treatment of Bondi, these coordinates are fixed at $\mathscr{I}^+$ where u is known to be a non-spacelike coordinate. For the purposes of matching the characteristic coordinates to an interior Cauchy slicing, it will be convenient to choose θ and ϕ on a closed surface at finite r, representing the interface. For such a situation, care must be taken to ensure that as the null geodesics map these coordinates to $\mathscr{I}^+$, u remains non-spacelike there. The analogy used by Bondi is that of a spotlight which projects a beam onto some surface distant enough that successive points have spacelike rather than timelike separation. This can be checked by verifying that the g_{00} component of the metric remains positive, and might be expected to cause particular difficulties for coordinate systems co-rotating with some interior matter source, or an interface within the Killing horizon of a Kerr metric (ie. subjected to frame dragging). There are indications, however, that such a spacelike shift at $\mathscr{I}^+$ may not pose great difficulties to numerical evolutions (Bishop et al. (1997), Brady et al. (1998)).

Field equations

Some very useful features of the Bondi-Sachs formulation become apparent when the vacuum field equations are written out in terms of the metric variables defined in the previous section. The set of ten equations can be grouped into the following classification, originally due to Bondi:

i. *main equations*:

   (a) *hypersurface equations* (4), $R_{rr} = R_{rA} = h^{AB}R_{AB} = 0$

   (b) *dynamical equations* (2), $R_{AB} - h_{AB}h^{CD}R_{CD}/2$

ii. *trivial equation* (1), $R_{ur} = 0$

iii. *supplementary conditions* (3), $R_{uu} = R_{uA} = 0$

The reasons for classifying the equations in this way is determined by the contracted Bianchi identities which can be used to show that the trivial equation is satisfied automatically as a consequence of the main equations. The supplementary conditions are also satisfied over the entire spacetime provided that they are satisfied on a single $r = constant$ slice.

The splitting of the main equations into two subgroups is determined by the fact that the 'hypersurface equations' contain no derivatives of the metric components with

respect to the retarded time u. This allows them to be evaluated on a single $u =$ *constant* hypersurface. The remaining pair, however, contain time derivatives of the spatial metric, $\partial_u h_{AB}$ and thus depend on information off of the slice.

The hypersurface equations take on the following form when written out schematically. First of all, from the R_{rr}, $R_{r\theta}$, $R_{r\phi}$ and $g^{AB}R_{AB}$ components respectively,

$$\partial_r \beta = F_1(\gamma, \delta), \tag{4.33a}$$

$$\partial_r P = F_2(\gamma, \delta, \beta), \tag{4.33b}$$

$$\partial_r Q = F_3(\gamma, \delta, \beta), \tag{4.33c}$$

$$\partial_r U = F_4(\gamma, \delta, \beta, P, Q), \tag{4.33d}$$

$$\partial_r W = F_5(\gamma, \delta, \beta, P, Q), \tag{4.33e}$$

$$\partial_r V = F_6(\gamma, \delta, \beta, U, W) \tag{4.33f}$$

where P and Q are intermediate variables defined by

$$\begin{pmatrix} P \\ Q \end{pmatrix} = r^4 e^{-2\beta} \begin{pmatrix} e^{2\gamma}\cosh(2\delta) & \sinh(2\delta) \\ \sinh(2\delta) & e^{-2\gamma}\cosh(2\delta) \end{pmatrix} \begin{pmatrix} \partial_r U \\ \partial_r W \end{pmatrix} = r^4 e^{-2\beta} h_{AB} \partial_r U^A. \tag{4.34}$$

The right-hand sides of equations (4.33a)-(4.33f) are functions which depend only on their specified arguments and their spacelike (r, θ, ϕ) derivatives. This suggests that if δ and γ are known on a single $u =$ *constant* slice, then β, U and W can be determined by successive integrations of (4.33a), (4.33b), (4.33c), and the inversion of (4.34). Then V can be determined on the slice by integrating the final hypersurface equation (4.33f). That is, knowledge of δ and γ on a single slice allows one to determine all of the other metric variables on that same slice.

The two evolution equations can be written in terms of auxillary variables

$$\phi_1 = r\cosh(2\delta)\partial_u \gamma \qquad \phi_2 = r\partial_u \delta \tag{4.35}$$

in the form:

$$\partial_r \phi_1 + f\phi_2 = F_7(\gamma, \delta, \beta, U, W, V), \tag{4.36a}$$

$$\partial_r \phi_2 - f\phi_1 = F_8(\gamma, \delta, \beta, U, W, V), \tag{4.36b}$$

where $f = 2\sinh(2\delta)\partial_r \gamma$. This is coupled set of differential equations which, when solved, yield the u derivatives of γ and δ. Once again, the right-hand sides of these equations depend only on the metric functions and their spacelike derivatives. Thus, if the metric functions have been determined over a given slice, then the time evolution of

γ and δ can be determined. This knowledge can be used to determine γ and δ at some nearby future time, at which point enough data exists that the process can be repeated to determine the remaining metric variables on the slice.

The entire Bondi process involves a total of eight first-order integrations in r and two in u, each of which introduces an integration 'constant'. The constants for the u integrations are fixed by initial data for δ and γ. The remainder can be dealt with as follows:

- Three (P_0, Q_0, V_0) can be specified as initial conditions on an $r = constant$ surface and integrated forward in time using the supplementary conditions.
- Two (U_0, W_0) can be set using an appropriate outgoing radiation condition at null infinity.
- One (β_0) can be set to zero by virtue of coordinate transformations provided the outgoing radiation condition is satisfied.
- Two $(\partial_u \delta_0, \partial_u \gamma_0)$ remain freely specifiable and, as discussed below, correspond to the quadrupole modes of gravitational radiation.

The entire evolution of the system is determined by the specification (γ, δ) on an initial $u = constant$ slice once the integration constants $(\partial_u \delta_0, \partial_u \gamma_0)$ have been fixed. In particular, if the latter are specified on some finite $r = constant$ 2-surface, they determine δ and γ on the entire surface all the way out to $\mathscr{I}^+$, prompting Bondi to christen their u-derivatives the *news functions*.

The presence of integration constants can be related to the form of the asymptotic expansions of the metric variables and will be discussed in more detail in the next section. For the moment, however, we note that the listed methods for reducing the (U_0, W_0, β_0) to zero depend on a choice of coordinates at $\mathscr{I}^+$. For the purpose of the CCM numerical code, this is not ideal, as it would be preferable to be able to fix the coordinates at the interface with the Cauchy region. As a result, in practice the restrictions on these variables will not be applied. Instead each of the functions of integration will be set by extracting data from the Cauchy interior at the $r = constant$ interface. The supplementary conditions become redundant, but can be used as a check on the accuracy of the integrations in much the same way as the constraint equations are used in the 3+1 formulation.

Asymptotic behaviour of the metric variables

An important feature of the original treatment of the Bondi metric was the careful analysis of the asymptotic behaviour of the metric functions. By restricting consideration to isolated sources, the metric was supposed to be asymptotically Minkowskian with only outgoing radiation. Bondi showed that these conditions could be used to set strict conditions on the falloff of each of the metric functions as $r \to \infty$.

For the moment, we will suppose only a very general falloff of each of the metric functions, so that as $r \to \infty$ on a single $u = constant$ hypersurface, Σ, we have

$$\delta \to \delta_0, \quad \gamma \to \gamma_0, \quad U \to U_0, \quad W \to W_0, \quad \beta \to \beta_0 \tag{4.37}$$

It can be shown (see Sachs (1962), Christodoulou and Klainerman (1993)) that a coordinate transformation ψ exists such that δ and γ are reduced to zero,

$$\tilde{\delta}_0 = \psi^* \delta_0 = 0, \qquad \tilde{\gamma}_0 = \psi^* \gamma_0 = 0. \tag{4.38}$$

This is equivalent to transforming the metric of the asymptotic S^2 to the standard metric on a sphere,

$$\tilde{h}_{AB} dx^A dx^B = d\tilde{\theta}^2 + \sin^2\theta d\tilde{\phi}^2. \tag{4.39}$$

A further coordinate transformation, χ, can be applied to reduce the asymptotic behaviour of U and W to

$$\tilde{U}_0 = \chi^* U_0 = 0, \qquad \tilde{W}_0 = \chi^* W_0 = 0, \tag{4.40}$$

while preserving the asymptotic form of δ and γ. Significantly, the applied transformations involve only equations on the hypersurface. The asymptotic behaviour of β can also be fixed through a coordinate transformation so that

$$\tilde{\beta}_0 = 0. \tag{4.41}$$

However the required coordinate transformation generally is u dependent. It corresponds to a deformation of Σ which can be viewed as a condition on the lapse, or a slicing condition.

In expanding the metric functions asymptotically, Bondi et al. (1962) came across the following disturbing behaviour: If a polynomial expansion was assumed asymptotically for δ and γ,

$$\delta \to \delta_0 + \frac{\delta_1}{r} + \frac{\delta_2}{r^2} + O(r^{-3}), \qquad \gamma \to \gamma_0 + \frac{\gamma_1}{r} + \frac{\gamma_2}{r^2} + O(r^{-3}), \tag{4.42}$$

then the corresponding expansions of U, W, and V involved terms logarithmic in r. They found, however, that they could remove these terms through the application of an intuitively reasonable condition, corresponding roughly to a Sommerfeld radiation condition. For general non-zero asymptotic δ and γ, this can be formulated as

$$h^2_{AB} - \frac{1}{2}h^{0CD}h^2_{CD}h^0_{AB} = 0, \tag{4.43}$$

where

$$h^0_{AB} = \lim_{r\to\infty} h_{AB}\Big|_\Sigma, \qquad h^2_{AB} = \lim_{r\to\infty} \frac{r^2}{2}\frac{\partial^2}{\partial r^2}(rh_{AB})\Big|_\Sigma, \tag{4.44}$$

or using the more specific falloffs of Eq. (4.38), the more familiar form

$$\lim_{r\to\infty} r^2\frac{\partial^2}{\partial r^2}(r\gamma)\Big|_\Sigma = 0, \qquad \lim_{r\to\infty} r^2\frac{\partial^2}{\partial r^2}(r\delta)\Big|_\Sigma = 0. \tag{4.45}$$

(Bondi et al. (1962)). These conditions form a restriction on the coefficients γ_2 and of the δ_2 of the expansions (4.42) which are enough to remove logarithmic terms from the asymptotic expansions of the other metric variables. The restrictions take the form

$$\gamma_2 = -2\tanh(2\delta_0)\gamma_1\delta_1, \qquad \delta_2 = \sinh(2\delta_0)\cosh(2\delta_0)\gamma_1^2, \tag{4.46}$$

d'Inverno and Vickers (1997), which are reduced to zero for the Bondi asymptotic expansions, $\gamma_0 = \delta_0 = 0$.

If these conditions hold, then it can be shown that the remaining metric variables have polynomial expansions:

$$\beta \to \beta_0 + \frac{\beta_1}{r} + \frac{\beta_2}{r^2} + O(r^{-3}), \tag{4.47a}$$

$$U \to U_0 + \frac{U_1}{r} + \frac{U_2}{r^2} + O(r^{-3}), \tag{4.47b}$$

$$W \to W_0 + \frac{W_1}{r} + \frac{W_2}{r^2} + O(r^{-3}), \tag{4.47c}$$

$$V \to V_{-2}r^2 + V_{-1}r + V_0 + \frac{V_1}{r} + \frac{V_2}{r^2} + O(r^{-3}), \tag{4.47d}$$

where the expansion coefficients are all functions of (u, θ, ϕ). The transformation, $\chi \circ \psi$, to coordinates for which the leading coefficients of γ, δ, U and W are zero can be written in terms of the expansion coefficients as:

$$\frac{d\tilde{\theta}}{\sin\tilde{\theta}} = e^{2\gamma_0}\cosh 2\delta_0\left(\frac{d\theta - U_0 du}{\sin\theta}\right) + \sinh 2\delta_0\left(d\phi - \frac{W_0}{\sin\theta}du\right), \tag{4.48a}$$

$$d\tilde{\phi} = d\phi - \frac{W_0}{\sin\theta}du, \tag{4.48b}$$

$$\tilde{r} = r/\omega, \tag{4.48c}$$

where

$$\omega = \sqrt{\cosh 2\delta_0} e^{\gamma_0} \frac{\sin\tilde{\theta}}{\sin\theta}. \tag{4.49}$$

The restriction, (4.41), of the asymptotic behaviour of β to a Bondi-type falloff for which $\tilde{\beta}_0 = 0$ is determined from the g_{ur} component of the metric. Under the given transformation,

$$-e^{-2\tilde{\beta}} = \tilde{g}^{\tilde{u}\tilde{r}} = g^{ab} \frac{d\tilde{u}}{dx^a} \frac{d\tilde{r}}{dx^b} = \frac{-e^{-2\beta}}{\omega}. \tag{4.50}$$

Thus we require

$$\beta_0 = -\frac{1}{2} \log \omega, \tag{4.51}$$

to ensure that each of the metric coefficients can be reduced to their Bondi-type asymptotics.

In summary, we have applied two restrictions on the asymptotic behaviour of the metric functions. The first is that they have a polynomial falloff as $r \to \infty$ without logarithic terms. This is equivalent to the imposition of an 'outgoing radiation condition', Eq. (4.43). The second restriction is that the spacetime should be sliced in such a way that the reduction of δ, γ, U and W to their asymptotic Bondi form also reduces β. This will enable us to more easily make the transformation to Bondi coordinates when it becomes neccessary to extract the outgoing radiation and mass parameters.

It should be noted that the necessity of applying these conditions has in recent years come into question. At the time of the original analysis, an appropriate definition of aysmptotic flatness had not been formulated. Instead, the spacetime was required to be asymptotically Minkowskian and the outgoing radiation condition considered a reasonable requirement. In the intervening years as the understanding of the nature of $\mathscr{I}^+$ has crystalised (see, for instance, the discussion in Wald (1984), or the definitions provided by Hawking and Ellis (1973) or Stewart (1990)), these conditions have been found to be excessively restrictive. (In particular, it can be shown that when the conditions for a metric to be conformally compactified are satisfied, then $\mathscr{I}^+$ is an incoming null surface.) Thus there can be no influx of null or timelike fields across $\mathscr{I}^+$ into the spacetime, and the outgoing radiation condition becomes redundant. Without this restriction, the asymptotic falloff of the metric variables becomes more complicated, as logarithmic terms are no longer prevented from appearing. Such spacetimes have been recently studied by Winicour (1985) and Chruściel et al. (1995). Note, however, since

the numerical solution of the equation proceeds in an outward direction and does not at any point make use of the asymptotic expansion of the metric variables or enforce any particular asymptotic behaviour, the presence of log terms will not influence the behaviour of evolution codes constructed from the Bondi equations.

5

The Southampton axisymmetric CCM code

The method of Cauchy-characteristic matching is a particularly difficult one to implement and test numerically, for it essentially involves the test of three independent codes. In order to have a reliable test of the interface, both the interior and exterior codes should be known to be stable as independent codes using (exact or artificial) boundary conditions. A successful test of the full CCM code would then require that when the codes are connected across an interface, a similar level of stability and accuracy should be obtained.

Axisymmetric Cauchy evolutions were first applied to the study of gravitational collapse of stellar formations by Nakamura and Sato (1981). Their scheme tracked the evolution of matter and the formation of a black hole, but was not accurate enough to compute the emitted gravitational radiation. More sophisticated approaches were taken by Evans (1986), for non-rotating matter configurations, and Abrahams et al. (1994) for the case of rotating collisionless (dust) particles. Dynamic axisymmetric fields in vacuum have been used in the study of the collapse of pure gravitational waves to form a black hole Abrahams and Evans (1992), Abrahams and Evans (1993), as well and the evolution of distorted, rotating and colliding black holes (Bernstein et al. (1994), Brandt and Seidel (1995), Anninos et al. (1995)) as a precursor and complement to the study of these problems in full 3D by the Grand Challenge Alliance. More recently, it has been suggested that reliable and efficient axisymmetric codes can be implemented by a reduction of a fully 3D code to a 'plane' with appropriate boundary conditions. The use of Cartesian coordinates avoids many of the axis problems commonly associated with numerical implementations of axisymmetric problems (Krivan et al. (1997), Alcubierre et al. (1999)).

In the mid-1980s, the numerical evolution of matter fields in axisymmetry was visited by Stark and Piran (1987). Their code was tested in both vacuum and matter configurations, and was used to compute the gravitational radiation emitted from a collapsing black hole based on a formalism developed in Bardeen and Piran (1983) which took special care to construct ADM-type equations based which could be reliably implemented in a stable manner. Further, their evolution variables were chosen for their well-defined behaviour at both the inner (r=0) and outer boundaries, and at the poles where axisymmetric codes have traditionally encountered difficulties. At the outer boundary, the choice of variables allowed the two independent polarisation amplitudes of emitted radiation to be easily extracted to good approximation. These strengths suggested that a code based on the methods of Stark and Piran would form a good basis for the Cauchy evolution module of an axisymmetric CCM code. The details of the the implementation of the Cauchy region are described below in Section 5.1.

The application of characteristic methods in axisymmetry has been far less common. In fact, because the characteristic field equations gain little in complexity in the transition from 2 to 3 dimensions, and because their numerical application was first studied somewhat later than the corresponding ADM-type evolutions (and hence at a time when computing power was less of a limiting factor), the tendency has been to move directly to the development of 3D characteristic codes, skipping axisymmetry entirely. A number of successes have been reported in this arena, especially in the work at the Pittsburgh Relativity Group, most notably in the stable evolution of a distorted black hole to more than 10000M (Gómez et al. (1998), Gómez (1998)). Their codes make use of the 'eth' formalism to construct coordinates and derivative operators free of the singularities normally associated with spherical coordinate systems. Evolution is carried out using a second order scheme based on the surface area of parallelograms in double-null coordinates (Gómez et al. (1992)). More recently, this model has been used in axisymmetry to accurately study the geometry of the event horizons of colliding black holes (Husa and Winicour (1999)).

For the Southampton matching code, it was decided that an independent code would be developed based on the Bondi-Sachs formalism described in the previous section. This would allow a degree of customisation of the coordinates and variables at the interface so as to provide a more convenient matching. In particular, by matching angular coordinates across the interface, the amount of interpolation between variables on the interior and exterior grids could be minimised. The cost entailed through the use

of polar coordinates are potential problems with behaviour of metric functions atthe origin. These are handled via appropriate expansions of the metric functions in these regions, as described below. Details of the equations used are given in Section 5.2.

In numerically passing data across the interface from Cauchy to characteristic regions, it is important to keep track of the relationship between the locations of the two regions' respective computational grid points. Points on the numerical Cauchy grid generally do not lie on points of the characteristic grid, however by fixing the coordinates on the exterior region relative to the interior, these points can be chosen to line up on a single $t = constant, r = constant$ surface (a line in axisymmetry where invariance in the ϕ direction is imposed). The metric data can then be carried across the interface via the standard coordinate transformations between regions, and are placed exactly on points of the opposing grid. This removes a common source of error, namely the interpolation of unevenly spaced data onto a numerical grid. The Southampton approach transforms both metric functions and their derivatives, where necessary, along a single line to provide appropriate boundary data for the opposing region. Modules for both extracting ($Cauchy \rightarrow characteristic$) and injecting ($Cauchy \leftarrow characteristic$) data have been developed. Information is passed in both directions as required, resulting in true 'matching' of data between the regions. The requisite equations describing the transformations are outlined in Section 5.3.

The Southampton axisymmetric CCM project is a collaborative effort carried out by Chris Clarke, Mark Dubal, Ray d'Inverno, James Vickers, and the author. The implementation of the Cauchy code as an independent entity was largely the work of Mark Dubal, and modified by the author for the purposes of attaching a characteristic interface. Chris Clarke began an implementation of the interface which was largely rewritten and extended by the author. The implementation of the characteristic code, as well as the design of the complete system of interacting codes comprises the main body of the author's original work which is described by this thesis.

5.1 The Stark-Piran axisymmetric 3+1 scheme

The formulation of the Einstein equations used by Stark and Piran (1987) follows the standard ADM treatment presented in the previous chapter with specific choices of shift and slicing conditions used to simplify the evolution system. In their notation, the line

element is written

$$ds^2 = -(\alpha^2 - N_\alpha N^\alpha)dt^2 - 2N_\alpha dx^\alpha dt + h_{\alpha\beta}dx^\alpha dx^\beta, \tag{5.1}$$

A local coordinate system, $(x^a) = (t, x^\alpha) = (t, r, \theta, \phi)$, which is adapted to the foliation is introduced so that

$$v^a = \alpha n^a + N^a = (1, 0, 0, 0), \quad N^a = (0, N^r, N^\theta, N^\phi), \quad h^{ab} = \delta^a{}_\alpha \delta^b{}_\beta h^{\alpha\beta}. \tag{5.2}$$

are the coordinate transport, shift, and spatial 3-metric (introduced in Section 4.1).

Coordinate conditions are used to simplify the form of the metric tensor and Einstein equations. In particular, θ and ϕ are chosen to represent spherical polar coordinates so that the off-diagonal components $h_{r\theta}$ and $h_{r\phi}$ vanish, and the spatial metric can be written in the form

$$h_{\alpha\beta}dx^\alpha dx^\beta = A^2 dr^2 + r^2(B^{-2}d\theta^2 + B^2 \sin^2\theta(d\phi + \xi \sin\theta)^2). \tag{5.3}$$

The quantity B is related to an auxiliary variable η via the definition

$$B^2 = 1 + \eta \sin^2\theta, \tag{5.4}$$

where η and ξ are the two independent polarisation amplitudes, h_+ and $h_\times$ respectively, of gravitational waves in the transverse traceless gauge,

$$h_+ := \eta \sin^2\theta, \qquad h_\times := -\sin^2\theta \int_0^t \alpha A^{-1} \xi, \tag{5.5}$$

for large values of the radial coordinate.

The parametrisation of the radial gauge requires some consideration. Bardeen and Piran (1983) suggest that for the system in question, some advantage can be gained if the r coordinate is fixed to satisfy the condition that the determinant of

$$\det h_{\alpha\beta} = r^4 \sin^2\theta, \tag{5.6}$$

ie. that surface elements of constant r have area $r^2 \sin\theta d\theta d\phi$. In particular, they find that this *radial gauge* leads to a parabolic equation for the shift component N^θ, in contrast to the elliptic equation associated with the more commonly used *isothermal gauge* (a generalisation of isotropic coordinates to non-spherically symmetric spacetimes).

With these coordinate choices in place, the metric data which are evolved are the following:

$$\begin{aligned} & & \alpha, & \quad \text{(lapse)} \\ \beta^r = N^r/r, \quad G = N^\theta/\sin\theta, & \quad N^\phi, & & \quad \text{(shift)} \\ & & A & \quad \text{(the radial metric component } \sqrt{h_{rr}}\text{)} \\ & \eta, \quad \xi, & & \quad \text{(wave-modes).} \end{aligned}$$

Extrinsic curvature components are projected onto a basis of orthonormal vectors,

$$e^\alpha_{\mathbf{1}} = [A^{-1}, 0, 0], \qquad e^\alpha_{\mathbf{2}} = [0, B/r, -\xi B \sin\theta/r], \qquad e^\alpha_{\mathbf{3}} = [0, 0, 1/(Br\sin\theta)], \tag{5.7}$$

so that the independent components of the extrinsic curvature in axially symmetry can be represented by the set

$$\begin{aligned} K_1 = K_{\mathbf{11}}, \qquad K_2 = K_{\mathbf{12}}, \qquad K_3 = K_{\mathbf{13}}, \\ K_+ = \frac{1}{2}(K_{\mathbf{33}} - K_{\mathbf{22}})/\sin^2\theta, \qquad K_\times = K_{\mathbf{23}}/\sin^2\theta, \end{aligned} \tag{5.8}$$

where the form of K_+ and $K_\times$ are chosen so that they represent the even and odd parity modes (in the linearised case) and are conjugate to η and ξ respectively.

Gauge conditions

An unconventional slicing condition is used in order to improve the behaviour of the metric variables at the interior ($r = 0$) and exterior boundaries, as well as simplify the integration of the field equations. Bardeen and Piran (1983) studied the condition of *polar slicing*, defined by

$$K = K^r{}_r, \tag{5.9}$$

extensively. In regards to the variables defined above, they find it has a number of advantages over the more commonly applied *maximal slicing* for which the trace of the extrinsic curvature is zero,

$$K = 0. \tag{5.10}$$

For the given system, polar slicing involves a parabolic equation for the lapse which is much less expensive to solve numerically than the corresponding elliptical equation in maximal slicing. Polar slicing also exhibits strong singularity avoiding properties, and when used in combination with the radial gauge results in a simple outer boundary

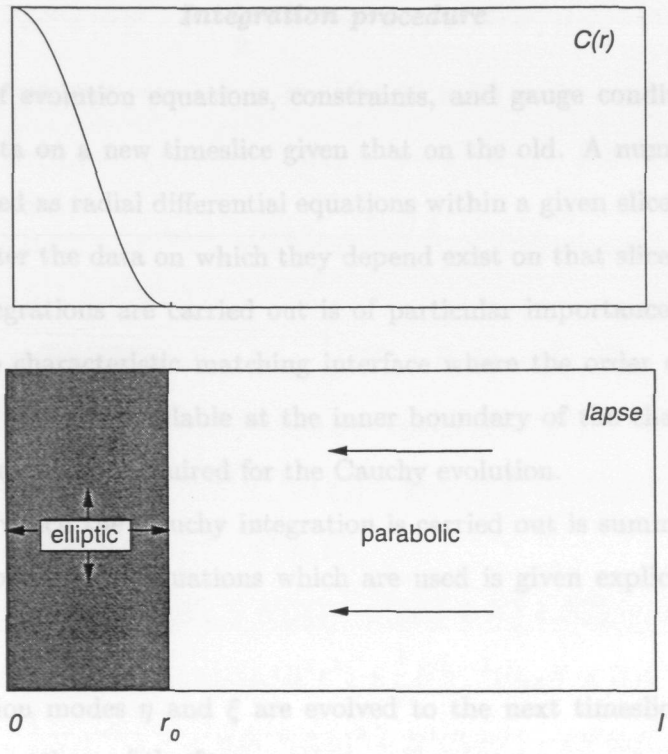


Figure 5.1: The function $C(r) = (1 - (r/r_0)^2)^{5/2}$ is used to implement a 'mixed' slicing condition, which is maximal at $r = 0$ and polar for $r > r_0$. The integration of the lapse over the Cauchy grid is carried out in two parts corresponding to an outer parabolic region and inner elliptic region where $C(r)$ is non-zero.

condition on the lapse (see below). At the origin (where maximal slicing has a perfectly well-defined lapse) the lapse is irregular for polar slicing.

In order to take advantage of the benefits of both types of slicing, Bardeen and Piran (1983) suggest that a *mixed slicing condition* be used. The trace of the extrinsic curvature is set as follows:

$$K = (1 - C(r))K_r{}^r, \tag{5.11}$$

where $C(r) = 1$ at the origin and decreases to zero before reaching the outer boundary. Significantly, the equation for the lapse is inward parabolic on outer region where the spacetime is polar sliced, and switches to an elliptic equation as $C(r)$ becomes non-zero (Bardeen and Piran (1983)). The lapse is thus solved inwards from the outer boundary, and its value at the edge of the elliptic region is used as a boundary condition for an elliptic solver acting over the inner maximally sliced region around the origin. The integration is represented schematically in Figure 5.1.

Integration procedure

A combination of evolution equations, constraints, and gauge conditions are used to determine the data on a new timeslice given that on the old. A number of the metric variables are solved as radial differential equations within a given slice, and as such can only be solved after the data on which they depend exist on that slice. Thus, the order in which the integrations are carried out is of particular importance and will also be significant at the characteristic matching interface where the order of operations will determine which data are available at the inner boundary of the characteristic region and at what stages data is required for the Cauchy evolution.

The order in which the Cauchy integration is carried out is summarised as follows. The particular form of the equations which are used is given explicitly in Stark and Piran (1987).

- The radiation modes η and ξ are evolved to the next timeslice using a pair of evolution equations of the form

$$\begin{aligned}\partial_t\eta = &-((1-x^2)(\partial_x G + 2NK_+) + 2xG)\eta - \partial_x G - 2NK_+ \\ &- r\beta^r\partial_r\eta + G(1-x^2)\partial_x\eta, \qquad (5.12)\\ \partial_t(\partial_r\xi) = &\,\partial_x((1-x^2)G) - \partial_r(r\beta^r))\xi_r - \partial_x(2NK_3Ar^{-1}B^{-1}) \\ &- \partial_r(2NK_\times B^{-2}) - r\beta^r\partial_r(\partial_r\xi) + G(1-x^2)\partial_x(\partial_r\xi). \qquad (5.13)\end{aligned}$$

  Note that rather than ξ itself, the evolution equation actually determines $\partial_r\xi$. In fact, the integration scheme can be carried out using only the latter, though analysis of the emitted radiation will require a calculation of ξ by a subsequent radial integration. The extrinsic curvature variables are also determined straightforwardly using the evolution equations,

$$\begin{aligned}\partial_t K_1 = &\,2N(1-x^2)(K_2^2+K_3^2) - 2AB(1-x^2)K_2\partial_x\beta^r - A^{-1}\partial_r(A^{-1}\partial_r N) \\ &- A^{-1}r^{-2}\partial_r((1-x^2)NB^2\partial_x A) - (1/2)NA^{-2}(B^{-4}(\partial_r\eta)^2 \\ &+ B^4(\partial_r\eta))(1-x^2)^2 + 2Nr^{-1}A^{-3}\partial_r A + NKK_1 \\ &- r\beta^r\partial_r K_1 + G(1-x^2)\partial_x K_1, \qquad (5.14a)\end{aligned}$$

$$\begin{aligned}\partial_t K_2 =&(-xG + NK + N((\frac{1}{2}C + 1)K_1 + (1-x^2)K_+))K_2 \\ &+ 2NK_3K_\times(1-x^2) + Br^{-1}\partial_x(A^{-1}\partial_r N) \\ &+ r(AB)^{-1}\partial_r G((\frac{1}{2}C+1)K_1 + (1-x^2)K_+) - BA^{-2}r^{-2}\partial_x(NA) \\ &+ \frac{1}{2}((1-x^2)\partial_x(A^{-1}N\partial_r\eta) - 4xNA^{-1}\partial_r\eta)B^{-1}r^{-1} - r\beta^r\partial_r K_2 \\ &+ G(1-x^2)\partial_x K_2, \end{aligned} \tag{5.14b}$$

$$\begin{aligned}\partial_t K_3 =&(-xG + NK + N(-K_1 - \frac{1}{2}CK_1 + K_+(1-x^2)))K_3 \\ &- 2NK_2K_\times(1-x^2) - r(AB)^{-1}\partial_r G K_\times(1-x^2) \\ &- ((1-x^2)\partial_x(NB^4A^{-1}(\partial_r\xi)) - 4xNB^4A^{-1}(\partial_r\xi))/(2Br) \\ &- r\beta^r\partial_r K_3 + G(1-x^2)\partial_x K_3, \end{aligned} \tag{5.14c}$$

$$\begin{aligned}\partial_t K_+ =&(-2xG + NK)K_+ - N(2(1-x^2)K_\times^2 + K_2^2 + K_3^2) - r(AB)^{-1}\partial_r G K_2 \\ &- \partial_r(r^2NA^{-1}\partial_r\eta)/(2AB^2r^2) + \frac{1}{2}B^2r^{-2}(\partial_{xx}N + NA^{-1}\partial_{xx}A) \\ &+ \frac{1}{2}NA^{-2}(1-x^2)(B^{-4}(\partial_r\eta)^2 + B^4(\partial_r\xi)^2) - r\beta^r\partial_r K_+ \\ &+ G(1-x^2)\partial_x K_+, \end{aligned} \tag{5.14d}$$

$$\begin{aligned}\partial_t K_\times =&(-2xG + NK + 2NK_+(1-x^2))K_\times \\ &- \frac{1}{2}A^{-1}(Br)^{-2}\partial_r(r^2NB^4A^{-1}(\partial_r\xi)) + r\partial_r G K_3/(AB) \\ &- r\beta^r\partial_r K_\times + G(1-x^2)\partial_x K_\times. \end{aligned} \tag{5.14e}$$

- The metric variable A is determined from the Hamiltonian constraint which for the radial gauge is an outward parabolic equation in r, and as such requires no outer boundary condition:

$$\partial_r A = \frac{1}{2}A^2r^{-1}\partial_x((1-x^2)B^2\partial_x A) + A(S_1 - 1) + A^{-1}(S_2 + 1) \tag{5.15}$$

where

$$S_1 = r^2((K_+^2 + K_\times^2)(1-x^2)^2 + (K_2^2 + K_3^2)(1-x^2) + \frac{3}{4}C^2K_1^2 + CKK_1) \tag{5.16a}$$

$$S_2 = \frac{1}{4}r^2(1-x^2)^2(B^{-4}(\partial_r\eta)^2 + B^4(\partial_r\xi)^2) \tag{5.16b}$$

The equation can be solved on the new slice using only the data which has been calculated to this point.

- The shift components β^r and G and the lapse, N, form a set of coupled equations on a slice which are integrated iteratively. The equations for G and N are inward parabolic in the polar sliced region, while β^r is determined algebraically once G and

N are known. At this point, initial data for the inward phase of the integration integration of N is required at the interface. The equations for G and β^r are determined from the shift conditions:

$$\partial_r G = \frac{1}{2} r^{-1} (AB)^2 \partial_{xx}((1-x^2)G) + C\partial_x(NK_1) - 4NK_2(AB)^{-1}, \tag{5.17a}$$

$$\beta^r = \frac{1}{2}\partial_x((1-x^2)G) + \frac{1}{2}NCK_1, \tag{5.17b}$$

while the lapse is given by

$$CA^{-1}\partial_r(A^{-1}\partial_r N) - 2A^{-2}r^{-1}\partial_r N + CA^{-1}r^{-2}\partial_x((1-x^2)AB^2\partial_x N) + (1-C)r^{-2}\partial_x((1-x^2)B^2\partial_x N) = S + \Lambda N, \tag{5.18}$$

where

$$S = r\partial_r C\beta^r K_1 + 2(1-C)AB(1-x^2)\partial_x\beta^r K_2, \tag{5.19a}$$

$$\Lambda = (1-C)(r^{-2}(1-A^{-2}) - \frac{1}{2}r^{-2}\partial_{xx}((1-x^2)^2\eta) + \frac{1}{4}A^{-2}(B^{-4}(\partial_r\eta)^2 + B^4(\partial_r\xi)^2)(1-x^2)^2) + (1+C)(K_+^2 + K_\times^2)(1-x^2)^2 + (3C-1)(K_2^2 + K_3^2)(1-x^2) + C(1-\frac{1}{4}C)(1+C)K_1^2, \tag{5.19b}$$

and the function C controls the slicing condition, as described in Section 5.1. We follow Stark and Piran and choose it to be of the form

$$C = (1 - (r/r_0)^2)^n \tag{5.20}$$

with r_0 a constant.

- The shift component N^ϕ is not required for the Stark-Piran integration procedure, however it can be found by performing a radial integration of the shift equation

$$\partial_r N^\phi = -2NK_3Ar^{-1}B^{-1} - \xi(1-x^2)\partial_r G. \tag{5.21}$$

Note that this procedure differs somewhat from that presented in Stark and Piran (1987) for the same system of equations. An examination of the equations determining the lapse N and two shift components β^r and G shows they are in fact coupled on a given $t = \textit{constant}$ slice. As a result, the advantages of using an inward parabolic equation for N in the polar-sliced region is offset by the fact that an iterative procedure with an initial guess for β^r will be required even in the parabolic region. In fact, this coupling between β^r and N is extended even to the determination of N at the interface as will be seen in Section 5.4.

Once the integrations are completed and the data is determined on a slice, a boundary condition is used to determine the radial derivatives of the metric variables at the outer edge of the grid.

Boundary conditions

Boundary conditions for the axially symmetric 3+1 system are discussed in detail in Bardeen and Piran (1983). There are three boundaries to be considered: the origin, the polar axis, and the outer radial boundary.

Regularity conditions are used to determine appropriate expansions for the metric variables in the neighbourhood of the origin and the polar axis. In fact, on the axis, the condition that the appropriate quantities vanish can be enforced automatically by choosing as variables functions weighted with appropriate exponents of $\sin\theta$.

For the standard Cauchy evolutions carried out by Stark and Piran, an outgoing wave condition was used at the outer boundary. For polar-sliced hypersurfaces, Bardeen and Piran (1983) show that as $r \to \infty$,

$$NA = 1 + O(r^{-2}), \tag{5.22}$$

which is the analogue of the Robin boundary condition. This condition is accurate to order r^{-2} (compared to the r^{-1} accuracy for the corresponding Dirichlet condition) and has the additional benefit of being simple to apply once A has been determined.

Of the other metric variables, A and G do not require an outer boundary condition, as they are integrated radially outwards. Outer boundary conditions for the wave modes (η, ξ) and their corresponding momenta $(K_+, K_\times)$ are set to be that of purely outgoing spherical radial waves for the Cauchy code on its own. Similarly, the remaining extrinsic curvature variables (K_1, K_2, K_3) vanish to order r^{-3} (Stark and Piran (1987)) and so are set to zero.

Note that these approximations are not required when the Cauchy code is interfaced with a characteristic code. Rather, waveforms passing outwards at the boundary will be transferred to the characteristic code for a full integration to $\mathscr{I}^+$, whereas quantities requiring outer boundary values (notably N) are provided these by the corresponding characteristic data passed inwards. Details of how this is accomplished are given in Section 5.3.

5.2 A Bondi-Sachs characteristic scheme

The implementation of the characteristic scheme in the Southampton code is somewhat simpler to describe due to the form of the Bondi equations. As described in the previous chapter, in the characteristic coordinate system the Bondi variables form a hierarchical system. On each slice, the data γ and δ as well as appropriate data at the interface are enough to determine the metric variables β, U, W, and V on a single $u = constant$ slice, as well as the u-derivatives of γ and δ which can be used to evolve the initial data forward.

The coordinates used in the characteristic region are those of Section 4.2 with one notable difference. The radial coordinate r is replaced by

$$y = 1/r, \tag{5.23}$$

so that null infinity is mapped to the surface $y = 0$ and the coordinate is well defined over the numerical grid provided that the inner boundary is located at some finite non-zero radius. For the convenience of matching with the Cauchy region, the function

$$x = \cos\theta \tag{5.24}$$

replaces the θ coordinate in the equations that follow.

With these coordinate choices, the Bondi field equations take the following explicit form:

$$\partial_y \beta = \frac{1}{2}((\partial_y \gamma)^2 \cosh^2 2\delta + (\partial_y \delta)^2) y, \tag{5.25a}$$

$$\begin{aligned} \partial_y P =& (-4\cosh^2 2\delta (\partial_y \gamma) x/\sqrt{1-x^2} + (2\cosh^2 2\delta (\partial_{xy}\gamma) \\ & - 4\cosh^2 2\delta (\partial_x \gamma)(\partial_y \gamma) - 4(\partial_x \beta)/y - 2(\partial_{xy}\beta) \\ & + 8\sinh 2\delta \cosh 2\delta (\partial_x \delta)(\partial_y \gamma) - 4(\partial_y \delta)(\partial_x \delta))\sqrt{1-x^2})/y^2, \end{aligned} \tag{5.25b}$$

$$\begin{aligned} \partial_y Q =& (-4((\partial_y \delta) + (\partial_y \gamma)\sinh 2\delta \cosh 2\delta) x/\sqrt{1-x^2} + (-4(\partial_y \gamma)(\partial_x \delta) \\ & + 2(\partial_{xy}\delta) - 4(\partial_x \gamma)(\partial_y \delta) - 4(\partial_x \gamma)(\partial_y \gamma)\sinh 2\delta \cosh 2\delta \\ & + 8(\partial_x \delta)(\partial_y \gamma)\cosh^2 2\delta + 2(\partial_{xy}\gamma)\sinh 2\delta \cosh 2\delta)\sqrt{1-x^2})e^{2\gamma}/y^2, \end{aligned} \tag{5.25c}$$

$$\begin{aligned}\partial_y V =&(e^{2\beta-2\gamma}(\cosh 2\delta((1-x^2)(+2((\partial_x\delta)^2+(\partial_x\gamma)^2-(\partial_x\gamma)(\partial_x\beta))+(\partial_x\beta)^2\\&+(\partial_{xx}\beta)-(\partial_{xx}\gamma))+x((\partial_x\gamma)-2(\partial_x\beta)+3(\partial_x\gamma))-1)\\&+\sinh 2\delta((1-x^2)((\partial_{xx}\delta)+(\partial_x\delta)(2(\partial_x\beta)-4(\partial_x\gamma)))-4x(\partial_x\delta)))\\&+\cosh 2\delta e^{-2\beta}(e^{-2\gamma}(\partial_y W)^2+e^{\gamma}(\partial_y U)^2)/4-(\partial_{xy}U)\sqrt{1-x^2}/2\\&+(\partial_y U)x/2\sqrt{1-x^2}+e^{-\beta}\sinh 2\delta(\partial_y U)(\partial_y W)/2)/y^2\\&+2((\partial_x U)\sqrt{1-x^2}-Ux/\sqrt{1-x^2})/y^3,\end{aligned}\tag{5.25d}$$

$$\begin{aligned}\partial_y\phi_1-f\phi_2=&-(2\sinh 2\delta V(\partial_y\delta)(\partial_y\gamma)+\cosh 2\delta(V(\partial_y\gamma)y+(\partial_y V)(\partial_y\gamma)))y^2/2\\&-\cosh 2\delta V(\partial_y\gamma)y/2+(e^{-2\beta}(e^{-2\gamma}(\partial_y W)^2-e^{2\gamma}(\partial_y U)^2)/2-e^{2\beta-2\gamma}((\partial_{xx}\beta)\\&+(\partial_x\beta)^2)(1-x^2)/2+(\sinh 2\delta((\partial_x U)(\partial_y\delta)\\&+2U((\partial_x\gamma)(\partial_y\delta)+(\partial_x\delta)(\partial_y\gamma))+e^{-2\gamma}(\partial_{xy}W)/4)+\cosh 2\delta((\partial_{xy}U)/4\\&+e^{-2\gamma}(\partial_x W)(\partial_y\delta)+(\partial_y U)(\partial_x\gamma)/2\\&+U(\partial_{xy}\gamma)+(\partial_x U)(\partial_y\gamma)))\sqrt{1-x^2}/2+(\sinh 2\delta(U(\partial_y\delta)+e^{-2\gamma}(\partial_y W)/4)\\&+\cosh 2\delta(e^{-2\gamma}W(\partial_y\delta)+(\partial_y U)/4-(\partial_y\gamma))/2)x/\sqrt{1-x^2})/y\\&+(-(e^{-2\gamma}\sinh 2\delta W+\cosh 2\delta U)x/\sqrt{1-x^2}/2-(\cosh 2\delta((\partial_x U)/2\\&+U(\partial_x\gamma))+e^{-2\gamma}\sinh 2\delta(\partial_x W)/2)\sqrt{1-x^2})/y^2,\end{aligned}\tag{5.25e}$$

$$\begin{aligned}\partial_y\phi_2+f\phi_1=&\,y^2(\cosh 2\delta\sinh 2\delta V(\partial_y\gamma)^2-((\partial_y\delta)(\partial_y V)+V(\partial_y\delta)y))/2-y(\partial_y\delta)V/2\\&+(e^{2\beta-2\gamma}\sinh 2\delta((\partial_{xx}\beta)+(\partial_x\beta)^2)(1-x^2)/2+((\partial_x U)(\partial_y\delta)/2\\&+e^{-2\gamma}(\partial_{xy}W)/4+(\partial_y U)(\partial_x\delta)/2-e^{-2\gamma}\cosh^2 2\delta(\partial_x W)(\partial_y\gamma)+U(\partial_{xy}\delta)\\&-\cosh 2\delta\sinh 2\delta(\partial_y\gamma)((\partial_x U)+2U(\partial_x\gamma)))\sqrt{1-x^2}\\&-e^{-2\beta}\sinh 2\delta(e^{2\gamma}(\partial_y U)^2+e^{-2\gamma}(\partial_y W)^2)/8\\&-e^{-2\beta}\cosh 2\delta(\partial_y U)(\partial_y W)/4+(e^{-2\gamma}(\partial_y W)/4-U(\partial_y\delta)/2\\&-e^{-2\gamma}\cosh^2 2\delta W(\partial_y\gamma)-\cosh 2\delta\sinh 2\delta U(\partial_y\gamma))x/\sqrt{1-x^2})/y\\&-(e^{-2\gamma}((\partial_x W)\sqrt{1-x^2}/2+Wx/\sqrt{1-x^2})\\&+U(\partial_x\delta)\sqrt{1-x^2})/y^2,\end{aligned}\tag{5.25f}$$

where the functions P and Q are those defined by Equations (4.34), namely

$$P=(-e^{2\beta+2\gamma}(\partial_y U)\cosh 2\delta-e^{\beta}(\partial_y W)\sinh 2\delta)/y^2,\tag{5.26a}$$

$$Q=(-e^{2\beta}(\partial_y U)\sinh 2\delta-e^{2\beta+2\gamma}(\partial_y W)\cosh 2\delta)/y^2\tag{5.26b}$$

and the auxiliary variables ϕ_1, ϕ_2, and f, are defined in terms of the u derivatives of γ and δ by

$$\phi_1 = \cosh 2\delta(\partial_u \gamma)/y, \tag{5.27}$$

$$\phi_2 = (\partial_u \delta)/y, \tag{5.28}$$

$$f = -2(\partial_y \gamma)\sinh 2\delta. \tag{5.29}$$

As noted in the previous chapter, the system of equations in the order

(5.25a), (5.25b), (5.25c), (5.26a), (5.26b), (5.25d), (5.25e), (5.25f),

is hierarchical in that all of the data required by the right-hand side of the later equations will be available if the earlier equations have been solved.

Initial data for each of the y integrations is supplied at the Cauchy interface, with one notable exception. We wish to match points on the characteristic grid with the Cauchy grid at the interface between the two. To accomplish this, the (y, θ, ϕ) coordinates on the characteristic region are fixed at the interface rather than $\mathscr{I}^+$, and as such will generally not result in Bondi-type asymptotic behaviour of the metric functions. A coordinate transformation to Bondi coordinates can, however, be carried out provided that the metric variable β satisfies the condition given by Equation (4.51) at $\mathscr{I}^+$. Enforcing this condition amounts to setting 'initial' data for the y-integration of β along a $u = \textit{constant}$ slice via Equation (5.25a). That is, in order to ensure that a Bondi-type coordinate system exists, we will choose to set β at $\mathscr{I}^+$ and integrate it *inwards* towards the interface with the Cauchy region.

To understand the influence of this on the Cauchy data at the interface, recall that it was the slicing condition in the Cauchy region which determined the form of boundary condition which would normally be used instead of the characteristic interface. In particular, the fact that the spacetime was polar sliced in the neighbourhood of the interface resulted in an approximate condition on the lapse, $(NA) \simeq 1$, as suggested by Equation (5.22). In Section 4.2, asymptotic expansions of the characteristic metric functions was related to the outgoing wave condition originally posed by Bondi et al. (1962). So the imposition of the condition on β at $\mathscr{I}^+$ can be seen as an extension of placing an outer boundary condition on the lapse in the Cauchy region. We will see in Section 5.3 that the Cauchy lapse will not be uniquely determined without first determining the characteristic β and imposing the polar slicing condition at the interface.

The evolution algorithm

The hierarchical structure of the Bondi equations leads to a very simple evolution system on a null slice. Taking into account the comments made in the previous section, the following scheme has been implemented to evolve the data on the characteristic grid:

1. The metric functions δ and γ are specified on an initial $u = constant$ retarded time slice. Values for δ_y and γ_y are determined from these for later use.

2. β is determined by the Bondi slicing requirement, Equation (4.51) at null infinity and integrated *inwards* over the grid to the Cauchy boundary.

3. The auxiliary variables P and Q are determined on the slice by *outward* integration of Equations (5.25b) and (5.25c) from the interface. Initial values for these integrations can be given in the form of values for $(\partial_y U)_I$ and $(\partial_y W)_I$ so that P and Q are determined on I via Equations (5.26a) and (5.26b).

4. U and W are determined on the slice by an *outward* integration of the inversions of Equations (5.26a) and (5.26b) to form equations for $\partial_y U$ and $\partial_y W$. Initial data for U and W are specified at the interface.

5. V is determined on the slice via an *outward* integration of Equation (5.25d) with initial data on the interface.

6. The coupled system given by Equations (5.25e) and (5.25f) are solved for ϕ_1 and ϕ_2. These are used to determine $\partial_u \gamma$ and $\partial_u \delta$ on the slice via Equations (5.27) and (5.28). Initial data is specified at the interface in the form of values for $(\partial_u \gamma)|_I$ and $(\partial_u \delta)|_I$.

7. The values of $\partial_u \gamma$ and $\partial_u \delta$ are used to integrate γ and δ to some future $u + \Delta u$. The procedure is then repeated from the beginning.

Finally, we note that the integration procedure as specified makes no use of the supplementary conditions described in Section 4.2. Enforcing these equations can be shown to be equivalent to specifying values for $\{P, Q, U, W, V\}$ on a $y = constant$ world-tube (see, for example Sachs (1962), d'Inverno and Vickers (1997)). Thus by making use of them, the amount of information which is required to be passed across the interface can be reduced. In fact, however, we take the position that in order to ensure a better correspondence between the numerical Cauchy and characteristic data, it is preferable to extract this information from the Cauchy interface. The supplementary conditions is

then used as a check on the data in much the same way that the constraints are used in Cauchy evolutions.

Behaviour at the boundaries

The characteristic grid has four boundaries at which the behaviour of the metric functions must be controlled. The innermost of these is the most significant for the purpose of Cauchy-characteristic matching, as it is located at the interface with the Cauchy region. The data required for the radial outward integration of the Bondi metric variables are placed here based on their values on the corresponding points on the Cauchy grid. The procedure for carrying this out is the subject of the next section.

The outer boundary for the characteristic grid is located at $\mathscr{I}^+$, corresponding to the surface $y = 0$ of the compactified radial coordinate. Since this is an ingoing null surface, data on $\mathscr{I}^+$ is not causally connected with data on the grid, and thus should not influence the numerical calculations. This is reflected by the fact that the integrations for the metric functions are, for the most part, performed in an outward direction. The finite difference equations use only data from grid points interior to the point being calculated, and so data at $\mathscr{I}^+$ need not be specified.

The exception to this is the calculation of β which, as mentioned in the previous section, is required to be of a form which allows transformations of the asymptotic metric functions to Bondi-type quantities defined at $\mathscr{I}^+$. This is accomplished by calculating β at $\mathscr{I}^+$ via Equation (4.51) and integrating inwards. Although this might be considered to violate causality, in fact the condition on β corresponds to a gauge choice on the slicing of the spacetime, and as such does not involve the propagation of a physical field. Further, the determination of β via Equation (5.25a) is in terms of the functions γ and δ which have been determined in a manner consistent with the causality. Thus it is not expected that the non-standard integration of β will have adverse affects with regards to the integration scheme.

The behaviour of the metric functions at the equator are determined by the assumed reflection symmetry of the metric in the Cauchy region, which necessarily carries over to the characteristic metric functions. In particular, an examination of individual Bondi metric components given by Equation (4.28) under the requirement of reflection symmetry across the equator suggests that γ, β, U and V are even across the equator,

whereas δ and W are odd.

The poles are traditionally the source of difficulties for numerical codes in spherical coordinates, for they represent a coordinate singularity in the metric components where $\sin\theta \to 0 (x \to 1)$. In order to ensure that no coordinate 'kinks' exist on the axis, it must be possible to choose a Minkowskian tangent metric there. The condition that the radius of a circle approach 2π results in the following behaviour of the metric functions:

$$\gamma \to (1-x^2)\hat{\gamma}(x), \qquad \delta \to (1-x^2)\hat{\delta}(x), \tag{5.30a}$$

$$U \to \sqrt{1-x^2}\hat{U}(x), \qquad W \to \sqrt{1-x^2}\hat{W}(x), \tag{5.30b}$$

$$\beta \to \hat{\beta}(x), \qquad V \to \hat{V}(x), \tag{5.30c}$$

where hatted quantities are regular functions of x as $x \to 1$. Thus the values of γ, δ, U, W as well as the derivatives $\partial_x\gamma$ and $\partial_x\delta$ go to zero as the axis is approached.

5.3 Data transfer across the interface

The CCM method requires that information be carried accurately across the interface between the interior Cauchy region and exterior characteristic region. The particular choice of integration scheme demands that the information be carried in both directions. Thus, data on the characteristic region provide an outer boundary condition for the Cauchy code, which the Cauchy data specifies interior values for outward integration of the Bondi metric quantities in the characteristic region.

A variety of problems must be overcome at the interface. Most significantly, although the interface is viewed as a world-tube on a single two-dimensional surface, not all of the data which are passed across the interface are local to the surface. In particular, for an interface at a constant radius, radial derivatives of the data are required to be passed. Numerical determination of these derivatives require that the data be known on both sides of the interface in the given coordinate system.

In the past, this difficulty has been overcome by placing data past the outer edge of a grid by interpolating data from the other grid (see Figure 5.2). This has been applied effectively by Dubal et al. (1995) in cylindrical symmetry. In order to maintain second order accuracy in the derivatives, a fourth order interpolation scheme was used. Note, however, that the ability of this code to perform the necessary interpolations to the desired accuracy benefitted from the high degree of symmetry of the problem under study, and in more general spacetimes the interpolation of points from one grid to the

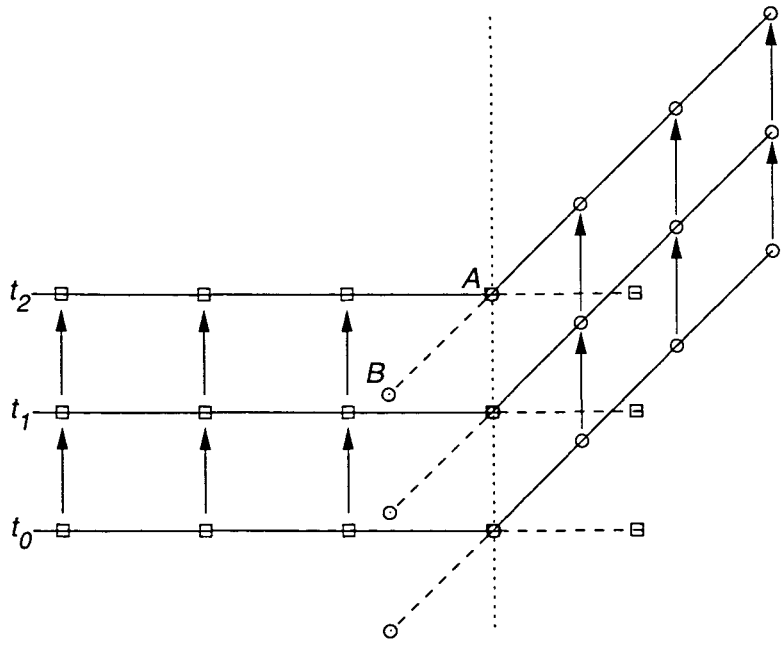


Figure 5.2: One method of determining the radial derivative of a characteristic function at a point A on the interface, is to place data at a ghost-point B via interpolation from the known Cauchy data. The complementary procedure can be used to determine radial derivatives on the Cauchy side of the interface.

other is a far more sensitive problem. Bishop et al. (1996) have reported success in the extraction of waves to a quasi-spherical characteristic region using such a 'thick' interface, and the technique has been suggested as a means of matching the stable Pittsburgh characteristic code with a Cauchy module based on the successors to the Grand Challenge effort (Winicour (1998)).

A somewhat different approach is taken by the Southampton axisymmetric CCM code.

- The equations for transforming both the metric functions *and* their derivatives between coordinates are applied at the interface. This means that ghost grid points are not needed in order to calculate derivatives there. Thus the matching is performed at a single $r = constant$ ($y = constant$) surface with no need for interpolations from points on the interior of either grid.

- A further simplification at the interface is introduced by matching the angular θ and ϕ coordinate in the characteristic region with that defined in the Cauchy region.

As a result of these considerations, data specified at the outermost grid points of the

Cauchy region are transformed directly on to grid points of the characteristic region with no need for interpolation, thus avoiding a large source of inaccuracy as well as computational complexity involved in the extraction/injection procedure.

For the purposes of the section, the coordinates on the Cauchy grid are labelled $x^a = (t, r, x, \phi)$ while those in the characteristic region are distinguished with a tilde, $\tilde{x}^a = (\tilde{u}, \tilde{y}, \tilde{x}, \tilde{\phi})$. Recall that following Stark and Piran (1987), the coordinate $x = \cos\theta$ is used as the angular coordinate, while axisymmetry implies that fields are independent of ϕ. The coordinate transformation between regions takes place at the interface, I, which is fixed to the constant coordinate location $r = 1$ in the Cauchy region, and $\tilde{y} = 1$ in the characteristic region. On I we also require that both the timelike coordinates and angular coordinates match, so that

$$t = \tilde{u}, \qquad x = \tilde{x}, \qquad \text{and} \qquad \phi = \tilde{\phi}. \tag{5.31}$$

The transformation of evolution data from one system to the other will require the transformation of both the metric components and their derivatives at some interface surface. The transformations from Cauchy g_{ab} to characteristic $\tilde{g}_{ab}$ can be found, as usual, by computing

$$\tilde{g}_{ab} = g_{cd}\frac{\partial x^c}{\partial \tilde{x}^a}\frac{\partial x^d}{\partial \tilde{x}^b}, \tag{5.32}$$

$$\partial_c \tilde{g}_{ab} = \partial_f g_{de}\frac{\partial x^d}{\partial \tilde{x}^a}\frac{\partial x^e}{\partial \tilde{x}^b}\frac{\partial x^f}{\partial \tilde{x}^c} + \left(\frac{\partial^2 x^d}{\partial \tilde{x}^a \partial \tilde{x}^c}\frac{\partial x^e}{\partial \tilde{x}^b} + \frac{\partial x^d}{\partial \tilde{x}^a}\frac{\partial^2 x^e}{\partial \tilde{x}^b \partial \tilde{x}^c}\right) g_{de}, \tag{5.33}$$

and corresponding transformations in the other direction found by switching $\sim$ with non-$\sim$ quantities. The detailed calculations involved in determining these quantities is outlined in d'Inverno and Vickers (1997), and only the relevant results are presented here except for the case of the transformation of $\partial_1 g_{01}$ and $\partial_1 g_{11}$, where a correction to the published derivation and result is required.

For the transformation from the Cauchy to characteristic coordinates, it is found that

$$\left(\frac{\partial x^a}{\partial \tilde{x}^b}\right)_I = \left(\frac{\partial x^a}{\partial \tilde{x}^0}, \frac{\partial x^a}{\partial \tilde{x}^1}, \frac{\partial x^a}{\partial \tilde{x}^A}\right)_I = (\delta^a{}_0, Y^a, \delta^A{}_0), \tag{5.34}$$

where Y^a is defined in terms of the Stark-Piran variables by

$$X^a = (1, \nu, G\sqrt{1-x^2}, N^\phi), \tag{5.35}$$

$$\nu = \beta^r + N/A. \tag{5.36}$$

The result of the application of this transformation (and its derivatives) via Equation (5.33) are the following expressions for the components of the characteristic metric components in terms of the Cauchy data at the interface:

$$\tilde{g}_{00} = g_{00}, \tag{5.37a}$$

$$\tilde{g}_{01} = \rho(g_{00} + \nu g_{01} - \gamma^{AB} g_{0A} g_{0B}), \tag{5.37b}$$

$$\tilde{g}_{0A} = g_{0A}, \tag{5.37c}$$

$$\tilde{g}_{AB} = g_{AB}, \tag{5.37d}$$

$$\partial_0 \tilde{g}_{00} = \partial_t g_{00}, \tag{5.37e}$$

$$\partial_0 \tilde{g}_{01} = \partial_t (\rho X^a g_{0a}), \tag{5.37f}$$

$$\partial_0 \tilde{g}_{0A} = \partial_t g_{0A}, \tag{5.37g}$$

$$\partial_0 \tilde{g}_{AB} = \partial_t g_{AB}, \tag{5.37h}$$

$$\partial_1 \tilde{g}_{00} = 2\partial_t (\rho X^a) g_{0a} + \rho X^a \partial_a g_{00}, \tag{5.37i}$$

$$\partial_1 \tilde{g}_{01} = \epsilon X^a g_{0a}, \tag{5.37j}$$

$$\partial_1 \tilde{g}_{0A} = \partial_0 (\rho X^a) g_{aA} + \partial_A (\rho X^a) g_{0a} + \rho X^a \partial_A g_{0a}, \tag{5.37k}$$

$$\partial_1 \tilde{g}_{AB} = \rho(\partial_A X^a g_{aB} + \partial_B X^a g_{aA} + X^a \partial_a g_A B), \tag{5.37l}$$

$$\partial_C \tilde{g}_{00} = \partial_C g_{00}, \tag{5.37m}$$

$$\partial_C \tilde{g}_{01} = \partial_C (\rho X^a g_{0a}), \tag{5.37n}$$

$$\partial_C \tilde{g}_{0A} = \partial_C g_{0A}, \tag{5.37o}$$

$$\partial_C \tilde{g}_{AB} = \partial_C g_{AB}, \tag{5.37p}$$

where

$$\rho = (\nu + \frac{1}{2}(xG - \sqrt{1 - x^2}\partial_x G))^{-1} \tag{5.38}$$

$$\epsilon = f(\xi, \eta, N, \beta^r, G, N^\phi, K_{ab}), \tag{5.39}$$

and γ^{AB} is the matrix inverse of g_{AB}, ie. $\gamma^{AB} g_{BC} = \delta^A{}_C$.

The inverse transformation from characteristic to Cauchy coordinates is given by

$$\left(\frac{\partial \tilde{x}^a}{\partial x^b}\right)_I = \left(\frac{\partial \tilde{x}^a}{\partial x^0}, \frac{\partial \tilde{x}^a}{\partial x^1}, \frac{\partial \tilde{x}^a}{\partial x^A}\right)_I = (\delta^a{}_0, \tilde{Y}^a, \delta^A{}_0). \tag{5.40}$$

where $\tilde{Y}^a$ is given by

$$\tilde{Y}^a = \left(-\tilde{\nu}^{-1}, (\tilde{\nu}\tilde{\rho})^{-1}, \tilde{\nu}^{-1}\tilde{g}^{AB}\tilde{g}_{0A}\right), \tag{5.41}$$

$$\tilde{\rho} = \frac{1}{\tilde{\nu} - \tilde{p}}, \tag{5.42}$$

$$\tilde{\nu} = \frac{1}{\tilde{g}_{01}}(\tilde{g}_{00}2\tilde{p}\tilde{g}_{01} - g^{AB}g_{0A}g_{0B}), \tag{5.43}$$

$$\tilde{p} = -\frac{1}{2}\partial_x(g^{2A}g_{0A}\sqrt{1-x^2}). \tag{5.44}$$

It is significant to note that the expression for $\tilde{p}$ can only be fixed once the slicing condition in the Cauchy region is known. In the case under consideration, polar slicing $\mathrm{tr}K = K^r{}_r$ is used in the region of the interface. It is shown in d'Inverno (1995) that in axial symmetry this condition can be re-expressed as a condition on the metric variables. In the present context, this can be written

$$g_{01} = -rpg_{11}, \tag{5.45}$$

with $p = \tilde{p}$ given by Equation (5.44).

In terms of the given transformation and its derivatives, the Cauchy metric components in terms of the characteristic components on the interface are given by

$$g_{00} = \tilde{g}_{00}, \tag{5.46a}$$

$$g_{01} = \frac{1}{\tilde{v}}\left(-\tilde{g}_{00} + \frac{\tilde{g}_{01}}{\tilde{\rho}} + \tilde{g}^{AB}\tilde{g}_{0A}\tilde{g}_{0B}\right), \tag{5.46b}$$

$$g_{0A} = \tilde{g}_{0A}, \tag{5.46c}$$

$$g_{11} = \frac{1}{\tilde{\nu}^2}\left(\tilde{g}_{00} - 2\frac{\tilde{g}_{01}}{\tilde{\rho}} - \tilde{g}^{AB}\tilde{g}_{0A}\tilde{g}_{0B}\right), \tag{5.46d}$$

$$g_{AB} = \tilde{g}_{AB} \tag{5.46e}$$

$$\partial_0 g_{00} = \partial_0 \tilde{g}_{00}, \tag{5.46f}$$

$$\partial_0 g_{01} = \partial_0(\tilde{Y}^a \tilde{g}_{0a}), \tag{5.46g}$$

$$\partial_0 g_{0A} = \partial_0 \tilde{g}_{0A}, \tag{5.46h}$$

$$\partial_0 g_{11} = \partial_0(\tilde{g}_{ab}\tilde{Y}^a\tilde{Y}^b), \tag{5.46i}$$

$$\partial_0 g_{00} = \partial_0 \tilde{g}_{AB} \tag{5.46j}$$

$$\partial_1 g_{00} = 2\partial_0 \tilde{Y}^a \tilde{g}_{0a} + \tilde{Y}^a \partial_a \tilde{g}_{00}, \tag{5.46k}$$

$$\partial_1 g_{01} = \tilde{\kappa}, \tag{5.46l}$$

$$\partial_1 g_{0A} = \partial_1 \tilde{g}_{0A}, \tag{5.46m}$$

$$\partial_1 g_{11} = \tilde{\eta}, \tag{5.46n}$$

$$\partial_1 g_{AB} = g_{a(A}\partial_{B)}\tilde{Y}^a + \tilde{Y}^a \partial_a \tilde{g}_{AB} \tag{5.46o}$$

$$\partial_C g_{00} = \partial_C \tilde{g}_{00}, \tag{5.46p}$$

$$\partial_C g_{01} = \partial_C (\tilde{g}_{0a}\tilde{Y}^a), \tag{5.46q}$$

$$\partial_C g_{0A} = \partial_C \tilde{g}_{0A}, \tag{5.46r}$$

$$\partial_C g_{11} = \partial_C (\tilde{g}_{ab}\tilde{Y}^a\tilde{Y}^b), \tag{5.46s}$$

$$\partial_C g_{AB} = \partial_C \tilde{g}_{AB} \tag{5.46t}$$

where

$$\tilde{\kappa} = (\tilde{p} + \tilde{Y}^a \partial_a \tilde{p})\tilde{g}_{ab}\tilde{Y}^a\tilde{Y}^b - \tilde{\eta} \tag{5.47}$$

and $\tilde{\eta}$ is determined by Equation (5.65), derived below.

As a final note, we emphasise that the calculations in this section are dependent on the chosen gauge in both the Cauchy and characteristic regions. In particular, a derivation of the parameter $\tilde{\nu}$ which determines the transformations to g_{01} and g_{11} in the Cauchy regions, shows that the value of ν is only uniquely fixed once a slicing condition on the Cauchy region is selected. Thus, for Cauchy systems in which other slicings are used, alternate forms of these transformations are required. This can be a difficult point, as it is not necessarily the case that a particular lapse equation will lead to an explicit condition on the metric components which can be used to fix the coordinate transformation. It is often the case that slicing conditions result in elliptic equations for the lapse (eg. maximal slicing) and the shift (eg. minimal strain shift, Smarr and York (1978), Brady et al. (1998)). These equations can only be solved once appropriate boundary values have been specified. Boundary values, however, can only be determined once the gauge variables themselves have been extracted to determine data on the characteristic grid. It is significant in the system presented here the values of U, W, and V on the characteristic region are determined at the interface from the metric components $\tilde{g}_{01}$ and $\tilde{g}_{0A}$. According to Equations (5.37a), these components can only be determined once the shift and lapse are known on the Cauchy side of the interface.

However a given choice of shift and lapse may depend for outer boundary conditions on the values of the characteristic U, W, and V.

In fact, for the evolution system in question, the issue can be sidestepped via a method presented in Section 5.3 which is consistent with the overall evolution scheme in the Cauchy region. The extent to which such methods can be found for generic slicings, however, is not clear.

The injection of $\partial_1 g_{11}$

In d'Inverno and Vickers (1997), a confusion between variables on the characteristic and Cauchy sides in Equations (154) to (156) of that paper, results in an erroneous calculation for the derivative of $\partial_1 g_{11}$ in Equation (160), as was pointed out by Chris Clarke (personal communication). The problem can be corrected via the following calculation.

To begin, recall that the Cauchy coordinates are written in the radial gauge and as such satisfy the determinant condition (5.6), namely

$$\det \gamma_{AB} = r^4 \sin^2\theta. \tag{5.48}$$

Differentiating this with respect to r gives

$$\gamma^{AB}\partial_r \gamma_{AB} r^4 \sin^2\theta = 4r^3 \sin^2\theta \qquad \Longrightarrow \qquad \gamma^{AB}\partial_r \gamma_{AB} = \frac{4}{r}, \tag{5.49}$$

and similarly, differentiating with respect to t and θ give

$$\gamma^{AB}\partial_t \gamma_{AB} = 0 \tag{5.50}$$

$$\gamma^{AB}\partial_\theta \gamma_{AB} = 2\cot\theta. \tag{5.51}$$

The Christoffel symbols of the first kind are defined by

$$\Gamma_{cab} = \frac{1}{2}(\partial_{(a}g_{b)c} - \partial_c g_{ab}) \tag{5.52}$$

Then, in terms of the above results, we have

$$\gamma^{AB}\Gamma_{1AB} = \frac{1}{2}(\gamma^{AB}(\partial_{(A}g_{B)1} - \partial_1 g_{AB})) = -\frac{1}{2}\gamma^{AB}\partial_1 \gamma_{AB} = -\frac{2}{r} \tag{5.53a}$$

$$\gamma^{AB}\Gamma_{0AB} = \gamma^{AB}\partial_B \beta_A \tag{5.53b}$$

$$\gamma^{AB}\Gamma_{CAB} = \gamma^{AB}\partial_B \gamma_{CA} - \delta^2{}_C \frac{x}{1-x^2} \tag{5.53c}$$

where we've used the fact that $g_{a1} = 0$ and $g_{0a} = \beta_a$ for the Cauchy metric. Taking the derivative of Equation (5.53a) with respect to r gives

$$\partial_1 \gamma^{AB} \Gamma_{1AB} + \gamma^{AB} \nabla_1 \Gamma_{1AB} = \frac{2}{r^2}, \tag{5.54}$$

which upon expanding the covariant derivative in the second term becomes

$$\partial_1 \gamma^{AB} \Gamma_{1AB} + \gamma^{AB} \gamma^{cd} \Gamma_c AB \Gamma_{d11} + \gamma^{AB} \partial_1 \Gamma_{1AB} = \frac{2}{r^2}. \tag{5.55}$$

The Christoffel symbols can now be expanded in terms of the metric variables. The first term becomes

$$T_1 = \partial_1 \gamma^{AB} \Gamma_{1AB} = \frac{1}{2} \gamma^{AC} \gamma^{BD} \partial_1 \gamma_{AB} \partial_1 \gamma_{CD}. \tag{5.56}$$

The second term on the left can more easily be expressed by first defining the following contractions of the Christoffel symbols:

$$Q_0 = \gamma^{AB} \Gamma_{0AB} = \gamma^{AB} \partial_A \beta_B, \tag{5.57}$$

$$Q_1 = \gamma^{AB} \Gamma_{1AB} = -\frac{1}{2} \gamma^{AB} \partial_1 \gamma_{AB} = -\frac{2}{r}, \tag{5.58}$$

$$Q_C = \gamma^{AB} \Gamma_{CAB} = \gamma^{AB} \partial_C \gamma_{AB} - \delta^2{}_C \cot\theta. \tag{5.59}$$

In terms of these variables, the second term becomes

$$T_2 = \gamma^{cd} \Gamma_{c11} Q_d = \frac{1}{2}(-\partial_0 \gamma_{11} + 2\partial_1 \beta_1) g^{0a} Q_a + \frac{1}{2} \partial_1 g^{1a} Q_a - \frac{1}{2} \partial_B \gamma_{11} g^{Ba} Q_a. \tag{5.60}$$

The final term on the right, which significantly involves an r derivatives of the Christoffel symbols, can be re-expressed in terms θ derivatives and the Riemann tensor through use of the Ricci identity,

$$\gamma^{AB} \partial_1 \Gamma_{1AB} = \gamma^{AB} (R_{1A1B} + \partial_A \Gamma_{11B}). \tag{5.61}$$

The result is expanded in terms of the metric quantities to yield

$$\begin{aligned} T_3 = & \frac{1}{2} \gamma^{AB} \partial_{AB} \gamma_{11} - \gamma^{AB} (g^{0a} \partial_{(1} \gamma_{A)0} \partial_{(1} \gamma_{B)0} + \frac{1}{4} g^{11} \partial_A \gamma_{11} \partial_B \gamma_{11} \\ & + \frac{1}{2} g^{1C} \partial_A \gamma_{11} \partial_1 \gamma_{BC} + \frac{1}{4} g^{CD} \partial_1 \gamma_{AC} \partial_1 \gamma_{BD}) - \gamma^{AB} R_{1A1B}. \end{aligned} \tag{5.62}$$

To this point, all calculations have been performed with the coordinates and metric components of the Cauchy system. However, using the transformations of the metric variables and their derivatives, which are known except for $\partial_1 g_{01}$ and $\partial_1 g_{11}$, we can rewrite the metric components in terms of their corresponding values in the characteristic

coordinates. Note the two functions for which the transformations are not known are related via Equation (5.45). In particular, taking the derivative of this equation yields

$$\partial_1 g_{01} = p g_{11} + \partial_1 p g_{11} + p \partial_1 g_{11}. \tag{5.63}$$

This is used to eliminate $\partial_1 g_{01}$ from T_2,

$$\begin{aligned} T_2 =& \frac{1}{2}(-\partial_0 \gamma_{11} - 2p\gamma_{11} - 2\gamma_1 1 \partial_1 p) g^{0a} Q_a + \partial_1 \gamma_{11}(-p g^{0a} Q_a + \frac{1}{2} g^{1a} Q_a) \\ & - \frac{1}{2} \partial_B g^{Ba} Q_a. \end{aligned} \tag{5.64}$$

Using this expression, and Equation (5.55) along with the expansions (5.56) and (5.62), we can solve for $\partial_1 g_{11}$ as

$$\begin{aligned} \partial_1 g_{11} = & (-p g^{0a} Q_a + \frac{1}{2} g^{1a} Q_a)^{-1} (2/r^2 - T_1 - \frac{1}{2}(-\partial_0 \gamma_{11} - 2\gamma_1 1 \partial_1 p) g^{0a} Q_a + \\ & \frac{1}{2} \partial_B \gamma_{11} g^{Ba} Q_a - T_3 - \gamma^{AB} R_{1A1B}). \end{aligned} \tag{5.65}$$

On I each of terms, here given in terms of Cauchy components, can be written in the characteristic variables by making the transformations given by Equations (5.46a)–(5.46t), making note that

$$\tilde{p} = p \partial_1 \tilde{p} = \partial_a p \tilde{e}_1^a \tilde{R}_{1A1B} = R_{aAbB} \tilde{e}_1^a \tilde{e}_1^b \tag{5.66}$$

(d'Inverno and Vickers (1997)). The result is that Equation (5.65) determines the value of $\tilde{\eta}$ referred to in Equation (5.46n) entirely in terms of characteristic variables at the interface.

Given the transformation equations (5.46a)–(5.46t) and (5.37a)–(5.37p), it is possible to determine the metric coefficients at the interface in one of the coordinate systems given the other. The actual variables used by Stark-Piran, and those of the Bondi metric, can be solved for the metric functions, as can the necessary radial derivatives required to provide data on the interface.

The only problem that might arise concerns whether enough information exists at a particular stage of the evolution so that the transformations can be carried out. In fact, a fully consistent procedure can be defined in which it can be shown that at each stage of the evolution, enough information exists to perform the required integrations on either the Cauchy or characteristic grids, and to carry out any required coordinate transformations in order to determine boundary information at the interface. The description of this procedure is the subject of the next section.

Determination of the Cauchy lapse from incomplete characteristic data

An important factor in practical implementation of any of the coordinate transformations is how much information is actually available in the form of calculated components. This problem is particularly acute with the Stark-Piran and Bondi systems where the calculation of each evolution variable can take place only once certain previous data have been calculated, so that the order of integration is fairly rigidly structured.

An examination of the metric components in the Cauchy region (see Appendix D) and the injection equations (5.46a)–(5.46t) shows that at the point at which boundary data for the inward integration of N is required, not enough information exists in the characteristic region to inject the metric components required to solve for N. However, note that the entire integration of N in the Cauchy region is coupled to the value of β^r, and as such requires an initial guess for this quantity. By extending this to the transformation at the interface, we can determine an outer boundary condition for N using characteristic data in combination with a value for β^r in the Cauchy region. The injection equation for g_{01} is given by Equation (5.46a),

$$g_{01} = \frac{1}{\tilde{\nu}}\left(-\tilde{g}_{00} + (\tilde{\nu} - \tilde{p})\tilde{g}_{01} + \tilde{g}^{AB}\tilde{g}_{0A}\tilde{g}_{0B}\right). \tag{5.67}$$

The expressions for g_{0a} and g_{AB} are transformed to their Cauchy equivalents via Equations (5.46a). The same can be done for the expanded form of $\tilde{\nu}$ resulting in the expression

$$\tilde{\nu} = \frac{1}{g_{01}}(g_{00} + 2\tilde{p}\tilde{g}_{01} - g^{AB}g_{0A}g_{0B}). \tag{5.68}$$

Finally, note that $\tilde{p} = p = \beta^r$ follows from the derivation of Equation (5.45) and the definition of p via Equation (5.44) respectively. This, along with Equations (5.68),(5.67), and (5.68), leads to the equation

$$g_{01}(g_{00} + 2\beta^r\tilde{g}_{01} - g^{AB}g_{0A}g_{0B}) = \beta^r\tilde{g}_{01}. \tag{5.69}$$

The values of the Cauchy metric components at I,

$$g_{00} = -N^2 + A^2(\beta^r)2 + g^{AB}g_{0A}g_{0B}, \tag{5.70}$$

$$g_{01} = -A^2\beta^r, \tag{5.71}$$

and the characteristic component,

$$\tilde{g}_{01} = -e^{2\beta}, \tag{5.72}$$

are substituted into this expression and the result solved for the lapse to give

$$N^2 = A^2(\beta^r)^2 + e^{2\beta}(e^{2\beta}/A - 2(\beta^r)) \tag{5.73}$$

where care must be taken to distinguish the Stark-Piran shift component β^r with the characteristic metric variable β. This is an equation for the lapse in terms of data which is already known at the interface (Cauchy A is determined in Step 2, characteristic β in Step 6) and the value of β^r which is not yet know exactly but results from the coupled system with N. Thus the transformation (5.73) at the interface is applied iteratively until a result is converged upon, just as is the equation for N and β^r over the entire Cauchy grid.

Once the lapse equations have been solved, all of the metric information is known on the Cauchy side of the interface. Thus enough information exists to extract the required inner boundary data for the remaining characteristic integrations. The only remaining task is to fix the boundary conditions of metric variables on the Cauchy grid (namely their radial derivatives) via injection so that a full set of data exists for the procedure to begin at the next timestep.

5.4 A scheme for coordinating Cauchy and characteristic codes

With the theoretical groundwork for both evolution systems laid, as well as a specification for how they can be matched, the only remaining task is to specify a consistent procedure for performing the integration as a complete system. A full procedure for the evolution of a spacetime constructed from 3+1 and characteristic regions which pass information back and forth across a boundary is conditioned by a number of requirements:

- The order of integration of the metric variables in the Cauchy interior;
- The boundary data required by the metric variables in the Cauchy region;
- The order of integration of the metric variables in the characteristic region;
- The boundary data required by the metric variables in the characteristic region;
- The information (in the form of calculable metric components g_{ab} and $\tilde{g}_{ab}$ required by each of the coordinate transformations from one region to the other.

Essentially, the order of operations must be specified in such a manner that each integration step possesses enough information in the form of boundary conditions that it can be carried out.

The integration of the Cauchy lapse function α provides a particular difficulty in that, unlike the other variables, it must be integrated inwards from the characteristic boundary. Its value is dependent on the characteristic parameter β, which it will be remembered is determined by the Bondi slicing condition, (4.51), at null infinity.

Beyond this, we note the following relevant points:

- The Cauchy variables η and ξ are sufficient to determine the transverse 2-metric g_{AB} at the interface. Each of these can be determined by evolution from the previous slice.

- The characteristic variables γ and δ can be determined from the metric components $\tilde{g}_{AB}$ at the interface.

- The transformation of g_{AB} to $\tilde{g}_{AB}$ on the characteristic side of the interface is given simply by

$$\tilde{g}_{AB} = g_{AB}. \tag{5.74}$$

- The Cauchy extrinsic curvature data $\{K_1, K_2, K_3, K_+, K_\times\}$ can be determined by evolution from the previous slice.

- The Cauchy variable $A = \sqrt{g_{rr}}$ is determined on the new slice via outward radial integration, and so requires no outer boundary value.

- The determination of the lapse, N, on the Cauchy slice is an inward radial integration in the polar sliced region which neighbours the interface, and thus requires an outer boundary condition. In order to set this, a value of the Bondi variable β is required at the interface.

- The value of β at the interface is set by solving the Bondi slicing condition (4.51) at $\mathscr{I}^+$ and integrating inwards via (5.25a). This involves the solution of an elliptic equation, (4.48a) for $\tilde{\theta}$ over the sphere at $\mathscr{I}^+$.

- Once N has been set, the remaining Cauchy variables $\{\beta^r, N^\phi, G\}$ can be determined on the new timeslice. These data can be extracted at the interface to determine inner boundary data for the outward integration of the remaining characteristic variables.

- The values of the characteristic variables and their derivatives at the interface can be injected to set outer boundary data for the variables on the Cauchy grid.

These considerations lead to the following consistent procedure for carrying out an integration over the full Cauchy and characteristic regions so that sufficient boundary information is supplied to each region as required. For each step of the procedure, a table listing which information is known in each part of the grid on both the original ($t = t_0$) and evolved ($t = t_1$) is given. The procedure begins by assuming a full knowledge of metric variables on the initial slice and is complete when the equivalent information has been placed on the evolved slice:

Time	*Cauchy*	*Interface*	*Characteristic*
t_1			
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

Given this data at t_0, new data (indicated in boldface) are determined on the t_1 slice by the following steps:

1. *Cauchy:* Using data at t_0, solve for the metric variables $\{\eta, \xi\}$ over the Cauchy grid at t_1 by integrating equations (5.12) and (5.13).

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\boldsymbol{\eta, \xi}$		
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

2. *Cauchy:* Solve for the extrinsic curvature components $\{K_+, K_\times, K_1, K_2, K_3\}$, over the Cauchy grid at t_1 by integrating equations (5.14a)–(5.14e).

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\eta, \xi, \boldsymbol{K_{ab}}$		
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

3. *Cauchy:* Solve for A over the Cauchy grid at t_1 by carrying out an outward integration of equation (5.15).

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\eta, \xi, K_{ab}, \boldsymbol{A}$		
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

4. *Interface*: Transform the 3-metric components $h_{\alpha\beta}$ (dependent on A, η, ξ) from 3+1 to Bondi coordinates using equation (5.37a) at the interface in order to determine initial data for the Bondi δ and γ.

Time	Cauchy	Interface	Characteristic
t_1	η, ξ, K_{ab}, A	δ, γ	
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

5. *Characteristic:* Perform an outward integration of $\{\delta, \gamma\}$, using the hypersurface evolution equations (5.25e) and (5.25f) and initial data at I supplied in the previous step.

Time	Cauchy	Interface	Characteristic
t_1	η, ξ, K_{ab}, A	δ, γ	$\boldsymbol{\delta, \gamma}$
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \boldsymbol{\delta_{,u}, \gamma_{,u}}$

6. *Characteristic:* Use the Bondi slicing condition and the values of δ and γ at $\mathscr{I}^+$ in order to determine initial data for β as described in Section 4.2. Integrate β inwards over the characteristic grid.

Time	Cauchy	Interface	Characteristic
t_1	η, ξ, K_{ab}, A	δ, γ	$\delta, \gamma, \boldsymbol{\beta}$
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

7. *Interface:* Transform the value of β at I (which determines the Bondi g_{01} component) via equation (5.37a) to determine a value for the lapse, α, on the interior (see the discussion below).

Time	Cauchy	Interface	Characteristic
t_1	η, ξ, K_{ab}, A	$(\beta^r), \delta, \gamma$	δ, γ, β
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

8. *Cauchy:*

   (a) Given a value for β^r, inject the characteristic β data to determine a value for the lapse, N, at the interface of the t_1 Cauchy grid.

   (b) Integrate the coupled equations (5.17a)–(5.18) for G, β^r, and N.

   The data for β^r from the second of these steps is used as input for the first step, and the process is repeated until convergence to a specified accuracy.

Time	Cauchy	Interface	Characteristic
t_1	$\eta, \xi, K_{ab}, A, \boldsymbol{G, \beta^r, N}$	$\boldsymbol{\beta^r}, \delta, \gamma$	δ, γ, β
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

9. *Interface:* The Cauchy metric components g_{ab} are fully determined at this stage, and are transformed to determine the remaining Bondi metric components via equations (5.37a), which in turn are used to determine initial values for the Bondi functions U, W, V.

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\beta^r, \delta, \gamma, \boldsymbol{U, U_{,r}, W, W_{,r}, V}$	δ, γ, β
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

10. *Characteristic:* An outward integration from the interface is used to determine $\{U, W, V\}$, over the characteristic grid at t_1.

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\beta^r, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, \boldsymbol{U, W, V}$
t_0	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

11. *Interface:* At this point all of the metric functions on each grid have been determined for t_1. Data from each side of the interface are used to determine the required derivatives of the metric functions at I so that the process can be repeated from Step 1.

Time	*Cauchy*	*Interface*	*Characteristic*
t_1	$\eta, \xi, \beta^r, K_{ab}, A, G, N$	$\boldsymbol{\eta, \xi, K_{ab}, A, G}, \beta^r, \boldsymbol{N}, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$
t_0	$\eta, \xi, K_{ab}, A, G, N, \beta^r$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V, \delta_{,u}, \gamma_{,u}$

12. At this stage, all of the data for both regions exists on the t_1 slice and the process can be repeated to evolve to a future t_2.

Time	*Cauchy*	*Interface*	*Characteristic*
$t+2$			
t_1	$\eta, \xi, K_{ab}, A, G, \beta^r, N$	$\eta, \xi, K_{ab}, A, G, \beta^r, N, \delta, \gamma, U, U_{,r}, W, W_{,r}, V$	$\delta, \gamma, \beta, U, W, V$

It can be shown that at all stages of the calculation enough data exists to perform the required integrations. The only difficulty results in providing initial data for N in the coupled solution of the shift and lapse equations in Step 8.

5.5 Numerical implementation

The scheme for matching a Stark-Piran type interior Cauchy system with a Bondi characteristic system which has been outlined in the previous sections is in the final stages

of being implemented as a numerical code by the Southampton Relativity Group. In general, standard finite differencing techniques are used with the goal of achieving second order accuracy and convergence as well as long term stability. The intention is that this should be achievable first by each of the interior and exterior regions, and then by the system as a whole.

Grid structure

The Cauchy grid

In setting up the grid for the Cauchy evolution, we follow closely the layout proposed by Stark and Piran (1987). Significantly, attention is paid to the relative placement of the individual variables on the grid so that their values and derivatives are centred correctly relative to each other depending on how they appear in Equations (5.12)–(5.21), as well as to their behaviour at the grid boundaries, in particular the polar axis.

Following Stark and Piran (1987), a system of two grids (labelled 'a' and 'b') in the radial direction are used. The distance between grid points is set at a uniform Δr for each grid. The two grids are offset by half a radial distance, $\Delta r/2$, so that points on the b-grid are centred between points on the a-grid and vice-versa. The purpose for the offset is to preserve the natural placement of the grid variables based on their determining equations. For instance, though the variable ξ can be placed on the a-grid, the variable which is actually evolved is its radial derivative $\partial_r \xi$ via Equation (5.13), which when finite differenced should be centred half-way between grid points, ie. on the b-grid. Because of the offset, both the a- and b-grids can not each have points at the origin and interface. We choose to place the origin ($r = 0$) on a point of the b-grid and the interface, I, on the a-grid.

A similar setup is used in the radial direction. Recall that the coordinate $x = \cos\theta$ is used as an angular coordinate and ranges from the value 0 at the equator to 1 at the polar axis. An angular a-grid with even spacing Δx is constructed to have points which straddle both the pole and the equator. This is so the equations for variables place on this grid will not be required to take values at the poles, where factors of $(1 - x^2)$ commonly cause degeneracy. An angular b-grid is offset by a distance $\Delta x/2$ from the a-grid. Both the pole and the equator sit on points of this grid.

The result is that individual variables are each placed on one of four grid structures defined by the two radial and two angular grids: aa, ab, ba, bb.

Derivatives at the boundaries are handled through the use of ghost-points beyond

the physically relevant sections of the grids. Values of functions at these points are determined by their expected behaviour at the boundaries, as discussed in Section 5.1. In particular, at the origin the expansions listed in Bardeen and Piran (1983) are enforced, while at the poles and equator the even/oddness of the variables are used to determine their values immediately beyond the boundary.

The entire grid layout for the Cauchy region, including ghost zones, is displayed in Figure 5.3.

The characteristic grid

In many senses the evolution system for the Bondi variables is much less complex than that in the Cauchy region, and as such a simpler grid structure is used. Rather than multiple offset grids, variables are placed at points on a single grid. The radial grid has points uniformly spaced at distances Δy, and both the interface and null infinity are points on the grid. The angular grid points are placed at values of the interior Cauchy grids so that data transferred to or from the interior will require a minimum of interpolation. Thus a grid spacing of $\Delta x/2$ is used, where Δx is the corresponding distance between points on each of the a and b Cauchy grids. Ghost zones are also used at the pole and equator, with values of grid variables set using the expansions of 5.2. A boundary condition is not set at $y = 0$ ($\mathscr{I}^+$), as the equations determining the grid variables on a slice are integrated in an outward direction, except for the Bondi variable β which is integrated inwards and whose value at $\mathscr{I}^+$ is set using the Bondi slicing condition as described in Sections 5.2 and 4.2. As such, radial derivatives of the grid functions are not needed at $\mathscr{I}^+$. Radial derivatives at the Cauchy interface are determined by extraction, and are used to place values of the grid functions on a ghost zone point. The layout of the characteristic grid is shown in Figure 5.5.

The interface

The interface exists along a single $r = y = constant$ line on each of the Cauchy and characteristic grid structures. The transformation equations listed in Section 5.3 are used to convert data known on the line represented by the interface on one grid to data on the corresponding line of the other. Figure 5.5 shows the mapping of interface points between the two regions.

A problem arises due to the complicated nature of the Stark-Piran grid structure. The b-grid does not have points on the interface, and thus interpolation is required to

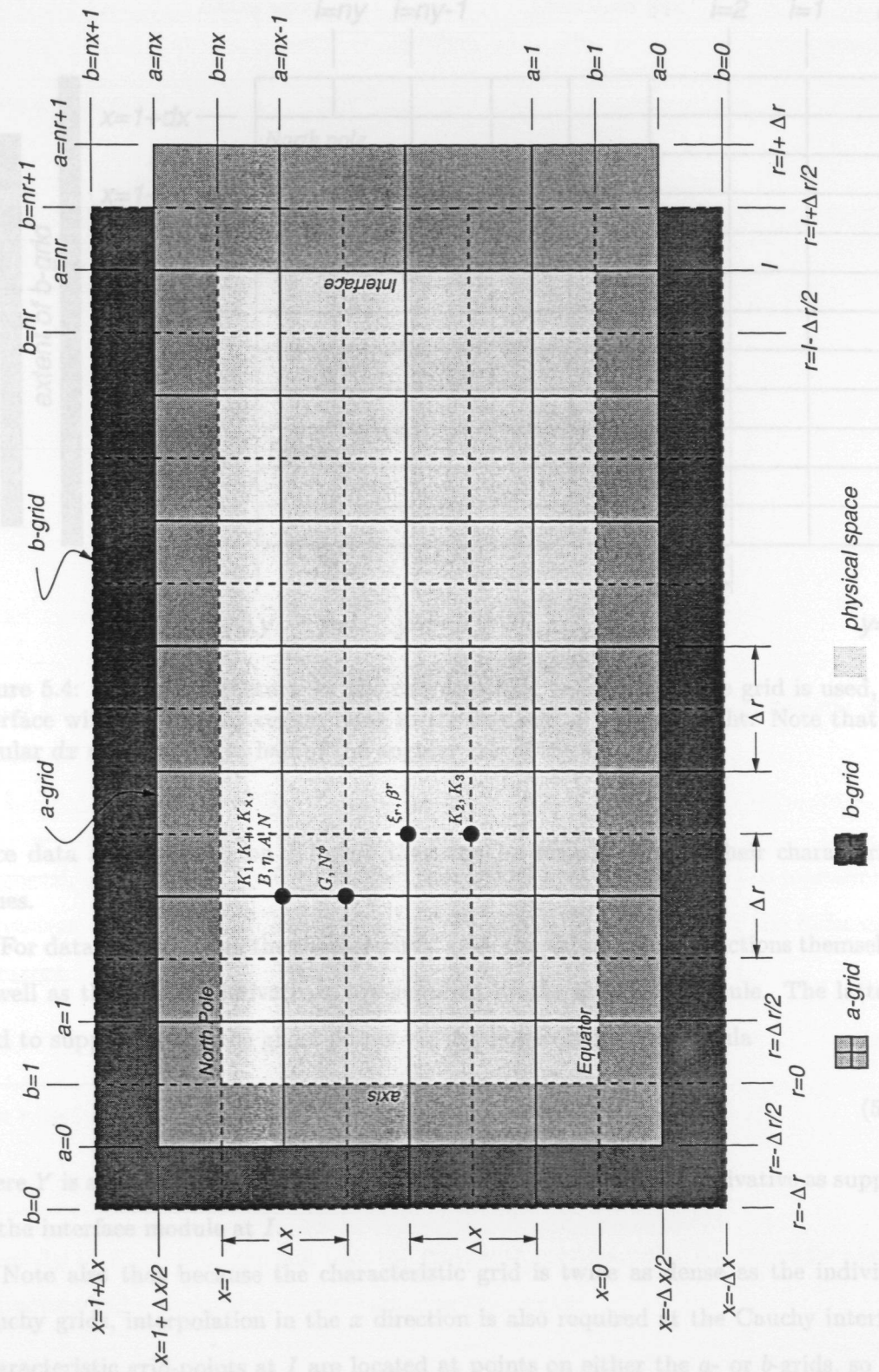


Figure 5.3: The grid structure of the Cauchy evolution scheme. Points on the a-grid (respectively in r and x) are indicated by and those on the b-grid by dotted lines. The placement of the Stark-Piran variables on each of the grids is shown. The physical region is the lightly shaded region in the centre, surrounded by ghost zones where data is placed to allow the calculation of derivatives at the boundaries of the grid. The interface with the characteristic region is located at a point on the radial a-grid.

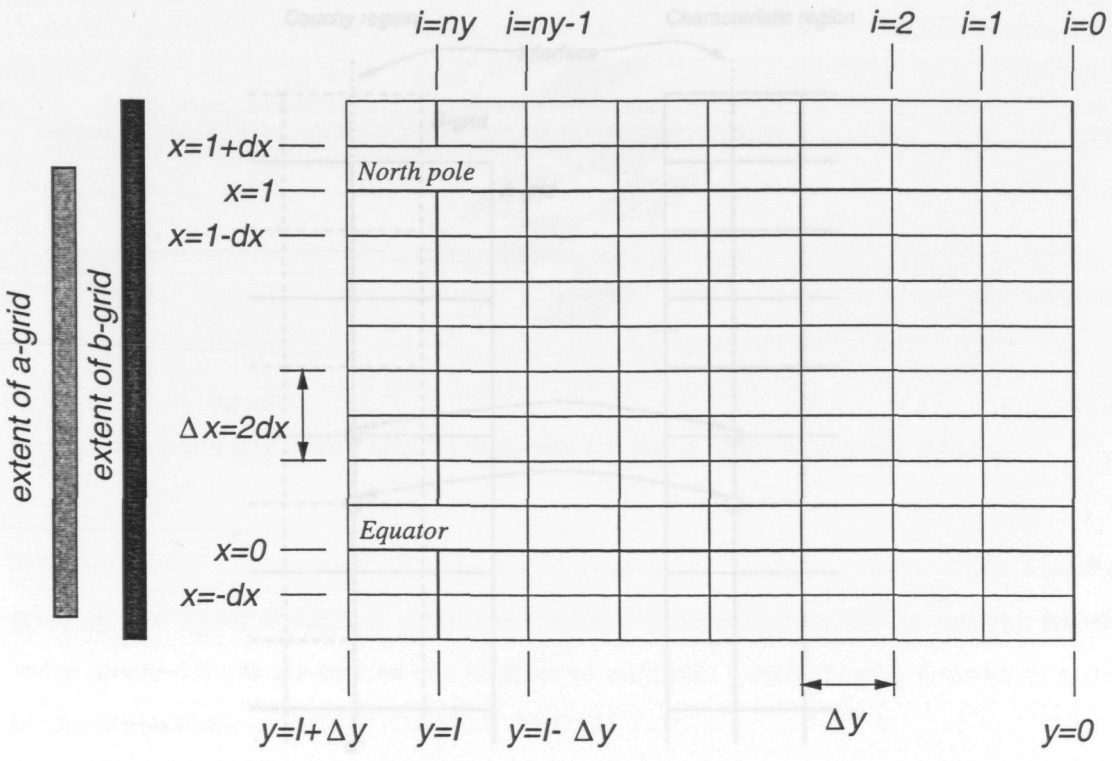


Figure 5.4: The grid structure for the characteristic region. A single grid is used, the interface with the Cauchy region lying to the left and $\mathscr{I}^+$ at the right. Note that the angular dx corresponds to half of the angular Δx of the Cauchy grid.

place data appropriately on I before they can be transformed to their characteristic values.

For data injected from the characteristic grid, the values of the functions themselves, as well as their radial derivatives, are supplied by the interface module. The latter is used to supply data at the ghost points via the finite difference formula

$$Y_{I+1} = Y_{I-1} + 2[\partial_r Y]_I \Delta r + O(r^2), \tag{5.75}$$

where Y is a generic grid variable and $[\partial_r Y]_I$ the value of its radial derivative as supplied by the interface module at I.

Note also that because the characteristic grid is twice as dense as the individual Cauchy grids, interpolation in the x direction is also required at the Cauchy interface. Characteristic grid-points at I are located at points on either the a- or b-grids, so that variables from the alternate grid must be interpolated if their value is required.

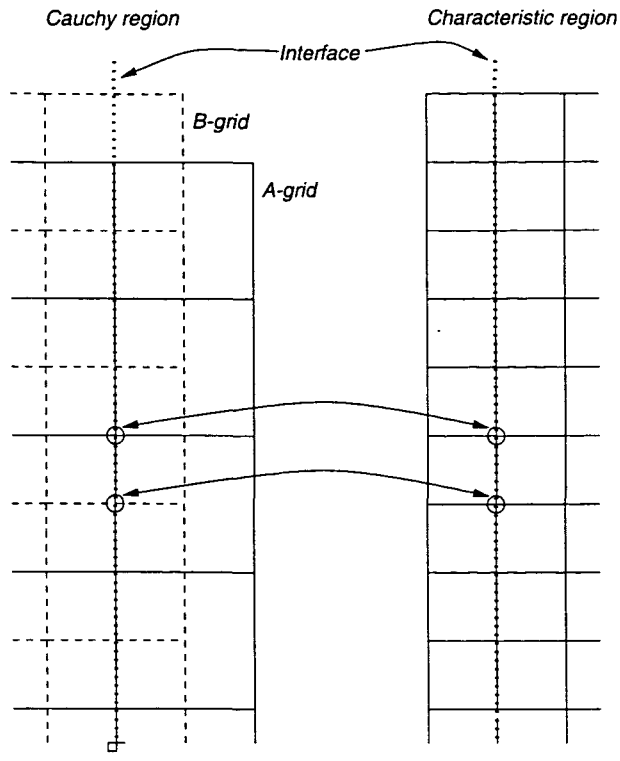


Figure 5.5: The correspondence between points of the Cauchy and characteristic grids.

Finite difference techniques

In general, standard second order finite differencing techniques are used to represent the derivatives of the variables on each of the two grids. In particular, if the differencing operators

$$\Delta Y_i = Y_{i+1} - Y_{i-1}, \qquad \Delta^2 Y_i = Y_{i+1} - 2Y_i + Y_{i-1}, \tag{5.76}$$

then r derivatives of a grid function Y at a point r_i are approximated by

$$[\partial_r Y]_i = \Delta Y_i / 2\Delta r, \qquad [\partial_{rr} Y]_i = \Delta Y_i / \Delta r^2, \tag{5.77}$$

where $\Delta y = y_{i+1} - y_i$ is constant over the grid, with corresponding equations for angular derivatives and y derivatives in the characteristic region.

Various interpolations are required, in particular at the interface where function values in the Cauchy region must be centred at locations of the characteristic grid points. Note, however, that by the choice of interface as an $r = constant$ line which has points on grids in both interior and exterior region, the difficulties usually encountered in performing interpolations between coordinate systems are largely avoided. The

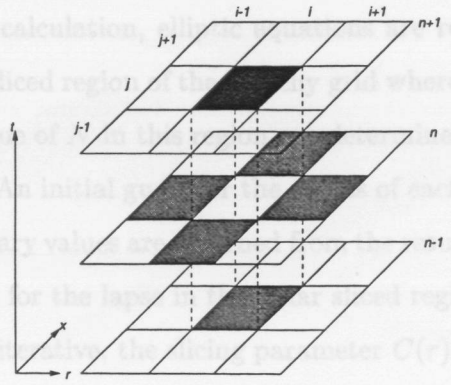


Figure 5.6: Computational molecule for the staggered leapfrog scheme.

location of the characteristic interface grid points are fixed on points of the Cauchy grids by the choice of angular coordinate in the characteristic region. In general, fourth order interpolations are carried out in order to maintain a second order truncation error in the derivatives.

The time evolution equations in both the Cauchy and characteristic regions are carried out using a staggered leapfrog technique in order to achieve second order accuracy in time. For a given variable Y known at timesteps n and $n-1$, the value at $n+1$ is calculated via

$$Y_{ij}^{n+1} = Y_{ij}^{n-1} + [\partial_t Y^n]_{ij} \Delta t / \Delta x, \tag{5.78}$$

where the value of $[\partial_t Y^n]_{ij}$ is determined by the evolution equation reduced to a finite differenced version centred at the grid point ij. Note that with the difference operators specified above, the computational molecule for the determination of this quantity extends from $i-1$ to $i+1$ in the radial direction and $j-1$ to $j+1$ in the angular direction, as shown in Figure 5.5.

The staggered leapfrog scheme is well known to be susceptible to 'mesh drifting' (see, for instance, Press et al. (1986)). To compensate for this, a dissipative term of the form

$$Y_0 = -\frac{1}{16}(Y_{i-2} - 4Y_{i-1} + 6Y_i - 4Y_{i+1} + Y_{i+2}) \tag{5.79}$$

has been found to remove the high frequency mode instabilities. Note that the term is of order four and as such does not influence the second order accuracy of the overall scheme Kreiss and Oliger (1973).

At two stages of the calculation, elliptic equations are required to be solved. The first is in the maximally sliced region of the Cauchy grid where the equation for the lapse becomes elliptic. The value of N in this region are determined by a standard successive over-relaxation method. An initial guess for the values of each variable is obtained from the previous slice. Boundary values are obtained from the result of the inward integration of the parabolic equation for the lapse in the polar sliced region. Although the solution of the elliptic problem is iterative, the slicing parameter $C(r)$ (see Section 5.1) is chosen so as to limit the elliptic solve to a small region around the origin, and convergence is shown to be rapid in tests of the independent Cauchy code.

A second elliptic integration takes place in the specification of the characteristic variable β at $\mathscr{I}^+$ via the Bondi slicing condition. Equation (4.48a) is solved along the $y = 0$ surface representing $\mathscr{I}^+$, which reduces to a line in axial symmetry. Once again, successive over-relaxation is used with a source provided by the values of γ and δ at $\mathscr{I}^+$ which have been determined from the previous timestep.

A full implementation of all three portions of the code has been carried out by Mark Dubal, Chris Clarke, Ray d'Inverno and the author. The Cauchy code has been well tested and reproduces the vacuum results of Stark and Piran (1987) using Robin outer boundary conditions. Teukolsky wave initial data have been used as the primary test bed (Teukolsky (1982)) with wave amplitudes given in terms of $y = t - r$ by the function

$$F(y) = Ay \exp(-y^2). \tag{5.80}$$

for a small amplitude A on the order of 10^{-3}. The waves are found to propagate off of the outer grid boundary. However, after a time corresponding to approximately a single wavelength, an instability in the elliptic region, originating at $r = 0$, begins to grow leading to late-time innacuracies which propagate outwards (see Figure 5.7). The source of this instability is currently being investigated.

The characteristic code is also fully implemented and has been tested by placing the data derived from exact solutions on the interior interface. In particular, the Schwarzschild solution and the boost-rotation symmetric solution of Bičák et al. (1988). The latter is defined only on an initial hypersurface, where given data in δ and γ, the characteristic code has been found to reproduce the exact solution accurately. Schwarzshild data can be evolved stably for an indefinite amount of time, maintaining accuracy to within machine error. Unfortunately, since this data is manifestly static, the dynamic evolution of the Bondi variables can not be adequately tested by this means. Small perturbations on the Schwarzschild data have also been tested, however. For instance,

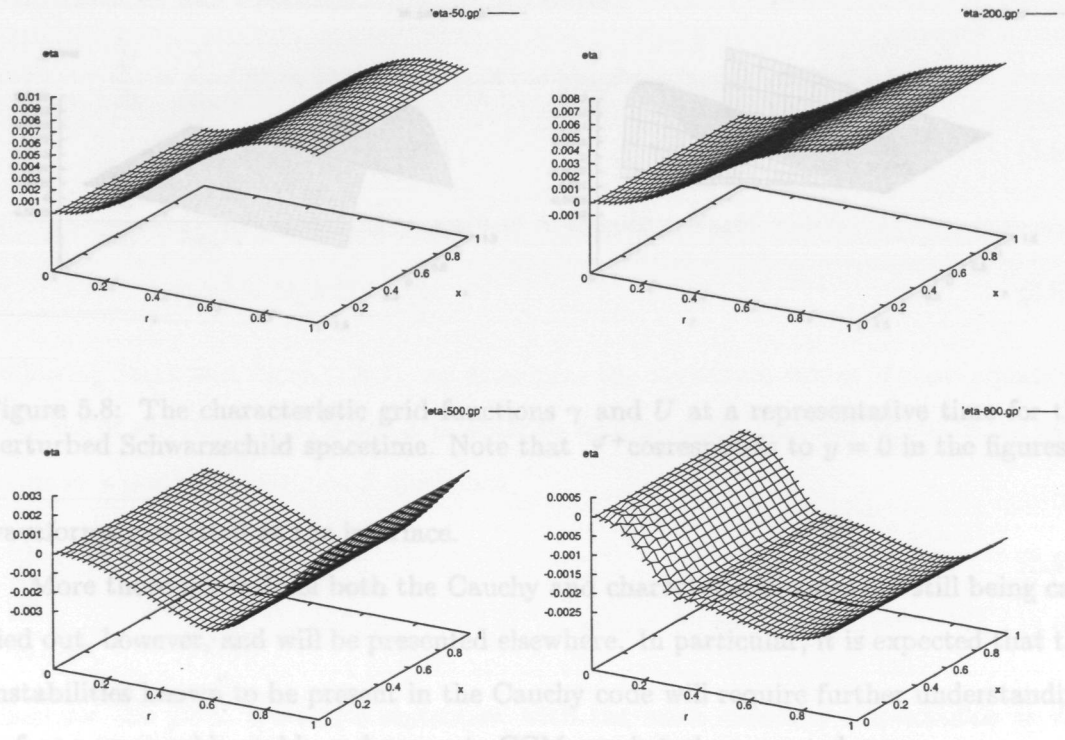


Figure 5.7: Teukolsky wave evolution using an amplitude parameter of $A = 0.001$. The metric function η is shown after 50, 200, 500, and 800 timesteps. Beyond approximately 700 timesteps an instability at the origin begins to affect the grid function calculation in the elliptic region. Though the run can be continued to 1400 timesteps, the results become progressively less accurate.

the data

$$\partial_u \gamma|_I = A e^{t^2 - t_{\max}^2} \sin\theta \tag{5.81}$$

specified at the interface results in an evolution of the metric variables, though not comparable to an exact solution (see Figure 5.8). Under such perturbations, the characteristic code has been found to evolve stably and smoothly until beyond the time when the perturbation has become negligible. It is optimistic to note that in these tests the specification of data for β at $\mathscr{I}^+$ has not been found to induce any problems in the evolution.

The interface is a difficult component to test in isolation, as a proper test requires an exact solution on either side. Since interpolations of metric variables are required, these solutions must be known in a region on either side of the line which forms the interface. For certain of the more simple transformation equations, arbitrary functional data can be given and the outcome of the transformation compared to what would be expected by a hand calculation. In general, however, errors in the transformation equations are expected to be made apparent on the integration of the codes in the form of spurious

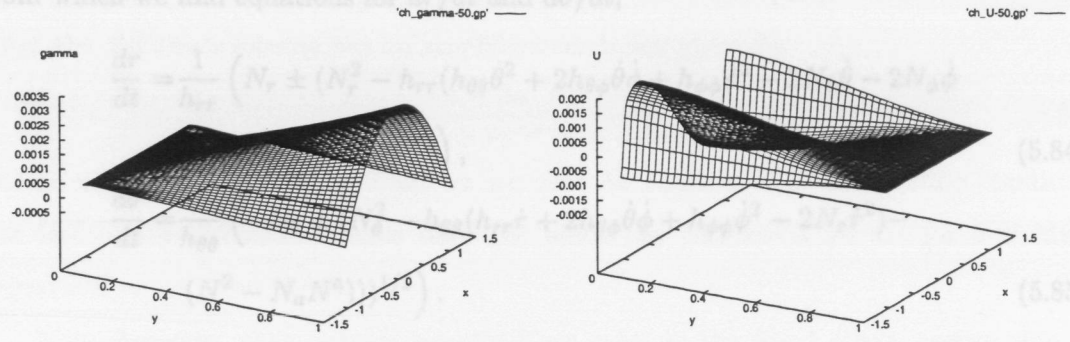


Figure 5.8: The characteristic grid functions γ and U at a representative time for the perturbed Schwarzschild spacetime. Note that $\mathscr{I}^+$corresponds to $y = 0$ in the figures.

waveforms generated at the interface.

More thorough tests of both the Cauchy and characteristic codes are still being carried out, however, and will be presented elsewhere. In particular, it is expected that the instabilities known to be present in the Cauchy code will require further understanding before a reasonably stable and accurate CCM run is to be expected.

Grid spacing and stability in the Cauchy region

Crucial to the stability of any numerical code is the relationship of the timestep, Δt, between evolved slices and the grid spacing, Δr and $r\Delta\theta$. For linear (or linearised) hyperbolic equations, a von Neumann stability analysis leads to the Courant criterion,

$$\Delta t \leq \frac{1}{v}\Delta x, \tag{5.82}$$

where v is the characteristic velocity. In geometric terms, this can be viewed as the statement that the grids must be set up so that the value of a grid function at a point is causally related to the data which determine it: The analytic domain of dependence of a point must lie within the numerical domain of dependence.

Using this as a guiding principle, a causality condition on the timestep can be developed by considering the equations of null geodesics for the given system. For the metric, Eq. (5.1), the null geodesics are determined from the equation

$$-(\alpha^2 - N_\alpha N^\alpha)dt^2 - 2N_\alpha dx^\alpha dt + h_{\alpha\beta}dx^\alpha dx^\beta = 0, \tag{5.83}$$

from which we find equations for dr/dt and $d\theta/dt$,

$$\frac{dr}{dt} = \frac{1}{h_{rr}}\Big(N_r \pm (N_r^2 - h_{rr}(h_{\theta\theta}\dot{\theta}^2 + 2h_{\theta\phi}\dot{\theta}\dot{\phi} + h_{\phi\phi}\dot{\phi}^2 - 2N_\theta\dot{\theta} - 2N_\phi\dot{\phi} - (N^2 - N_aN^a)))^{1/2}\Big), \tag{5.84}$$

$$\frac{d\theta}{dt} = \frac{1}{h_{\theta\theta}}\Big(-N_\theta \pm (N_\theta^2 - h_{\theta\theta}(h_{rr}\dot{r} + 2h_{\theta\phi}\dot{\theta}\dot{\phi} + h_{\phi\phi}\dot{\phi}^2 - 2N_r\dot{r}^2) - (N^2 - N_aN^a)))^{1/2}\Big). \tag{5.85}$$

Following Stark and Piran (1987), we determine the maximum values of these equations in terms of variations of $\dot{r}$, $\dot{\theta}$ and $\dot{\phi}$ in order to determine the furthest extent of the light cones at a point in the r and θ directions,

$$\frac{dr}{dt} \leq \max|N^r \pm N/A|, \tag{5.86}$$

$$\frac{d\theta}{dt} \leq \max|-N_\theta/h_{\theta\theta} \pm (N_\theta^2/h_{\theta\theta} + N_r^2/h_{rr} + N_\phi^2 h_{\phi\phi} + (N^2 - N_aN^a))^{1/2}/h_{\theta\theta}^{1/2}|. \tag{5.87}$$

Thus for the given system, a statement with the same causality implications as Eq. (5.82) is

$$\Delta t \leq \min\left|\frac{(1-x^2)^{-1/2}\Delta x}{d\theta/dt|_{\max}}, \frac{\Delta r}{dr/dt|_{\max}}\right|. \tag{5.88}$$

In principle, the parabolic equations for A, G and N impose analogous restrictions on the allowed relationships between the sizes of the radial and angular grid sizes. The equations for these variables (compare Eqs. (5.15), (5.17a), (5.18)) are of the form

$$\partial_r G = X_G\partial_{xx}\left((1-x^2)G\right) + S_G, \tag{5.89}$$

$$\partial_r A = X_A\partial_x\left((1-x^2)B^2\partial_x A\right) + S_A, \tag{5.90}$$

$$\partial_r N = -X_N\partial_x\left((1-x^2)B^2\partial_x N\right) + \Lambda N + S_N. \tag{5.91}$$

A 2-iteration Crank-Nicholson scheme is used for the finite differencing of these equations. The implementation of the scheme replaces the partial differential equation with variable coefficient $D(r)$,

$$\frac{\partial u}{\partial r} = D(r,x)\frac{\partial^2 u}{\partial x^2} \tag{5.92}$$

by the (implicit) finite differenced equation

$$\frac{u_j^{n+1} - u_j^n}{\Delta r} = \frac{D_{j+1/2}^{n+1}(u_{j+1}^{n+1} + u_j^{n+1})}{2\Delta x^2} - \frac{D_{j-1/2}^{n+1}(u_j^{n+1} + u_{j-1/2}^{n+1})}{2\Delta x^2} + \frac{D_{j+1/2}^{n}(u_{j+1}^{n} - u_j^{n})}{2\Delta x^2} - \frac{D_{j-1/2}^{n}(u_j^{n} - u_{j-1}^{n})}{2\Delta x^2}. \tag{5.93}$$

Substituting a test solution $u_j^n = \xi^n e^{ijk\Delta x}$, as usual for a von Neumann analysis, we find that the difference scheme has an amplification factor given by

$$\xi = \frac{1 - 2\alpha^n \sin^2 k\Delta x/2}{1 + 2\alpha^{n+1} \sin^2 k\Delta x/2} \tag{5.94}$$

where $\alpha^n = (D^n_{j+1} + D^n_{j-1})$. Thus we see that for positive D^n, the stability condition for the Crank-Nicholson scheme does not place any restriction on Δr for individual equations of the form (5.92).

Note, however, that certain complications arise in the Stark-Piran system due to the coupled nature of the parabolic equations. In particular, since the integrations of N and G are coupled, and one is inward stable while the other is outward stable, a more sophisticated analysis should be conducted in order to determine potential stability problems for given choices of Δr and Δx.

We make one further note on the usefulness of applying a von Neumann analysis to the full system of Stark-Piran equations in order to identify potentially unstable modes. Two types of instabilities can occur. Numerical instabilities result from rapidly growing solutions to the finite difference equations which are not solutions of the corresponding continuum equations. The establishment of von Neumann conditions such as those discussed above is aimed at reducing the effect of such modes. Alternatively, the continuum equations might admit modes which are absent in the desired solution, but are excited by numerical perturbations. Examples of the analysis of these modes for various evolution systems are given by Scheel et al. (1998), and Alcubierre et al. (1999).

The role of constraint violating modes also remains controversial (Choptuik (1991), Frittelli (1997)). As noted in Section 4.1, though analytically the constraint equations can be expected to hold for all times if they hold on a single slice, this is not necessarily true of the finite differenced versions of the evolution equations. Unstable modes in the numerical equations can drive the constraints away from their initial zero value (Detweiler (1987)). To avoid such instabilities, constraint enforcing terms have been proposed, such as adding terms of the form

$$+ \epsilon \left({}^{(3)}R + (\mathrm{tr}K)^2 - K_{ab}K^{ab} \right) \tag{5.95}$$

(where ϵ is a parameter and the term vanishes when the constraints are satisfied) at an appropriate point in the evolution system as a constraint 'driver' term. Analyses of such methods have, however, shown that the stability of the system can depend crucially on the value chosen for the parameter ϵ which controls the extent to which the constraint is blended in to the system (Alcubierre et al. (1999)).

It seems, then, that the stability of the system can be determined by the manner in which the constraint equations are applied. A stability analysis should be able to shed some light on whether the Stark-Piran system of equations inherits any such unstable modes which would require special treatment to remove.

In practice, the implementation of the Stark-Piran code considered here calculates the momentum constraint equation for given timesteps as the evolution and outputs its L_2 norm as a check on its error. The Hamiltonian constraint, on the other hand, is used in the determination of the metric variable A. As a check on this value, the evolution equation for A,

$$\frac{dA}{dt} = -(\partial_r(r\beta^r) + NK_1)A \tag{5.96}$$

can be integrated for comparison. This provides a check on a stable evolution to ensure that it is producing a result consistent with the full set of Einstein equations. However, it will not detect the presence of unstable modes in the numerical system which can grow to halt the calculation.

Conclusions

The algebraic classification of spacetimes

In the first part of this thesis, a methodology was developed for classifying exact solutions of the Einstein Field equations. The system that was described is based upon the Cartan-Karlhede methods. These methods were examined in more detail than has previously been published, however, and particular techniques were proposed for dealing with the numerous problems that are faced.

In Section 2.3, the notion of a standard form for the Weyl spinor was generalised to arbitrary symmetric spinors. Algorithmic techniques for determining these standard forms, as well as the transformations required to convert a spinor with arbitrary coefficients to one of these forms. The techniques have been designed to be both efficient in terms of the number of computations required, and also to simplify the form of the final expressions. The result of applying these transformations to the curvature spinor and its derivatives is a full algebraic classification of the exact solution. As well as elucidating symmetry properties of the spacetime, this classification can also be used to determine the equivalence between pairs of spacetimes in a coordinate invariant manner.

A number of difficulties in the Cartan-Karlhede methods have also been identified. In particular, the transformation coefficients for the standard form of an algebraically general spacetime arise as a result of solving a high-order polynomial. Although techniques exist up to 4th order, the resulting coefficients can be unwieldy both in carrying out the computations and in their later interpretation. This problem arises already at the first stage (Petrov classification) of the procedure. As a result, all of the spacetimes which have been classified in practice have been algebraically special or specified in a carefully chosen frame. Past the first stage of the classification procedure, however, the standard forms developed in Section 2.3 ensure that no more than a quadratic ever need be calculated.

Though the methods for are not difficult to specify, the sheer number of calculations

that must be carried out is enormous. Thus, a set of computer algebra tools, described in Chapter 3, was developed to determine the algebraic symmetries of a spacetime, derivative operators associated with spinors, and perform transformations on spinor components. As a result of this work, the GRTensor computer algebra system has been extended to include a number of tools useful to researchers in the field of exact solutions.

The ultimate objective is to use GRTensor as the front end of the computer data base of exact solutions Skea (1997b), so that the members of the user community themselves will be able to submit candidate solutions to update the database.

Numerical integrations via Cauchy-characteristic matching

The accurate representation of outer boundaries of a numerical grid is an acute problem for modern approaches to numerical relativity. As highly nonlinear problems such as the merger of black hole pairs are achieving greater success, the standard approaches to handling boundary conditions are becoming an increasing source of error, both for the evolution itself, and for the interpretation of data which is produced.

In this thesis is presented a structure for solving this problem in axial symmetry. The development of three independent codes is presented, for each of the Cauchy, characteristic regions, and that of the matching interface between them. Further, the solutions to the problems that need to be handled by such a code, in the form of gauge conditions and data transfer at the interface, have been presented.

A procedure for transforming both the metric coefficients and their derivatives between Cauchy and characteristic components was outlined in 5.3. The significance of this is that through use of these transformations, a complicated interpolation between grids over a region can be avoided, as all of the data is transformed on a single $r = constant$ line.

For the particular axisymmetric Cauchy code of Stark and Piran (1987), a number of issues arise in attempting to provide a characteristic outer boundary. A consistent evolution strategy on the Cauchy grid, differing somewhat from the original presentation of Stark and Piran in its determination of a number of the grid variables, was presented in Section 5.1. This minimum number of variables includes a component which can be related to the choice of slicing at $\mathscr{I}^+$. As such, it is fixed using a condition specified in Section 4.2 that the asymptotic form of the metric variables be transformable to that of the original studies by Bondi. Even with this specification, however, a value for the lapse can not be obtained at the interface via a simple transformation at the

interface. Instead, Section 5.3 showed that a combination of interior and exterior data consistent with the evolution scheme on the Cauchy region is sufficient to provide an outer boundary. The issue highlights a particular problem with the implementation of any Cauchy-characteristic scheme, namely that of being able to provide the information at the interface when it is needed. This problem is expected to be particularly acute in situations in which an elliptic equation must be solved for the shift and the lapse.

Finally, Section 5.4 tied the two systems of equations together with the transformation equations at the interface to provide a scheme for the evolution of the full CCM system. The system is consistent in the sense that appropriate data in the form of boundary conditions at the interface is present at each stage of the integration.

Details of the numerical implementation of the system were given at the end of Chapter 5. The fact that the numerical implementation is nearing completion but has yet to be properly tested gives certain scope for future work. The axisymmetric code represents a step in a ladder of increased complexity for the Southampton CCM project, which first used single dimensional codes as testbeds for the CCM concept, has moved to two dimensions, and will progress to three. Fortunately many of the equations developed here, in particular in the characteristic region and at the interface, require very little modification in the progression to three dimensions and potentially Cartesian coordinates. The problems that have been identified here will also need to be faced in more general codes and it is hoped that the experiences gained here can provide a strong background for their solution.

Appendices

A

An algorithm for the determination of the Petrov type

The presentation of an algorithm for the determination of the Petrov type by Letniowski and McLenaghan (1988) builds on the original work of d'Inverno and Russell-Clark (1971) and later extensions to be found in Fitch (1971), Hon (1975), Åman et al. (1991). It's main advantage is the simplicity of its representation. In many cases one can determine the Petrov type simply by knowing which of the Weyl spinor components are zero. In other cases, a simple check using a polynomial formed from the non-zero indices suffices to distinguish the Petrov type. Thus, a table is constructed with each row representing a possible zero/non-zero component configuration, listed in binary order, and the tests required to determine the Petrov type. A code to calculate the Petrov type, then, would apply only the tests corresponding to the initial Weyl spinor configuration. The amount of coding can be reduced by recognizing that a number of cases are identical once the $\Psi_0 \leftrightarrow \Psi_4$ and $\Psi_1 \leftrightarrow \Psi_3$ components are interchanged.

When applying this type of algorithm within a computer algebra system, however, some care on the part of the user is required. In order to arrive at its conclusion, the system must establish whether certain polynomials formed from the Weyl spinor components are equal to zero. Whether this can be done accurately is strongly dependent on the ability of the computer algebra system to simplify the given polynomials appropriately. For instance, the computer algebra system Maple will only give a zero result for

$$\sqrt{1-x} - i\sqrt{x-1}$$

once the routine `radsimp()` has been applied. It is important, then, that the system be able to present its internal calculations to the user for checking.

The complete algorithm is listed in Table A.1, with some special cases given below. A derivation of the listed tests can be found in Letniowski and McLenaghan (1988).

Case	Ψ_0	Ψ_1	Ψ_2	Ψ_3	Ψ_4	*Analysis*
0	0	0	0	0	0	Type 0
1	0	0	0	0	N	Type N
2	0	0	0	N	0	Type II
3	0	0	0	N	N	Type III
4	0	0	N	0	0	Type D
5	0	0	N	0	N	Type II

Case	Ψ_0	Ψ_1	Ψ_2	Ψ_3	Ψ_4	*Analysis*
6	0	0	N	N	0	Type II
7	0	0	N	N	N	$2{\Psi_3}^2 - 3\Psi_2\Psi_4 = 0$ { *true* ⇒ Type D; *false* ⇒ Type II
8	0	N	0	0	0	Type II (see 2.)
9	0	N	0	0	N	Type I
10	0	N	0	N	0	Type I
11	0	N	0	N	N	$27{\Psi_4}^2\Psi_1 + 64{\Psi_3}^3 = 0$ { *true* ⇒ Type II; *false* ⇒ Type I
12	0	N	N	0	0	Type II (see 6.)
13	0	N	N	0	N	${\Psi_1}^2\Psi_4 + 2{\Psi_2}^3 = 0$ { *true* ⇒ Type II; *false* ⇒ Type I
14	0	N	N	N	0	$9{\Psi_2}^2 - 16\Psi_1\Psi_3 = 0$ { *true* ⇒ Type II; *false* ⇒ Type I
15	0	N	N	N	N	Type I, II or III (see Case 15, below)
16	N	0	0	0	0	Type N (see 1.)
17	N	0	0	0	N	Type I
18	N	0	0	N	0	Type I (see 9.)
19	N	0	0	N	N	$\Psi_0{\Psi_4}^3 - 27{\Psi_3}^4 = 0$ { *true* ⇒ Type II; *false* ⇒ Type I
20	N	0	N	0	0	Type II (see 5.)
21	N	0	N	0	N	$9{\Psi_2}^2 - \Psi_4\Psi_4 = 0$ { *true* ⇒ Type D; *false* ⇒ Type I
22	N	0	N	N	0	Type I or II (see 13.)
23	N	0	N	N	N	Type I, II or III (see Case 23, below)
24	N	N	0	0	0	Type III (see 3.)
25	N	N	0	0	N	Type I or II (see 19.)
26	N	N	0	N	0	Type I or II (see 11.)
27	N	N	0	N	N	Type I, II, III or D (see Case 27, below)
28	N	N	N	0	0	Type II or D (see 7.)
29	N	N	N	0	N	Type I, II or III (see 23.)
30	N	N	N	N	0	Type I, II or III (see 15.)
31	N	N	N	N	N	Type I, II, III, N or D (see Case 31, below)

Case 15:

Define

$$I := 3\Psi_2{}^2 - 4\Psi_1\Psi_3,$$

$$F_1 := \Psi_2\Psi_3 - 3\Psi_1\Psi_4,$$

$$F_2 := 9\Psi_2\Psi_4 - 8\Psi_3{}^2,$$

$$\hat{D} := 3F_1{}^2 + 2IF_2.$$

The following tests lead to the Petrov type:
if $I = 0$ then
    if $F_1 = 0$ then Type II,
    otherwise Type I,
otherwise
    if $F_1 = 0$ or F_2=0 then Type I
    otherwise if $\hat{D} = 9$ then Type I
    otherwise Type I.

Case 23:

Define

$$I := \Psi_0\Psi_4 + 3\Psi_2{}^2,$$

$$\hat{J} := 4\Psi_2\Psi_4 - 3\Psi_3{}^2,$$

$$F_3 := \Psi_0\hat{J} - 2\Psi_2 I,$$

$$\tilde{D} := \Psi_4 I^2 - 3\hat{J}F_3.$$

The following tests lead to the Petrov type:
if $I = 0$ then
    if $\hat{J} = 0$ then Type III,
    otherwise Type I,
otherwise
    if $\hat{J} = 0$ or F_3=0 then Type I,
    otherwise if $\tilde{D} = 0$ then Type II,
    otherwise Type I.

Case 27:

Define

$$I := \Psi_0\Psi_4 + 2\Psi_1\Psi_3,$$

$$J := -\Psi_0\Psi_3{}^2 - \Psi_1{}^2\Psi_4,$$

$$D := I^3 - 27J^2,$$

$$U := \Psi_0\Psi_4 + 2\Psi_1\Psi_3,$$

$$V := \Psi_0\Psi_3{}^2 - \Psi_1{}^2\Psi_4,$$

$$W := \Psi_0\Psi_4 - 16\Psi_1\Psi_3.$$

The following tests lead to the Petrov type:
if $V = 0$ then
    if $U = 0$ then Type D,
    otherwise if $W = 0$ then Type II,
    otherwise Type I,
otherwise if $I = 0$ then
    if $J = 0$ then Type III,
    otherwise if $D \neq 0$ or $J = 0$ then Type I,
    otherwise if $D = 0$ then Type II,
    otherwise Type I.

Case 31:

Define

$$H := \Psi_0\Psi_2 - \Psi_1^2,$$

$$F := \Psi_0\Psi_3 - \Psi_1\Psi_2,$$

$$A := \Psi_1\Psi_3 - \Psi_2^2,$$

$$E := \Psi_0\Psi_4 - \Psi_2^2,$$

$$Q := 37\Psi_2^2 + 27\Psi_1\Psi_3,$$

$$I := E - 4A,$$

$$G := \Psi_0 F - 2\Psi_1 H,$$

$$J := \Psi_4 H - \Psi_3 F + \Psi_2 A,$$

$$S := \Psi_0^2 I - 3H^2,$$

$$Z := \Psi_0^2 I - 12H^2,$$

$$D := I^3 - 27J^2.$$

The following tests lead to the Petrov type:
if $H = 0$ then
    if $F = 0$ then
        if $E = 0$ then Type N,
        otherwise Type I,
    otherwise,
        if $E = 0$ then if $Q = 0$ then Type II,
            otherwise Type I,
        otherwise
            if $I = 0$ or $D \neq 0$ Type I,
            otherwise Type II,
otherwise
    if $I = 0$ then if $J = 0$ then Type III,
        otherwise Type I,
    otherwise if $G = 0$ then
        if $Z = 0$ then Type D,
        otherwise if $S = 0$ then Type II,
        otherwise Type I,
    otherwise if $J = 0$ or $D \neq 0$ then Type I,
    otherwise Type II.

B

Transformations to standard form for given Petrov types

The spin matrices which will transform a general Weyl tensor into the canonical forms listed in Table 2.1 are given below. The transformation to be used is determined by the initial zero/non-zero state of the individual Weyl tensor components and the Petrov type. Transformations are decomposed into a pair of null rotations about each of the Weyl spinors and a scaling spin-boost as follows:

$$\begin{pmatrix} \lambda & 0 \\ 0 & \lambda^{-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix} \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda & \lambda\alpha \\ \beta/\lambda & (\beta\alpha+1)/\lambda \end{pmatrix} \tag{B.1}$$

The following table lists an appropriate choice of α, β and λ which will bring the Weyl spinor into its standard form. For most algebraically special cases, the expressions are of a simple form and are stated explicitly. The more general type I and II configurations (in particular cases 19, 23, 27, 31) require the solution of a quartic, and it will only be for very special values of the Weyl components that a practically useful dyad transformation can be found.

	Weyl components	*Type*	*Transformation*
1	(0,0,0,0,N)	N	$\alpha = 0$ $\beta = 0$ $\lambda = \Psi_4^{1/4}$
2	(0,0,0,N,0)	III	$\alpha = 0$ $\beta = 0$ $\lambda = \Psi_3^{1/2}$
3	(0,0,0,N,N)	III	$\alpha = 0$ $\beta = -\Psi_3\Psi_4/4$ $\lambda = \Psi_3^{1/2}$
4	(0,0,N,0,0)	D	$\alpha = 0$ $\beta = 0$ $\lambda = 1$

	Weyl components	*Type*	*Transformation*
5	(0,0,N,0,N)	II	$\alpha = 0$ $\beta = 0$ $\lambda = \Psi_4^{1/4}$
6	(0,0,N,N,0)	II	$\alpha = 0$ $\beta = -\Psi_3/3\Psi_2$ $\lambda = (-2\Psi_3^2/3\Psi_2)^{1/4}$
7	(0,0,N,N,N)	D	$\alpha = 0$ $\beta = -\Psi_3/3\Psi_2$ $\lambda = 1$
		II	$\alpha = 0$ $\beta = -\Psi_3/3\Psi_2$ $\lambda = (-2\Psi_3^2/3\Psi_2 + \Psi_4)^{1/4}$
8	(0,N,0,0,0)	—	(equivalent to case 2 under basis interchange)
9	(0,N,0,0,N)	I	$\alpha = 0$ $\beta = (-\Psi_4/4\Psi_1)^{1/3}$ $\lambda = 1$ (result is type 14.)
10	(0,N,0,N,0)	I	$\alpha = -(\Psi_1/\Psi_3)^{1/2}$ $\beta = (1/2)(\Psi_3/\Psi_1)^{1/2}$ $\lambda = (-1)^{3/4}(4\Psi_3/\Psi_1^3)^{1/8}/2$
11	(0,N,0,N,N)	II	$\alpha = -8\Psi_3/3\Psi_4$ $\beta = 5\Psi_4/16\Psi_3$ $\lambda = (1/2)(-2\Psi_4/3)^{1/4}$
	(0,N,0,N,N)	I	$\alpha = 0$ $\beta = -(1/6)(-(3\sigma)^{2/3} + 12\Psi_3)/(3\sigma)^{1/3}\sqrt{\Psi_1}$ $\lambda = 1$ where $\sigma = -9\Psi_4\Psi^{1/2} + \sqrt{3}(64\Psi_3^3 + 27\Psi_4^2\Psi_1)^{1/2}$ (result is type 14.)
12	(0,N,N,0,0)	—	(equivalent to case 6 under basis interchange)
13	(0,N,N,0,N)	II	$\alpha = -\Psi_1/\Psi_2$ $\beta = 2\Psi_2/3\Psi_1$ $\lambda = (2\Psi_2^3/3\Psi_1^2)^{1/4}$
		I	$\alpha = 0$

	Weyl components	*Type*	*Transformation*
			$\beta = (\sigma^{2/3} + \Psi_2^2 - \Psi_2\sigma^{1/3})/2\sigma^{1/3}\Psi_1$ $\lambda = 1$ where $\sigma = -\Psi_1^2\Psi_4 - \Psi_2^3 + \Psi_4^{1/2}(\Psi_1^2\Psi_4 + 2\Psi_2^3)^{1/2}\Psi_1$ (result is type 14.)
14	(0,N,N,N,0)	II	$\alpha = -4\Psi_1/3\Psi_2$ $\beta = 3\Psi_2/8\Psi_1$ $\lambda = (3/2)(\Psi_2^3/12\Psi_1^2)^{1/4}$
		I	$\alpha = (\Psi_1/\Psi_3)^{1/2}$ $\beta = -(1/2)(\Psi_3/\Psi_1)^{1/2}$ $\lambda = (\Psi_3/2\Psi_1(4\sqrt{\Psi_1\Psi_3} + 3\Psi_2))^{1/4}$
15	(0,N,N,N,N)	III	$\alpha = -2\Psi_1/\Psi_2$ $\beta = \Psi_2/2\Psi_1$ $\lambda = \Psi_2/2(-\Psi_1)^{1/2}$
		II	$\alpha = 0$ $\beta = -\Psi_2/2\Psi_1$ $\lambda = 1$ (result is type 11.)
15		I	$\alpha = 0$ $\beta = 3^{2/3}\frac{(3^{2/3}\sigma^{2/3} - 12\Psi_1\Psi_3 + 9\Psi_2(\Psi_2 - \sigma^{1/3}))}{18\sigma^{1/3}\Psi_1}$ $\lambda = 1$ where $\sigma = 18\Psi_1\Psi_2\Psi_3 - 9\Psi_1^2\Psi_4 - 9\Psi_2^3 + \Psi_1\sqrt{3\gamma}$, $\gamma = 64\Psi_1\Psi_3^3 - 36\Psi_2^2\Psi_3^2 - 108\Psi_1\Psi_2\Psi_3\Psi_4$ $+27\Psi_1^2\Psi_4^2 + 54\Psi_2^3\Psi_4$
16	(N,0,0,0,0)	—	(equivalent to case 1 under basis interchange)
17	(N,0,0,0,N)	I	$\alpha = 0$ $\beta = 0$ $\lambda = \Psi_0^{-1/4}$
18	(N,0,0,N,0)	—	(equivalent to case 9 under basis interchange)
19	(N,0,0,N,N)	II	$\alpha = -3\Psi_3/\Psi_4$ $\beta = 2\Psi_4/9\Psi_3$ $\lambda = (\Psi_4/9)^{1/4}$
		I	α is root of $\Psi_0 z^4 + 4\Psi_3 z + \Psi_4 = 0$ $\beta = 0$

	Weyl components	*Type*	*Transformation*
			$\lambda = 1$ (result is type 15.)
20	(N,0,N,0,0)	—	(equivalent to case 5 under basis interchange)
21	(N,0,N,0,N)	D	$\alpha = -(-\Psi_0/3\Psi_2)^{1/2}$ $\beta = -(1/2)(-3\Psi_2/\Psi_0)^{1/2}$ $\lambda = 1$
		I	$\alpha = 0$ $\beta = 0$ $\lambda = \Psi_0^{-1/4}$
22	(N,0,N,N,0)	—	(equivalent to case 13 under basis interchange)
23	(N,0,N,N,N)	III	$\alpha = -2\Psi_4/3\Psi_3$ $\beta = 3\Psi_3/8\Psi_4$ $\lambda = -3\Psi_3^{3/2}/\sqrt{2}\Psi_4$
		I,II	α is root of $\Psi_4 z^4 + 4\Psi_3 z^3 + 6\Psi_2 z^2 + \Psi_0 = 0$ $\beta = 0$ $\lambda = 1$ (result is type 15.)
24	(N,N,0,0,0)	—	(equivalent to case 3 under basis interchange)
25	(N,N,0,0,N)	—	(equivalent to case 19 under basis interchange)
26	(N,N,0,N,0)	—	(equivalent to case 11 under basis interchange)
27	(N,N,0,N,N)	D	$\alpha = (1+\sqrt{3})\Psi_0/2\Psi_1$ $\beta = -\Psi_1/\sqrt{3}\Psi_0$ $\lambda = 1$
		III	$\alpha = -\Psi_0/2\Psi_1$ $\beta = \Psi_1/\Psi_0$ $\lambda = -2\Psi_1^{3/2}/\Psi_0$
		I,II	$\alpha = 0$ $\beta = -\Psi_1/\Psi_0$ $\lambda = 1$ (result is type 23.)
28	(N,N,N,0,0)	—	(equivalent to case 7 under basis interchange)
29	(N,N,N,0,N)	—	(equivalent to case 23 under basis interchange)
30	(N,N,N,N,0)	—	(equivalent to case 15 under basis interchange)

	Weyl components	*Type*	*Transformation*
31	(N,N,N,N,N)	N	$\alpha = -\Psi_4/\Psi_3$ $\beta = 0$ $\lambda = \Psi_3/\Psi_4^{3/4}$
		D	$\alpha = \Psi_0/t_+$ $\beta = -t_+t_-/2\sqrt{3}\Psi_0(\Psi_1^2 - \Psi_0\Psi_2)^{1/2}$ $\lambda = t_+/2\sqrt{3}(\Psi_1^2 - \Psi_0\Psi_2)^{1/2}$ where $t_\pm = -\Psi_1 \pm \sqrt{3}(\Psi_1^2 - \Psi_0\Psi_2)^{1/2}$
		III	$\alpha = i((\Psi_0\Psi_2 - \Psi_1^2)/(\Psi_2\Psi_4 - \Psi_3^2))^{1/4}$ $\beta = \Psi_4/4(\Psi_3^2 - \Psi_2\Psi_4)^{1/2}$ $\lambda = (\Psi_3^2 - \Psi_2\Psi_4)^{1/4}$
		I,II	α is root of $\Psi_4 z^4 + 4\Psi_3 z^3 + 6\Psi_2 z^2 + 4\Psi_1 z + \Psi_0 = 0$ $\beta = 0$ $\lambda = 1$ (result is type 15.)

C

A classification of the Edgar-Ludwig metrics

The following is a transcript of Maple output displaying the use of the **spinor** package to classify the Edgar-Ludwig conformally flat pure radiation metrics Wils (1989), Koutras (1992), Skea (1997a).

```
> restart: grt():
```

GRTensorII Version 1.65b (R4)

5 December 1997

Developed by Peter Musgrave, Denis Pollney and Kayll Lake

Copyright 1994 – 1997 by the authors.

Latest version available from : http://astro.queensu.ca/~grtensor/

Defaults read from /home/dp/grii/lib4//grtensor.ini

```
> grlib ( classify ):

Cartan-Karlhede classification routines
Version 0.9 4 Dec 1997
> qload ( EL );

Calculated ds for EL (.021 sec.)
```

Default spacetime = EL

For the EL spacetime :

Coordinates

x(*up*)

$x^{a} = [u, w, x, y]$

Line element

$$ds^2 = (2\,\mathrm{f}(u)\,x\,\mathrm{g}(u)\,y + 2\,\mathrm{f}(u)\,x\,\mathrm{h}(u) + 2\,\mathrm{f}(u)\,x^3 + 2\,\mathrm{f}(u)\,x\,y^2 - w^2)\ du^2 + 2\,x\ du\ d\ w - 2\,w\ du\ d\ x - dx^2 - dy^2$$

```
> nptetrad ():
```

For the EL spacetime :

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 (covariant components)

$$\mathrm{NPl}(dn)$$

$$l_a = [x,\ 0,\ 0,\ 0]$$

$$\mathrm{NPn}(dn)$$

$$n_a = \left[\frac{1}{2}\,\frac{2\,\mathrm{f}(u)\,x\,\mathrm{g}(u)\,y + 2\,\mathrm{f}(u)\,x\,\mathrm{h}(u) + 2\,\mathrm{f}(u)\,x^3 + 2\,\mathrm{f}(u)\,x\,y^2 - w^2}{x},\ 1,\ -\frac{w}{x},\ 0\right]$$

$$\mathrm{NPm}(dn)$$

$$m_a = \left[0,\ 0,\ -\frac{1}{2}\sqrt{2},\ -\frac{1}{2}\,I\,\sqrt{2}\right]$$

$$\mathrm{NPmbar}(dn)$$

$$mbar_a = \left[0,\ 0,\ -\frac{1}{2}\sqrt{2},\ \frac{1}{2}\,I\,\sqrt{2}\right]$$

```
The null tetrad has been stored as e(bdn,dn).
> classify();

Basis/tetrad related object definitions
Last modified 5 February 1997
Lambda is zero.
WeylSp is zero.

Fixing RicciSp under {SL(2,C)} transformations.
RicciSp can be modified by the following transformations: {Boost,
lnswap}
RicciSp is in canonical form for the lnswap isotropy.
RicciSp is not in canonical form for Boost transformations.
Transforming RicciSp.
RicciSp is in standard form.
Dyad transformation stored as: dyfix[0]
The rotated frame has been named EL0.
Default metric is now EL0
Isotropies of the EL0 frame: {Spin, Null2D}
Fixed spinors: {RicciSp}

Level 0 frame fixing completed.
Remaining isotropy: {Spin, Null2D}

Found 0 independent functions.
Level 0 completed.
Continue? (1=yes [default], other=no):
```

```
> 1;


Fixing DPhi under {Spin, Null2D} transformations.
DPhi can be modified by the following transformations: {NullR, Spin}
DPhi is in canonical form for the Spin isotropy.
DPhi is not in canonical form for NullR transformations.
Transforming DPhi.
DPhi is in standard form.
Dyad transformation stored as: dyfix[1]
The rotated frame has been named EL01.
Default metric is now EL01
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp}

Level 1 frame fixing completed.
Remaining isotropy: {NullI}

Found 1 independent function.
Level 1 completed.
Continue? (1=yes [default], other=no):
> 1;


Fixing D2Phi under {NullI} transformations.
D2Phi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, D2Phi}


Fixing DalPhi under {NullI} transformations.
DalPhi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, D2Phi, DalPhi}

Level 2 frame fixing completed.
Remaining isotropy: {NullI}

Found 3 independent functions.
Level 2 completed.
Continue? (1=yes [default], other=no):
> 1;


Fixing D3Phi under {NullI} transformations.
D3Phi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, D2Phi, D3Phi, DalPhi}


Fixing DalDPhi under {NullI} transformations.
DalDPhi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, DalDPhi, D2Phi, D3Phi, DalPhi}

Level 3 frame fixing completed.
Remaining isotropy: {NullI}

Found 4 independent functions.
Level 3 completed.
Continue? (1=yes [default], other=no):
```

```
> 1;


Fixing D4Phi under {NullI} transformations.
D4Phi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, DalDPhi, D2Phi, D4Phi, D3Phi, DalPhi}


Fixing DalD2Phi under {NullI} transformations.
DalD2Phi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, DalDPhi, D2Phi, DalD2Phi, D4Phi,
D3Phi, DalPhi}


Fixing Dal2Phi under {NullI} transformations.
Dal2Phi is invariant under {NullI} transformations.
Isotropies of the EL01 frame: {NullI}
Fixed spinors: {DPhi, RicciSp, DalDPhi, D2Phi, DalD2Phi, Dal2Phi,
D4Phi, D3Phi, DalPhi}

Level 4 frame fixing completed.
Remaining isotropy: {NullI}

Found 4 independent functions.
Level 4 completed.
Continue? (1=yes [default], other=no):
> 1

1

EL01 has been fully classified.
Results have been stored as the EL01 spacetime.
The isotropy of the spacetime is: {NullI}
Independent functions are stored in the global variable:
grG_indep_fns_[EL01]
> idisplay ();
```

Independent components for the EL01 spacetime

$$\Re((D^3\,\Phi)_{5,5}) = \frac{25}{24}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))^3}{\mathrm{f}(u)^{9/2}\,x^{3/2}} + \frac{5}{12}\,\frac{\sqrt{2}\,w^3}{\mathrm{f}(u)^{3/2}\,x^{9/2}} - \frac{5}{8}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,w^2}{\mathrm{f}(u)^{5/2}\,x^{7/2}} + \frac{5}{6}\,\frac{I\sqrt{2}\,w\,\mathrm{g}(u)}{\sqrt{\mathrm{f}(u)}\,x^{5/2}}$$
$$+ \frac{5}{3}\,\frac{I\sqrt{2}\,w\,y}{\sqrt{\mathrm{f}(u)}\,x^{5/2}} - \frac{5}{6}\,\frac{\sqrt{2}\,w\,\mathrm{h}(u)}{\sqrt{\mathrm{f}(u)}\,x^{7/2}} + \frac{1}{4}\,\frac{\sqrt{2}\,w\,(\frac{\partial^2}{\partial u^2}\,\mathrm{f}(u))}{\mathrm{f}(u)^{5/2}\,x^{5/2}} - \frac{5}{6}\,\frac{\sqrt{2}\,w\,\mathrm{g}(u)\,y}{\sqrt{\mathrm{f}(u)}\,x^{7/2}} - \frac{5}{6}\,\frac{\sqrt{2}\,w\,y^2}{\sqrt{\mathrm{f}(u)}\,x^{7/2}}$$
$$- \frac{5}{12}\,\frac{I\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,\mathrm{g}(u)}{\mathrm{f}(u)^{3/2}\,x^{3/2}} - \frac{5}{6}\,\frac{I\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,y}{\mathrm{f}(u)^{3/2}\,x^{3/2}} + \frac{5}{12}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,\mathrm{h}(u)}{\mathrm{f}(u)^{3/2}\,x^{5/2}}$$
$$- \frac{5}{4}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,(\frac{\partial^2}{\partial u^2}\,\mathrm{f}(u))}{\mathrm{f}(u)^{7/2}\,x^{3/2}} + \frac{5}{12}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,\mathrm{g}(u)\,y}{\mathrm{f}(u)^{3/2}\,x^{5/2}} + \frac{5}{12}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,y^2}{\mathrm{f}(u)^{3/2}\,x^{5/2}}$$
$$- \frac{1}{4}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{g}(u))\,y}{\sqrt{\mathrm{f}(u)}\,x^{5/2}} + \frac{1}{4}\,\frac{\sqrt{2}\,(\frac{\partial^3}{\partial u^3}\,\mathrm{f}(u))}{\mathrm{f}(u)^{5/2}\,x^{3/2}} - \frac{1}{4}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{h}(u))}{\sqrt{\mathrm{f}(u)}\,x^{5/2}} + \frac{5}{6}\,\frac{\sqrt{2}\,w}{\sqrt{\mathrm{f}(u)}\,x^{3/2}} - \frac{5}{12}\,\frac{\sqrt{2}\,(\frac{\partial}{\partial u}\,\mathrm{f}(u))}{\mathrm{f}(u)^{3/2}\,\sqrt{x}}$$

$$\Re((D\,\Phi)_{2,3}) = \frac{\sqrt{2}}{x}$$

$$\begin{aligned}\Re((D^2\,\Phi)_{4,4}) &= -\frac{5}{9}\,\frac{(\frac{\partial}{\partial u}\,\mathrm{f}(u))^2}{\mathrm{f}(u)^3\,x} - \frac{5}{18}\,\frac{(\frac{\partial}{\partial u}\,\mathrm{f}(u))\,w}{\mathrm{f}(u)^2\,x^2} + \frac{5}{18}\,\frac{w^2}{\mathrm{f}(u)\,x^3} + \frac{1}{2}\,\frac{\frac{\partial^2}{\partial u^2}\,\mathrm{f}(u)}{\mathrm{f}(u)^2\,x} \\ &\quad - \frac{1}{2}\,\frac{\mathrm{g}(u)\,y}{x^2} - \frac{1}{2}\,\frac{\mathrm{h}(u)}{x^2} + \frac{5}{2} - \frac{1}{2}\,\frac{y^2}{x^2}\end{aligned}$$

$$\Re((D^2\,\Phi)_{3,4}) = \frac{1}{3}\,\frac{\frac{\partial}{\partial u}\,\mathrm{f}(u)}{\mathrm{f}(u)^{3/2}\,x^{3/2}} - \frac{2}{3}\,\frac{w}{\sqrt{\mathrm{f}(u)}\,x^{5/2}}$$

D

Metric components for the Stark-Piran and Bondi systems

The following tables list the values of the Stark-Piran metric components in terms of the variables used in Stark and Piran (1987) and those used in d'Inverno and Vickers (1996). For convenience of comparisons at the interface, the Bondi metric components of d'Inverno and Vickers (1997) are also listed.

$$\text{Stark \& Piran} \longleftrightarrow \text{Rd'I \& JAV}$$

$$N \longleftrightarrow \alpha,$$

$$N_i \longleftrightarrow \beta_\mu,$$

$$h_{ij} \longleftrightarrow \gamma_{\mu\nu},$$

$$B^2 = 1 + \eta \sin^2\theta \longleftrightarrow \gamma_{33}/r^2 \sin^2\theta,$$

$$\xi \longleftrightarrow \gamma_{23}/r^2 B^2 \sin^3\theta,$$

$$A^2 \longleftrightarrow \gamma_{11},$$

$$\beta^r = N^r/r \longleftrightarrow -\beta^1/r,$$

$$G = N^\theta/\sin\theta \longleftrightarrow -\beta^2/\sin\theta,$$

$$N^\phi \longleftrightarrow -\beta^3.$$

Metric components:

	Rd'I & JAV	Stark & Piran	Bondi
g_{00}	$-(\alpha^2-\beta^\mu\beta_\mu)$	$-N^2+r^2[A^2(\beta^r)^2+\sin^2\theta([B^{-2}$ $+B^2\xi^2\sin^4\theta]G^2+2B^2\xi\sin^2\theta GN^\phi$ $+B^2(N^\phi)^2)]$	$-(r^{-1}Ve^{2\beta}-r^2U^2e^{2\gamma}\cosh 2\delta-$ $2r^2UW\sinh 2\delta-r^2W^2e^{-2\gamma}\cosh 2\delta)$
g_{01}	$\gamma_{11}\beta^1$	$-rA^2\beta^r$	$-e^{2\beta}$
g_{02}	$\gamma_{2A}\beta^A$	$-r^2\sin\theta([B^{-2}+B^2\xi^2\sin^4\theta]G$ $+B^2\xi\sin^2\theta N^\phi)$	$-r^2(Ue^{2\gamma}\cosh 2\delta+W\sinh 2\delta)$
g_{03}	$\gamma_{3A}\beta^A$	$-r^2B^2\sin^2\theta(\xi\sin^2\theta G+N^\phi)$	$-r^2\sin\theta(U\sinh 2\delta+We^{-2\gamma}\cosh 2\delta)$
g_{11}	γ_{11}	A^2	0
g_{1A}	0	0	0
g_{22}	γ_{22}	$r^2(B^{-2}+B^2\xi^2\sin^4\theta)$	$r^2e^{2\gamma}\cosh 2\delta$
g_{23}	γ_{23}	$r^2B^2\xi\sin^3\theta$	$r^2\sin\theta\sinh 2\delta$
g_{33}	γ_{33}	$r^2B^2\sin^2\theta$	$r^2e^{-2\gamma}\sin^2\theta\cosh 2\delta$
g^{00}	$-1/\alpha^2$	$-1/N^2$	0
g^{01}	β^1/α^2	$-r\beta^r/N^2$	$-e^{-2\beta}$
g^{02}	β^2/α^2	$-G\sin\theta/N^2$	0
g^{03}	β^3/α^2	$-N^\phi/N^2$	0
g^{11}	$\frac{1}{\gamma_{11}}-\frac{(\beta^1)^2}{\alpha^2}$	$\frac{1}{A^2}-\frac{r^2(\beta^r)^2}{N^2}$	$Ve^{-2\beta}/r$
g^{12}	$\frac{-\beta^1\beta^2}{\alpha^2}$	$\frac{-r\sin\theta\beta^rG}{N^2}$	$-Ue^{-2\beta}$
g^{13}	$\frac{-\beta^1\beta^3}{\alpha^2}$	$\frac{-r\beta^rN^\phi}{N^2}$	$-We^{-2\beta}/\sin\theta$
g^{22}	$\frac{\gamma_{33}}{r^4\sin^2\theta}-\frac{(\beta^2)^2}{\alpha^2}$	$\frac{B^2}{r^2}-\frac{\sin^2\theta G^2}{N^2}$	$e^{-2\gamma}\cosh 2\delta/r^2$
g^{23}	$\frac{-\gamma_{23}}{r^4\sin^2\theta}-\frac{\beta^2\beta^3}{\alpha^2}$	$\frac{-B^2\xi\sin\theta}{r^2}-\frac{\sin\theta GN^\phi}{N^2}$	$-\sinh 2\delta/r^2\sin\theta$
g^{33}	$\frac{\gamma_{22}}{r^4\sin^2\theta}-\frac{(\beta^3)^2}{\alpha^2}$	$\frac{(B^{-2}+B^2\xi^2\sin^4\theta)}{r^2\sin^2\theta}-\frac{(N^\phi)^2}{N^2}$	$e^{2\gamma}\cosh 2\delta/r^2\sin^2\theta$

Bibliography

Abrahams, A. M., C. B. Cook, S. L. Shapiro, and S. A. Teukolsky (1994). Solving einstein's equations for rotating space-times: Evolution of relativistic star clusters. *Phys. Rev. D49*, 5153.

Abrahams, A. M. and C. R. Evans (1992). Trapping a geon: Black hole formation by an imploding gravitational wave. *Phys. Rev. D46*, 4117.

Abrahams, A. M. and C. R. Evans (1993). Critical behavior and scaling in vacuum axisymmetric gravitational collapse. *Phys. Rev. Lett. 70*, 2980.

Alcubierre, M., G. Allen, B. Brügmann, E. Seidel, and W.-M. Suen (1999). Towards an understanding of the stability properties of the 3+1 evolution equations in general relativity. *LANL eprint archive*. gr-qc/9908079.

Alcubierre, M., B. Brügmann, D. Holz, E. Seidel, R. Takahashi, and J. Thornburg (1999). Symmetry without symmetry: Numerical simulation of axisymmetric systems using cartesian grids. *LANL eprint archive*. gr-qc/9908012.

Åman, J. E. (1986). Manual for CLASSI – classification programs for geometries in general relativity. Technical report, University of Stockholm.

Åman, J. E., R. A. d'Inverno, G. C. Joly, and M. A. H. MacCallum (1991). Quartic equations and classification of Riemann tensors in general relativiy. *Gen. Rel. Grav. 23*, 1023–1056.

Åman, J. E. and A. Karlhede (1980). A computer-aided clomplete classification of geometries in general relativity. First results. *Phys. Lett. A80*, 229–231.

Anninos, P., E. Seidel, R. Price, J. Pullin, and W.-M. Suen (1995). Headon collision of two black holes: Comparison of different approaches. *Phys. Rev. D52*, 4462–4480.

Arnowitt, R., S. Deser, and C. W. Misner (1962). The dynamics of general relativity. In L. Witten (Ed.), *Gravitation: an introduction to current research*, Chapter 7, pp. 227–265. John Wiley & Sons.

Bardeen, J. M. and T. Piran (1983). General relativisitic axisymmetric rotating systems: Coordinates and equations. *Phys. Rep. 96*, 205–250.

Bernstein, D., D. Hobill, E. Seidel, L. Smarr, and J. Towns (1994). Numerically generated axisymmetric black hole space-times: Numerical methods and code tests. *Phys. Rev. D50*, 5000.

Bishop, N. T. (1992). Some aspects of the characteristic initial value problem in numerical relativity. In R. d'Inverno (Ed.), *Approaches to Numerical Relativity*, Cambridge, pp. 20–33. Cambridge University Press.

Bishop, N. T., R. Gomez, P. R. Holvorcem, R. A. Matzner, P. Papadopoulos, and J. Winicour (1996). Cauchy-Characteristic Matching: A New Approach to Radiation Boundary Conditions. *Phys. Rev. Lett. 76*, 4303–4306.

Bishop, N. T., R. Gomez, L. Lehner, M. Maharaj, and J. Winicour (1997). High-powered gravitational news. *Phys. Rev. D56*, 6298–6309. gr-qc/9708065.

Bishop, N. T., R. Gómez, L. Lehner, and J. Winicour (1996). Cauchy-characteristic extraction in numerical relativity. *Phys. Rev. D54*, 6153–6165.

Bičák, J., P. Reilly, and J. Winicour (1988). Boost-Rotation Symmetric Gravitational Null Cone Data. *Gen. Rel. Grav. 20*, 171–181.

Bondi, H., M. G. J. van der Burg, and A. W. K. Metzner (1962). Gravitational waves in general relativity VII. waves from axi-symmetric isolated systems. *Proc. Roy. Soc. Lond. A269*, 21–52.

Brady, P. R., J. D. E. Creighton, and K. S. Thorne (1998). Computing the merger of black-hole binaries: The ibbh problem. *Phys. Rev. D58*, 061501. gr-qc/9804057.

Brandt, S. R. and E. Seidel (1995). The evolution of distorted rotating black holes. 1: Methods and tests. *Phys. Rev. D52*, 856–869. gr-qc/9412072.

Brans, C. H. (1965). Invariant approach to the geometry of spaces in general relativity. *J. Math. Phys. 6*, 95–102.

Carminati, J. and R. G. McLenaghan (1991). Algebraic invariants of the riemann tensor in a four-dimensional lorentzian space. *J. Math. Phys. 32*, 3135–3140.

Cartan, E. (1946). *Leçons sur la géométrie des espaces de Riemann*. Paris: Gauthier–Villars.

Choptuik, M. W. (1991). Consistency of finite-difference solutions of einstein's equations. *Phys. Rev. D44*, 3124–3135.

Choptuik, M. W. (1993). Universality and scaling in gravitational collapse of a massless scalar field. *Phys. Rev. Lett. 70*, 9–12.

Christodoulou, D. and S. Klainerman (1993). *The global nonlinear stability of Minkowski space*. Princeton, NJ: Princeton University Press.

Christoffel, E. B. (1869). über die Transformation der homogenen Differentialausdrücke und der Krümmung höherer Mannigfaltigkeiten. *J. Reine Angew. Math. 70*, 46.

Chruściel, P., M. A. H. MacCallum, and D. Singleton (1995). Gravitational waves in general relativity XIV. Bondi expansions and the 'polyhomogeneity' of $\mathscr{I}^+$. *Philos. Trans. R. Soc. London, Ser A350*, 113–137.

Clarke, C. J. S. and R. A. d'Inverno (1994). Combining Cauchy and characteristic numerical evolutions in curved coordinates. *Class. Quantum Grav. 11*, 1463–1468.

Clarke, C. J. S., R. A. d'Inverno, and J. A. Vickers (1995). Combining Cauchy and characteristic codes. I. The vacuum cylindrically symmetric problem. *Phys. Rev. D52*, 6863–6867.

Collins, J. M. (1991). The Karlhede classification of type N vacuum spacetimes. *Class. Quantum Grav. 8*, 1859–1869.

Collins, J. M., R. A. d'Inverno, and J. A. G. Vickers (1990). The Karlhede classification of type D spacetimes. *Class. Quantum Grav. 7*, 2005–2015.

Collins, J. M., R. A. d'Inverno, and J. A. G. Vickers (1991). Upper-bounds for the Karlhede classification of type D vacuum spacetimes. *Class. Quantum Grav. 8*, L215–L217.

Cormack, W. J. and G. S. Hall (1979). Riemannian Curvature and the Classification of the Riemann and Ricci Tensors in Space-Time. *Int. Jrn. Theor. Phys. 18*, 279–289.

Detweiler, S. (1987). Evolution of the constraint equations in general relativity. *Phys. Rev. D35*, 1095–1099.

d'Inverno, R. A. (1995). Polar slicing in axially symmetric systems. *Class. Quantum Grav. 12*, L75–L80.

d'Inverno, R. A. and R. A. Russell-Clark (1971). CLAM – its function, structure and implementation. *Computer Journal 17*, 229.

d'Inverno, R. A. and J. Stachel (1978). Conformal two-structure as the gravitational degrees of freedom in general relativity. *J. Math. Phys. 19*, 2447–2460.

d'Inverno, R. A. and J. A. Vickers (1996). Combining Cauchy and characteristic codes. III. The interface problem in axial symmetry. *Phys. Rev. D54*, 4919–4928.

d'Inverno, R. A. and J. A. Vickers (1997). Combining Cauchy and characteristic codes. IV. The characteristic field equations in axial symmetry. *Phys. Rev. D56*, 772–784.

Dubal, M. R., R. A. d'Inverno, and C. J. S. Clarke (1995). Combining Cauchy and characteristic codes. II. The interface problem for vacuum cylindrical symmetry. *Phys. Rev. D52*, 6868–6881.

Evans, C. (1986). An approach for calculating axisymmetric gravitational collapse. In J. Centrella (Ed.), *Dynamical Spacetimes and Numerical Relativity*, pp. 3–39. Cambridge: Cambridge University Press.

Fitch, J. P. (1971). *An algebraic manipulator*. Ph. D. thesis, Cambridge University.

Friedrich, H. (1981). On the regular and asymptotic characteristic initial value problem for Einstein's vacuum field equations. *Proc. Roy. Soc. Lond. A375*, 169.

Friedrich, H. and A. D. Rendall (2000). The Cauchy problem for Einstein equations. *LANL eprint archive*. gr-qc/0002074.

Friedrich, H. and J. M. Stewart (1983). Characteristic initial data and wavefront singularities in general relativity. *Proc. Roy. Soc. Lond. A385*, 345–371.

Frittelli, S. (1997). Note on the propagation of the constraints in standard 3+1 general relativity. *Phys. Rev. D55*, 5992–5996.

Geroch, R. P. (1977). Asymptotic structure of space-time. In F. P. Esposito and L. P. Witten (Eds.), *Asymptotic Structure of Space-Time*. New York: Plenum.

Gómez, R., R. Isaacson, and J. Winicour (1992). Evolution of scalar fields from characteristic data. *jcompphys 98*, 11.

Gómez, R., P. Laguna, P. Papadopoulos, and J. Winicour (1996). Cauchy-characteristic evolution of Einstein-Klein-Gordon systems. *Phys. Rev. D54*, 4719–4727.

Gómez, R., L. Lehner, R. L. Marsa, and J. Winicour (1998). Moving black holes in 3d. *Phys. Rev. D57*, 4778–4788.

Gómez, R., R. L. Marsa, and J. Winicour (1997). Black hole excision and matching. *Phys. Rev. D56*, 6310–6319.

Gómez, R., P. Papadopoulos, and J. Winicour (1994). Null cone evolution of axisymmetric vacuum space-times. *J. Math. Phys. 35*, 4184–4204.

Gómez, R. e. a. (1998). Stable characteristic evolution of generic three-dimensional single-black-hole spacetimes. *Phys. Rev. Lett. 80*, 3915–3918.

Hawking, S. W. and G. F. R. Ellis (1973). *The large scale structure of space-time.* Cambridge: Cambridge University Press.

Hon, E. (1975). Application of REDUCE system to some problems in general relativity. Master's thesis, University of Waterloo.

Husa, S. and J. Winicour (1999). Asymmetric merger of black holes. *Phys. Rev. D60*, 4019–4032.

Joly, G. C. and M. A. H. MacCallum (1990). Computer-aided classification of the Ricci tensor in general relativity. *Class. Quantum Grav. 7*, 541–556.

Karlhede, A. (1979). A review of the equivalence problem. Technical report, University of Stockholm.

Karlhede, A. (1980). A Review of the Geometrical Equivalence of Metrics in General Relativity. *Gen. Rel. Grav. 12*, 693–707.

Kinnersley, W. (1969). Type D Vacuum Metrics. *J. Math. Phys. 10*, 1195.

Koutras, A. (1992). A spacetime for which the Karlhede invariant classification requires the 4th covariant derivative of the Riemann tensor. *Class. Quantum Grav. 9*, L143–L145.

Kramer, D., H. Stephani, H. E., and M. MacCallum (1980). *Exact Solutions of Einstein's Field Equations.* Cambridge: Cambridge University Press.

Kreiss, H. and J. Oliger (1973). Methods for the approximate solutions of time dependent problems. Technical report, Global Atmospheric Research Program.

Krivan, W., P. Laguna, P. Papadopoulos, and N. Andersson (1997). Dynamics of perturbations of rotating black holes. *Phys. Rev. D56*, 3395–3404. gr-qc/9702048.

Letniowski, F. W. and R. G. McLenaghan (1988). An improved algorithm for quartic equation classification and Petrov classification. *Gen. Rel. Grav. 20*, 463–484.

MacCallum, M. A. H. and J. E. Åman (1986). Algebraically independent nth derivatives of the riemann curvature spinor in a general spacetime. *Class. Quantum Grav. 3*, 1133–1141.

Machado Ramos, M. P. and J. A. G. Vickers (1996). Invariant differential operators and the Karlhede classification of type N vacuum solutions. *Class. Quantum Grav. 13*, 1589–1600.

Musgrave, P. and K. Lake (1996). Junctions and thin shells in general relativity using computer algebra: I. The Darmois-Israel formalism. *Class. Quantum Grav. 13*, 1885–1899.

Musgrave, P. and K. Lake (1997). Junctions and thin shells in general relativity using computer algebra: I. The null formalism. *Class. Quantum Grav. 14*, 1285–1294.

Nakamura, T. and T. Sato (1981). General relativisitic collapse of rotating stars. *Phys. Lett. A86*, 318–320.

Newman, E. T. and R. Penrose (1962). An approach to gravitational radiation by a method of spin coefficients. *J. Math. Phys. 3*, 896–902. (Errata 4:998 (1963)).

Penrose, R. (1963). Asymptotic properties of fields and spacetimes. *Phys. Rev. Lett. 10*, 66–68.

Penrose, R. and W. Rindler (1984). *Spinors and space-time*, Volume 1. Cambridge: Cambridge University Press.

Penrose, R. and W. Rindler (1986). *Spinors and space-time*, Volume 2. Cambridge: Cambridge University Press.

Petrov, A. Z. (1954). Classification of spaces defined by gravitational fields. *Uch. Zap. Kazan Gos. Univ. 114*, 55–69. English translation: Tran. No. 29, Jet Propulsion Lab. Cal. Inst. Tech. Pasadena, 1963.

Pirani, F. A. E. (1965). Introduction to gravitational radiation theory. In A. Trautman, F. A. E. Pirani, and H. Bondi (Eds.), *Lectures on general relativity and gravitation: Brandeis summer institute in theoretical physics*, Volume 1, pp. 251–369. Englewood Cliffs, NJ: Prentice Hall, Inc.

Pollney, D. (1996). Identities among sp-invariants. Queens' University, Kingston.

Pollney, D., P. Musgrave, K. Santosuosso, and K. Lake (1996). Algorithms for computer algebra calculations in spacetime: I The calculation of curvature. *Class. Quantum Grav. 13*, 2289–2309.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1986). *Numerical Recipes: The art of Scientific Computing.* Cambridge: Cambridge University Press.

Reula, O. A. (1998). Hypberbolic methods for einstein's equations. *Living Reviews 1*, 1998–3.

Sachs, R. K. (1962). Gravitational waves in general relativity VIII. waves in asymptotically flat space-time. *Proc. Roy. Soc. Lond. A270*, 103–126.

Scheel, M. A., T. W. Baumgarte, G. B. Cook, S. L. Shapiro, and S. A. Teukolsky (1998). Treating instabilities in a hyperbolic formulation of einstein's equations. *Phys. Rev. D58*, 044020.

Schmidt, H.-J. (1994). Why do all the curvature invariants of a gravitational wave vanish? In G. A. Sardanashvily (Ed.), *New frontiers in gravitation*, pp. 337–344. Palm Harbour, FL: Hadronic Press. gr-qc/9404037.

Schmidt, H.-J. (1998). Consequences of the noncompactness of the Lorentz group. *Int. J. Theor. Phys 37*, 691–696. gr-qc/9512007.

Segré, C. (1884). Sulla teoria e sulla classificazione delle omografie in uno spazio lineare ad un numero qualunque di dimensioni. *Memoria della R. Accad. Lincei serie 3a XIX*, 127–148.

Seixas, W. (1991). Extensions to the computer-aided classification of the Ricci tensor. *Class. Quantum Grav. 8*, 1577–1585.

Sjödin, K. R. P., U. Sperhake, and J. A. Vickers (2000). Dynamic cosmic strings. *LANL eprint archive*. gr-qc/0002096.

Skea, J. (1996). Standard forms for the ricci spinor in CLASSI. Private communication.

Skea, J. E. F. (1997a). The invariant classification of conformally flat pure radiation spacetimes. *Class. Quantum Grav. 14*, 2393–2404.

Skea, J. E. F. (1997b). Online invariant classification database. WWW site: `http://edradour.symbcomp.uerj.br/`.

Skea, J. E. F. (1997c). Type N spacetimes whose invariant classifications require the fourth covariant derivative of the Riemann tensor. *Class. Quantum Grav. 14*, 2947–2950.

Smarr, L. and J. W. York (1978). Kinematical conditions in the construction of spacetime. *Phys. Rev. D15*, 2529–2551.

Stark, R. F. and T. Piran (1987). A general relativistic code for rotating axisymmetric configurations and gravitational radiation: Numerical methods and tests. *Comput. Phys. Rep. 5*, 221–264.

Stewart, J. (1990). *Advanced general relativity.* Cambridge: Cambridge University Press.

Stewart, J. M. and H. Friedrich (1982). Numerical Relativity I: The characteristic initial value problem. *Proc. Roy. Soc. Lond. A384*, 427–454.

Tariq, N. and B. O. J. Tupper (1975). A class of algebraically general solutions of the einstein-maxwell equations for non-null electromagnetic fields. *grg 6*, 345.

Teukolsky, S. A. (1982). Linearized quadrupole waves in general relativity and the motion of test particles. *Phys. Rev. D26*, 745.

Thomas, T. Y. (1934). *The Differential Invariants of Generalized Spaces.* London: Cambridge University Press.

van Stockum, W. J. (1937). The gravitational field of a distribution of particles rotating about an axis of symmetry. *Proc. Roy. Soc. Edinburgh A57*, 135.

Wald, R. M. (1984). *General Relativity.* Chicago: University of Chicago Press.

Wils, P. (1989). Homogeneous and conformally ricci flat pure radiation fields. *Class. Quantum Grav. 6*, 1243–1251.

Winicour, J. (1985). Logarithmic asymptotic flatness. *Found. Phys. 15*, 605–615.

Winicour, J. (1998). Characteristic Evolution and Matching. *Living Reviews 1*, 1998–5.

Winicour, J. (1999). The characteristic treatment of black holes. *LANL eprint archive*. gr-qc/9911106.

York, J. W. (1979). Kinematics and dynamics of general relativity. In L. Smarr (Ed.), *Sources of gravitational radiation*, pp. 175–201. Cambridge: Cambridge University Press.