UNIVERSITY OF Southampton

University of Southampton Research Repository
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

http://eprints.soton.ac.uk

University of Southampton
Faculty of Engineering, Science and Mathematics
School of Mathematics

Well-posed formulations and stable finite differencing schemes for numerical relativity

by

Ian Hinder

Submitted for the degree of Doctor of Philosophy

September 2005

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING, SCIENCE AND MATHEMATICS

SCHOOL OF MATHEMATICS

DOCTOR OF PHILOSOPHY

WELL-POSED FORMULATIONS AND STABLE FINITE DIFFERENCING SCHEMES FOR NUMERICAL RELATIVITY

BY IAN HINDER

This work concerns the evolution equations of general relativity; their mathematical properties at the continuum level, and the properties of the finite difference schemes used to approximate their solution in numerical simulations.

Stability results for finite difference approximations of partial differential equations which are first order in time and first order in space are well-known. However, systems which are first order in time and second order in space have been more successful in the field of numerical relativity than fully first order systems. For example, binary black hole simulations are accurate for much longer times. Hence, a greater understanding of the stability properties of these systems is desirable. An example of such a system is the NOR (Nagy, Ortiz and Reula) [47] formulation of general relativity. We present a proof of the stability of a finite difference approximation of the linearized NOR evolution system. The new tools used to prove stability for second order in space systems are described, along with the simple example of the wave equation.

In order to implement and compare different formulations of the Einstein equations in numerical simulations, the equations must be expanded from abstract tensor relations into components, discretized, and entered into a computer. This process is aided enormously by the use of *automated code generation*. I present the *Kranc* software package which we have written to perform these tasks.

It is expected, by analogy with the wave equation, that numerical simulations of systems which are first order in time and second order in space will be more accurate than those of fully first order systems. We present a quantitative comparison of the accuracy of formulations of the fully nonlinear Einstein equations and determine that for linearized gravitational waves, this prediction is verified. However, the same cannot be said for other test cases, and it is concluded that certain problems with the second order in space formulations make them behave worse than fully first order formulations in these test cases.

Contents

List of Figures

List of Tables

Declaration of authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of a university or other institute of higher learning, except where due acknowledgement is made in the text.

I also declare that the intellectual content of this thesis is the product of my own work. Parts of this work have been published before submission as [37] and [20].

Ian Hinder
19th September, 2005

Acknowledgements

I would firstly like to thank my supervisor, Dr. Carsten Gundlach, and my adviser, Prof. James Vickers, for their expertise and guidance throughout my studies. Always available and willing to answer questions and give advice, they have been a tremendous help to me.

I would like to thank my collaborators Dr. Sascha Husa and Dr. Christiane Lechner at the Albert Einstein Institute, Germany, for their hard work and dedication in co-writing our code generation software. My thanks go as well to Dr. Gioel Calabrese in Southampton who not only worked with me on the stability theory contained in this work, but was also a constant source of helpful ideas and insight.

For help with the Cactus software, I am indebted to (among others) Dr. Ian Hawke, Dr. Denis Pollney, Dr. Erik Schnetter and Dr. Jonathan Thornburg. Ian Hawke has also kindly proof-read a late draft of this manuscript.

Finally, I thank my parents, Alun and Carole Hinder, for their support throughout my education and continual encouragement to pursue my academic interests.

Notation and conventions

Conventions

Quantity	Symbol
Extrinsic curvature	$K_{ab} \equiv - \perp \nabla_a n_b$
Riemann tensor	$R^e{}_{fgh}\omega_e = \nabla_h \nabla_g \omega_f - \nabla_g \nabla_h \omega_f$
Timelike vector	$g_{ab} X^a X^b < 0$

Notation

Quantity	Symbol
Imaginary constant	$i = \sqrt{-1}$
Matrix norm	$\|A\|$
Hermitian conjugate	A^*
Continuum function	$u(t, x)$
Semidiscrete function	$v_j(t)$
Grid function	v_j^n
Semidiscrete symbol	$\hat{P}$
Fully discrete symbol	$\hat{Q}$
Grid spacing	h
Time step	k
Courant factor	$\lambda = k/h$
Number of grid points	N
Number of spatial dimensions	d
Four-metric	g_{ab}
Three-metric	γ_{ij}

Tensor index conventions

The abstract index convention of [49] and [64] will be used; expressions with Latin indices $(a, b, c, \ldots)$ denote abstract tensor indices. Expressions with Greek indices $(\mu, \nu, \rho, \ldots)$ denote component expressions. In the sections concerning numerical analysis, Latin indices will be used on grid functions to denote components of the tensors they represent.

Chapter 1

Introduction

1.1 Numerical Relativity

The field of numerical relativity is concerned with obtaining approximate solutions to Einstein's equations by making use of numerical analysis and computer simulations (for general reviews, see [43, 2]) . The major application is to astrophysics, where the physics of highly relativistic bodies is poorly understood. There is an extensive effort underway to detect gravitational waves from highly dynamical events. One example of such an event is the orbit and merger of two black holes (see review in [12]), which is expected to provide a strong gravitational wave signal [48]. Gravitational waves are so weak that a technique called *matched filtering* is required to detect them . This needs a *template* signal to be provided, and this is compared with the detector data to give a probability that such an event occurred (e.g. [24]). Numerical relativity simulations can be used to provide such templates.

The early stages of the orbit can be modelled using so-called *post Newtonian* methods [14, 25] which involve using the first few terms in the expansion of the GR solution about the Newtonian solution in powers of v/c (where v is some characteristic velocity for the system). This technique applies to weak-field situations. After the merger, the resulting black hole is approximately the Kerr solution, and Einstein's equations can be linearized about this solution and a mode analysis of the perturbations can be performed to extract the wave signal [52, 10, 9]. However, between the early stages of the orbit and the aftermath of the merger, neither post-Newtonian nor black hole perturbation theory calculations are good approximations to the physics. During this period, solutions to the fully nonlinear

equations are required.

In numerical relativity, one provides suitable initial data for the metric and stress-energy tensor (for example, two orbiting black holes) at a given time [23], and uses the Einstein equations to provide a solution at later times, from which the gravitational wave signal can be extracted. The work of [22, 18] shows that such a solution will exist (at least for a finite time) and will be unique. Whilst apparently straightforward, this procedure is fraught with difficulties.

There are two requirements on the initial data. Firstly, it must be a physically realistic model for the astrophysical event under study. Secondly, the data must satisfy the *constraint equations* of general relativity. One must choose which spacelike surface of spacetime to use as the initial slice, and what coordinates to use on that slice. Providing physically realistic initial data is not straightforward. For example, to provide initial data for two orbiting black holes, it is not possible to simply add together two Kerr solutions, as the resulting spacetime will not satisfy Einstein's equations, due to their nonlinearity.

Once initial data in some coordinate system has been provided, it is necessary to specify how the coordinates map to different points in the evolved spacetime. Different choices will have different properties. For example, some choices may lead to coordinate singularities after a finite time.

There is more than one way to write the Einstein equations as a time evolution problem. In addition to the choice of gauge, the evolution equations can be modified by using the constraint equations. This leads to a multitude of *formulations* of the Einstein equations, each of which are the same when the constraints are satisfied, but which have different solutions when they are not. Some of these formulations are *hyperbolic*, which means that the speeds of propagation of features in the solution are finite, and the initial value problem is what is called *well-posed* (for a review of hyperbolic formulations, see [53]).

1.2 Stability of numerical implementations of well-posed formulations

The Einstein equations consist of a set of ten coupled nonlinear second order partial differential equations. In order to solve the initial value (time evolution) problem, the fully second order system is usually written as a first order in time system, modelled on the

Arnowitt-Deser-Misner (ADM) decomposition [7, 65]. Such systems can be evolved directly [57, 11], or a further reduction from second to first spatial order can be performed (see, for example, [29, 36, 6, 38]).

An important issue is the mathematical *well-posedness* of a formulation of the Einstein equations. A problem is well-posed if a solution exists, is unique, and depends continuously on any prescribed data. Local (for finite time) existence and uniqueness have been shown in [22, 18]. The requirement of continuity, often called stability, implies that small fluctuations in the prescribed data should not lead to arbitrarily large fluctuations in the solution, a property essential for a physical theory to have predictive power, and also for numerical implementation. Chapter 4 in [34] and Chapter 2 in [39] give comprehensive treatments.

There are many techniques for obtaining a numerical solution to a system of PDEs. One approach is to consider only a finite grid of points, and to replace derivatives in the equations with differences between the values of the function at different points. The discretized equations are then solved. This approach will be used in this work, and is called the method of *finite differences*. We follow Chapter 5 of [34] for our treatment of this subject, and the notation used will be the same.

The solution of the resulting set of finite difference equations will be an approximation to the solution to the continuum problem. It is necessary that the error in the numerical solution at a time t tends to zero as the grid spacing is reduced. This is called *convergence* (for the finite difference schemes that we consider, the rate of convergence will be polynomial).

For linear systems, a necessary condition for convergence is *stability* of the discretized equations. This is the discrete analogue of well-posedness, and means that the norm of the discrete solution can be bounded in terms of its initial value, independently of the initial data.

Whereas the theory of Cauchy problems for fully first order systems of partial differential equations is understood, in terms of well-posedness at the continuum and the stability of finite difference approximations, the theory of second order in space hyperbolic systems is less well developed. The recent improvement in the understanding of second order in space formulations of Einstein's equations at the continuum [54, 47, 31, 32, 11] has not been matched by developments concerning finite difference approximations of such systems (see, however, [41, 58]). Given that these systems have fewer variables, fewer constraints, and typically smaller errors (see [41] and Section 7.6.1), it is desirable to better appreciate their properties.

For discretizations of linear systems that are first order in time and first order in space, it is possible to show stability for various common discretizations. However, there is very little, if anything, in the literature concerning proofs of numerical stability for discretizations of Einstein-type equations in first order in time and second order in space form. To our knowledge, the study of such systems has not been performed before.

I discuss progress in this area made in collaboration with Gioel Calabrese, and present a proof of stability for one particular formulation of Einstein's equations, known as NOR. When the equations are linearized about a Minkowski background in Cartesian coordinates, the standard discretization of these equations is conditionally stable (i.e. it is stable with a small enough Courant factor, k/h, where h and k are the space and time coordinate spacing of the numerical grid respectively). The proof is general and relies on a *discrete reduction to first order in Fourier space*, and a set of conditions are derived that can be applied to any second order scheme to determine whether or not the system is stable. Stability is defined with respect to a discrete norm that contains difference operators.

1.3 Automated code generation

In order to calculate solutions of the Einstein equations numerically, the particular equation system must be entered into a computer. A typical evolution system (the BSSN equations) consists of 18 coupled, nonlinear, partial differential equations. They are algebraically complicated to write down, even using abstract index notation. Before solving the equations, they must be expanded into components, which further complicates the problem. It takes a long time to write such a system of equations by hand in a traditional programming language such as C or Fortran. Further, it is easy to make mistakes, and debugging such a code is difficult. To address this problem, we decided to implement a method of taking a description of the abstract index initial value problem and converting it automatically into the C or Fortran code necessary to solve the equations numerically. We used Mathematica as the basis for our system, and we call the resulting package Kranc, for *KRanc Assembles Numerical Code*. This package has been used for all the numerical experiments in this work, and it was written in collaboration with a group at the Albert Einstein Institute. The Kranc output code is based on the Cactus [30, 5, 59, 19] problem solving infrastructure.

1.4 Formulation comparison

Given the large number of formulations of the Einstein equations that have been proposed, for example [7, 38, 57, 11, 29, 27, 54, 55, 47], it is desirable to compare the suitability of these formulations for numerical simulations. There is a project [3] underway between various numerical relativity groups around the world with the aim of doing this numerically. The project is called "Apples with Apples", as the emphasis is on providing *standardized* tests. Care must be taken when comparing different formulations to keep the specifics of the test the same, as a comparison between two numerical runs can be influenced by factors other than the formulation.

We use testbeds from this project to perform quantitative comparisons of the accuracy of numerical implementations of the NOR, BSSN and ST formulations. The tests are designed to model different parts of an astrophysical spacetime, including oscillatory gauge dynamics, linear gravitational waves, and the strong field region around a singularity. These testbeds are restricted to using periodic boundary conditions so that there are no complications arising from the presence of artificial boundaries. A later stage of the project will introduce tests with artificial boundaries. The exact solutions to the testbeds are one dimensional (i.e. they have plane symmetry) for simplicity.

We test the numerical convergence of these nonlinear systems about the exact solutions in the testbeds; for nonlinear systems stability is not defined and convergence is what is ultimately required. We also attempt to determine whether the improvement in accuracy exhibited by a finite difference approximation of the wave equation when written in second order in space form as opposed to fully first order form is reflected in the Einstein equations.

1.5 Thesis overview

In Chapter 2, the $3+1$ decomposition of Einstein's equations is presented; the equations are written in a form (the ADM equations) which is manifestly an initial value problem. The important notion of the *well-posedness* of such a problem is introduced in Chapter 3, and modifications to the ADM equations are described which make the problem well-posed. In Chapter 4, the technique of *finite differencing* for numerically solving partial differential equations is described, along with definitions of *stability* and *convergence* of the associated numerical schemes. Chapters 2–4 constitute review material.

In Chapter 5 we introduce the concept of a *discrete symmetrizer* and the techniques we have developed for showing stability of first order in time, second order in space finite difference schemes. The main result is a proof of the stability of the linearized NOR system. Chapter 6 describes our Kranc package for automated code generation, and in Chapter 7 we describe numerical experiments showing results for convergence and accuracy tests of several formulations about testbed solutions.

Chapter 2

3+1 decomposition of Einstein's equations

2.1 Motivation

Consider a spacetime $(\mathcal{M}, g_{ab})$ where $\mathcal{M}$ is a four dimensional manifold and g_{ab} is the spacetime metric on that manifold (see e.g. [35, 64] for an introduction to the theory of general relativity). Einstein's equations

$$G_{ab} \equiv {}^{(4)}R_{ab} - \frac{1}{2}{}^{(4)}R g_{ab} = \kappa {}^{(4)}T_{ab} \tag{2.1}$$

describe geometrically the behaviour of the curvature of a space-time and how this is related to its matter content. G_{ab} is the Einstein tensor associated with g_{ab}, ${}^{(4)}R_{ab}$ is the Ricci tensor, and ${}^{(4)}T_{ab}$ is the stress energy tensor representing the energy and matter content of the spacetime. The ${}^{(4)}$ indicates that the curvature tensors are those of the four dimensional spacetime, as we will be considering three dimensional quantities later. This work will be concerned with vacuum general relativity, hence the matter terms T_{ab} will be neglected. In this situation, (2.1) is equivalent to

$${}^{(4)}R_{ab} = 0 \tag{2.2}$$

Writing (2.2) in terms of partial derivatives associated with a coordinate basis, the structure of the equations becomes apparent:

$${}^{(4)}R_{\mu\nu} \equiv \frac{1}{2}g^{\sigma\rho}(g_{\sigma\nu,\mu\rho} + g_{\mu\rho,\sigma\nu} - g_{\sigma\rho,\mu\nu} - g_{\mu\nu,\sigma\rho}) + g^{\sigma\rho}(\Gamma^{m}{}_{\mu\rho}\Gamma_{m\sigma\nu} - \Gamma^{m}{}_{\mu\nu}\Gamma_{m\sigma\rho}) \tag{2.3}$$

$$\Gamma^{\mu}{}_{\nu\sigma} \equiv \frac{1}{2}g^{\mu\rho}(g_{\rho\nu,\sigma} + g_{\rho\sigma,\nu} - g_{j\sigma,\rho}) \tag{2.4}$$

Due to the symmetry of ${}^{(4)}R_{ij}$, this is a set of 10 equations. It forms a system of second order partial differential equations for the metric components g_{ab}. This is not manifestly an initial value problem. For an astrophysical simulation, initial data is provided at a time t and the solution to the Einstein equations is required at later times.

2.2 Foliations of spacetime

Given a four dimensional spacetime $(\mathcal{M}, g_{ab})$, the first step is to introduce a time coordinate. This is a slight loss of generality, as only manifolds of the form $\mathbb{R} \times \Sigma$, or particular patches of this form in a general manifold, can be described. It would have been possible to define a global one-form ω_a and this could have been used to generate a time function locally. We will be satisfied with a concept of time which is local. This can be provided by a scalar function t satisfying

$$g^{ab}\nabla_a t \nabla_b t < 0 \tag{2.5}$$

This is the requirement that surfaces of constant t (called *slices*) are spacelike. Each of these surfaces is the image of a three dimensional manifold Σ_t under an *embedding* $\phi_t : \Sigma_t \to \mathcal{M}$. See Chapter 2 of [35] for more details. The set of surfaces Σ_t is called a *foliation*, and each Σ_t is a *spacelike hypersurface*.

The *lapse* α describes the rate of change of proper time with coordinate time in a direction normal to the slices:

$$-\alpha^{-2} \equiv g^{ab}\nabla_a t \nabla_b t \tag{2.6}$$

2.3 Projections in and across the slices

The unit normal n^a to the slice is defined as

$$n^a \equiv -g^{ab}\alpha\nabla_b t \tag{2.7}$$

This satisfies $n^a n_a = -1$, and for any vector in the slice (i.e. s^a such that $s^a\nabla_a t = 0$), $n_a s^a = 0$. Such vectors are described as *spatial* in the context of this decomposition. The set of tensors which give zero when contracted with n^a are described as *spatial*, and these

tensors can be identified with tensors on Σ_t by using the map ϕ_t and the metric g_{ab}. Hence spatial tensors S can be considered to be objects defined on Σ_t, which means that they have only three dimensional degrees of freedom. The remaining degrees of freedom are constrained by the condition $S \cdot n = 0$.

The four dimensional metric g_{ab} induces a metric γ_{ab} on Σ_t via the relation

$$\gamma_{ab} \equiv g_{ab} + n_a n_b \tag{2.8}$$

This metric is spatial ($\gamma_{ab} n^a = 0$) and positive definite.

$\gamma^a{}_b$ is a *projection operator* and projects tensors into the slice. We use the notation

$$\perp T_{ab} \equiv \gamma^c{}_a \gamma^d{}_b T_{cd} \tag{2.9}$$

to represent a projected tensor, with the obvious generalization to contravariant and mixed tensors. Such tensors give zero when contracted with n^a. Indices on spatial tensors can be raised and lowered with either g_{ab} or γ_{ab}.

The covariant derivative ∇ on the spacetime can be used to induce a covariant derivative on the slice. For spatial tensors S (indices are suppressed here), this covariant derivative is defined as

$$D_a S \equiv \perp \nabla_a S \tag{2.10}$$

It can be shown that

$$D_a \gamma_{bc} = 0 \tag{2.11}$$

hence D is the metric connection of the induced metric γ_{ab}.

The *extrinsic curvature*, the tensor representing how the embedded slice curves with respect to the spacetime, is defined as

$$K_{ab} \equiv - \perp \nabla_a n_b \tag{2.12}$$

Whilst not obvious from the definition, this tensor is symmetric. This quantity can be written as

$$K_{ab} = -\frac{1}{2} \mathcal{L}_n \gamma_{ab} \tag{2.13}$$

The *acceleration* vector of the slice is defined as

$$a^a \equiv n^b \nabla_b n^a \tag{2.14}$$

This is also a spatial vector. Starting from $\nabla_{[a}\nabla_{b]}t = 0$, and using (2.7), then projecting with n^b and $\gamma^a{}_c$, we obtain $D_c \ln \alpha = a_c$.

2.4 Coordinate evolution: lapse and shift

In order to define the slice, a time function t has been introduced. $x^0 \equiv t$ will be used to denote the time coordinate, and spatial coordinates on a particular slice will be denoted x^i where $i = 1, 2, 3$. Once coordinates have been assigned to the initial slice, they need to be propagated to subsequent slices. The integral curves of a timelike vector field t^a are used for this purpose. Points on the same integral curve of t^a are defined to have the same spatial coordinates. These curves are parameterized using coordinate time t; hence

$$t^a \nabla_a t = 1 \tag{2.15}$$

The vector field t^a can be decomposed in terms of the unit normal to the slice and an arbitrary spatial vector β^a.

$$t^a = r n^a + \beta^a \tag{2.16}$$

The value of r is obtained from (2.15):

$$r = \alpha \tag{2.17}$$

hence

$$t^a = \alpha n^a + \beta^a \tag{2.18}$$

So the change in spatial coordinates from one slice to the next is completely determined by an arbitrarily prescribed spatial vector field β^a, which we call the *shift vector.*

In these adapted coordinates, the metric can be shown to have the following component forms:

$$g_{\mu\nu} = \begin{pmatrix} -\alpha^2 + \beta^k\beta_k & \beta_i \\ \beta_i & \gamma_{ij} \end{pmatrix} \qquad g^{\mu\nu} = \begin{pmatrix} -\alpha^{-2} & \beta^i/\alpha^2 \\ \beta^i/\alpha^2 & \gamma^{ij} - \beta^i\beta^j/\alpha^2 \end{pmatrix} \tag{2.19}$$

Spatial vectors have components:

$$v^\mu = \begin{pmatrix} 0 \\ v^i \end{pmatrix} \qquad v_\mu = \begin{pmatrix} \beta^k v_k \\ v_i \end{pmatrix} \tag{2.20}$$

and spatial tensors have components:

$$S_{\mu\nu} = \begin{pmatrix} \beta^k\beta^l S_{kl} & \beta^k S_{ki} \\ \beta^k S_{ki} & S_{ij} \end{pmatrix} \qquad S^{\mu\nu} = \begin{pmatrix} 0 & 0 \\ 0 & S^{ij} \end{pmatrix} \tag{2.21}$$

So spatial tensors are completely determined by their spatial part, and in a time evolution, we only need to consider evolution equations for the spatial components; the remaining components can be reconstructed if necessary from these. Since γ_{ab} and K_{ab} are spatial, they can be represented by their spatial parts and a specification of lapse and shift. So these tensors can be described by 3×3 symmetric matrices, and γ_{ij} and K_{ij} have six components each.

2.5 Projection of Einstein's equations; the ADM formulation

Einstein's equations (2.3) are to be rewritten as an initial value problem on the foliation introduced above. The three-metric describes the geometry of a given slice, and the extrinsic curvature, a first derivative of the metric in the direction normal to the slice, describes how the metric changes from one slice to the next.

γ_{ij} and K_{ij} will be provided on the initial slice, and these quantities will be evolved to later times using the Einstein equations. In order to do this, the vacuum Einstein equations are projected in directions tangent to the slice.

The starting point is the vacuum Einstein equations:

$$^{(4)}R_{ab} = 0 \tag{2.22}$$

In order to derive the time evolution equations, this equation is projected on all its spatial indices.

The relation between the Riemann tensor, $^{(3)}R^i{}_{jkl}$, on the slice and the projection of the

Riemann tensor, ${}^{(4)}R^{a}{}_{bcd}$, of the spacetime can be derived as follows. By definition, for a spatial vector Y^a,

$$ {}^{(3)}R^{a}{}_{bcd}Y^b = D_cD_dY^a - D_dD_cY^a \tag{2.23}$$

Using (2.10) to write D in terms of ∇ and γ_{ab}, replacing derivatives of n^a with K_{ab} using (2.12), and using the fact that $Y^a n_a = 0$, the *Gauss equation* is obtained:

$$ {}^{(3)}R^{a}{}_{bcd} = {}^{(4)}R^{e}{}_{fgh}\gamma^{a}{}_{e}\gamma^{f}{}_{b}\gamma^{g}{}_{c}\gamma^{h}{}_{d} - K^{a}{}_{c}K_{bd} + K^{a}{}_{d}K_{bd} \tag{2.24}$$

So the Riemann tensors on the slice and in the spacetime are related via the extrinsic curvature of the slice.

Contracting on a and c, the Ricci tensor of the slice is obtained

$$ {}^{(3)}R_{bd} = {}^{(4)}R_{fh}\gamma^{f}{}_{b}\gamma^{h}{}_{d} + {}^{(4)}R^{e}{}_{fgh}n^{g}n_{e}\gamma^{f}{}_{b}\gamma^{h}{}_{d} - KK_{bd} + K^{a}{}_{d}K_{ba} \tag{2.25}$$

The time-space-time-space projected Riemann tensor is obtained by starting from

$$ {}^{(4)}R^{e}{}_{fgh}n_e = \nabla_h\nabla_g n_f - \nabla_g\nabla_h n_f \tag{2.26}$$

and contracting with $n^g\gamma^{f}{}_{b}\gamma^{h}{}_{d}$. Again, derivatives of n^a are replaced with K_{ab} using (2.12), and $n^a\nabla_a K_{bc}$ is replaced using the Lie derivative:

$$ {}^{(4)}R^{e}{}_{fgh}n^{g}n_{e}\gamma^{f}{}_{b}\gamma^{h}{}_{d} = \mathcal{L}_n K_{bd} + D_d a_b + a_b a_d + K_{gb}K^{g}{}_{d} \tag{2.27}$$

Substituting this expression into (2.25), the spatially projected 4 dimensional Ricci tensor is obtained in terms of 3 dimensional quantities:

$$ {}^{(4)}R_{fh}\gamma^{f}{}_{b}\gamma^{h}{}_{d} = {}^{(3)}R_{bd} - D_d a_b - a_b a_d + KK_{bd} - K^{a}{}_{d}K_{ba} - \mathcal{L}_n K_{bd} \tag{2.28}$$

Projecting Einstein's equations in the spatial directions yields

$$ \gamma^{a}{}_{c}\gamma^{b}{}_{d}\,{}^{(4)}R_{ab} = 0 \tag{2.29}$$

Hence, in terms of the Lie derivative in the n^a direction,

$$ \mathcal{L}_n K_{bd} = R_{bd} - D_d a_b - a_b a_d + KK_{bd} - K^{a}{}_{d}K_{ba} \tag{2.30}$$

The Lie derivative with respect to n^a can be replaced with a Lie derivative with respect to t^a using (2.18). In combination with (2.13), the ADM [7] (Arnowitt, Deser and Misner) evolution equations for vacuum general relativity are

$$\mathcal{L}_t \gamma_{ab} = -2\alpha K_{ab} + \mathcal{L}_\beta \gamma_{ab} \tag{2.31}$$

$$\mathcal{L}_t K_{ab} = -D_a D_b \alpha + \alpha[{}^{(3)}R_{ab} - 2K_{ac}K^c{}_b + K_{ab}K^c{}_c] + \mathcal{L}_\beta K_{ab} \tag{2.32}$$

This derivation is based upon that presented in [65].

The Einstein equations have been projected in the space-space directions to obtain the ADM evolution equations. The projections in the time-time and time-space directions yield

$$H \equiv {}^{(3)}R + K^2 - K_{ab}K^{ab} = 0 \tag{2.33}$$

$$M^a \equiv D_b(K^{ab} - \gamma^{ab}K) = 0 \tag{2.34}$$

Whilst (2.31)–(2.32) refer to how the metric and extrinsic curvature change from one slice to the next (they are called *evolution equations*), (2.33)–(2.34) are conditions which must be satisfied on every slice (*constraint equations*). It can be shown that if these constraints are satisfied on one slice, then the evolution equations guarantee that they will be satisfied on subsequent slices.

Chapter 3

Well-posedness of the Cauchy problem for PDEs

When considering a system of partial differential equations in a certain set of variables, the *Cauchy problem* is the following: given values for the variables at an initial time for all spatial coordinates, obtain a solution to the equations at a later time. The Cauchy problem is called *well-posed* if a solution exists (at least for a finite time), is unique, and depends continuously on the initial data. In a physical theory, changing the initial conditions of a process should only change the outcome by an amount that can be controlled by making the change in the initial conditions smaller. This property should be reflected in the field equations of the theory. From the point of view of a numerical simulation, this property is essential, as at every time step a small error is introduced. If this could have an arbitrarily large effect on the solution, then there would be no guarantee of obtaining an approximation convergent to the exact solution.

By writing the Einstein equations in *harmonic coordinates* ($\nabla^a \nabla_a x^\mu = 0$), it has been shown [22] that the system has a well-posed Cauchy problem, as the equations are wave equations in these coordinates. However, these coordinates are not suitable for numerical simulations as they lead to the formation of coordinate singularities in finite time (however, it should be noted that a formulation of the Einstein equations using *generalized* harmonic coordinates has shown a large degree of success in tackling the binary black hole problem; see [50, 51]). The ADM formulation of the Einstein equations has been introduced already. The combined system of evolution and constraint equations forms a well-posed problem. However, numerical evolutions typically use only the evolution equations (this is called

free evolution), and the constraints are monitored to assess the accuracy of the numerical solution. The set of ADM *evolution* equations written in fully first order form has been shown [38] to have an ill-posed Cauchy problem. Since convergence of a numerical scheme relies on well-posedness of the continuum problem, free numerical evolutions of ADM should not converge.

However, the decomposition of the Einstein equations into evolution and constraint equations is not unique. Multiples of the constraints can be added to the evolution equations. This changes the nature of the free evolution problem, but the physical solutions (those satisfying the constraints) remain the same. It turns out (see e.g. [38]) that it is possible to make the free evolution problem well-posed by introducing auxiliary variables and adding multiples of the constraints to the evolution equations.

In this chapter, the definition of well-posedness of linear constant-coefficient partial differential equations is reviewed, as are algebraic conditions that can be used for testing the well-posedness of a given PDE. General systems are considered first, then the specialization to first order systems is made. The necessary framework for discussing second order in space systems like the Einstein equations is then introduced, and an outline of the procedure for generalizing the results to variable coefficient and nonlinear problems follows. A description of three well-posed formulations of the Einstein equations concludes the chapter; these are the formulations that will be compared in Chapter 7.

3.1 Spaces and norms

The set $L_2(\mathbb{R}^d)$ (L_2 for short) is the set of Lebesgue integrable functions defined on $\mathbb{R}^d$:

$$L_2(\mathbb{R}^d) = \left\{ u : \mathbb{R}^d \to \mathbb{C} : \int_{\mathbb{R}^d} |u(x)|^2 \, dx < \infty \right\} \tag{3.1}$$

The L_2 norm is defined by

$$||u|| \equiv \sqrt{\int_{\mathbb{R}^s} |u(x)|^2 \, dx} \tag{3.2}$$

It will be necessary to consider *vector valued* functions $u : \mathbb{R}^d \to \mathbb{C}^m$; the above definitions are extended by replacing $|u(x)|$ under the integral by the Euclidean norm (see Appendix A) on $\mathbb{C}^m$.

3.2 Constant coefficient Cauchy problems

In this work we will be dealing with initial value (or Cauchy) problems of the form

$$\frac{\partial}{\partial t}u(t,x) = P\left(\frac{\partial}{\partial x}\right)u(t,x) \tag{3.3}$$

$$u(0,x) = f(x) \tag{3.4}$$

in d spatial dimensions, where $x \in \mathbb{R}^d$, $u = (u^{(1)}, u^{(2)}, \ldots u^{(m)})^T$ and P is a linear, constant coefficient, differential operator of order p. We consider only the cases $p = 1$ and $p = 2$. Furthermore, we assume that the eigenvalues of the symbol of the differential operator, $\hat{P}(i\omega)$, which is obtained by replacing $\partial/\partial x_j$ in $P(\partial/\partial x)$ with $i\omega_j$, for $j = 1, 2, \ldots, d$, have real part uniformly bounded from below and above. We are thus excluding parabolic systems, but we are allowing for systems like the wave equation written as a first order in time, second order in space system. For simplicity we focus on solutions that are 2π-periodic in all spatial coordinate directions. Thus the initial data, $f(x)$, is chosen so that it satisfies this property.

Definition 3.2.1. *Problem (3.3)–(3.4) is well-posed with respect to a norm $\|\cdot\|$ if for all smooth periodic initial data f there is a unique smooth spatially periodic solution and there are constants α and K, independent of f, such that*

$$\|u(t,\cdot)\| \leq Ke^{\alpha t}\|f\| \tag{3.5}$$

(Definition 4.1.1 in [34])

Exponential growth must be allowed if one wants to treat problems with lower order terms. For first order hyperbolic systems the L_2 norm

$$\|w\|^2 = \int_0^{2\pi} \ldots \int_0^{2\pi} |w(x)|^2 dx_1 \ldots dx_d \tag{3.6}$$

is usually used in (3.5). We will see later that the second order systems we study in this work require the use of a different norm.

This definition of well-posedness is difficult to apply in practice, so there is a theorem in

Fourier space that is easier to use. Writing $u(t,x)$ as a Fourier series,

$$u(t,x) = (2\pi)^{-d/2} \sum_{\omega=-\infty}^{\infty} e^{i\omega^* x} \hat{u}(t,\omega) \tag{3.7}$$

problem (3.3)–(3.4) can be written as

$$\frac{\partial}{\partial t}\hat{u}(t,\omega) = \hat{P}(i\omega)\,\hat{u}(t,\omega) \tag{3.8}$$

$$\hat{u}(0,\omega) = \hat{f}(\omega) \tag{3.9}$$

with formal solution

$$\hat{u}(t,\omega) = e^{\hat{P}(i\omega)t}\hat{f}(\omega) \tag{3.10}$$

which in physical space gives the formal solution of (3.3)–(3.4) as

$$u(t,x) = (2\pi)^{-d/2} \sum_{\omega=-\infty}^{\infty} e^{i\omega^* x} e^{\hat{P}(i\omega)t}\hat{f}(\omega) \tag{3.11}$$

Theorem 3.2.1. *Well-posedness in the L_2 norm is equivalent to there being constants K, α such that, for all ω,*

$$|e^{\hat{P}(i\omega)t}| \leq Ke^{\alpha t} \tag{3.12}$$

where $|A| = \sup_{|u|=1}|Au|$ is the matrix (operator) norm of a matrix A (see Appendix A). (Theorem 4.5.1 in [34])

The main result for general systems is presented in the following theorem:

Theorem 3.2.2. *Well-posedness of the Cauchy problem in the L_2 norm is also equivalent to the existence of constants α, $K > 0$ and of Hermitian matrices $\hat{H}(\omega)$ satisfying, for every ω,*

$$\begin{aligned} &K^{-1}I \leq \hat{H}(\omega) \leq KI \\ &\hat{H}(\omega)\hat{P}(i\omega) + \hat{P}^*(i\omega)\hat{H}(\omega) \leq 2\alpha\hat{H}(\omega) \end{aligned} \tag{3.13}$$

where $\hat{P}^*$ represents the Hermitian conjugate of $\hat{P}$. (Theorem 4.5.8 in [34])

Inequalities for matrices are defined in Appendix A. The last inequality gives an energy estimate for each Fourier mode and the estimate in physical space, (3.5), follows from

Parseval's relation,

$$\|u(t,\cdot)\|^2 = \sum_{\omega} |\hat{u}(t,\omega)|^2 \tag{3.14}$$

As shown in Lemma 2.3.5 in [39], the existence of $\hat{H}(\omega)$ is not affected by the addition of a constant matrix to $\hat{P}(i\omega)$. Suppose $P(\partial/\partial x)$ is a constant coefficient operator, and $B \in \mathbb{C}^{m,m}$ is a constant matrix, and let $P_0(\partial/\partial x) = P(\partial/\partial x) + B$. The Cauchy problem is well-posed for $u_t = Pu$ if and only if it is well-posed for $u_t = P_0 u$. Therefore, undifferentiated terms on the right hand side of the equations can be ignored in the analysis of well-posedness.

If (3.3) is modified by adding a forcing (inhomogeneous) term to the right hand side,

$$\frac{\partial}{\partial t}u(t,x) = P\left(\frac{\partial}{\partial x}\right)u(t,x) + F(t,x) \tag{3.15}$$

then well-posedness of the homogeneous problem (3.3)–(3.4) implies well-posedness of the inhomogeneous problem, in the sense that it satisfies the modified estimate

$$\|u(t,\cdot)\| \leq K\left(e^{\alpha t}\|f\| + \varphi^*(\alpha,t)\max_{0\leq\tau\leq t}\|F(t,\cdot)\|\right) \tag{3.16}$$

where

$$\varphi^*(\alpha,t) = \begin{cases} \frac{1}{\alpha}(e^{\alpha t}-1), & \text{if } \alpha \neq 0 \\ t, & \text{if } \alpha = 0 \end{cases} \tag{3.17}$$

(Theorem 4.7.2 in [34]).

If (3.13) is satisfied with $\hat{H}\hat{P} + \hat{P}^*\hat{H} = 0$ then $\hat{H}$ is called a *symmetrizer*. Consider the time evolution of the quantity $E(t,\omega) \equiv \hat{u}^*(t,\omega)\hat{H}(\omega)\hat{u}(t,\omega)$:

$$\frac{d}{dt}\hat{u}^*\hat{H}\hat{u} = 2\mathrm{Re}\left[\hat{u}^*\hat{H}\frac{d}{dt}\hat{u}\right] = 2\mathrm{Re}\left[\hat{u}^*\hat{H}\hat{P}\hat{u}\right] = \hat{u}^*(\hat{H}\hat{P} + \hat{P}^*\hat{H})\hat{u} \tag{3.18}$$

So for each ω, the following statements are equivalent:

- $\frac{d}{dt}\hat{u}^*\hat{H}\hat{u} = 0$
- $\hat{H}\hat{P} + \hat{P}^*\hat{H} = 0$

So if $\hat{H}$ is a symmetrizer, E is a *conserved energy* of the system.

To construct $\hat{H}$ one can proceed as follows. Assume the existence of a matrix T such that $T^{-1}\hat{P}T = \Lambda$ is diagonal with imaginary elements. Then the quantity $\hat{u}^*\hat{H}\hat{u}$, where $\hat{H} = T^{-1*}DT^{-1}$ and D is a positive definite matrix which commutes with Λ, is conserved by the system $\partial_t\hat{u} = \hat{P}\hat{u}$. Defining the *characteristic variables* of $\hat{P}$ to be $\hat{w} \equiv T^{-1}\hat{u}$ (these are individually conserved: $\partial_t|\hat{w}_i|^2 = 0$), we see that to construct a conserved quantity one can take $\hat{w}^*D\hat{w}$. (For $D = I$ this corresponds to adding the squared absolute values of the characteristic variables.) For $\hat{H}$ to be a symmetrizer it remains to be established that $K^{-1}|\hat{u}|^2 \leq \hat{u}^*\hat{H}\hat{u} \leq K|\hat{u}|^2$.

3.3 First order hyperbolic systems

For $p = 1$, system (3.3) can be written as

$$\partial_t u(t,x) = A^i \partial_i u(t,x) + B \tag{3.19}$$

where A_i and B are constant matrices. The *symbol* of the principal part (that part not containing derivatives) of system (3.19) is $\hat{P} = i\omega_i A^i$. The system is said to be *strongly hyperbolic* if the corresponding Cauchy problem is well-posed in the L_2 norm (i.e. if $\hat{H}(\omega)$ exists). Strong hyperbolicity is equivalent to $\hat{P}$ being uniformly diagonalizable with imaginary eigenvalues (i.e. the matrix T which diagonalizes $\hat{P}$ satisfies $|T||T^{-1}| \leq C$ for C independent of ω). The system is said to be weakly hyperbolic if the eigenvalues of $\hat{P}$ are imaginary. If $\hat{H}(\omega) = I$, the system is said to be *symmetric hyperbolic*. If $\hat{H}(\omega) = H$ is independent of ω, then we say that the system is *symmetrizable hyperbolic*. Note that symmetrizable hyperbolic systems are often also called symmetric hyperbolic by some authors. For a symmetrizable hyperbolic system, the change of variables $\tilde{u} = H^{1/2}u$ brings the system into symmetric hyperbolic form.

We define the characteristic speeds in the direction ω_i to be the eigenvalues of $\hat{P}(i\omega)$ divided by $i\omega$. If $\hat{H} = H$ is independent of ω (i.e. symmetrizable hyperbolicity), and there are no lower order terms in (3.3) then there exists a conserved energy in the domain Ω.

$$\begin{aligned} E(t) &\equiv \int_\Omega u^*(t,x)Hu(t,x)\, d^dx \qquad (3.20)\\ \frac{dE}{dt} &= \int_\Omega u^*HA^i\partial_i u + \partial_i u^*A^{i*}Hu\, d^dx = \int_\Omega u^*HA^i\partial_i u + \partial_i u^*HA^i u\, d^dx \end{aligned}$$

$$= \int_\Omega \partial_i (u^* H A^i u)\, d^d x = \int_{\partial\Omega} u^* H A^i u n_i \, dS \tag{3.21}$$

where we have used $HA^i = A^{i*}H$ and the divergence theorem, and n_i is a unit vector normal to the boundary element dS. So the change of the energy in Ω is dependent only on the solution at the boundary $\partial\Omega$. In the case of periodic boundaries, or Cauchy problems where all fields fall off to zero sufficiently rapidly at infinity, the surface integral is zero. When artificial boundaries are considered, estimates of this form can sometimes be used for symmetrizable hyperbolic systems to prove well-posedness of the initial boundary value problem.

If H depends on ω (only strong hyperbolicity), the translation back to physical space is not so straightforward, and a strongly hyperbolic system satisfies the estimate $\|u(t,\cdot)\| \leq K\|u(0,\cdot)\|$ with a constant $K \geq 1$.

3.4 Second order systems

The ADM evolution equations (2.31)–(2.32) are first order in time and second order in space. A model for this system is the first order in time, second order in space, wave equation:

$$\partial_t \phi(t,x) = \pi(t,x) \tag{3.22}$$
$$\partial_t \pi(t,x) = \partial_x^2 \phi(t,x) \tag{3.23}$$

This is obtained from the fully second order wave equation $\phi_{tt}(t,x) = \phi_{xx}(t,x)$ by the introduction of the variable $\pi(t,x) \equiv \phi_t(t,x)$. This mimics the ADM equations with ϕ representing γ_{ij} and π representing K_{ij}. Consider the initial data

$$\phi_0(x) = \sin(\omega x) \tag{3.24}$$
$$\pi_0(x) = 0 \tag{3.25}$$

This generates the solution

$$\phi(t,x) = \sin(\omega x)\cos(\omega t) \tag{3.26}$$
$$\pi(t,x) = -\omega \sin(\omega x)\sin(\omega t) \tag{3.27}$$

with L_2 norm

$$||u(t)||^2 = \int_0^{2\pi} |\phi|^2 + |\Pi|^2 dx = \|u(0)\|^2 \left[\cos^2(\omega t) + \omega^2 \sin^2(\omega t)\right] \tag{3.28}$$

hence $\|u(t)\|$ can be made arbitrarily large in comparison with $\|u(0)\|$ by choosing large enough ω, contradicting well-posedness in L_2. We will see later that the system *is* well-posed in a norm that contains derivatives of ϕ.

3.5 Well-posedness of second order in space systems: first order reduction

It is possible for the Cauchy problem for a first order in time and second order in space system of equations to be ill-posed in the L_2 norm, but well-posed in a norm which contains additional derivatives. We analyse the well-posedness of the Cauchy problem for such systems by using the analytical tool of a *reduction to first order*. This will be done in Fourier space, so that the number of additional variables being introduced is minimized [40].

Consider system (3.3) with $p = 2$ and suppose that it can be written in the form

$$\partial_t \boldsymbol{u} = P\boldsymbol{u} \qquad \boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix} \qquad P = \begin{pmatrix} A^i\partial_i + B & C \\ D^{ij}\partial_i\partial_j + E^i\partial_i + F & G^i\partial_i + J \end{pmatrix} \tag{3.29}$$

where the evolved variables have been split into two types. The column vector u represents those that are differentiated twice (in space) and v represents those that are not. In P a sum over repeated indices is assumed. Not all second order in space systems can be written in this form (for example, $u_t = u_{xx}$). This form is general enough to include all the first order in time, second order in space systems that we have considered that can be reduced to first order in space. Fourier transforming this system, we obtain

$$\partial_t \hat{\boldsymbol{u}} = \hat{P}\hat{\boldsymbol{u}} \qquad \hat{\boldsymbol{u}} = \begin{pmatrix} \hat{u} \\ \hat{v} \end{pmatrix} \hat{P} = \begin{pmatrix} i\omega A^n + B & C \\ -\omega^2 D^{nn} + i\omega E^n + F & i\omega G^n + J \end{pmatrix} \tag{3.30}$$

where $M^n \equiv M^i n_i$ and $\omega_i \equiv |\omega| n_i$ and $\omega \equiv |\omega|$. We define the *second order principal symbol*

to be

$$\hat{P}' = \begin{pmatrix} i\omega A^n & C \\ -\omega^2 D^{nn} & i\omega G^n \end{pmatrix} \tag{3.31}$$

We now state the main new result of this subsection.

Theorem 3.5.1. *If there exists $\hat{H}(\omega) = \hat{H}^*(\omega)$ such that the* energy $\hat{\boldsymbol{u}}^* \hat{H} \hat{\boldsymbol{u}}$ *is conserved by the principal system*

$$\partial_t \hat{\boldsymbol{u}} = \hat{P}' \hat{\boldsymbol{u}} \tag{3.32}$$

and $\hat{H}$ satisfies

$$K^{-1} I_\omega \leq \hat{H} \leq K I_\omega, \quad I_\omega \equiv \begin{pmatrix} \omega^2 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.33}$$

where K is a positive scalar constant, then the solution of (3.29) satisfies the estimate

$$\begin{aligned} &\|\boldsymbol{u}(t,\cdot)\| \leq K e^{\alpha t} \|\boldsymbol{u}(0,\cdot)\| \\ &\|\boldsymbol{u}\|^2 \equiv \int |u|^2 + \sum_{i=1}^{d} |\partial_i u|^2 + |v|^2 d^d x \end{aligned} \tag{3.34}$$

and the problem is well-posed in this norm. ($\hat{H}\hat{P}' + \hat{P}'^* \hat{H} = 0$ is equivalent to the conservation of $\hat{\boldsymbol{u}}^* \hat{H} \hat{\boldsymbol{u}}$.)

Proof The proof proceeds via a pseudo-differential reduction to first order [47]. This involves the introduction of a new variable $\hat{w} = i\omega\hat{u}$. By taking a time derivative of this definition, we obtain the enlarged system in which the second derivative of $\hat{u}$ has been replaced with a first derivative of $\hat{w}$. We reduce the order of the system as much as possible so that any occurrence of $i\omega\hat{u}$ is replaced with $\hat{w}$. This particular first order reduction is

$$\partial_t \hat{\boldsymbol{u}}_R = \hat{P}_R \hat{\boldsymbol{u}}_R, \qquad \hat{\boldsymbol{u}}_R = \begin{pmatrix} \hat{u} \\ \hat{w} \\ \hat{v} \end{pmatrix} \tag{3.35}$$

$$\hat{P}_R = \begin{pmatrix} B & A^n & C \\ 0 & i\omega A^n + B & i\omega C \\ F & i\omega D^{nn} + E^n & i\omega G^n + J \end{pmatrix}$$

This system is equivalent to the second order system (3.30) only when the *auxiliary constraints*

$$\hat{C}(t,\omega) \equiv \hat{w}(t,\omega) - i\omega\hat{u}(t,\omega) = 0 \tag{3.36}$$

are satisfied. It can be shown that $\partial_t \hat{C} = B\hat{C}$ so if these constraints are satisfied initially, then they are satisfied for all time. They are said to be *propagated* by the first order evolution equations.

If this system admits a matrix $\hat{H}_R$ satisfying (3.13) then the solutions satisfy the estimates

$$|\hat{\boldsymbol{u}}_R(t,\omega)| \leq Ke^{\alpha t}|\hat{\boldsymbol{u}}_R(0,\omega)| \tag{3.37}$$

where $|\hat{\boldsymbol{u}}_R|^2 \equiv |\hat{u}|^2 + |\hat{w}|^2 + |\hat{v}|^2$, for arbitrary initial data and ω. Specifically, the estimate holds for solutions which satisfy the auxiliary constraints and therefore correspond to solutions of the second order system. The estimate in Fourier space,

$$\|\hat{\boldsymbol{u}}_R(t,\cdot)\|^2 = \sum_{\omega} \left(|\hat{u}|^2 + \omega^2|\hat{u}|^2 + |\hat{v}|^2\right) h^d \tag{3.38}$$

$$= \sum_{\omega} \left(|\hat{u}|^2 + \sum_{i=1}^{d} |i\omega_i\hat{u}|^2 + |\hat{v}|^2 \right) h^d \leq K^2 e^{2\alpha t}\|\hat{\boldsymbol{u}}_R(0,\cdot)\|^2 \tag{3.39}$$

implies by Parseval's relation the estimate in real space

$$\|\boldsymbol{u}(t,\cdot)\| \leq Ke^{\alpha t}\|\boldsymbol{u}(0,\cdot)\| \tag{3.40}$$

$$\|\boldsymbol{u}\|^2 \equiv \int |u|^2 + \sum_{i=1}^{d} |\partial_i u|^2 + |v|^2 d^d x$$

So the existence of $\hat{H}_R$ for a first order pseudo-differential reduction implies the well-posedness of the second order system with respect to a norm containing derivatives.

We have still to show that we can find an $\hat{H}_R$ for (3.35). Whether or not this is the case is independent of the *lower order* terms $\hat{P}_R$ contains. A calculation similar to Lemma 2.3.5 in [39] shows that if $\hat{P}(\omega)$ admits an $\hat{H}_R$, then so will $\hat{P}(\omega) + B(\omega)$, where $B(\omega)$ is any matrix which satisfies $|B| + |B^*| \leq C$ for C independent of ω. In other words, the terms that are not multiplied by $i\omega$ can be dropped from (3.35), giving the principal symbol of the first

order reduction

$$\hat{P}'_R = \begin{pmatrix} 0 & 0 & 0 \\ 0 & i\omega A^n & i\omega C \\ 0 & i\omega D^{nn} & i\omega G^n \end{pmatrix} \tag{3.41}$$

without affecting the well-posedness. The principal symbols of the second order system, (3.31), and the first order pseudo-differential reduction, (3.41), are related by

$$\hat{P}'_R = \begin{pmatrix} 0 & 0 \\ 0 & T\hat{P}'T^{-1} \end{pmatrix} \qquad T \equiv \begin{pmatrix} 0 & 1 \\ i\omega & 0 \end{pmatrix} \tag{3.42}$$

(Note that T^{-1} does not exist for $\omega = 0$. However, in this case, $\hat{P}'_R = 0$, and admits the identity as a symmetrizer.) By assumption, there exists $\hat{H}(\omega) = \hat{H}^*(\omega)$ such that $\hat{\boldsymbol{u}}^*\hat{H}\hat{\boldsymbol{u}}$ is conserved by the principal system $\partial_t\hat{\boldsymbol{u}} = \hat{P}'\hat{\boldsymbol{u}}$ and satisfies (3.33). This $\hat{H}$ satisfies $\hat{H}\hat{P}' + \hat{P}'^*\hat{H} = 0$, and it is straightforward to show that

$$\hat{H}_R \equiv \begin{pmatrix} 1 & 0 \\ 0 & T^{-1*}\hat{H}T^{-1} \end{pmatrix} \tag{3.43}$$

satisfies $\hat{H}_R = \hat{H}_R^*$ and $\hat{H}_R\hat{P}'_R + \hat{P}'^*_R\hat{H}_R = 0$. Further, by noting that $T^*T = I_\omega$, using (3.33) one can show that $\hat{H}_R$ satisfies $K^{-1}I \leq \hat{H}_R \leq KI$. Hence we have found a symmetrizer of $\hat{P}'_R$ and the result has been proved. It can also be shown that $\hat{P}'_R$ is diagonalizable with the same eigenvalues as $\hat{P}'$, plus as many zeroes as there are components of u. □

To construct $\hat{H}$ one can use the characteristic variables of $\hat{P}'$, as described at the end of Section 3.2. Note that this analysis did not require that the auxiliary constraint propagation problem be well-posed. These constraints are merely a tool for the analysis of the system. When evolving the second order system they are identically zero. An alternative to the pseudo-differential reduction method is to perform a fully differential reduction by introducing a new variable in physical space for each derivative (see, for example, [54, 11]).

3.6 Well-posedness of variable coefficient linear Cauchy problems

Consider a variable coefficient problem

$$\frac{\partial}{\partial t}u(t,x) = P\left(t,x,\frac{\partial}{\partial x}\right)u(t,x) \equiv \sum_{j=1}^{d} A_j(t,x)\frac{\partial u}{\partial x_j} \tag{3.44}$$

$$u(0,x) = f(x) \tag{3.45}$$

If one freezes the coefficients at a particular spacetime point (t_0, x_0) one obtains a constant coefficient problem. The *localization principle* is the statement that if all such frozen coefficient problems are strongly hyperbolic, then the variable coefficient problem is well-posed. It is not known if this principle holds, but it is known that if the frozen coefficient problems are strongly hyperbolic and the symmetrizer of the variable coefficient problem is smooth, then the variable coefficient problem will be strongly hyperbolic according to the following definition.

Definition 3.6.1. *In the following problem, equation (3.44) is called* strongly hyperbolic *if there exists a Hermitian matrix function*

$$H(t,x,\omega) \geq 0, \quad x \in \mathbb{R}^d, t \geq 0, \omega \in \mathbb{R}^d, |\omega| = 1, K^{-1}I \leq H \leq KI \tag{3.46}$$

which is C^∞ smooth in all its arguments, and satisfies

$$H(t,x,\omega)P(t,x,i\omega) + P^*(t,x,i\omega)H(t,x,\omega) = 0 \tag{3.47}$$

(Definition 2 in 2.7.2 in [39])

So for first order systems, the definition of strong hyperbolicity is extended to the variable coefficient case simply by requiring smoothness of $H(t,x,\omega)$. According to Theorem 2.7.1 in [39], this definition of strong hyperbolicity is sufficient for well-posedness, though it is not known whether the requirement for smoothness of the symmetrizer is necessary.

3.7 Well-posedness of nonlinear Cauchy problems

Consider nonlinear problems of the form

$$\frac{\partial}{\partial t}u(t,x) = P\left(t,x,u,\frac{\partial}{\partial x}\right)u(t,x)+F \equiv \sum_{j=1}^{d} A_j(t,x,u)\frac{\partial u}{\partial x_j}+F(t,x) \tag{3.48}$$

$$u(0,x) = f(x) \tag{3.49}$$

Such a problem is called *quasi-linear* as it is linear in the highest derivatives.

Definition 3.7.1. *The system (3.48)–(3.49) is called* symmetric hyperbolic *if for all x, t and u, the matrices $A_j(t,x,u)$ are Hermitian (Definition 4.9.1 in [34]).*

Existence and uniqueness of solutions to (3.48)–(3.49) must be shown on a case-by-case basis; this can be done by an iteration scheme for linear problems (see Section 4.9 in [34]).

Definition 3.7.2. *Let $\tilde{u}$ be a smooth solution to (3.48)–(3.49). We call the quasi-linear initial value problem well-posed for a time interval $0 \le t \le T$ at the solution $\tilde{u}$, if*

1. *there is a neighbourhood $\mathcal{N}$ of $(\tilde{F},\tilde{f})$ defined by a suitable norm $\|\cdot\|_N$:*

$$\sup_t \|\tilde{F}-F\|_N < \eta, \quad \|\tilde{f}-f\|_N < \eta \tag{3.50}$$

   where η is a sufficiently small constant, such that (3.48)–(3.49) has a smooth solution for all $(F,f) \in \mathcal{N}$, and

2. *there is a constant K such that*

$$\sup_{0\le t\le T} \|\tilde{u}-u\|_N \le K\eta \tag{3.51}$$

(Definition 4.9.3 in [34])

Theorem 3.7.1. *If the system*

$$\frac{\partial}{\partial t}\tilde{u}(t,x) = P\left(t,x,\tilde{u},\frac{\partial}{\partial x}\right)\tilde{u}(t,x)+F \tag{3.52}$$

$$\tilde{u}(0,x) = \tilde{f}(x)\,, \tag{3.53}$$

is symmetric hyperbolic at $\tilde{u}$, then it is well-posed at $\tilde{u}$ (Theorem 4.9.3 in [34]).

3.8 Reformulations of Einstein's equations

Since the ADM evolution system is ill-posed, once might ask if there are alternative ways of writing the Einstein equations in time evolution form for which the evolution equations are well-posed. Typically, this can be done by making changes of variables and modifying the evolution equations by adding parameterized multiples of the constraint equations to the right hand sides. For certain choices of these parameters, the resulting evolution equations have a well-posed Cauchy problem. In this section, three modifications to the ADM equations are described.

The BSSN [11] and [57] system is used by many groups to perform numerical simulations, and there is evidence (e.g. [1, 11]) that simulation lifetimes are greater than when ADM is used. In [54], the BSSN system with prescribed shift and algebraic lapse condition $\alpha = (\det\gamma)^{\sigma}e^{Q}$ was reduced to first order and it was shown that the resulting first order system is symmetrizable hyperbolic and hence admits a well-posed Cauchy problem (here and later, Q represents an a-priori specified function of spacetime, independent of the evolved variables). In [13], the result was extended to more general slicing conditions of the form $\partial_t\alpha = -\alpha F(\alpha, K, x^{\mu})$. In [32], symmetrizable hyperbolicity was defined directly for second order in space systems based upon the existence of energy estimates, and BSSN with an algebraic lapse $\alpha = (\det\gamma)^{\sigma/2}Q$ and prescribed shift was shown to satisfy this definition.

The NOR system is a simplification of BSSN and was introduced in [47], where it was shown to have a strongly hyperbolic *pseudo-differential* reduction to first order. This is sufficient for the second order system to have a well-posed Cauchy problem. The lapse was $\alpha = (\det\gamma)^{b}Q$ and the shift was fixed. In [32], the NOR system was shown to satisfy the definition of symmetrizable hyperbolicity based on energy estimates. As far as we know, the NOR system has not been implemented numerically before this work. We consider two parameterizations of the NOR system; NOR-A and NOR-B. In [33], the NOR-B system is shown to have a principle part which is essentially the same as that of the BSSN system.

The ST [55] formulation is a fully first order system and is a special case of one of the family of formulations introduced in [38], which in turn is a generalization of the Einstein-Christoffel formulation of [6]. The hyperbolicity of the ST system can be analysed directly using the techniques of the previous sections. With appropriate choices of parameters, it is symmetrizable hyperbolic and hence admits a well-posed Cauchy problem.

3.8.1 BSSN

The BSSN formulation is derived from the ADM equations by defining a new set of variables and adding the constraints to the evolution equations.

The 3-metric γ_{ij} is conformally rescaled to have unit determinant, and the determinant is evolved separately.

$$\tilde{\gamma}_{ij} = e^{-4\phi}\gamma_{ij} \tag{3.54}$$
$$e^{4\phi} = \gamma^{1/3} \equiv (\det \gamma_{ij})^{1/3} \tag{3.55}$$

The extrinsic curvature K_{ij} is rescaled by the same factor, and is split into its trace and trace-free parts; these are evolved separately as well.

$$\tilde{A}_{ij} = e^{-4\phi}(K_{ij} - \frac{1}{3}\gamma_{ij}K) \tag{3.56}$$

The resulting evolution equations for ϕ and K are

$$\partial_t \phi = -\frac{1}{6}\alpha K \tag{3.57}$$
$$\partial_t K = -e^{-4\phi}\left[\tilde{D}^i\tilde{D}_i\alpha + 2\partial_i\phi\tilde{D}^i\alpha\right] + \alpha\left(\tilde{A}^{ij}\tilde{A}_{ij} + \frac{1}{3}K^2\right) \tag{3.58}$$

where the Hamiltonian constraint has been used to eliminate the Ricci scalar in (3.58). $\tilde{\gamma}_{ij}$ and $\tilde{A}_{ij}$ inherit evolution equations from those of γ_{ij} and K_{ij}:

$$\partial_t \tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} \tag{3.59}$$
$$\partial_t \tilde{A}_{ij} = e^{-4\phi}\left[\alpha\tilde{R}_{ij} + \alpha R^{\phi}_{ij} - \tilde{D}_i\tilde{D}_j\alpha + 4\partial_{(i}\phi\tilde{D}_{j)}\alpha\right]^{\mathrm{TF}} + \alpha K\tilde{A}_{ij} - 2\alpha\tilde{A}_{ik}\tilde{A}^k_j \tag{3.60}$$

where the superscript TF denotes the trace free part (the trace free part of a tensor T_{ij} with respect to a metric γ_{ij} is defined as $T^{\mathrm{TF}}_{ij} \equiv T_{ij} - \frac{1}{3}T_{mn}\gamma^{mn}\gamma_{ij}$ which implies $\gamma^{ij}\left(T^{\mathrm{TF}}_{ij}\right) = 0$ with respect to either metric). The Ricci tensor in (3.60) is computed by splitting it into two parts. The first part is the Ricci tensor of the conformal metric $\tilde{\gamma}_{ij}$. The second part is the contribution from the conformal factor.

$$R_{ij} = \tilde{R}_{ij} + R^{\phi}_{ij} \tag{3.61}$$
$$R^{\phi}_{ij} = -2\tilde{D}_i\tilde{D}_j\phi - 2\tilde{\gamma}_{ij}\tilde{D}^k\tilde{D}_k\phi + 4\tilde{D}_i\phi\tilde{D}_j\phi - 4\tilde{\gamma}_{ij}\tilde{D}^k\phi\tilde{D}_k\phi \tag{3.62}$$

In addition to the conformal traceless decomposition, an important ingredient of the BSSN system is the introduction of the *conformal connection functions*, $\tilde{\Gamma}^i$. These are defined by

$$\tilde{\Gamma}^i \equiv \tilde{\gamma}^{jk}\tilde{\Gamma}^i{}_{jk} = -\partial_j \tilde{\gamma}^{ij} \tag{3.63}$$

The $\tilde{\Gamma}^i$ are introduced as new evolved variables, and are used to remove derivatives of the divergence of $\tilde{\gamma}_{ij}$ from the system. Whenever such a term appears, it is replaced with a derivative of $\tilde{\Gamma}^i$. This happens in the computation of the conformal Ricci tensor.

$$\tilde{R}_{ij} = -\frac{1}{2}\tilde{\gamma}^{kl}\partial_k\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k - \tilde{\Gamma}_{(ij)k}\partial_j\tilde{\gamma}^{jk} + \tilde{\gamma}^{ls}(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)ks} + \tilde{\Gamma}^k_{is}\tilde{\Gamma}_{klj}) \tag{3.64}$$

This equation for the Ricci tensor is different to the usual one; a term is missing because $\det\tilde{\gamma}_{ij} = 1$. An evolution equation for $\tilde{\Gamma}^i$ is obtained by taking the time derivative of (3.63) and commuting space and time derivatives.

$$\partial_t\tilde{\Gamma}^i = -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha\left[(m-1)\partial_k\tilde{A}^{ki} - \frac{2m}{3}\tilde{D}^iK + m(\tilde{\Gamma}^i{}_{kl}\tilde{A}^{kl} + 6\tilde{A}^{ij}\partial_j\phi)\right] \tag{3.65}$$

In (3.65), the momentum constraint has been added to the right hand side with parameter m. Setting m to 1 removes the divergence of $\tilde{A}_{ij}$. We choose to evolve the lapse α (an evolved, rather than prescribed, lapse allows greater freedom in choosing slicings). In this work, we only consider harmonic slicing ($\nabla^a\nabla_a t = 0$). The resulting evolution equation for α is:

$$\partial_t\alpha = -\alpha^2 K \tag{3.66}$$

We now present the full set of BSSN evolution equations:

$$\partial_t\phi = -\alpha K/6 \tag{3.67}$$

$$\partial_t\tilde{\gamma}_{ij} = -2\alpha\tilde{A}_{ij} \tag{3.68}$$

$$\partial_t K = -e^{-4\phi}\left[\tilde{D}^i\tilde{D}_i\alpha + 2\partial_i\phi\tilde{D}^i\alpha\right] + \alpha\left(\tilde{A}^{ij}\tilde{A}_{ij} + \frac{1}{3}K^2\right) \tag{3.69}$$

$$\partial_t\tilde{A}_{ij} = e^{-4\phi}\left[\alpha\tilde{R}_{ij} + \alpha R^\phi_{ij} - \tilde{D}_i\tilde{D}_j\alpha + 4\partial_{(i}\phi\tilde{D}_{j)}\alpha\right]^{\mathrm{TF}} + \alpha K\tilde{A}_{ij} - 2\alpha\tilde{A}_{ik}\tilde{A}^k_j \tag{3.70}$$

$$\partial_t\tilde{\Gamma}^i = -2\tilde{A}^{ij}\partial_j\alpha + 2\alpha\left[(m-1)\partial_k\tilde{A}^{ki} - \frac{2m}{3}\tilde{D}^iK + m(\tilde{\Gamma}^i{}_{kl}\tilde{A}^{kl} + 6\tilde{A}^{ij}\partial_j\phi)\right] \tag{3.71}$$

$$\tilde{R}_{ij} = -\frac{1}{2}\tilde{\gamma}^{kl}\partial_k\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\tilde{\Gamma}^k - \tilde{\Gamma}_{(ij)k}\partial_j\tilde{\gamma}^{jk} + \tilde{\gamma}^{ls}(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)ks} + \tilde{\Gamma}^k_{is}\tilde{\Gamma}_{klj}) \tag{3.72}$$

$$R^{\phi}_{ij} = -2\tilde{D}_i\tilde{D}_j\phi - 2\tilde{\gamma}_{ij}\tilde{D}^k\tilde{D}_k\phi + 4\tilde{D}_i\phi\tilde{D}_j\phi - 4\tilde{\gamma}_{ij}\tilde{D}^k\phi\tilde{D}_k\phi \quad (3.73)$$

$$\partial_t\alpha = -\alpha^2 K \quad (3.74)$$

The constraint equations in the BSSN variables are:

$$H \equiv e^{-4\phi}\tilde{\gamma}^{ij}(\tilde{R}_{ij} + R^{\phi}_{ij}) - \tilde{A}_{ij}\tilde{A}^{ij} + \frac{2}{3}K^2 \quad (3.75)$$

$$M_i \equiv 6\tilde{\gamma}^{jl}\tilde{A}_{li}\phi_{,j} + \tilde{\gamma}^{lk}\tilde{D}_k\tilde{A}_{li} - \frac{2}{3}\partial_i K \quad (3.76)$$

Choosing $m = 1$ leads to a well-posed Cauchy problem for these equations [13].

3.8.2 NOR

Nagy, Ortiz and Reula suggested modifications to the ADM system such that it can be made strongly hyperbolic whilst remaining in second order form. The system we use includes the slight adjustments of [32]. Additionally, we use an evolved lapse.

The variable f_i is defined as

$$f_i = \gamma^{kl}(\gamma_{ik,l} - \frac{1}{2}\rho\gamma_{kl,i}) \quad (3.77)$$

with parameter ρ. This introduces the new constraint G_i where

$$G_i := f_i - \gamma^{kl}(\gamma_{ik,l} - \frac{1}{2}\rho\gamma_{kl,i}) \quad (3.78)$$

Starting from the ADM evolution equations, an evolution equation for f_i is obtained by differentiating (3.77) and commuting space and time derivatives. The Hamiltonian and momentum constraints are added with parameters c and b, and derivatives of the f_i definition constraint G_i are added with parameters a and a':

$$\partial_t\gamma_{ij} = -2\alpha K_{ij} \quad (3.79)$$

$$\partial_t K_{ij} = -D_iD_j\alpha + \alpha(R^{(3)}_{ij} - 2K_{ik}K_{lj}\gamma^{kl} + K_{ij}K) + \frac{1}{2}a(G_{i,j} + G_{j,i}) + (cH + a'G_{k,l}\gamma^{kl})\gamma_{ij} \quad (3.80)$$

$$\partial_t f_i = \alpha K^{kl}(2\gamma_{ik,l} - \rho\gamma_{kl,i}) - \gamma^{kl}\left[2(\alpha K_{ik})_{,l} - \rho(\alpha K_{kl})_{,i}\right] + 2bM_i \quad (3.81)$$

$$\partial_t\alpha = -\alpha F(\alpha, K, x^i) \quad (3.82)$$

The variables γ_{ij}, K_{ij}, f_i and α are evolved. Due to the symmetries of γ_{ij} and K_{ij}, this leads to 16 evolved variables. We write the Ricci tensor entirely in terms of γ_{ij}; f_i is only used where it appears as part of G_i.

For harmonic slicing, the lapse source function is

$$F(\alpha, K, x^i) = \alpha K \tag{3.83}$$

We identify two specific sets of parameters. Choosing

$$a = 1, \quad b = 1, \quad a' = 0, \quad \rho = 2/3, \quad c = 0 \tag{3.84}$$

we refer to the system as NOR-A. Setting $a' = 1$ and $c = 1/3$ leads to a system which we call NOR-B. Both of these systems are symmetric hyperbolic, in the sense of [32], as shown in [33]. Note that choosing parameters

$$a = 0, \quad b = 0, \quad a' = 0, \quad \rho = 0, \quad c = 0 \tag{3.85}$$

leads to standard ADM.

3.8.3 ST

Instead of leaving the Einstein equations in second order in space form, it is possible to introduce new variables for the derivatives of those quantities which are differentiated twice and obtain a fully first order reduction. We describe the formulation of Sarbach and Tiglio, referred to as ST.

Starting from the ADM equations, the variables which are differentiated twice are γ_{ij} and α. New variables for the first derivatives of these are defined:

$$d_{kij} \equiv \partial_k \gamma_{ij} \tag{3.86}$$

$$A_i \equiv \partial_i \alpha \tag{3.87}$$

as well as the contractions

$$b_j \equiv d_{kij}\gamma^{ki} \tag{3.88}$$

$$d_k \equiv d_{kij}\gamma^{ij} \tag{3.89}$$

The definitions (3.86)–(3.87) lead to the constraints

$$C_{kij} \equiv d_{kij} - \partial_k \gamma_{ij} = 0 \tag{3.90}$$

$$C_{lkij} \equiv \partial_{[l} d_{k]ij} = 0 \tag{3.91}$$

where the first is the definition constraint of d_{kij} and the second comes from the fact that successive partial derivatives of γ_{ij} commute.

An evolution equation for d_{kij} is formed using the ADM evolution equation for γ_{ij} and commuting space and time derivatives. The general form for the evolved lapse is

$$\partial_t \alpha = -\alpha F(\alpha, K, x^\mu) \tag{3.92}$$

where F is an arbitrary function of its arguments. For harmonic slicing ($\nabla^a \nabla_a t = 0$) with zero shift we have

$$F = \alpha K \tag{3.93}$$

leading to

$$\partial_t \alpha = -\alpha^2 K \tag{3.94}$$

Taking a time derivative of A_i leads to an evolution equation

$$\partial_t A_i = \alpha \left[-K A_i - (K_{mn,i} \gamma^{mn} - K^{mn} d_{imn}) + \xi M_i \right] \tag{3.95}$$

where the momentum constraint $M_i = D^a K_{ai} - D_i K$ has been added to the right hand side with parameter ξ. The evolution equation for γ_{ij} is unchanged:

$$\partial_t \gamma_{ij} = -2\alpha K_{ij} \tag{3.96}$$

but the equation for K_{ij} is modified by the addition of $C_{a(ij)b}$ and the Hamiltonian constraint $H \equiv \frac{1}{2}(R - K_{ab}K^{ab} + K^2)$. The resulting equation is

$$\begin{aligned} \partial_t K_{ij} = \alpha \big[& R_{ij} - \partial_{(i} A_{j)} + \Gamma^k{}_{ij} A_k - \\ & A_i A_j - 2 K_{ia} K^a_j + K K_{ij} + \gamma_{ij} H + \zeta \gamma^{ab} C_{a(ij)b} \big] \end{aligned} \tag{3.97}$$

Taking a time derivative of the definition of d_{kij} and adding the constraints M_j leads to

$$\partial_t d_{kij} = \alpha \left[-2\partial_k K_{ij} - 2A_k K_{ij} + \eta \gamma_{k(i} M_{j)} + \chi \gamma_{ij} M_k \right] \tag{3.98}$$

The Ricci tensor R_{ij} is

$$\begin{aligned} R_{ij} &= \frac{1}{2}\gamma^{ab}(-\partial_a d_{bij} + \partial_a d_{(ij)b} + \partial_{(i} d_{|ab|j)} - \partial_{(i} d_{j)ab}) + \\ &\quad \frac{1}{2} d_i{}^{ab} d_{jab} + \frac{1}{2}(d_k - 2b_k)\Gamma^k{}_{ij} - \Gamma^k{}_{lj}\Gamma^l{}_{ik} \end{aligned} \tag{3.99}$$

Wherever they appear, the Christoffel symbols are computed according to

$$\Gamma^k{}_{ij} = \frac{1}{2}\gamma^{kl}(2d_{(ij)l} - d_{lij}) \tag{3.100}$$

The choice of parameters

$$\gamma = -1/2, \quad \zeta = -1, \quad \eta = 2, \quad \xi = -\chi/2, \quad \chi = -1 \tag{3.101}$$

leads [55] to a symmetric hyperbolic system.

3.9 Summary

In this chapter, the notion of well-posedness of the Cauchy problem for a system of partial differential equations has been described. For fully first order systems, textbook characterizations of well-posedness based on algebraic properties of the coefficients in the equations have been presented.

It has been shown here how some systems which are first order in time and second order in space can be written in a general form and reduced to first order in space. If the reduced system is well-posed, then the second order system will also be well-posed, but in a norm containing derivatives. The well-posedness of a second order in space system can be analysed by considering the algebraic properties of the second order system, without explicitly performing the reduction to first order.

Three reformulations of the ADM Einstein equations were reviewed, and choices of their parameters were given which have been shown to lead to well-posedness of their Cauchy problems.

Chapter 4

Finite difference approximations to time evolution PDEs

Given a time evolution problem for which there is no known analytical solution, and the desired solution is not a perturbation of a known exact solution, numerical methods can be used to find an approximate solution. In this chapter, the method of finite differences will be introduced. For a finite difference scheme to be useful, it must have the property of *convergence* of the numerical solution to the exact solution as the computational grid is refined. The *Lax theorem* for linear, constant coefficient systems states that this will happen if the scheme is *consistent* with the continuum equation and is *stable*. These terms will be explained below. Essentially, consistency ensures that the finite difference scheme approximates the correct continuum equation, and stability controls the growth of the solution in time. Consistency is usually easy to show, but stability is not. Some theorems are reviewed which make it easier to show stability for various schemes.

4.1 Notation and definitions

Only problems which are periodic in the spatial coordinates will be considered in this work. This means that Fourier series can be used to represent the solutions. Problems with artificial boundaries will not be considered. Consider a time evolution problem of the form:

$$\frac{\partial}{\partial t}u(t,x) = P\left(t,x,u,\frac{\partial}{\partial x}\right)u(t,x) \tag{4.1}$$

$$u(0,x) \;=\; f(x) \tag{4.2}$$

where $x \in \mathbb{R}^d$, $t \geq 0 \in \mathbb{R}$, and $u(t,x) \in \mathbb{R}^m$. Much of the theory developed below is for linear problems with constant coefficients, though Einstein's equations are nonlinear with variable coefficients. The results of this section can often be extended to systems with variable coefficients [34].

The spatial coordinates will be $(x^{(1)}, x^{(2)}, \ldots, x^{(d)})$ where d is the number of spatial dimensions. The *numerical grid* is

$$x_j = (x^{(1)}_{j_1}, x^{(2)}_{j_2}, \ldots, x^{(d)}_{j_d}) = (j_1 h_1, j_2 h_2, \ldots, j_d h_d) \tag{4.3}$$

for j_i from 0 to $N_i - 1$, and h_i the spatial interval between grid points. A time step k is chosen so that the time interval over which a solution will be found is discretized into the points $0, k, 2k, \ldots, t_n = nk$. The numerical approximation to the solution of (4.1)–(4.2) will be represented by $v^n_j \in \mathbb{R}^m$, where v is a *grid function.*

A grid function v is a member of the space $\mathbb{C}^{N_1,N_2,\ldots,N_d}$. We will be considering *discrete norms* on this space. For example, the discrete $l_{2,h}$ norm is

$$||v||^2_h \equiv \sum_j |v_j|^2 h^d \tag{4.4}$$

where h^d is notation for $h_1 h_2 \ldots h_d$ and j is a multi-index (see Appendix B) for the grid points. A suffix h on a norm indicates that the norm depends on the grid spacing h.

In order to solve problem (4.1)–(4.2) using finite difference methods, it is necessary to construct a *scheme.* This is an algebraic relation between the values of v at different points which—given suitable initial data—is sufficient for determining the grid function at every point. There are many such schemes for each equation, and they can have different properties. However, it is necessary to be able to make the *error* (the difference between the discrete solution and the continuum solution) arbitrarily small by increasing the resolution of the grid (i.e. by decreasing h_i and k). A scheme satisfying this property is said to *converge.*

This work will be concerned with one-step explicit schemes of the form

$$v^{n+1}_j = Q v^n_j \tag{4.5}$$

where now j is a multi-index $j_1 j_2 \dots j_d$ and Q is a $\mathbb{C}^{m,m}$ matrix of *grid function operators* (i.e. Q maps from grid functions to grid functions). The spatial grid point multi-index j in v_j^n will often be dropped if the grid function as a whole is meant, or if a relation can be interpreted pointwise. The value of the exact solution evaluated at points on the numerical grid will be written as

$$u_j^n \equiv u(t_n, x_j) \tag{4.6}$$

4.2 Convergence, consistency and stability

We require that the difference between the solution to the finite difference scheme and the solution to the continuum equations should approach zero as the grid spacing is decreased. This is called *convergence* and it is defined as follows:

Definition 4.2.1. *The difference scheme (4.5) approximating the partial differential equation (4.1) is* convergent *of order (p, q) in a discrete norm $||\cdot||$ if for any t, as $(n+1)k$ converges to t,*

$$\|v^n - u^n\|_h = O(h^p) + O(k^q) \tag{4.7}$$

provided that the initial data is accurate of order p to $u(0, x)$,

$$\|v^0 - u^0\|_h = O(h^p) \tag{4.8}$$

in a discrete norm $\|\cdot\|_h$. (See Definition 2.2.3 in [61] and Section 5.1 in [34])

It is necessary that the scheme approximates the correct differential equation; i.e. it is *consistent* with the equation. The level to which this holds is called the *order of accuracy* of the scheme to the equation. The *local truncation error* τ_j^n of the scheme is defined by

$$u^{n+1} = Qu^n + k\tau^n \tag{4.9}$$

Definition 4.2.2. *The difference scheme (4.5) is accurate of order (p, q) to the partial differential equation (4.1) in a discrete norm $||\cdot||$ if the truncation error satisfies*

$$\|\tau^n\|_h = O(h^p) + O(k^q) \tag{4.10}$$

The scheme is called consistent *if $p > 0$ and $q > 0$. (Definition 2.3.3 in [61])*

This is not enough to prove convergence, however. The extra ingredient necessary is *stability*.

Definition 4.2.3. *The difference scheme (4.5) is said to be stable in a norm $||\cdot||$ if there exist positive constants h_{i0} and k_0, and non-negative constants K and α so that*

$$\|v^{n+1}\|_h \leq Ke^{\alpha t_n}||v^0|| \tag{4.11}$$

for $t_n \geq 0$, $0 < h_i \leq h_{i0}$ and $0 < k \leq k_0$ (where $t_n = (n+1)k$) for all initial grid functions u^0. (Definition 2.4.1 in [61])

4.2.1 Forcing terms

It can be shown (Theorem 5.1.1 in [34]) that if (4.5) is stable, then the addition of a forcing term to the right hand side requires the estimate on the solution to be modified. The modified scheme

$$v_j^{n+1} = Qv_j^n + kF_j^n \tag{4.12}$$

satisfies the estimate

$$\|v^n\|_h \leq K\left(e^{\alpha t_n}\|v^0\|_h + \varphi_h^*(\alpha, t_n)\max_{0\leq\nu\leq n-1}\|F^\nu\|_h\right) \tag{4.13}$$

where

$$\varphi_h^*(\alpha, t_n) = \sum_{\nu=0}^{n-1} e^{\alpha(t_n - t_{\nu+1})k} \tag{4.14}$$

4.3 Lax equivalence theorem

The Lax theorem states that a consistent, one-step, difference scheme is convergent if and only if it is stable. We will show that consistency and stability are sufficient for convergence.

Theorem 4.3.1. *Consider the continuum problem (4.1)–(4.2) and a finite difference scheme (4.5). If the scheme is consistent of order (p, q) with the continuum problem, and it is stable, then the scheme will be convergent. (Theorem 5.1.3 in [34])*

Proof. The *error*, w, is defined to be a grid function which is the difference between the numerical and exact solutions,

$$w_j^n \equiv v_j^n - u_j^n \tag{4.15}$$

It satisfies the scheme

$$\begin{aligned} w^{n+1} &= v^{n+1} - u^{n+1} = Qv^n - Qu^n - k\tau^n && (4.16)\\ &= Qw^n - k\tau^n && (4.17) \end{aligned}$$

So the error satisfies the same scheme as the solution but with a forcing term due to the truncation error. If the scheme (4.5) is stable, then the addition of a forcing term leads to the estimate

$$\|w^n\|_h \leq K\left(e^{\alpha t_n}\|w^0\|_h + \varphi_h^*(\alpha, t_n) \max_{0\leq\nu\leq n-1}\|\tau^\nu\|_h\right) \tag{4.18}$$

If the scheme is consistent with order of accuracy (p, q),

$$\|\tau^n\|_h = O(h^p) + O(k^q) \tag{4.19}$$

and the initial data v^0 is accurate of order (p, q) to the exact initial data,

$$\|w^0\|_h = O(h^p) + O(k^q) \tag{4.20}$$

then we have

$$\|w^n\|_h = O(h^p) + O(k^q) \tag{4.21}$$

as required. □

4.4 Conditions for stability

4.4.1 Fourier representation

Definition 4.2.3 is hard to apply in practice, as working with the grid function operator Q is awkward. Instead, it is possible to work in Fourier space, where the grid function operator

becomes a simple matrix function of frequency, and can be manipulated using standard matrix techniques. Using the material in Appendix C, substituting

$$v_j = \frac{1}{N}\sum_{\xi} \hat{v}(\xi)e^{ij\xi} \tag{4.22}$$

into (4.5), the *Fourier transformed difference scheme* is obtained:

$$\hat{v}^{n+1}(\xi) = \hat{Q}(\xi)\hat{v}^n(\xi) \tag{4.23}$$

The matrix $\hat{Q}(\xi)$ can be used in the following theorem:

Theorem 4.4.1. *The difference scheme (4.5) is stable with respect to the $||\cdot||$ norm if and only if there exist positive constants h_{i0} and k_0 and non-negative constants K and α so that*

$$|\hat{Q}^n(\xi)| \leq Ke^{\alpha t_n} \tag{4.24}$$

for all $t_n > 0$, $h_r \leq h_{r0}$, $k \leq k_0$ and $\xi_r = -\pi + 2\pi/N_r, \ldots, \pi$, $r = 0, 1, \ldots, d$. (Proposition 2.4.2 in [61])

4.4.2 The von Neumann condition

The condition

$$\sigma(\hat{Q}) \leq e^{\alpha k} \tag{4.25}$$

is called the von Neumann condition. For a scalar equation, it is both necessary and sufficient for stability, but for a system of equations, it is only a necessary condition (Theorem 6.2.2 in [61]). However, if $\hat{Q}$ is uniformly diagonalizable, it is also sufficient. More precisely, assume that there exists a non singular matrix $T(\xi)$ with $|T||T^{-1}| \leq C$ for C independent of ξ, such that

$$T^{-1}\hat{Q}T = \Lambda = \text{diag}(q_1, q_2, \ldots, q_m) \tag{4.26}$$

then the von Neumann condition is both necessary and sufficient for stability.

$$|\hat{Q}^n| = |T\Lambda^n T^{-1}| \leq |T||T^{-1}||\Lambda^n| = |T||T^{-1}|\sigma(\hat{Q})^n \leq Ce^{\alpha t_n}$$

In particular, if $\hat{Q}$ is normal (i.e. $[\hat{Q}^*, \hat{Q}] = 0$), as would be the case if it were Hermitian or anti-Hermitian, then it can be unitarily diagonalized, and the diagonalizing matrix T has

unit norm. Hence

$$|\hat{Q}^n| = |T^*\Lambda^n T| = |\mathrm{A}|^n = \sigma(\hat{Q})^n$$

4.4.3 Lower order terms

As shown in Theorem 5.1.2 in [34], the stability of (4.5) is not affected by replacing $\hat{Q}(\xi) \rightarrow \hat{Q}(\xi) + \hat{B}$ where B does not depend on ξ. This means that the stability of a scheme with lower order terms can be determined by considering the stability of the scheme with those terms removed. This simplifies the analysis.

4.5 Finite Difference Operators

In order to construct a numerical scheme consistent with a particular partial differential equation, partial derivatives are usually replaced with grid function operators called *finite difference operators*. The following definitions are given in 1D for simplicity. The generalization to 3D is straightforward.

$$E_+ v_j \equiv v_{j+1} \tag{4.27}$$

$$E_- v_j \equiv v_{j-1} \tag{4.28}$$

$$D_+ v_j \equiv \frac{v_{j+1} - v_j}{h} \tag{4.29}$$

$$D_- v_j \equiv \frac{v_j - v_{j-1}}{h} \tag{4.30}$$

$$D_0 v_j \equiv \frac{v_{j+1} - v_{j-1}}{2h} \tag{4.31}$$

All the operators above can be written as polynomials in E_+ and E_-, which commute. Hence all these operators commute. A simple relation is

$$D_0 = \frac{1}{2}(D_+ + D_-) \tag{4.32}$$

The difference operators approximate partial derivatives to varying orders of accuracy:

$$D_+ v_j = v'(x_j) + O(h) \tag{4.33}$$

$$D_- v_j = v'(x_j) + O(h) \tag{4.34}$$

$$D_0 v_j = v'(x_j) + O(h^2) \tag{4.35}$$

$$D_+ D_- v_j = v''(x_j) + O(h^2) \tag{4.36}$$

$$D_0 D_0 v_j = v''(x_j) + O(h^2) \tag{4.37}$$

4.6 The Method of Lines

It is usually very useful to construct a numerical scheme in two stages. The first is the discretization of the spatial derivatives occurring on the right hand side of (4.1), and the second is the integration in time of the resulting set of ODEs. This is called the *method of lines.* It is not to be confused with the method of lines used for solving PDEs by integrating along characteristic curves.

Firstly, $u(t,x)$ is replaced with a grid function $v_j(t)$ which depends continuously on time, and P is replaced with a grid function operator (for notational convenience also called P) to obtain:

$$\frac{d}{dt} v_j(t) = P(t, x_j, v(t)) \tag{4.38}$$

This is called the *semidiscrete* system of evolution equations. It is necessary that P is a consistent representation of the continuum operator (see Section 4.2). A choice of P is known as a *spatial discretization.* For example, the following replacements can be made:

$$\begin{aligned} \partial_i &\rightarrow D_{0i} \\ \partial_i \partial_j &\rightarrow \begin{cases} D_{0i} D_{0j} & \text{if } i \neq j \\ D_{+i} D_{-i} & \text{if } i = j \end{cases} \end{aligned} \tag{4.39}$$

We refer to this as the *standard second order accurate discretization.* (4.38) is a set of coupled ordinary differential equations for the grid functions v_j in the variable t. There are $m \times n^d$ equations which can be solved approximately by using a standard ODE integrator.

In this work we restrict our attention to the following three ODE integrators: 3rd and 4th order Runge-Kutta (RK3 and RK4), and iterative Crank-Nicolson (ICN) [60]. Consider a system of ordinary differential equations

$$\frac{dy}{dt}(t) = f(t, y(t)) \tag{4.40}$$

where $y(t) \in \mathbb{C}^q$. The time integrators are:

RK3

$$\begin{aligned} k_1 &= kf(t_n, y^n) \\ k_2 &= kf(t_n + k/2, y^n + k_1/2) \\ k_3 &= kf(t_n + 3k/4, y^n + 3k_2/4) \\ y^{n+1} &= y^n + (2k_1 + 3k_2 + 4k_3)/9 \end{aligned}$$

RK4

$$\begin{aligned} k_1 &= kf(t_n, y^n) \\ k_2 &= kf(t_n + k/2, y^n + k_1/2) \\ k_3 &= kf(t_n + k/2, y^n + k_2/2) \\ k_4 &= kf(t_n + k, y^n + k_3) \\ y^{n+1} &= y^n + (k_1 + 2k_2 + 2k_3 + k_4)/6 \end{aligned}$$

ICN

$$\begin{aligned} k_1 &= kf(t_n, y^n) \\ k_2 &= kf(t_n + k/2, y^n + k_1/2) \\ k_3 &= kf(t_n + k/2, y^n + k_2/2) \\ y^{n+1} &= y^n + k_3 \end{aligned}$$

where the k_i are intermediate quantities in $\mathbb{C}^q$ and k is the time step as usual. (See Section 4.6.1 for a further discussion of ICN.). ICN, RK3 and RK4 are accurate of order two, three and four respectively, i.e. the error at time t is $O(k^p)$ where p is the order of accuracy.

These time integrators can be used to integrate the semidiscrete equations, and the resulting scheme is fully discrete (both time and space have been discretized) and suitable for programming on a computer. The semidiscrete equations (4.38) can be integrated to obtain

$$v_j^{n+1} = Q(t_n, x_j^n, v_j^n) \tag{4.41}$$

We note one useful property of these time integrators, which will be used for the linear problems we analyse in the following chapter. If the right hand side of the semidiscrete

system is linear and has no explicit dependence on t, then we can write

$$f(t_n, y^n) = Fy^n \tag{4.42}$$

where F is a $q \times q$ constant matrix. In this case, we can expand y^{n+1} as follows. For ICN:

$$y^{n+1} = \left(1 + 2\sum_{r=1}^{3} \frac{F^r}{2^r}\right) y^n \tag{4.43}$$

and for pth order Runge-Kutta:

$$y^{n+1} = \left(\sum_{r=0}^{p} \frac{F^r}{r!}\right) y^n \tag{4.44}$$

So in each case the solution operator is a polynomial $\mathcal{P}$ in F:

$$y^{n+1} = \mathcal{P}(F)y^n \tag{4.45}$$

and the semidiscrete equations (4.38) can be integrated to obtain

$$v_j^{n+1} = \mathcal{P}(P)v_j^n \equiv Qv_j^n \tag{4.46}$$

4.6.1 Further discussion of iterative Crank-Nicolson

Consider again (4.40). The simplest scheme for solving this equation is the Euler method:

$$\frac{y^{n+1} - y^n}{k} = f(t_n, y^n) \tag{4.47}$$

However, this scheme is only first order accurate, and in fact when it is used for solving the standard second order accurate discretization of the advection equation $u_t = u_x$, the method is unstable. Note the asymmetry in the equation: the time derivative is evaluated at both time steps but the right hand side is evaluated only at time step n. Replacing the right hand side with its value averaged over the two time steps leads to the Crank-Nicolson scheme, which is stable for a large class of semidiscrete problems:

$$\frac{y^{n+1} - y^n}{k} = \frac{1}{2}\left[f(t_{n+1}, y^{n+1}) + f(t_n, y^n)\right] \tag{4.48}$$

Note that this scheme is implicit; it is not in general possible to solve directly for y^{n+1} in terms of y^n. One way to solve this algebraic equation is by *iteration*. A trial solution $y_{(i)}^{n+1}$ is chosen, and this is used in evaluating the right hand side. The resulting value of y^{n+1} is used as a better approximation. The iteration scheme is

$$\frac{y_{(i+1)}^{n+1} - y^n}{k} = \frac{1}{2}\left[f(t_{n+1}, y_{(i)}^{n+1}) + f(t_n, y^n)\right] \tag{4.49}$$

We use $y_{(0)}^{n+1} = y^n$ as initial data for the scheme. In principle, for a small enough value of k, this scheme will converge on the exact solution. However, as pointed out in [60], stopping after three iterations (using $y_{(3)}^{n+1}$ as y^{n+1}) leads to a stable *explicit* scheme. This is the scheme presented as ICN above.

4.7 Round-off errors

Most numerical analysis concerns the properties of exact solutions to finite difference equations. However, when attempting to solve such equations on a computer, even algebraic computations are carried out only approximately, and the result of numerical operations is stored with only a finite precision. The method used by modern computers for storing real numbers is called *floating point representation* and the set of mathematical operations performed on these approximations is called *floating point arithmetic*. The error in the solution caused by the use of floating point arithmetic in solving a finite difference equation is called *roundoff error*, as the numbers have been rounded in order to store them with finite precision.

In [34], the effect of roundoff error is said to be equivalent to adding a forcing term to the right hand side of the finite difference approximation. Suppose that at each time step, an error is made, and the size of the error is characterized by some constant ϵ. For a linear system with constant coefficients,

$$v_j^{n+1} = Qv_j^n \tag{4.50}$$

becomes

$$\tilde{v}_j^{n+1} = Q\tilde{v}_j^n + \epsilon_j^n \qquad ||\epsilon^n|| \leq \epsilon \tag{4.51}$$

where ϵ_j^n is the roundoff error grid function. The error due to this forcing term, $w^n \equiv \tilde{v}^n - v^n$ satisfies

$$w_j^{n+1} = Q\tilde{w}_j^n + \epsilon_j^n \tag{4.52}$$

In other words, the error satisfies the same equation as the solution. If (4.50) is stable, then (4.52) will satisfy the estimate

$$||w^n||_h \leq C(t_n)\frac{\epsilon}{k} \tag{4.53}$$

where $C(t_n)$ is some function independent of h and k. Note that as the resolution of the simulation is increased, k decreases and $||w^n||_h$ becomes large. So the effect of roundoff error cannot in general be reduced by increasing the resolution of the simulation. One way to think about this is to consider that the error is made at each time step, so as the number of time steps becomes larger with increasing resolution, the error made becomes larger as well. With modern computers, $\epsilon \sim 10^{-15}$, and for stable linear problems, it is rarely significant. However, for nonlinear schemes, it is possible that roundoff error can become the dominant contribution to the solution at sufficiently late times.

4.8 Artificial dissipation

It is possible that a particular discretization of a PDE will be unstable. Some such discretizations can be stabilized by the use of *artificial dissipation*. Extra terms are added to the right hand side of the semidiscrete equations which go to zero as the grid spacing is reduced with a particular polynomial power (i.e. the consistency of the scheme and the order of accuracy are maintained). These terms have the effect of damping high frequencies in the numerical solution, and this can sometimes result in a stable scheme. When necessary for stability, we use Kreiss-Oliger type artificial dissipation with parameter σ:

$$\partial_t v(t) = F(v(t); t, x) - \sigma \sum_i h_i^3 (D_{+i} D_i)^2 v \tag{4.54}$$

This is for second order accurate schemes.

4.9 Summary

In this chapter, the method of *finite differences* has been introduced for solving time dependent partial differential equations. Essential for obtaining a solution is the property of *convergence* of the numerical scheme, which for linear systems requires *stability*. A necessary and sufficient condition for stability in Fourier space has been given, as well as a simpler necessary condition (the von Neumann condition). The *method of lines* for separating the spatial discretization from the time integration has been described, as well as several difference operators and time integrators. Roundoff errors due to the use of floating point arithmetic have been described, and it has been pointed out that these can sometimes be important.

Chapter 5

Numerical stability for finite difference approximations of Einstein's equations

5.1 Introduction

For systems which are first order in time and first order in space, and which are well-posed in the L_2 norm, much is known about the stability of the associated finite difference schemes. However, little can be found in the literature about the stability of "ADM-type" systems; i.e. those that are first order in time but second order in space, and are well-posed only in norms containing derivatives. As at the continuum, the simplest example is the one dimensional first order in time, second order in space wave equation:

$$\partial_t \phi(t,x) = \pi(t,x) \tag{5.1}$$

$$\partial_t \pi(t,x) = \partial_x^2 \phi(t,x) \tag{5.2}$$

where $t, x \in \mathbb{R}$, $\phi(t,x), \pi(t,x) \in \mathbb{R}$. As stated in Section 3.4, this system is ill-posed in L_2. By introducing new variables sufficient to make the equations first order in space, the first order system can be shown to be well-posed in L_2. Hence, the second order system is well-posed in a norm containing derivatives;

$$\int_0^{2\pi} |\phi|^2 + |\Pi|^2 + |\partial_i \phi|^2 \, dx$$

If the spatial derivatives are replaced with finite difference operators (time remains continuous), the *semidiscrete problem* is

$$\frac{d}{dt}\phi_j(t) = \pi_j(t)\,, \tag{5.3}$$

$$\frac{d}{dt}\pi_j(t) = D_+D_-\phi_j(t) \tag{5.4}$$

As for the continuum, by providing a suitable family of initial data, the $l_{2,h}$ norm,

$$||v||_h^2 \equiv \sum_j |v_j|^2 h^d \tag{5.5}$$

of the solution at a time t can be made arbitrarily large, contradicting the existence of an estimate

$$\|v^n\|_h \leq Ke^{\alpha t_n}\|f\|_h$$

for all initial data f.

By following an analogous procedure to that used in the continuum case, it is possible to prove stability of the fully discrete system in a norm containing difference operators, specifically the norm

$$||v||_h^2 \equiv \sum_i \left(\phi_i^2 + \pi_i^2 + (D_+\phi_i)^2\right) h$$

In this chapter, we introduce the idea of a *discrete reduction to first order*. This is used to reduce the second order in space finite difference scheme to a fully first order version, which can be analysed with standard techniques. As at the continuum, if a first order discrete reduction is stable, then the original second order discrete system is stable in a discrete norm containing difference operators (we call such norm a D_+ norm). This is not a necessary condition, but it is sufficient.

We find it convenient to introduce the concept of a *discrete symmetrizer* as a tool for proving stability for certain systems. Proofs of stability for the standard discretization of the wave equation in d dimensions are presented, as well as for the NOR formulation of Einstein's equations linearized about Minkowski spacetime in Cartesian coordinates. Courant limits for the wave equation and NOR are obtained; these are the maximum values of k/h that give stability. This chapter is based on joint work [20] done with Gioel Calabrese and

Sascha Husa.

5.2 Convergence

Ultimately, we require that a numerical scheme converge to the exact solution. The Lax theorem (Section 4.3) shows that consistency

$$||\tau^n|| = O(h^p) + O(k^q) \tag{5.6}$$

and stability

$$||v^n|| \leq Ke^{\alpha t_n}||v^0|| \tag{5.7}$$

imply convergence

$$||v^n - u(t^n, \cdot)|| = O(h^p) + O(k^q) \tag{5.8}$$

when the initial data is of the correct order of accuracy

$$||v^0 - u(0, \cdot)|| = O(h^p) \tag{5.9}$$

but the norm used must be the same throughout. So consistency and stability in the D_+ norm implies convergence in the same norm. However, we should note that if

$$||\cdot||_1 \leq ||\cdot||_2 \tag{5.10}$$

then convergence in $||\cdot||_2$ implies convergence in $||\cdot||_1$, since

$$||v^n - u(t^n, \cdot)||_1 \leq ||v^n - u(t^n, \cdot)||_2 \leq K(t)(h^p + k^q) \tag{5.11}$$

Since the L_2 norm is always smaller than the D_+ norm, we see that convergence in the D_+ norm implies convergence in the L_2 norm. Note however that the initial data must be accurate in the D_+ norm; this is equivalent to accuracy in the L_2 norm for smooth initial data, but not for non-smooth data.

5.3 Discrete symmetrizer

Consider a semidiscrete scheme of the form:

$$\frac{d}{dt}v_j(t) = Pv_j(t) \tag{5.12}$$

By taking the discrete Fourier transform (see Appendix C),

$$\hat{v}(\xi) = \sum_j v_j e^{-i\langle j,\xi\rangle} \tag{5.13}$$

we obtain

$$\frac{d}{dt}\hat{u}(t,\xi) = \hat{P}(\xi)\hat{u}(t,\xi) \tag{5.14}$$

where the *symbol* of P is $\hat{P}(\xi)$, a matrix function of frequency. Consider the case of a time integration scheme such that the fully discrete finite difference operator can be written as a polynomial in the semidiscrete operator (see Section 4.6)

$$\hat{v}^{n+1}(\xi) = \mathcal{P}(k\hat{P})\hat{v}^n(\xi) = \hat{Q}(\xi)\hat{v}^n(\xi) \tag{5.15}$$

with $\hat{Q}(\xi) = \mathcal{P}(k\hat{P})$ the amplification matrix of the fully discrete system. So far, a system is stable if and only if there exist $K, \alpha \geq 0$ such that

$$|\hat{Q}^n(\xi)| \leq Ke^{\alpha t_n} \tag{5.16}$$

Further, if the amplification matrix $\hat{Q}(\xi)$ is uniformly diagonalizable, then the von Neumann condition,

$$\sigma(\hat{Q}) \leq e^{\alpha k} \tag{5.17}$$

is both necessary and sufficient for stability. However, it is possible for a discretization to be (conditionally) stable without $\hat{Q}$ being uniformly diagonalizable.

Lemma 5.3.1. *Suppose that $\hat{H}(\xi)$ are Hermitian matrices such that*

$$K^{-1}I \leq \hat{H}(\xi) \leq KI \tag{5.18}$$

$$|\hat{Q}|_{\hat{H}} \leq e^{\alpha k} \tag{5.19}$$

where K is a positive constant. Then the scheme is stable.

Proof. Stability using (5.16) is obtained via

$$|\hat{Q}^n| \leq K|\hat{Q}^n|_{\hat{H}} \leq K|\hat{Q}|_{\hat{H}}^n \leq Ke^{\alpha t_n} \tag{5.20}$$

where (A.34), (A.4) and (5.19) have been used. As a corollary, it can be seen that the von Neumann condition is satisfied:

$$\sigma(\hat{Q}) = \sigma(\hat{H}^{1/2}\hat{Q}\hat{H}^{-1/2}) \leq |\hat{H}^{1/2}\hat{Q}\hat{H}^{-1/2}| = |\hat{Q}|_{\hat{H}} \leq e^{\alpha k} \tag{5.21}$$

which follows from the fact that the eigenvalues of similar matrices are the same, that the spectral radius of a matrix is less than or equal to its norm, and (A.23).

□

This result is difficult to apply in practice, as matrix norms are difficult to calculate. The following shows that if $\hat{H}$ is a *discrete symmetrizer* of the semidiscrete symbol $\hat{P}$, then the matrix energy norm required is equal to the spectral radius.

Lemma 5.3.2. *Suppose there exist Hermitian matrices $\hat{H}(\xi)$ such that*

$$K^{-1}I \leq \hat{H}(\xi) \leq KI\,, \tag{5.22}$$

$$(\hat{H}(\xi)\hat{P}(\xi))^* = -\hat{H}(\xi)\hat{P}(\xi)\,, \tag{5.23}$$

Then we say that $\hat{H}(\xi)$ is a discrete symmetrizer *of $\hat{P}(\xi)$, and $|\hat{Q}|_{\hat{H}} = \sigma(\hat{Q})$.*

Proof. Using (5.23), the matrices $\hat{H}^{1/2}\hat{P}\hat{H}^{-1/2}$ can be seen to be anti-Hermitian, hence they can be diagonalized by unitary matrices $\hat{S}(\xi)$. This implies that the matrices $\hat{H}^{-1/2}(\xi)\hat{S}(\xi)$ diagonalize $\hat{P}(\xi)$. Since $\hat{Q}(\xi)$ is a polynomial in $\hat{P}(\xi)$, these same matrices diagonalize $\hat{Q}(\xi)$. So,

$$|\hat{Q}|_{\hat{H}} = |\hat{H}^{1/2}\hat{Q}\hat{H}^{-1/2}| = |S^{-1}\hat{H}^{1/2}\hat{Q}\hat{H}^{-1/2}S| = \sigma(\hat{Q}) \tag{5.24}$$

The above calculation has used (A.23), (A.6), and the fact that the $\hat{H}^{-1/2}(\xi)\hat{S}(\xi)$ diagonalize $\hat{Q}(\xi)$. Also, the norm of a diagonal matrix is equal to its spectral radius. □

So if a discrete symmetrizer exists, and

$$\sigma(\hat{Q}) \leq e^{\alpha k} \tag{5.25}$$

then the system is stable. This is still a condition on the fully discrete system; calculations can be simplified by considering the semidiscrete system.

Lemma 5.3.3. *For the time integrators considered in this work, using the fact that* $\hat{Q} = \mathcal{P}(k\hat{P})$*, one can show that if the eigenvalues of* $\hat{P}(\xi)$ *are imaginary, as is the case for all problems we have studied, then*

$$\sigma(k\hat{P}) \leq \alpha_0 \Leftrightarrow \sigma(\hat{Q}) \leq 1 \tag{5.26}$$

where $\alpha_0 = 2$ *for ICN,* $\sqrt{8}$ *for 4RK,* $\sqrt{3}$ *for 3RK. This condition is called* local stability on the imaginary axis *in [42].*

Putting together all the above, we obtain the following result:

Theorem 5.3.1 (Discrete symmetrizer theorem). *If all the following are true*

- *The eigenvalues of* $\hat{P}$ *are imaginary*
- *There exists a discrete symmetrizer* $\hat{H}$*; i.e. a Hermitian matrix satisfying (5.22)–(5.23)*
- *The semidiscrete symbol satisfies* $\sigma(kP) \leq \alpha_0$

then

$$|\hat{Q}^n(\xi)| \leq Ke^{\alpha t_n} \tag{5.27}$$

which is sufficient for stability.

5.4 Conserved energy

Suppose that there exist matrices $\hat{H}(\xi)$ which are Hermitian and positive definite. Then

$$\begin{aligned} \frac{d}{dt}|\hat{v}|^2_{\hat{H}} &= 2\mathrm{Re}\left[\hat{v}^*\hat{H}\frac{d}{dt}\hat{v}\right] && (5.28)\\ &= 2\mathrm{Re}\left[\hat{v}^*\hat{H}\hat{P}\hat{v}\right] && (5.29)\\ &= \hat{v}^*(\hat{H}\hat{P} + \hat{P}^*\hat{H})\hat{v} && (5.30) \end{aligned}$$

So for each frequency ξ, the following statements are equivalent:

- $\frac{d}{dt}|\hat{v}|^2_{\hat{H}} = 0$
- $\hat{H}\hat{P} + \hat{P}^*\hat{H} = 0$

Since checking (5.23) can be difficult, the requirement $\frac{d}{dt}|\hat{v}|^2_{\hat{H}} = 0$ can be used instead.

To construct $\hat{H}$ one can proceed as follows. Assume the existence of a matrix T such that $T^{-1}\hat{P}T = \Lambda$ is diagonal with imaginary elements. Then the quantity $\hat{v}^*\hat{H}\hat{v}$, where $\hat{H} = T^{-1*}DT^{-1}$ and D is a positive definite matrix which commutes with Λ, is conserved by the system $\partial_t\hat{v} = \hat{P}\hat{v}$. Defining the *characteristic variables* of $\hat{P}$ to be $\hat{w} \equiv T^{-1}\hat{v}$ (these are individually conserved: $\partial_t|\hat{w}_i|^2 = 0$), we see that to construct a conserved quantity one can take $\hat{w}^*D\hat{w}$. (For $D = I$ this corresponds to adding the squared absolute values of the characteristic variables.) For $\hat{H}$ to be a symmetrizer it remains to be established that $K^{-1}|\hat{v}|^2 \leq \hat{v}^*\hat{H}\hat{v} \leq K|\hat{v}|^2$.

5.5 Discrete reduction to first order

In order to analyse the stability of systems such as the second order in space, first order in time formulations of Einstein's equations, we introduce a technique which we call *discrete reduction to first order*. This is analogous to the procedure performed at the continuum for analysing well-posedness. The reduction is done by introducing auxiliary variables which are equal to discrete derivatives of quantities that are differentiated twice. In this way, theorems for first order systems can be used. As in the continuum case, only sufficient conditions for stability are obtained by this method. Care must be taken when considering consistency and convergence, as the fact that the auxiliary constraints are identically satisfied must be used to obtain the correct orders of accuracy.

The semidiscrete finite difference approximation of

$$\partial_t \boldsymbol{u} = P\boldsymbol{u} \qquad \boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{5.31}$$

$$P = \begin{pmatrix} A^i\partial_i + B & C \\ D^{ij}\partial_i\partial_j + E^i\partial_i + F & G^i\partial_i + J \end{pmatrix}$$

(this is (3.29)) can be written as

$$\frac{d}{dt}\boldsymbol{v} = P\boldsymbol{v} \qquad \boldsymbol{v} = \begin{pmatrix} u \\ v \end{pmatrix} \tag{5.32}$$
$$P = \begin{pmatrix} A^i D_i^{(1)} + B & C \\ D^{ij} D_{ij}^{(2)} + E^i D_i^{(1)} + F & G^i D_i^{(1)} + J \end{pmatrix}$$

where $D_i^{(1)}$ is a discretization of the first derivative in the i direction and $D_{ij}^{(2)}$ is a discretization of the second derivative in the i and j directions. For example, the standard second order accurate discretization would have

$$D_i^{(1)} = D_{0i} \qquad D_{ij}^{(2)} = \begin{cases} D_{0i} D_{0j} & \text{if } i \neq j \\ D_{+i} D_{-i} & \text{if } i = j \end{cases} \tag{5.33}$$

The principal symbol of the semidiscrete system is

$$\hat{P}' = \begin{pmatrix} A^i \hat{D}_i^{(1)} & C \\ D^{ij} \hat{D}_{ij}^{(2)} & G^i \hat{D}_i^{(1)} \end{pmatrix} \tag{5.34}$$

where

$$\hat{D}_i^{(1)} = \frac{i}{h} \sin \xi_i \quad \hat{D}_{ij}^{(2)} = \begin{cases} -\dfrac{1}{h^2} \sin \xi_i \sin \xi_j & \text{if } i \neq j \\ -\dfrac{4}{h^2} \sin^2 \dfrac{\xi_i}{2} & \text{if } i = j \end{cases} \tag{5.35}$$

for the standard second order discretization. The *pseudo-discrete* first order reduction is obtained by defining

$$\hat{w} \equiv i\Omega \hat{u} \qquad \Omega^2 = \sum_{i=1}^{d} |\hat{D}_{+i}|^2 \tag{5.36}$$

The reduced system is

$$\frac{d}{dt}\hat{\boldsymbol{v}}_R = \hat{P}_R \hat{\boldsymbol{v}}_R \qquad \hat{\boldsymbol{v}}_R = \begin{pmatrix} \hat{u} \\ \hat{w} \\ \hat{v} \end{pmatrix} \tag{5.37}$$
$$P_R = \begin{pmatrix} B & (i\Omega)^{-1} A^i \hat{D}_i^{(1)} & C \\ 0 & A^i \hat{D}_i^{(1)} + B & i\Omega C \\ F & (i\Omega)^{-1} (D^{ij} \hat{D}_{ij}^{(2)} + E^i \hat{D}_i^{(1)}) & G^i \hat{D}_i^{(1)} + J \end{pmatrix}$$

The discrete auxiliary constraint is preserved by the time integrator, and there is a one-to-one mapping between solutions of the second order fully discrete system and those of the constraint-satisfying reduced system.

Making use of Theorem 5.1.2 of [34] the terms which correspond to the continuum lower order terms can be dropped from $\hat{P}_R$ without affecting the stability of the fully discrete system, provided that $(i\Omega)^{-1}\hat{D}_i^{(1)}$, $k\hat{D}_i^{(1)}$ and $k\Omega^{-1}\hat{D}_{ij}^{(2)}$ are bounded. This guarantees that the assumptions of the theorem are satisfied. This is true for the second and fourth order accurate standard discretizations.

The result for stability of the fully discrete problem is analogous to that for well-posedness at the continuum.

Theorem 5.5.1. *If there exists $\hat{H}(\xi) = \hat{H}^*(\xi)$ such that the energy $\hat{\boldsymbol{v}}^*\hat{H}\hat{\boldsymbol{v}}$ is conserved by the semidiscrete principal system $\partial_t\hat{\boldsymbol{v}} = \hat{P}'\hat{\boldsymbol{v}}$ and $\hat{H}$ satisfies*

$$K^{-1}I_\Omega \leq \hat{H} \leq KI_\Omega \qquad I_\Omega \equiv \begin{pmatrix} \Omega^2 & 0 \\ 0 & 1 \end{pmatrix} \tag{5.38}$$

where K is a positive scalar constant, then it is possible to construct a discrete symmetrizer for the first order reduction with no lower order terms. So if, in addition, the principal symbol $\hat{P}'$ satisfies $\sigma(k\hat{P}') \leq \alpha_0$, the fully discrete system (including lower order terms) is stable with respect to the norm

$$\|\boldsymbol{v}\|_{h,D_+}^2 \equiv \|u\|_h^2 + \|v\|_h^2 + \sum_{i=1}^{d} \|D_{+i}u\|_h^2 \tag{5.39}$$

i.e. the solution satisfies the estimate

$$\|\boldsymbol{v}^n\|_{h,D_+} \leq Ke^{\alpha t_n}\|\boldsymbol{v}^0\|_{h,D_+} \tag{5.40}$$

Again, $\hat{H}$ can be constructed from the characteristic variables of $\hat{P}'$, as described at the end of Section 5.4.

5.6 Stability of first order strongly hyperbolic systems

It is convenient to define the following quantities,

$$\chi_q^2 = \sum_{i=1}^{d} \sin^q \frac{\xi_i}{2} \qquad \chi^2 = \sum_{i=1}^{d} \sin^2 \xi_i \tag{5.41}$$

Consider a constant coefficient first order strongly hyperbolic system in d spatial dimensions

$$\frac{\partial}{\partial t} u(t,x) = \sum_{k=1}^{d} A^k \frac{\partial}{\partial x^k} u(t,x) \tag{5.42}$$

where $x \in \mathbb{R}^d, t \in \mathbb{R}, u(t,x) \in \mathbb{R}^m$.

We assume that the system is strongly hyperbolic and that it admits a symmetrizer, i.e. there exists a matrix $\hat{H}(\omega)$ in Fourier space, such that $\hat{H}(\omega)\hat{P}(i\omega) + \hat{P}^*(i\omega)\hat{H}(\omega) = 0$, where $\hat{P}(i\omega) = i\sum_{i=1}^{d} \omega_i A^i$. The discrete symbol associated with the standard second order accurate discretization of this system is

$$\hat{P}_h(\xi) = \frac{i}{h} \sum_{i=1}^{d} A^i \sin \xi_i = \hat{P}(ih^{-1} \sin \xi)$$

where we attached the subscript h to the discrete symbol to distinguish it from that of the continuum. We now construct the discrete symmetrizer

$$\hat{H}_h(\xi) \equiv \hat{H}(h^{-1} \sin \xi) \tag{5.43}$$

Conditions (5.22)–(5.23) are satisfied and condition (5.26) is sufficient for stability. The latter becomes $\sigma(k\hat{P}) = \lambda\chi\sigma(A(n)) \leq \alpha_0$, where $A(n) = \sum_{i=1}^{d} n_i A^i$, $n_i = \chi^{-1} \sin \xi_i$, so that $\sum_{i=1}^{d} n_i^2 = 1$. Since this inequality must hold for all ξ_i, and the quantity χ reaches its maximum value $\sqrt{d}$ at $\xi_i = \pm\pi/2$, we obtain the stability condition

$$\lambda \leq \frac{\alpha_0}{\sigma(A(n))\sqrt{d}} \tag{5.44}$$

In the symmetrizable hyperbolic case one can take the discrete symmetrizer to be the same

as that of the continuum (which, by definition, is independent of ω)

$$\hat{H}_h(\xi) = H \tag{5.45}$$

This analysis of first order strongly hyperbolic systems shows that if the characteristic speeds depend neither on the direction nor on the dimensionality of the problem, i.e. if $\sigma(A(n))$ depends neither on n nor on d, then the Courant limit has a $1/\sqrt{d}$ dependence.

5.7 Stability of the first order in time and second order in space wave equation

In this section we discuss stability properties of an approximation of the d dimensional wave equation written as a first order in time and second order in space system

$$\partial_t \phi(t,x) = \Pi(t,x) \tag{5.46}$$

$$\partial_t \Pi(t,x) = \sum_{i=1}^{d} \partial_i^2 \phi(t,x) \tag{5.47}$$

In the introduction we pointed out that the Cauchy problem for this system is not well-posed in L_2. One can expect that a direct application of Definition 4.2.3, which is based on the discrete L_2 norm, to a scheme approximating (5.46)–(5.47) would lead to the conclusion that the scheme is unstable. The first order reduction, however, is well-posed in L_2 (it is symmetric hyperbolic), hence the second order system satisfies an energy estimate with respect to

$$\|\boldsymbol{u}(\cdot,t)\|^2 = \int |\phi(x,t)|^2 + |\Pi(x,t)|^2 + \sum_{i=1}^{d} |\partial_i \phi(x,t)|^2 \, \mathrm{d}^d x \tag{5.48}$$

In this section we show stability for the standard discretization of this system, both by the pseudo-discrete reduction method given in Section 5.5, and by a direct discrete reduction in physical space. The two methods give equivalent results.

Following the method of lines, we first discretize space and leave time continuous,

$$\frac{d}{dt}\phi_j(t) = \Pi_j(t) \tag{5.49}$$

$$\frac{d}{dt}\Pi_j(t) = \sum_{i=1}^{d} D_{+i}D_{-i}\phi_j(t) \tag{5.50}$$

Using the technique described in Section 5.5, we see that the (principal) symbol of the second order semidiscrete problem

$$\hat{P} = \begin{pmatrix} 0 & 1 \\ -\Omega^2 & 0 \end{pmatrix} \qquad T^{-1} = \begin{pmatrix} i\Omega & 1 \\ -i\Omega & 1 \end{pmatrix} \tag{5.51}$$

has purely imaginary eigenvalues $\pm i\Omega$. The matrix T diagonalizes $\hat{P}$. The sum of the squared moduli of the characteristic variables gives the conserved energy (here $D = 1/2I$)

$$\hat{\boldsymbol{v}}^*(T^{-1})^* D T^{-1}\hat{\boldsymbol{v}} \equiv |i\Omega\hat{\phi}|^2 + |\hat{\Pi}|^2 = \Omega^2|\hat{\phi}|^2 + |\hat{\Pi}|^2 \tag{5.52}$$

By taking $K = 1$ in (5.38) we see that we have numerical stability with respect to the discrete norm

$$\|\boldsymbol{v}\|^2_{h,D_+} = \sum_j (\phi_j^2 + \Pi_j^2 + \sum_{i=1}^{d} (D_{+i}\phi_j)^2) h^d \tag{5.53}$$

provided that the von Neumann condition

$$\lambda \leq \alpha_0/(2\sqrt{d}) \tag{5.54}$$

which follows from $\sigma(k\hat{P}) = k\Omega = 2\lambda\chi_2 \leq \alpha_0$, is satisfied.

We now illustrate a different method for proving stability of this system. A *discrete reduction to first order* can be performed before going to Fourier space. We introduce the quantities

$$X_j^{(i)} = D_{+i}\phi_j \tag{5.55}$$

and obtain the reduced system

$$\frac{d}{dt}\phi_j(t) = \Pi_j(t) \tag{5.56}$$

$$\frac{d}{dt}\Pi_j(t) = \sum_{i=1}^{d} D_{-i}X_j^{(i)}(t) \tag{5.57}$$

$$\frac{d}{dt}X_j^{(i)}(t) = D_{+i}\Pi_j(t) \tag{5.58}$$

Notice that only if (5.55) is identically satisfied is the reduced system equivalent to the

original one. It is important to check whether the evolution equations (5.56)–(5.58) are compatible with this requirement. Let $C_j^{(i)}(t) \equiv X_j^{(i)} - D_{+i}\phi_j$. If we prescribe initial data such that $C_j^{(i)}(0) = 0$, then at later times $C_j^{(i)}(t) = 0$. This is a consequence of the fact that

$$\frac{d}{dt}C_j^{(i)}(t) = \frac{d}{dt}(X_j^{(i)}(t) - D_{+i}\phi_j(t)) = 0 \tag{5.59}$$

There is a one-to-one correspondence between solutions of (5.49)–(5.50) and those of (5.55)–(5.58). Furthermore, one can check that the time integrator does not spoil the propagation of the constraints.

Ignoring lower order terms, the symbol associated with the reduced system (5.56)–(5.58) is anti-Hermitian, therefore (5.23) is satisfied with $\hat{H} = 1$. The non-trivial eigenvalues of $\hat{P}$ are $\pm i\Omega$, the same as those of the original system (5.49)–(5.50). This proves that (5.54) is a necessary and sufficient condition for stability with respect to the discrete norm (5.53).

This specific discrete reduction to first order, and the pseudo-discrete reduction to first order described in Section 5.5 give equivalent results.

5.7.1 Fourth order accuracy

In hyperbolic problems a fourth order accurate spatial discretization requires significantly fewer grid points per wavelength for a given error (see [34]). The stability proof for the fourth order accurate discretization of the d-dimensional wave equation

$$\frac{d}{dt}\phi_j(t) = \Pi_j(t) \tag{5.60}$$

$$\frac{d}{dt}\Pi_j(t) = \sum_{i=1}^{d} D_{+i}D_{-i}\left(1 - \frac{h^2}{12}D_{+i}D_{-i}\right)\phi_j(t) \tag{5.61}$$

is similar to the second order accurate case. The discrete symbol and diagonalizing matrix are

$$\hat{P} = \begin{pmatrix} 0 & 1 \\ -\Delta^2 & 0 \end{pmatrix}, \qquad T^{-1} = \begin{pmatrix} i\Delta & 1 \\ -i\Delta & 1 \end{pmatrix} \tag{5.62}$$

where $\Delta^2 = \frac{4}{h^2}\sum_{i=1}^{d}\sin^2\frac{\xi_i}{2}\left(1 + \frac{1}{3}\sin^2\frac{\xi_i}{2}\right)$, has purely imaginary eigenvalues $\pm i\Delta$. Taking $D = 1/2I$ we get the conserved quantity

$$(T^{-1}\hat{\boldsymbol{v}})^* D\hat{T}^{-1}\hat{\boldsymbol{v}} = \Delta^2|\hat{\phi}|^2 + |\hat{\Pi}|^2 \tag{5.63}$$

Since $\Omega^2 \leq \Delta^2 \leq \frac{4}{3}\Omega^2$, by taking $K = 4/3$ in (5.38) we see that we have numerical stability with respect to the norm (5.53) provided that the principal symbol $\hat{P}$ satisfies $\sigma(k\hat{P}) \leq \alpha_0$. This gives a stability limit of $\lambda \leq \sqrt{3}\alpha_0/(4\sqrt{d})$.

5.7.2 A note about the D_0 norm and the D_0^2 discretization

Replacing the one sided difference operators D_{+i} with centred difference operators D_{0i} in the norm (5.53) leads to difficulties, as the D_0 norm does not capture the highest frequency mode. In fact, it is possible to construct a family of solutions of (5.49)–(5.50) proportional to $(-1)^j$ for which the D_0 energy estimate fails. For this purpose it is sufficient to consider $\phi_j(t) = (-1)^j \cos(2t/h)$, $\Pi_j(t) = -2/h(-1)^j \sin(2t/h)$, which gives

$$\frac{\|\boldsymbol{v}(t)\|_{h,D_0}}{\|\boldsymbol{v}(0)\|_{h,D_0}} = \left(\cos^2\frac{2t}{h} + \frac{4}{h^2}\sin^2\frac{2t}{h}\right)^{1/2} \tag{5.64}$$

where $\|\boldsymbol{v}(t)\|^2_{h,D_0} = \sum_j(\phi_j^2 + \Pi_j^2 + (D_0\phi_j)^2)h$. It it not possible to find constants K and α such that the ratio is bounded by $Ke^{\alpha t}$, independently of the space step h.

It has been suggested that the use of D_0^2 rather than D_+D_- for the second spatial derivatives may improve the stability properties of a second order in space scheme [16, 8]. To investigate this we study the wave equation in one space dimension discretized as

$$\frac{d}{dt}\phi_j(t) = \Pi_j(t) \tag{5.65}$$

$$\frac{d}{dt}\Pi_j(t) = D_0^2\phi_j(t) \tag{5.66}$$

The eigenvalues of $k\hat{P}$ are $\pm i\lambda\sin\xi$, which shows that the von Neumann condition is satisfied as long as $\lambda \leq \alpha_0$. Both the stencil and the maximum time step compatible with the von Neumann condition are twice what they are for the D_+D_- discretization. However, for a given spatial resolution the numerical speed of propagation has an error which is four times that of the D_+D_- case (see Section 7.6.1).

So far, we have only shown that the scheme is unstable if $\lambda > \alpha_0$. By looking at the discrete symbol

$$\hat{P}(\xi) = \begin{pmatrix} 0 & 1 \\ -\frac{1}{h^2}\sin^2\xi & 0 \end{pmatrix} \tag{5.67}$$

we see that there might be a problem for $|\xi| = \pi$. In this case the symbol is not diago-

nalizable. To explicitly show that the system (5.65)–(5.66) is unstable with respect to the norm

$$\|\boldsymbol{v}\|_{h,D_+}^2 = \sum_j \left(\phi_j^2 + \Pi_j^2 + (D_+\phi_j)^2\right) h \tag{5.68}$$

it is sufficient to consider the family of initial data $\phi_j(0) = 0, \Pi_j(0) = (-1)^j$, generating the solution $\phi_j(t) = (-1)^j t, \Pi_j(t) = (-1)^j$. As $h \to 0$ the ratio

$$\frac{\|\boldsymbol{v}(t)\|_{h,D_+}}{\|\boldsymbol{v}(0)\|_{h,D_+}} = \left(1 + t^2 + \frac{4t^2}{h^2}\right)^{1/2} \tag{5.69}$$

grows without bound.

Had we chosen the D_0-norm, however, we would have concluded that the scheme satisfies the required estimate. This is because this norm does not capture the highest frequency mode $\phi_j = (-1)^j$. A desirable requirement of a norm is that if a scheme is stable with respect to that norm, then it will remain stable with respect to the same norm when perturbed with lower order terms (independently of how these are discretized). The modified problem

$$\frac{d}{dt}\phi_j(t) = \Pi_j(t) \tag{5.70}$$

$$\frac{d}{dt}\Pi_j(t) = D_0^2\phi_j(t) - D_+\phi_j(t) \tag{5.71}$$

admits the family of exponentially growing solutions $\phi_j(t) = (-1)^j \exp(\sqrt{2/h}t)$, $\Pi_j(t) = (-1)^j\sqrt{2/h}\exp(\sqrt{2/h}t)$ which leads to unbounded growth in the ratio

$$\frac{\|\boldsymbol{v}(t)\|_{h,D_0}}{\|\boldsymbol{v}(0)\|_{h,D_0}} = \exp\left(\sqrt{\frac{2}{h}}t\right) \tag{5.72}$$

If we want to be able to decide whether a scheme is stable or not just by looking at the principal part of the discrete system, then we must conclude that the D_0-energy is not a suitable energy.

We note that the requirement that stability should not depend on how lower order terms are discretized was crucial. If we restrict ourselves to the perturbation $D_0\phi_j$, then the scheme is still stable with respect to the D_0-energy. If one wants to be able to discretize lower order terms freely, as we do, then one is forced to reject the D_0^2 discretization.

Clearly it is the presence of high frequency modes that makes the D_0^2 discretization unstable with respect to the D_+ norm. The introduction of a mechanism that damps high frequency

modes, such as artificial dissipation, may restore stability. In the system

$$\begin{aligned}\frac{d}{dt}\phi_j &= \Pi_j - \sigma h^3 (D_+ D_-)^2 \phi_j \\ \frac{d}{dt}\Pi_j &= D_0^2 \phi_j - \sigma h^3 (D_+ D_-)^2 \Pi_j\end{aligned}$$

the same family of initial data used to prove instability of (5.65)–(5.66) gives

$$\frac{\|\boldsymbol{v}(t)\|_{h,D_+}}{\|\boldsymbol{v}(0)\|_{h,D_+}} = (1 + t^2 + 4t^2/h^2)^{1/2} e^{-16\sigma t/h} \tag{5.73}$$

which does not grow without bound.

5.8 Stability of the linearized NOR system

The NOR formulation of Einstein's equations linearized about Minkowski space with zero shift and unperturbed densitized lapse ($\alpha = \sqrt{\det(\gamma_{ij})}$) has the form

$$\partial_t \gamma_{ij} = -2K_{ij} \tag{5.74}$$

$$\partial_t K_{ij} = \partial_{(i} w_{j)} - \frac{1}{2}\partial^k \partial_k \gamma_{ij} \tag{5.75}$$

$$\partial_t w_i = 0 \tag{5.76}$$

This system corresponds to the one in [32] with the choice of parameters $a = b = \sigma = 1$. We also choose $\rho = 2$ as this removes a mixed second derivative term which causes complications in the analysis of the semidiscrete system. Fourier transforming the semidiscrete system obtained with the standard second order accurate discretization yields

$$\frac{d}{dt}\hat{\gamma}_{ij} = -2\hat{K}_{ij} \tag{5.77}$$

$$\frac{d}{dt}\hat{K}_{ij} = \frac{i}{h}\sin\xi_{(i}\hat{w}_{j)} + \frac{2}{h^2}\chi_2^2\hat{\gamma}_{ij} \tag{5.78}$$

$$\frac{d}{dt}\hat{w}_i = 0 \tag{5.79}$$

Introducing $\hat{X}_{ij} = \frac{i}{h}\chi_2\hat{\gamma}_{ij}$ reduces the system to first order. After dropping lower order terms (those not depending on ξ_i) we get

$$\frac{d}{dt}\hat{\gamma}_{ij} = 0 \tag{5.80}$$

$$\frac{d}{dt}\hat{K}_{ij} = \frac{i}{h}\sin\xi_{(i}\hat{w}_{j)} - \frac{2i}{h}\chi_2\hat{X}_{ij} \tag{5.81}$$

$$\frac{d}{dt}\hat{w}_i = 0 \tag{5.82}$$

$$\frac{d}{dt}\hat{X}_{ij} = -\frac{2i}{h}\chi_2\hat{K}_{ij} \tag{5.83}$$

The eigenvalues of the symbol are imaginary. A conserved energy for this system will lead to a discrete symmetrizer. The obvious starting point is

$$E_0 \equiv \sum_\xi \sum_{ij}(|\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + \sum_{ij}|\hat{X}_{ij}|^2) + \sum_i |\hat{w}_i|^2 \tag{5.84}$$

From now on, the summation signs over i and j will be dropped, and these sums will be implicit. The time derivative of E_0 is

$$\frac{dE_0}{dt} = \sum_\xi 2\mathrm{Re}\left[\frac{i}{h}\hat{K}^*_{ij}\sin\xi_{(i}\hat{w}_{j)}\right] \tag{5.85}$$

Hence E_0 is not a conserved energy. Defining $E = E_0 + E_1$, the requirement that E is conserved is equivalent to

$$\frac{dE_1}{dt} = -\frac{dE_0}{dt} = \sum_\xi 2\mathrm{Re}\left[-\frac{i}{h}\hat{K}^*_{ij}\sin\xi_{(i}\hat{w}_{j)}\right] \tag{5.86}$$

Since $\hat{w}_j$ is a constant, using (5.83),

$$E_1 = -\sum_\xi 2\mathrm{Re}\left[\hat{X}^*_{ij}\sin\xi_{(i}\hat{w}_{j)}/(2\chi_2)\right] \tag{5.87}$$

and

$$E = \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + |\hat{w}_i|^2 - 2\mathrm{Re}\left[\hat{X}^*_{ij}\sin\xi_{(i}\hat{w}_{j)}/(2\chi_2)\right] \tag{5.88}$$

and $dE/dt = 0$. Since $\partial_t\hat{w}_i = 0$, $|\hat{w}_i|^2$ can be multiplied by a constant $c > 0$ and E will still be conserved. The reason for this will become apparent later. E is thus redefined as

$$E = \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + c|\hat{w}_i|^2 - 2\mathrm{Re}\left[\hat{X}^*_{ij}\sin\xi_i\hat{w}_j/(2\chi_2)\right] \tag{5.89}$$

For E to represent a symmetrizer ($E = \hat{u}^* \hat{H}(\xi)\hat{u}$), it must be equivalent to the identity. Defining

$$L \equiv u^* I u = \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + |\hat{w}_i|^2 \tag{5.90}$$

we require that $\exists\, K$ such that

$$K^{-1} L \leq E \leq K L \tag{5.91}$$

The following inequalities will be used:

$$2\mathrm{Re}\,[a^* b] \leq \gamma |a|^2 + |b|^2/\gamma \tag{5.92}$$

$$2\mathrm{Re}\,[a^* b] \geq -\gamma |a|^2 - |b|^2/\gamma \tag{5.93}$$

for $a, b \in \mathbb{C}$ and $\gamma \in \mathbb{R} > 0$. Using (5.93) multiplied by -1,

$$-2\mathrm{Re}\left[\hat{X}^*_{ij} \sin\xi_i \hat{w}_j/(2\chi_2)\right] \leq |\hat{X}_{ij}|^2 + |\sin\xi_i \hat{w}_j/(2\chi_2)|^2 \tag{5.94}$$

Using this with (5.88), we have that

$$E \leq \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + c|\hat{w}_i|^2 + |\hat{X}_{ij}|^2 + |\sin\xi_i \hat{w}_j/(2\chi_2)|^2 \tag{5.95}$$

$$= \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + c|\hat{w}_i|^2 + |\hat{X}_{ij}|^2 + |\hat{w}_j|^2 \chi^2/(4\chi_2^2) \tag{5.96}$$

$$= \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + 2|\hat{X}_{ij}|^2 + |\hat{w}_j|^2 \left(c + \frac{\chi^2}{4\chi_2^2}\right) \tag{5.97}$$

By using

$$\frac{\chi^2}{\chi_2^2} = \frac{\sum_i \sin^2\xi_i}{\sum_i \sin^2(\xi_i/2)} \leq 1 \tag{5.98}$$

we obtain

$$E \leq \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + 2|\hat{X}_{ij}|^2 + |\hat{w}_j|^2 \left(c + \frac{\chi^2}{4\chi_2^2}\right) \tag{5.99}$$

$$\leq \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + 2|\hat{X}_{ij}|^2 + |\hat{w}_j|^2 \left[c + \frac{1}{4}\right] \tag{5.100}$$

$$\leq 2\left[c + \frac{1}{4}\right] \sum_{\xi} |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + |\hat{w}_j|^2 \tag{5.101}$$

$$\leq 2\left[c + \frac{1}{4}\right] L \tag{5.102}$$

The other part of inequality (5.91) proceeds as follows. Using (5.93) with $a = \sin\xi_i\hat{w}_j/(2\chi_2)$, $b = \hat{X}^*_{ij}$ and $\gamma = 2$,

$$-2\mathrm{Re}\left[\hat{X}^*_{ij}\sin\xi_i\hat{w}_j/(2\chi_2)\right] \geq -\frac{1}{2}|\hat{X}_{ij}|^2 - 2|\sin\xi_i\hat{w}_j/(2\chi_2)|^2 \tag{5.103}$$

Hence

$$\begin{aligned} E &\geq \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + c|\hat{w}_i|^2 - \frac{1}{2}|\hat{X}_{ij}|^2 - 2|\sin\xi_i\hat{w}_j/(2\chi_2)|^2 && (5.104)\\ &\geq \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + \frac{1}{2}|\hat{X}_{ij}|^2 + \left[c - \chi^2/(2\chi_2^2)\right]|\hat{w}_j|^2 && (5.105)\\ &\geq \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + \frac{1}{2}|\hat{X}_{ij}|^2 + \left[c - \frac{1}{2}\right]|\hat{w}_j|^2 && (5.106) \end{aligned}$$

Setting $c = \frac{3}{2}$,

$$\begin{aligned} E &\geq \sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + \frac{1}{2}|\hat{X}_{ij}|^2 + |\hat{w}_j|^2 && (5.107)\\ &\geq \frac{1}{2}\sum_\xi |\hat{\gamma}_{ij}|^2 + |\hat{K}_{ij}|^2 + |\hat{X}_{ij}|^2 + |\hat{w}_j|^2 && (5.108)\\ &\geq \frac{1}{2}L && (5.109) \end{aligned}$$

Hence

$$\begin{aligned} \frac{1}{2}L \leq E \leq \frac{7}{2}L && (5.110)\\ \frac{2}{7}L \leq E \leq \frac{7}{2}L && (5.111) \end{aligned}$$

as required.

By theorem 5.3.1, the system is stable provided that the von Neumann condition

$$\lambda \leq \frac{\alpha_0}{2\sqrt{d}} \tag{5.112}$$

is satisfied.

So the main result of this section is that the standard discretization of the NOR equations linearized about a Minkowski background in Cartesian coordinates is stable with respect

to the norm

$$\sum_{\text{gridpoints}} \left(\sum_{ij} \gamma_{ij}^2 + \sum_{ij} K_{ij}^2 + \sum_{i} w_i^2 + \sum_{kij} (D_{+k}\gamma_{ij})^2 \right) h^3 \tag{5.113}$$

provided (5.112) is satisfied.

5.9 The ADM system

With a densitized lapse function, $\alpha = \sqrt{\det(\gamma_{ij})}$, the ADM equations linearized around Minkowski in Cartesian coordinates take the form

$$\partial_t \gamma_{ij} = -2K_{ij} \tag{5.114}$$

$$\partial_t K_{ij} = \partial_k \partial_{(i}\gamma_{j)k} - \frac{1}{2}\partial^k \partial_k \gamma_{ij} - \partial_i \partial_j t \tag{5.115}$$

The symbol $\hat{P}(i\omega)$ of (5.114)–(5.115) is not diagonalizable and neither is that of its differential or pseudo-differential reduction. The family of solutions in which the only non vanishing components are $\gamma_{1A} = \sin(\omega x)t$, $K_{1A} = -\sin(\omega x)/2$, where $A = 2, 3$, can be used to show ill-posedness explicitly. It gives

$$\frac{\|\boldsymbol{u}(t,\cdot)\|}{\|\boldsymbol{u}(0,\cdot)\|} = \left(1 + 4t^2 + 4\omega^2 t^2\right)^{1/2} \tag{5.116}$$

where $\|u(t,\cdot)\|^2 = \|\gamma_{ij}(t,\cdot)\|^2 + \|K_{ij}(t,\cdot)\| + \|\partial_k \gamma_{ij}(t,\cdot)\|^2$. The ratio cannot be bounded by $Ke^{\alpha t}$ with K and α independent of ω.

To see that the second order accurate standard discretization is unstable we take $\gamma_{1A} = (-1)^j t$ and $K_{1A} = (-1)^{j+1}/2$. As in the continuum, the ratio

$$\frac{\|\boldsymbol{v}(t)\|_{h,D_+}}{\|\boldsymbol{v}(0)\|_{h,D_+}} = \left(1 + 4t^2 + 16\frac{t^2}{h^2}\right)^{1/2} \tag{5.117}$$

cannot be bounded. We can nevertheless compute the von Neumann condition, which is given by

$$\lambda \leq \frac{\sqrt{3}\alpha_0}{2\sqrt{7d}} \tag{5.118}$$

In [3] stability tests were performed with the nonlinear version of this formulation. The domain used consisted of a thin channel, with an even number N of grid points in one

spatial direction and 3 grid points in the other two directions. A one-dimensional von Neumann analysis gives the limit $\lambda \leq \frac{\sqrt{3}\alpha_0}{2\sqrt{7}}$. However, this would not capture the fact that there could be exponentially growing modes with non trivial dependence in the two thin directions. Figure 2 in [3] confirms that with a Courant factor of $\lambda = 1/2$ there is a von Neumann instability.

We have not yet understood whether the lack of diagonalizability of the symbol, either at the continuum or the semidiscrete level, implies that the system will fail to be well-posed (stable) in the obvious norms. We suspect that this is the case.

Although the symbol associated with the continuum system (5.114) and (5.115) has three Jordan blocks of size two for any $\omega \neq 0$, interestingly, the symbol associated with the semidiscrete problem obtained with the standard second order accurate discretization can have rather different properties. For Fourier modes travelling in directions parallel to the axis the continuum result still holds. Thus, 1D stability tests will exhibit linear frequency dependent growth. Fourier modes for which one and only one of the frequencies vanishes (2D case, excluding directions parallel to the axes) have a single Jordan block of size 2. Again, one can expect to see frequency dependent linear growth in stability tests, but not as clearly. Finally, for Fourier modes with $\xi_i \neq 0$ for $i = 1, 2, 3$ the symbol is diagonalizable. Hence it can be difficult to experimentally see frequency dependent growth in 3D, as the fraction of modes which grow in a frequency dependent manner is proportional to $1/N$. This might be one of the reasons that the ADM system was not immediately dismissed by numerical relativists.

5.10 Summary

In this chapter, the concept of a *discrete symmetrizer* has been introduced for proving stability of a fully first order finite difference scheme. The concept of the *absolute stability region* of a time integrator has been used to allow conclusions concerning stability of the fully discrete scheme to be made from properties of the semidiscrete scheme.

We have introduced the idea of a *discrete reduction to first order in Fourier space* by analogy with the continuum technique, and have shown how stability properties of a second order in space system can be determined.

As examples, stability is shown for standard discretizations of first order strongly hyperbolic systems, as well as the first order in time and second order in space wave equation. The

NOR formulation of the Einstein equations has been linearized about Minkowski spacetime in Cartesian coordinates, and the standard discretization of this system has been shown to be stable. We believe that this is the first time that this has been done for the Einstein equations in first order in time and second order in space form.

Chapter 6

The Kranc package for automated code generation

6.1 Introduction

When programming a computer to solve a system of partial differential equations, a language such as C or Fortran is typically used. These languages do not have built-in support for abstract mathematical constructs such as tensors. For simple systems such as the wave equation, it is feasible to write the necessary computer code by hand. However, when more complicated systems, such as the full 3D nonlinear Einstein equations, are required, this task becomes daunting. Whilst not impossible, the process is long and error-prone, and the resulting code can be difficult to debug and maintain.

A collaboration between Sascha Husa and Christiane Lechner at the Albert Einstein Institute and myself resulted in a suite of Mathematica packages (about 6000 lines in total) which accepts tensorial evolution equations in abstract index notation, and generates either C or Fortran code for performing the numerical evolution using the Cactus infrastructure (see below). The Mathematica suite is called *Kranc* for "KRanc Assembles Numerical Code". Kranc has so far been used to implement the 3D nonlinear ADM, NOR, BSSN, ST and Z4 formulations of the Einstein equations, in addition to standard test cases such as the wave equation, the Klein-Gordon equation, and Maxwell's equations. The development of the Kranc package is described in [37].

6.2 Cactus

The *Cactus Computational Toolkit* is an open-source problem solving environment originally developed in the numerical relativity community. It is arranged as a central *flesh* and a collection of modules called *thorns* which all communicate with the flesh. Many thorns are provided, and the user writes additional thorns in C or Fortran which solve their particular physics problem. Cactus is particularly suited to the numerical solution of time dependent partial differential equations.

Kranc is concerned with taking an abstract mathematical description of a system of PDEs and producing working computer code. It does this by generating Cactus thorns, allowing use of all the infrastructure provided by Cactus.

For example, Kranc makes uses of existing Cactus thorns which provide:

- Parameter file parsing.
- Memory management for variables associated with the computational grid.
- Scheduling of parts of the code based upon parameters.
- Standard efficient time integrators such as fourth order Runge-Kutta and iterative Crank-Nicolson via the *MoL* thorn written by I. Hawke.
- Mesh refinement [56]; i.e. using variable resolution across the numerical grid, so that the computational resources are focused on interesting parts of the simulation.
- Automatic parallelization of the code to run across multiple processors on a supercomputer or cluster, both to improve computational speed and to use larger grids than can be stored in the memory of a single node.
- Output of grid variables to permanent storage in a structured format.

These tasks are completely divorced from the physics and numerical analysis side of the problem, but are necessary in most numerical codes.

6.2.1 Cactus for numerical relativity

In the context of numerical relativity, any Cactus code can take advantage of thorns providing:

- Initial data for the Einstein equations. For example, the *Exact* thorn can compute initial data for a large number of spacetimes, including Minkowski, Schwarzschild and Kerr in various coordinate systems.
- Algorithms for quickly finding apparent [62] and event [26] horizons in dynamical spacetimes.
- Standard methods for extracting gravitational wave information.
- Standard *ADM* grid variables. The *ADMBase* thorn defines variables for the three-metric and extrinsic curvature that can be used to communicate data between thorns. For example, thorns for calculating initial data can populate these variables, and thorns for performing wave extraction or other analysis tasks can use these variables to determine details of the spacetime. In this way, these types of thorns do not need to be aware of the precise variables used by the time evolution system. The evolution system translates between the ADM variables and the evolution variables.

6.3 Overview of the Kranc system

There are five types of Kranc thorn:

- A *base thorn* defines the grid functions which the simulation will use.
- A *MoL thorn* computes the right hand sides of the evolution equations so that the time integrator can compute the evolved variables at the next time step. This is the most important type of thorn as it determines the system of partial differential equations being solved. The time integration methods in Cactus require that those grid functions containing evolved variables must be registered as such, and the MoL thorn performs this registration.
- A *setter thorn* performs a user-specified calculation at each point of the grid. This will typically set certain grid variables as functions of others, and can be used for various purposes including making a change of variables or computing intermediate quantities from evolved variables.
- A *translator thorn* is a special case of a setter thorn which converts between the evolved variables and some other set of variables (for example, the ADMBase variables used by initial data and analysis thorns in the context of numerical relativity)

- An *evaluator thorn* computes quantities such as norms and constraints that are used in the analysis of the constructed solution. The calculations are invoked only when the quantities concerned are output to permanent storage, which improves efficiency when output is not required at each time step.

These five thorn types allow complete codes to be assembled.

Common to many of these thorn types is the idea of assigning new values to grid functions in a loop over grid points based upon evaluating expressions involving other grid functions. To encapsulate this, we define a Kranc structure called a *calculation*. Calculations contain lists of assignment statements for different grid functions, and these are evaluated at each point on the grid. Calculations can also contain temporary variables called *shorthands* into which are placed intermediate expressions which are used later in the calculation. Many of the thorn types are based around calculation structures.

6.4 Kranc Design

Kranc is composed of several Mathematica packages. Each of these human readable scripts performs a distinct function. The user only needs to be concerned with calling functions from the KrancThorns package. This package contains functions for creating the different types of Kranc thorn.

The diagram in Figure 6.4 illustrates the relationships between the Kranc packages KrancThorns, TensorTools, CodeGen, Thorn and MapLookup, which are described in the following subsections. Separating the different logically independent components of Kranc into different packages promotes code reuse. For example, none of the thorn generation packages need to know anything about tensors, and none of the packages other than CodeGen need to know the programming language in which the thorn is being generated (C or Fortran). We have chosen to define several types of thorn (setter, evaluator, *etc.*) but the mechanics of producing a thorn implemented in Thorn and CodeGen are completely independent of this decision.

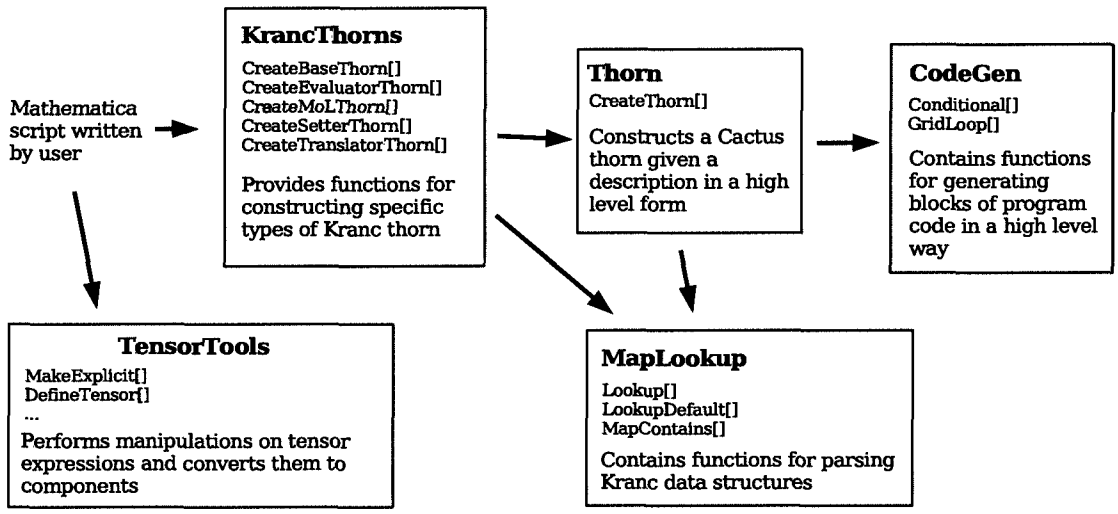

Figure 6.1: Relationships between Kranc packages: Each block represents a package, with the main functions it provides indicated with square brackets. An arrow indicates that one package calls functions from another

6.4.1 Package: KrancThorns

The different types of Cactus thorn used in a Kranc arrangement are the MoL, setter, base, translator and evaluator thorns. The KrancThorns package provides functions to create thorns of these types given high level descriptions. These functions are the ones directly called by users. Internally the KrancThorns package uses the Thorn package to create the Cactus thorns.

Types of arguments

Mathematica allows two types of arguments to be passed to a function. *positional arguments* and *named arguments* (referred to in the Mathematica book as *optional arguments*). It is possible for some named arguments to be omitted from a function call; in this case a suitable default will be chosen. Positional arguments are useful when there are few arguments to a function, and their meaning is clear in the calling context. Named arguments are preferred when there are many arguments, as the argument names are given explicitly in the calling context.

For each type of Kranc thorn, there is a function to create it (`Create*Thorn`). There is a certain set of named arguments ("Common named arguments") which can be passed to any of these functions (e.g. the name of the Thorn to create, where to create it, etc). Then,

for each type of thorn, there is a specific set of named arguments specifically for that thorn type. All of the functions accept some positional arguments as well.

Common data structures

Kranc consists of several packages which need to pass data between themselves in a structured way. Mathematica does not have the concept of a C++ class or a C structure, in which collections of named objects are grouped together for ease of manipulation. Instead, we have defined a *Kranc structure* as a list of rules of the form *key* `->` *value*. We have chosen to use the Mathematica rule symbol "`->`" for syntactic convenience. For example, one might describe a person using a "Person" structure as follows:

```
alice = {Name -> "Alice",
         Age -> 20,
         Gender -> Female}
```

Once a structure has been built up, it can be parsed with the `lookup` function in the MapLookup package. `lookup[structure, key]` returns the value in `structure` corresponding to `key`. For example, `lookup[alice, Age]` would return the number 20. This usage mirrors what is known as an association list (or alist) in LISP style languages. Based on this concept a number of data structures have been defined which will be used to describe the thorns to construct. Each of these data structures is introduced below.

Data structure: PartialDerivatives

The user can define partial derivative operators and associated finite difference approximations of these operators. This allows different discretizations of the PDE system.

A finite difference operator maps grid functions to grid functions. We restrict to those operators which are polynomials in *shift* operators. In one dimension, the shift operator E_+ is defined as

$$E_+ v_j \equiv v_{j+1} \tag{6.1}$$

It is clear that

$$(E_+)^n v_j = v_{j+n} \tag{6.2}$$

and negative powers n take on the obvious meaning. In three dimensions, there is one shift operator for each dimension:

$$E_{+1}v_j \equiv v_{j+(100)} \qquad E_{+2}v_j \equiv v_{j+(010)} \qquad E_{+3}v_j \equiv v_{j+(001)} \tag{6.3}$$

where here $j = j_1 j_2 j_3$ is a multi-index.

The PartialDerivatives structure is a list of definitions of partial derivative operators in terms of finite difference approximations:

```
{ name[i_, j_, ...]  -> defn, ... }
```

where *name* is the name for the partial derivative, and *defn* is an algebraic expression in shift operators representing the difference operator. The shift operator E_{+i} is written as `shift[i]`. The form `spacing[i]` can be used in *defn* to represent the grid spacing in the i direction. The parameters `i, j, ...` are used in *defn* to represent the direction of differentiation for the first, second, etc. derivatives. Partial derivatives with the same name but a different number of arguments (i.e. for first and second derivatives) are allowed in the PartialDerivatives structure.

Since the definitions of the difference operators are written in terms of Mathematica expressions, higher level operators can be constructed from `shift` and `spacing`. For example, Kranc predefines

```
DPlus[n_]  := (shift[n] - 1)/spacing[n];
DMinus[n_] := (1 - 1/shift[n])/spacing[n];
DZero[n_]  := (DPlus[n] + DMinus[n])/2;
```

As an example, we give here a PartialDerivatives structure containing the definition of the standard second order accurate difference operators, as well as the D_0^2 discretization.

```
derivs = {
  PDstandard2nd[i_] -> DZero[i],
  PDstandard2nd[i_, j_] -> DPlus[i] DMinus[j],
  PDzero2nd[i_] -> DZero[i],
  PDzero2nd[i_, j_] -> DZero[i] DZero[j]
}
```

In a calculation, a partial derivative is written in the form

```
name[gridfunction, i, j, ...]
```

For example, a one dimensional advection equation $\partial_t u = \partial_x u$ with semidiscrete form $\partial_t v_j = D_{01} v_j$ could be described as

```
dot[v] -> PDstandard2nd[v,1]
```

The PartialDerivatives structure can also be used to define operators for artificial dissipation. Given a semidiscrete scheme

$$\partial_t v(t)_j = F_j(v(t);t) \tag{6.4}$$

we can add Kreiss-Oliger style artificial dissipation by modifying the scheme to read

$$\partial_t v_j(t) = F_j(v(t);t) - \sigma \sum_i h_i^3 (D_{+i} D_i)^2 v_j \tag{6.5}$$

We define a differencing operator `Diss2nd` in the PartialDerivatives structure with no directional arguments

```
Diss2nd[] -> - sigma Sum[spacing[i]^3 (DPlus[i] DMinus[i])^2,
                         {i, 1, 3}]
```

using the standard Mathematica function for summations. An evolution equation representing the advection equation with dissipation could then be written as

```
dot[v] -> PDstandard2nd[v,1] + Diss2nd[v]
```

A PartialDerivatives structure is given as an argument to the thorn generation functions.

Data structure: GroupDefinition

A `GroupDefinition` structure lists the grid functions that are members of a specific Cactus group. A list of such structures should be supplied to all the KrancThorns functions so that Kranc can determine which group each grid function belongs to.

Data structure: Calculation

Calculation structures are the core of the Kranc system. The user provides a list of equations of the form

variable -> *expression*

When the calculation is performed, for each point in the grid, *expression* is evaluated and placed into the grid function *variable*. Here *expression* may contain partial derivatives of grid functions which have been defined in a PartialDerivatives structure.

The user may specify intermediate (non-grid) variables called *shorthands* which can be used as *variable* for precomputing quantities which will be used later in the calculation. To identify these variables as shorthands, they must be listed in a *Shorthands* entry of the Calculation.

The arrangement of the terms in the equations can have a marked effect on both compile time and run time. It is often helpful to tell Mathematica to collect the coefficients of certain types of term, rather than expanding out entire expressions. To this end, the user can include a *CollectList* entry in a calculation; this is a list of variables whose coefficients should be collected during simplification.

There is the facility for performing multiple loops in a single calculation structure; this can be used to set a grid function in one loop, then evaluate derivatives of it in a later loop. For this reason, the equations are given as a list of lists of equations.

Note that the system is not designed to allow the same grid function to be set more than once in a single loop of a calculation.

The following example is taken from the Kranc implementation of the NOR formulation. It is a calculation which describes the time evolution equation for the lapse α in harmonic slicing. It uses the TensorTools package to represent tensorial quantities.

```
lapseEvolveCalc = {
  Shorthands -> {trK, hInv[ua,ub]},
  Equations ->
    {{
       hInv[ua,ub] -> MatrixInverse[h[ua,ub]],
       trK -> K[la,lb] hInv[ua,ub],
       dot[alpha] -> alpha^2 trK
    }}
};
```

See Appendix D for further information.

Data structure: GroupCalculation

A `GroupCalculation` structure associates a group name with a `Calculation` which is used to update the grid functions in that group. This is used when creating evaluator thorns, where the calculations are triggered by requests for output for specific groups.

6.4.2 Package: TensorTools

The TensorTools package was written specifically for the Kranc system, though it is in no way tied to it. It is necessary to perform certain operations on tensorial quantities, and there was no free software available which met the requirements.

TensorTools has the following features:

- It expands covariant derivatives in terms of partial derivatives and Christoffel symbols (more than one covariant derivative can be defined)
- It expands Lie derivatives in terms of partial derivatives
- Dummy indices can be automatically relabelled to avoid conflicts
- Abstract tensor expressions can be converted into component expressions

Representation of tensor quantities

Tensorial expressions are entered in the same syntax as is used by MathTensor, a commercial tensor manipulation package which can be used instead of TensorTools. An abstract tensor consists of a *kernel* and an arbitrary number of abstract *indices*, each of which can be *upper* or *lower*. Abstract indices are alphabetical characters (a-z, A-Z) prefixed with either an l or a u depending on whether the index is considered to be lower or upper. The tensor is written using square brackets as

```
kernel [ indices separated by commas ]
```

For example, $T_a{}^b$ would be written as `T[la,ub]`. There is no automatic index raising or lowering with any metric. Entering a tensorial expression causes it to be displayed in standard mathematical notation:

```
In :=   T[la,lb]
```

Out = T_{ab}

Internally, tensors are represented as `Tensor[`*kernel*`, TensorIndex[`*label*`, `*type*`], ...]` where *label* is the alphabetical index, and *type* is either "u" or "l" depending on the position of the index. This representation helps in pattern matching, and allows TensorTools to identify whether a certain object is a tensor or not.

Expansion of tensorial expressions into components

As an example, the TensorTools function `MakeExplicit` converts an expression containing abstract tensors into a list of component expressions:

```
In :=   MakeExplicit[T[la, lb]g[ub, uc]]
```

```
Out =   { g11 T11  + g21 T12 + g31 T13, g12 T11 + g22 T12 + g32 T13,
          g13 T11  + g23 T12 + g33 T13, g11 T21 + g21 T22 + g31 T23,
          g12 T21  + g22 T22 + g32 T23, g13 T21 + g23 T22 + g33 T23,
          g11 T31  + g21 T32 + g31 T33, g12 T31 + g22 T32 + g32 T33,
          g13 T31  + g23 T32 + g33 T33}
```

Note here that there is no distinction made between upper and lower indices in the component form. TensorTools was written mainly for automated code generation rather than symbolic manipulation; different kernels should be used for the different forms if this is a problem.

Covariant derivatives

TensorTools allows the user to define more than one covariant derivative. The following defines a covariant derivative operator `CD` with Christoffel symbol `H`:

```
DefineConnection[CD,H]
```

The function `CDtoPD` is used to replace covariant derivatives with partial derivatives in any expression:

```
In :=   CDtoPD[CD[V[ua],lb]]
```

$$\texttt{Out} = \quad V^a{}_{,\,b} + H^a{}_{bc} V^c$$

The function `MakeExplicit` will automatically do this before converting expressions into components. In order to convert an expression containing a covariant derivative into components, TensorTools first simplifies the expression. In the following, x and y represent expressions which may contain tensorial indices. The following steps are performed to simplify the expression:

- Replace any high order covariant derivatives with repeated application of a first order covariant derivative. This ensures that we only need to know how to evaluate a first derivative.

$$\nabla_d \nabla_a V^b \rightarrow \nabla_d (\nabla_a V^b)$$

- Replace the covariant derivative of a product using the Leibniz rule:

$$\nabla_a (xy) \rightarrow (\nabla_a x) y + x (\nabla_a y)$$

- Replace the covariant derivative of a sum using the linearity property:

$$\nabla_a (x + y) \rightarrow \nabla_a x + \nabla_a y$$

- Replace the covariant derivative of an arbitrary expression containing tensorial indices with its expansion in terms of a partial derivative and Christoffel symbols, one for each index in the expression: e.g.

$$\nabla_a V^b \to \partial_a V^b + \Gamma^b{}_{ac} V^c$$

Lie derivatives

The Lie derivative of an expression x with respect to a vector V is written

```
Lie[x,V]
```

where V has been registered using `DefineTensor` and is written *without* indices. The function `LieToPD` is used to replace Lie derivatives with partial derivatives:

```
In :=   LieToPD[Lie[T[ua,lb], V]]
```

$$\texttt{Out} = \quad T^a{}_{b,c} V^c + T^a{}_c V^c{}_{,b} - T^c{}_b V^a_{,c}$$

Lie derivatives of products and sums are supported. The function `MakeExplicit` will automatically perform this replacement before converting expressions into components.

Automatic dummy index manipulation

When two expressions both containing a dummy index b are multiplied together, one dummy index is relabelled so as not to conflict with any other index in the resulting expression:

```
In :=   (T[la, lb]g[ub, uc])v[ub, ld, lb]
```

$$\texttt{Out} = \quad T_{ab} g^{bc} V^e{}_{de}$$

This requires that every multiplication be checked for tensorial operands. This can be a performance problem, so the feature can be enabled and disabled with `SetEnhancedTimes[True]` and `SetEnhancedTimes[False]`. It is enabled by default.

6.4.3 Package: CodeGen

During the development of the Kranc system, we explored two different approaches to generating Cactus files using Mathematica as a programming language. Initially, a very straightforward system was used whereby C statements were included almost verbatim in the Mathematica script and output directly to the thorn source file. This approach has two main deficiencies:

- The same block of text might be used in several places in the code. When a bug is fixed in one place, it must be fixed in all.
- It is not easy to alter the language that is produced. For example, it is difficult to output both C and Fortran.
- The syntax in the Mathematica source file is ugly, with lots of string concatenation, making it difficult to read and edit.

The CodeGen package provides functions to solve these problems. To address the first problem, Mathematica functions are used to represent each block of code. This allows the block to be customized by giving the function arguments. By making this abstraction, it became very easy to change between outputting C and Fortran.

Fundamental to the system is the notion of a *block*; in Mathematica terms this can be either a string or a list of blocks (this definition is recursive). All the CodeGen functions return blocks, and the lists are all flattened and the strings concatenated when the final source file is generated. This is because it is syntactically easier in the Mathematica source file to write a sequence of statements as a list than to concatenate strings.

Many programming constructs are naturally block-structured; for example, C `for` loops need braces after the block of code to loop over. For this reason, it was decided that CodeGen functions could take as arguments any blocks of code which needed to be inserted on the inside of such a structure.

6.4.4 Package: Thorn

The Thorn package is used by all the different thorn generators to construct the final Cactus thorn. It takes care of the mechanics of writing files to storage and parsing the Kranc structures necessary for writing parameter configuration files, grid function definitions etc.

6.5 Implementation of the NOR formulation

The NOR evolution equations (3.79)–(3.82) are entered in the Kranc system as follows:

```
{
(* Shorthands *)
deth -> hDet,
invdeth -> 1 / deth,
hInv[ua,ub] -> MatrixInverse[h[ua,ub]],

trK -> K[la,lb] hInv[ua,ub],
gamma[ua, lb, lc] -> 1/2 hInv[ua,ud] (PD[h[lb,ld], lc]
  + PD[h[lc,ld], lb] - PD[h[lb,lc],ld]) ,
R[li,lj] -> 1/2 hInv[uk,ul] ( PD[h[lk,lj],li,ll] + PD[h[li,ll],lk,lj] -
                              PD[h[lk,ll],li,lj] - PD[h[li,lj],lk,ll]) +
            hInv[uk,ul](gamma[um,li,ll] gamma[un,lk,lj] h[ln,lm] -
                        gamma[um,li,lj] gamma[un,lk,ll] h[lm,ln]),
G[li,lj] -> PD[f[li],lj] + hInv[uk,um] hInv[ul,un] PD[h[lm,ln], lj]
            (PD[h[li,lk],ll] - (1/2) rho PD[h[lk,ll],li])
            - hInv[uk,ul]
            (PD[h[li,lk],ll,lj] - (1/2) rho PD[h[lk,ll],li,lj]),

DK[lk,ll,lj] -> PD[K[lk,ll],lj] - gamma[um,lj,lk] K[lm,ll] -
                gamma[um,lj,ll] K[lk,lm],

RS -> hInv[ui,uj] R[li,lj],
norham -> RS +  (trK)^2 - K[li,lj] K[lk,ll] hInv[uk,ui] hInv[ul, uj],
M[ui] -> - hInv[ui,uk] hInv[uj,ul] DK[lk,ll, lj] +
         hInv[ui,uj] hInv[uk,ul] DK[lk,ll,lj],

(* The Evolution equations *)
dot[h[la,lb]] -> 2 alpha K[la,lb] + Diss[h[la,lb]],

dot[K[la,lb]] ->
  -(-DDalpha[lb,la] + alpha (R[la,lb] - 2 K[la,lc] K[ld,lb] hInv[ud,uc] +
                             K[la,lb] trK) +
  (1/2) a alpha (G[la,lb] + G[lb,la]) +
```

```
  c (alpha norham + aprime alpha G[lk,ll] hInv[uk,ul]) h[la,lb]) +
  Diss[K[la,lb]],

dot[f[li]] -> -2 alpha hInv[uk,um] hInv[ul,un] K[lm,ln] PD[h[li,lk],ll] +
              rho alpha hInv[uk,um] hInv[ul,un] K[lm,ln] PD[h[lk,ll],li] +
              2 Dalpha[ll] K[li,lj] hInv[uj, ul] +
              2 alpha hInv[uk,ul] PD[K[li,lk], ll] -
              rho Dalpha[li] trK -
              rho alpha hInv[uk,ul] PD[K[lk,ll],li] +
              2 b alpha M[uj] h[lj,li] + Diss[f[li]]
}
```

This comes to about 40 lines. The generated C code is over 1200 lines long. Note that for historical reasons the sign convention for K_{ij} is opposite to that used in the rest of this work.

6.6 Summary

In this chapter, the Kranc software has been introduced. This software was written to allow a user to write an abstract tensorial description of a time evolution PDE and have Kranc generate the C or Fortran code necessary for solving the finite difference equations. Kranc uses the Cactus problem-solving infrastructure as a basis for the generated code. The design of the Kranc system has been discussed, and several of the important data structures have been described, including representations of calculations and custom finite differencing operators. Representation and manipulation of tensorial quantities via the TensorTools package is explained, as well as the abstraction of programming constructs provided by the CodeGen package for automatic code generation. Finally, the example of the fully nonlinear NOR evolution equations is given in Kranc syntax.

The Kranc software has allowed us to implement many formulations of the Einstein equations in a fraction of the time it would have taken to write the code by hand.

Chapter 7

Numerical comparisons between formulations of the Einstein equations

7.1 Introduction

There is a large number of 3+1 formulations of the Einstein equations, many of which have been shown to have a well-posed Cauchy problem. In order to choose a formulation for performing numerical simulations, it is desirable to understand the properties of the different formulations and the numerical schemes used to implement them.

Whilst well-posedness for many first order in time, first order in space formulations has long been established (e.g. [38], [29], [27], [54], [55]), this has only recently been achieved for formulations which are first order in time but second order in space (e.g. [13], [47], [32], [33]).

There are several reasons to suspect that second order in space systems may be more suited to numerical evolutions than fully first order systems.

- Second order systems have fewer variables. This means that the storage requirements during the computer simulation are lower, so higher resolution can be achieved for the same computational resources than is possible with a fully first order system.

- In reducing the Einstein equations to a fully first order form, the solution space is enlarged and the reduced system is subject to additional *auxiliary* constraints. It could be argued that having more constraints means that there is more potential for error.

- There are many reductions to first order which can be made, and the choice is usually determined by a set of parameters. Different choices of parameters lead to different numerical evolutions; dynamical adjustment of the parameters based upon the behaviour of the solution has been suggested in [45] in order to improve accuracy. Evolving the second order system directly does not lead to this complication.

- Whilst very high accuracy has been achieved for single black hole spacetimes using fully first order systems (e.g. [38]), there have been no results for binaries evolved using these systems. All of the recent successes (e.g. [17, 4, 51]) in the solution of the binary black hole problem have been achieved through the use of second order in space systems.

- Taking the wave equation in first order in time, second order in space form, and comparing it with a fully first order reduction, it can be shown that the standard finite difference approximation for the fully first order reduction leads to errors in the wave propagation speeds which are four times larger than the second order in space system. We investigate numerically in this chapter whether the same is true for the full Einstein equations.

In this work, we consider the ADM [7], BSSN [57, 11], NOR-A, NOR-B [47, 32, 33] and ST [55] formulations of the Einstein equations and construct finite difference approximations of each. Test cases from the Apples with Apples project [3] are used to obtain quantitative measures of the relative accuracy of these formulations. Specifically, we investigate the following issues.

- By performing numerical simulations, we test experimentally the convergence of the finite difference approximations to the solutions of the partial differential equations for each test case. The initial data is perturbed with random noise (as suggested in [3]) in order that the initial data contains all discrete Fourier frequencies. This makes it easier to see instability. The material of Chapter 5 is essential when considering this type of convergence test for systems which are second order in space, as the correct norm must be used when considering the accuracy of the initial data.

- We investigate whether the improvement in accuracy of the second order wave equation compared with first order is also observed for the fully nonlinear Einstein equations.
- The NOR system contains some of the features of BSSN, but it is simpler. We compare two variants of NOR (NOR-A and NOR-B) with BSSN, in order to see if there is any advantage in using BSSN over the simpler NOR systems.

7.2 The Apples with Apples tests

The *Apples with Apples* project [3] introduces a set of numerical tests which can be applied to formulations of the Einstein equations for the purpose of comparing them. As the goal was to encourage participation and cooperation by as many groups as possible, these tests are designed to be straightforward to implement. Periodic boundary conditions are used so that the issue of specifying boundary conditions for artificial boundaries does not complicate the test, though in physics simulations artificial boundaries are very important. All of the tests have initial data with plane symmetry, leading to essentially one dimensional problems. The project is called *Apples with Apples* as the aim is to compare different formulations in a standard setting.

In this work, we have deviated in several ways from the original test prescription, as described in Section 7.4.

We use fourth order Runge Kutta as the time integrator and, except where otherwise stated, we use the standard second order accurate centred finite difference approximation for spatial derivatives. We use 50 grid points as the base resolution, as this is sufficient to resolve the solutions for each test. We use a multiplier $\rho = 1, 2, 4, \ldots$ when additional resolutions are required (e.g. for convergence testing) leading to grids of size 50ρ.

7.2.1 Gauge wave

This test evolves Minkowski spacetime in sinusoidally perturbed Cartesian coordinates. It is a nonlinear test, in the sense that the solution is not a small perturbation of Minkowski spacetime.

The exact solution is prescribed as follows:

$$\begin{aligned} \gamma_{xx} &= 1 - A\sin\left(\tfrac{2\pi(x-t)}{d}\right) & \gamma_{yy} &= 1 \quad \gamma_{zz} = 1 \\ K_{xx} &= -\tfrac{\pi A}{d}\cos\left(\tfrac{2\pi(x-t)}{d}\right)\left(1 + A\sin\left(\tfrac{2\pi(x-t)}{d}\right)\right)^{-1/2} & K_{yy} &= 0 \quad K_{zz} = 0 \\ \alpha &= \sqrt{\det\gamma} \end{aligned} \tag{7.1}$$

The off-diagonal components of γ_{ij} and K_{ij} are zero. The lapse α is compatible with *harmonic slicing*; $\nabla^a\nabla_a t = 0$ and the shift β^a is zero. We choose an amplitude of $A = 10^{-1}$. The evolution is within the nonlinear regime for this amplitude. The coordinate domain is $x \in [-0.5, +0.5), y = 0, z = 0$ and the grid points have coordinates $x_j = -0.5 + jh$ where $h = 1/N$ and $N = 50\rho$ with $\rho = 1, 2, 4$.

7.2.2 Linear Waves

This test is aimed at determining whether a formulation is capable of evolving weak travelling gravitational waves.

$$\begin{aligned} \gamma_{xx} &= 1 \quad \gamma_{yy} = 1 + b & \gamma_{zz} &= 1 - b \\ K_{xx} &= 0 \quad K_{yy} = \tfrac{1}{2}\partial_t b & K_{zz} &= -\tfrac{1}{2}\partial_t b \\ \alpha &= 1 \quad b \equiv A\sin\left(\tfrac{2\pi(x-t)}{d}\right) \end{aligned} \tag{7.2}$$

The off-diagonal components of γ_{ij} and K_{ij} are zero. Note that this is not an exact solution to the nonlinear equations. It is a solution to the linearized equations. Only when A is small will the two be similar. $A = 10^{-8}$ is chosen to ensure that the quadratic terms in the evolution equations are of the order of numerical round-off; hence the numerical solution should be a solution of the linearized equations to a good approximation. Similarly, the lapse α is compatible with harmonic slicing $\nabla^a\nabla_a t = 0$ only at the linearized level. The shift β^a is zero. The grid and coordinates are the same as for the gauge wave.

7.2.3 Gowdy

Gowdy spacetimes are nonlinear cosmological solutions of the Einstein equations. This test is a special case called *polarized Gowdy*. This is a strong gravity test, and it is not a perturbation of Minkowski; it is not flat and it is nonlinear. The explicit initial data is

given below; see [3] for more details. The grid and coordinates are the same as for the gauge wave, except that z rather than x is the principal direction.

$$\begin{aligned}
\gamma_{xx} &= te^{P} & K_{xx} &= -\tfrac{1}{2}t^{1/4}e^{-\lambda/4}e^{P}(1+tP_{,t}) \\
\gamma_{yy} &= te^{-P} & K_{yy} &= -\tfrac{1}{2}t^{1/4}e^{-\lambda/4}e^{-P}(1-tP_{,t}) \\
\gamma_{zz} &= t^{-1/2}e^{\lambda/2} & K_{zz} &= \tfrac{1}{4}t^{-1/4}e^{\lambda/4}(t^{-1}-\lambda_{,t}) \\
\alpha &= \sqrt{\gamma_{zz}} = t^{-1/4}e^{\lambda/4}
\end{aligned}$$

where

$$P = J_0(2\pi t)\cos(2\pi z) \tag{7.3}$$

$$\begin{aligned}
\lambda &= -2\pi t J_0(2\pi t)J_1(2\pi t)\cos^2(2\pi z) + 2\pi^2 t^2[J_0^2(2\pi t) + J_1^2(2\pi t)] \\
&\quad -\frac{1}{2}\left\{(2\pi)^2[J_0^2(2\pi) + J_1^2(2\pi)] - 2\pi J_0(2\pi)J_1(2\pi)\right\}
\end{aligned} \tag{7.4}$$

The off-diagonal components of γ_{ij} and K_{ij} are zero. The cosmological singularity is at $t = 0$ in these coordinates. The shift β^a is zero, but the lapse α is *not* compatible with harmonic slicing. We evolve from $t = 1$ to test the behaviour of the scheme when the evolved variables grow very large; we call this *expanding Gowdy*. We also evolve the scheme backwards in time from $t = 1$ to $t = 0$ to determine how accurately the scheme reproduces solutions close to a singularity. This test is called *collapsing Gowdy*. In [3], the collapsing Gowdy test is performed using harmonic slicing, leading to an infinite coordinate time before reaching the singularity. We choose to use the lapse associated with the exact solution above so that the singularity is reached in finite coordinate time.

7.3 Coordinate conditions

With zero shift, the Bona-Masso lapse condition [15] is

$$\partial_t \alpha = -\alpha^2 f(\alpha) K \tag{7.5}$$

We use a more general slicing condition (see, for example, [55]):

$$\partial_t \alpha = -\alpha F(\alpha, K, x^\mu) \tag{7.6}$$

as this is the form used by the ST formulation, and it encompasses all the slicing conditions we want to implement. For NOR, harmonic slicing is implemented as

$$\partial_t \alpha = -\alpha(\alpha \gamma^{ij} K_{ij}) \tag{7.7}$$

and the Gowdy lapse is

$$\partial_t \alpha = -\alpha(t^{-1} + \alpha \gamma^{ij} K_{ij}) \tag{7.8}$$

which is chosen to be compatible with the exact solution for the Gowdy spacetime. For BSSN, the same slicing conditions are used, except that $\gamma^{ij} K_{ij}$ is replaced with the evolved variable K. For ST, we use

$$\begin{aligned} \partial_t \alpha &= -\alpha(\alpha \gamma^{ij} K_{ij}) & (7.9) \\ \partial_t A_i &= \alpha(-\gamma^{ij} K_{ij} A_i - (K_{mn,i} \gamma^{mn} - K^{mn} d_{imn}) + \xi M_i) & (7.10) \end{aligned}$$

for harmonic slicing and

$$\begin{aligned} \partial_t \alpha &= -\alpha(t^{-1} + \alpha \gamma^{ij} K_{ij}) & (7.11) \\ \partial_t A_i &= \alpha(-\gamma^{ij} K_{ij} A_i - (K_{mn,i} \gamma^{mn} - K^{mn} d_{imn}) + \xi M_i) & (7.12) \end{aligned}$$

for Gowdy slicing (recall that $A_i = \partial_i \alpha$ for the ST formulation).

7.4 Differences between our tests and the Apples with Apples specifications

Our use of the Apples with Apples tests is slightly different to the prescriptions made in [3].

We use fourth order Runge-Kutta (RK4) as the time integrator instead of iterative Crank Nicolson (ICN). Our work has shown that the Courant factor necessary for stability (at least in the linearized case) of the second order in space formulations of the Einstein equations is larger for RK4 than for ICN. We will in future wish to test schemes which are overall fourth order accurate; ICN is only second order accurate and hence is unsuitable for these tests. It is desirable to use the same time integrator for all tests (for example, the amount

of dissipation added by the time integrator is the same), so RK4 is chosen.

For convergence testing, the $50\rho \times 3 \times 3$ grid is used. However, in order to compare accuracy, a one dimensional grid is used with a one dimensional finite difference scheme. This gives an improvement in efficiency of a factor of 9. At the level of the finite difference equations, the two are equivalent for one dimensional initial data. However, when round-off errors are considered, the two are no longer equivalent (see Section 4.7).

For the *robust stability test*, [3] suggests adding noise of amplitude ϵ with

$$\epsilon \in (-10^{-10}/\rho^2, +10^{-10}/\rho^2) \tag{7.13}$$

We modify this in two ways. Firstly, the lowest resolution amplitude is changed from 10^{-10} to 10^{-3}. In [3] it is argued that using very small amplitude perturbations of Minkowski causes quadratic terms in the equations to be below the level of round-off error (10^{-15}), leading to a finite difference scheme which is essentially linear. We use a higher amplitude so that we test the fully nonlinear scheme, and so that round-off errors do not interfere with our interpretation of the results. Secondly, the ρ^2 is replaced with ρ^q where q depends both on the order of accuracy of the finite difference scheme, and on whether the variable concerned is differentiated twice in the second order in space equations (see Section 7.5).

The original test specification gives conflicting information about the coordinates of the numerical grid. We choose the coordinates so that upon refinement of the grid, the points on the refined grid coincide with points on the unrefined grid. This allows convergence testing without the need to interpolate points between grids.

7.5 Convergence

Given a finite difference approximation which is consistent with a partial differential equation and is accurate of order p, we require that the solution v_j^n to the finite difference equations converges to the solution $u(t,x)$ to the partial differential equations; i.e. for sufficiently high resolution,

$$\|v^n - u(t,\cdot)\|_h = O(h^p) \tag{7.14}$$

provided that the initial data is accurate of order p to $u(0, x)$,

$$\|v^0 - u(0, \cdot)\|_h = O(h^p) \tag{7.15}$$

in a norm $\|\cdot\|_h$. We wish to determine under what circumstances schemes for the various formulations of the Einstein equations are convergent for the different Apples with Apples tests. For a linear system with constant coefficients and no forcing terms, one can consider the definition of stability in Section 4.2 and test this experimentally. The Lax theorem can then be used to infer convergence. The stability test can lead to conclusions that can be used to determine convergence for all initial data. We have not been able to find a definition of numerical stability for a nonlinear finite difference scheme that can be related to its convergence by a Lax-type theorem. Hence, we directly test convergence for specific exact solutions. It is not possible to prove convergence using a finite number of numerical experiments for two reasons:

- It would be necessary to test all resolutions as $h \to 0$;
- The definition of convergence does not assume exact initial data, so all initial data of the correct order of accuracy would have to be tested.

It has been shown (for example in [21]) that an ill-posed formulation, which does not admit a convergent finite difference approximation, may appear to be convergent if smooth initial data is used (strictly speaking, initial data where only the lowest frequency discrete Fourier modes have non-zero amplitude) or if run times are short. In real simulations, high frequency modes can become populated, both due to the effects of round-off error and the nonlinearity of the equations causing coupling of Fourier modes. In order to test convergence in a more demanding way, we choose initial data which is not smooth, but is accurate to the correct order. This is achieved by adding random noise to the smooth initial data such that (7.15) is satisfied.

For second order in space formulations, the norm $\|\cdot\|_h$ will contain finite difference operators (we refer to it as a D_+ norm). Whilst it is true that if (7.14) is satisfied, then the same estimate in the L_2 norm follows, the requirement (7.15) on the order of accuracy of the initial data must still hold in the D_+ norm.

For each system to be tested, we perform short runs at four different resolutions parameterized by $\rho = 1, 2, 4, 8$ where the grid size is $50\rho \times 3 \times 3$. The grid spacing is $h = 1/(50\rho)$.

We plot the D_+ norm of the error multiplied by ρ^p as a function of time (p is the order of accuracy of the scheme; i.e. 2 or 4). If (7.14) is satisfied, then this quantity will be bounded above by a function of time that is independent of ρ. If this is not the case, and it appears that the trend is for increasing ρ to lead to larger rescaled errors, then we conclude that the scheme is not convergent. Note that in [3], the *robust stability test* (evolution of Minkowski spacetime with random noise superimposed on the initial data) is run until $t = 1000$ whereas we run until $t = 0.1$. We have found that it is easier to identify non-convergent schemes by increasing the number of grid points (and hence increasing the highest discrete frequency present) than by increasing the run time with only lower frequencies present.

The norms used for monitoring convergence are as follows. For NOR we use

$$||v||^2 \equiv \sum_{\text{gridpoints}} \left(\sum_{ij} \gamma_{ij}^2 + \sum_{ij} K_{ij}^2 + \sum_{i} f_i^2 + \alpha^2 + \sum_{kij} (D_{+k}\gamma_{ij})^2 + \sum_{i} (D_{+i}\alpha)^2 \right) h^3 \tag{7.16}$$

(h is the grid spacing). This is the $l_{2,h}$ norm obtained from a discrete first order reduction, and is the norm in which the linearized scheme with densitized lapse has been proven to be stable. The system that we call ADM is actually a parameterization of NOR, so it includes the f_i variables. For convenience, we use the NOR norm for ADM.

For BSSN, we use

$$||v||^2 \equiv \sum_{\text{gridpoints}} \left(\sum_{ij} \tilde{\gamma}_{ij}^2 + \phi^2 + K^2 + \alpha^2 \sum_{ij} \tilde{A}_{ij}^2 + \sum_{i} \tilde{\Gamma}_i^2 + \sum_{kij} (D_{+k}\tilde{\gamma}_{ij})^2 + (D_+\phi)^2 + \sum_{i} (D_{+i}\alpha)^2 \right) h^3 \tag{7.17}$$

Since the ST formulation is first order in space, we use the standard $l_{2,h}$ norm:

$$||v||^2 \equiv \sum_{\text{gridpoints}} \left(\sum_{ij} g_{ij}^2 + \sum_{ij} K_{ij}^2 + \sum_{ijk} (d_{ijk})^2 + N^2 + \sum_{i} (A_i)^2 \right) h^3 \tag{7.18}$$

7.5.1 ADM

The continuum ADM system is ill-posed, and the finite difference approximation of the linearized system is unstable, so we expect that the nonlinear finite difference approximation will fail to be convergent.

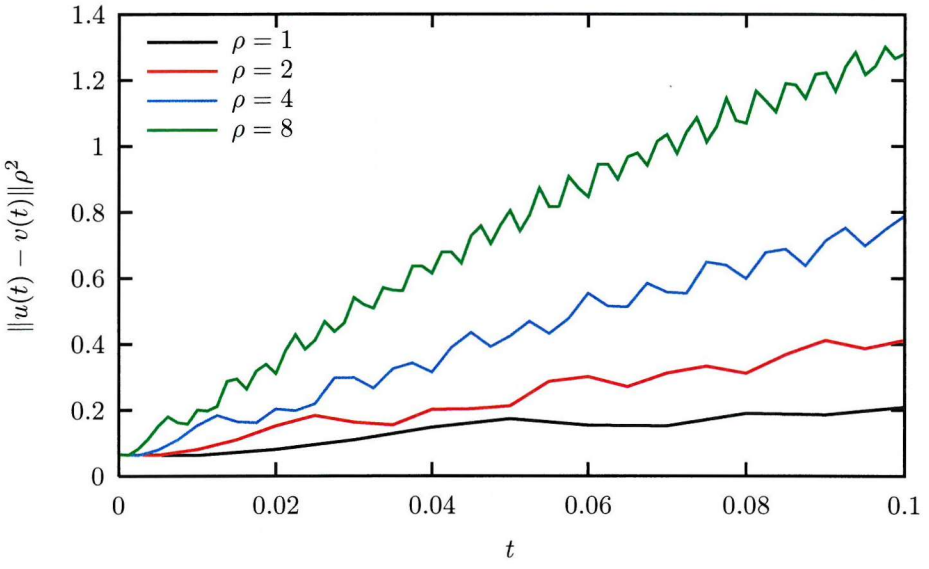

Figure 7.1: Convergence test, ADM, Minkowski

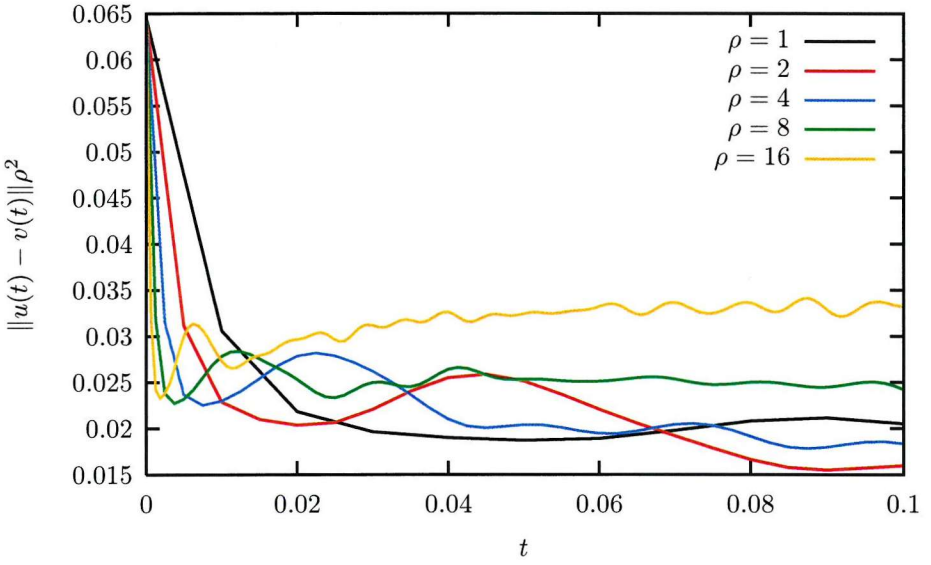

Figure 7.2: Convergence test, ADM with dissipation parameter $\sigma = 0.2$, Minkowski

Figures 7.1 and 7.2 show that both with and without artificial dissipation (see Section 4.8), the second order accurate finite difference approximation is not convergent.

7.5.2 NOR

Figures 7.3–7.10 show that both NOR-A and NOR-B without artificial dissipation appear to be convergent for Minkowski, the gauge wave, and expanding and collapsing Gowdy. The figures look identical, but on closer inspection they are not in fact the same. The random noise added to the solution is in fact only pseudo-random, and is the same for each test. We have verified that changing the random data (by altering the seed value given to the random number generator) produces different output.

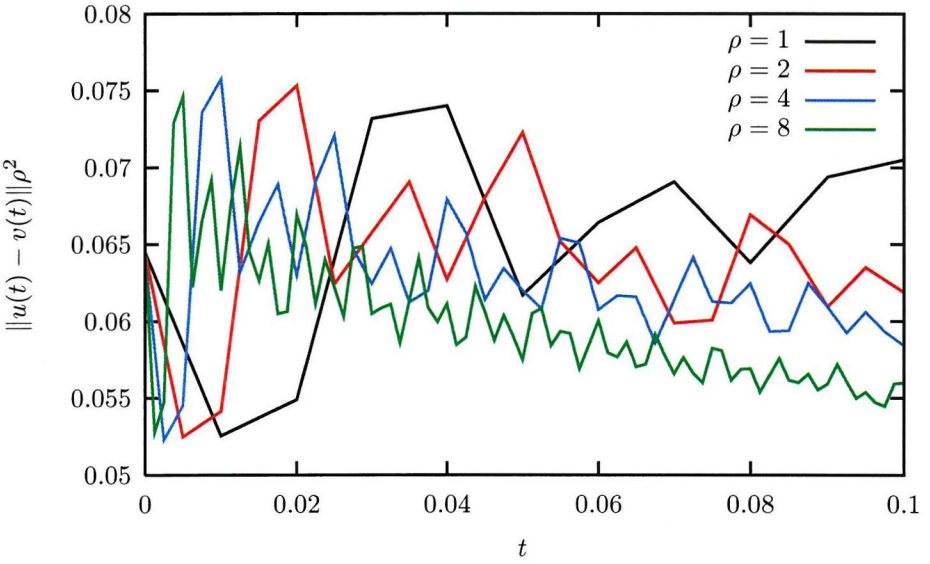

Figure 7.3: Convergence test, NOR-A, Minkowski

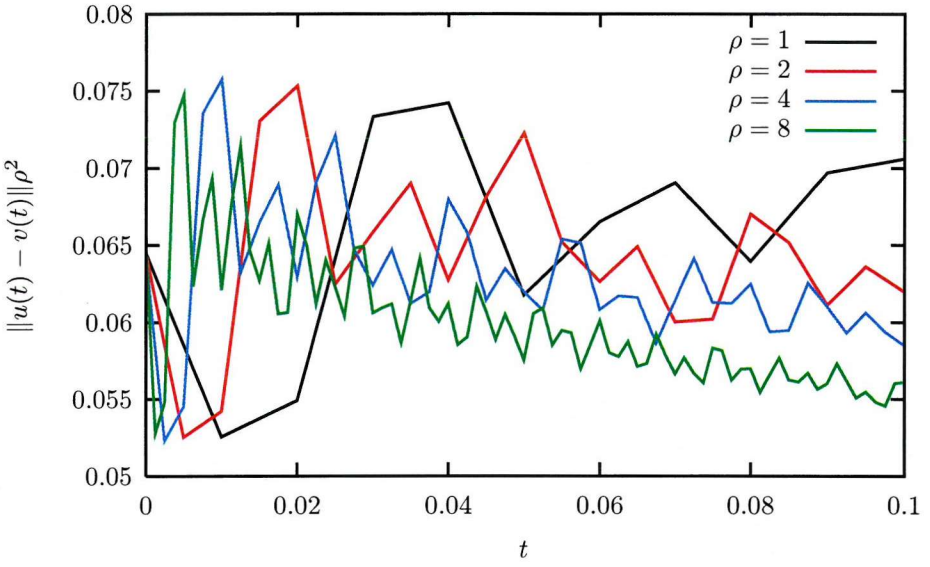

Figure 7.4: Convergence test, NOR-A, gauge wave

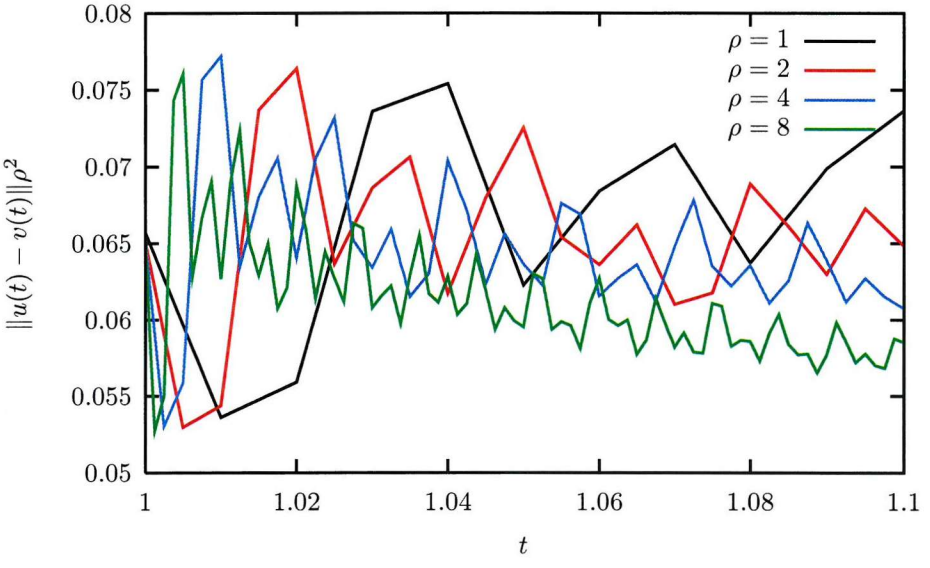

Figure 7.5: Convergence test, NOR-A, Gowdy

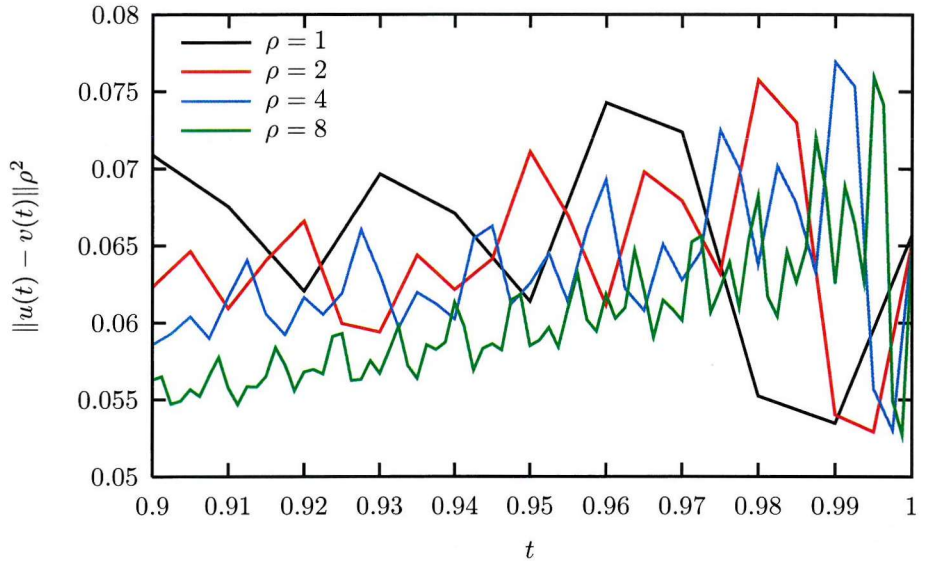

Figure 7.6: Convergence test, NOR-A, collapsing Gowdy

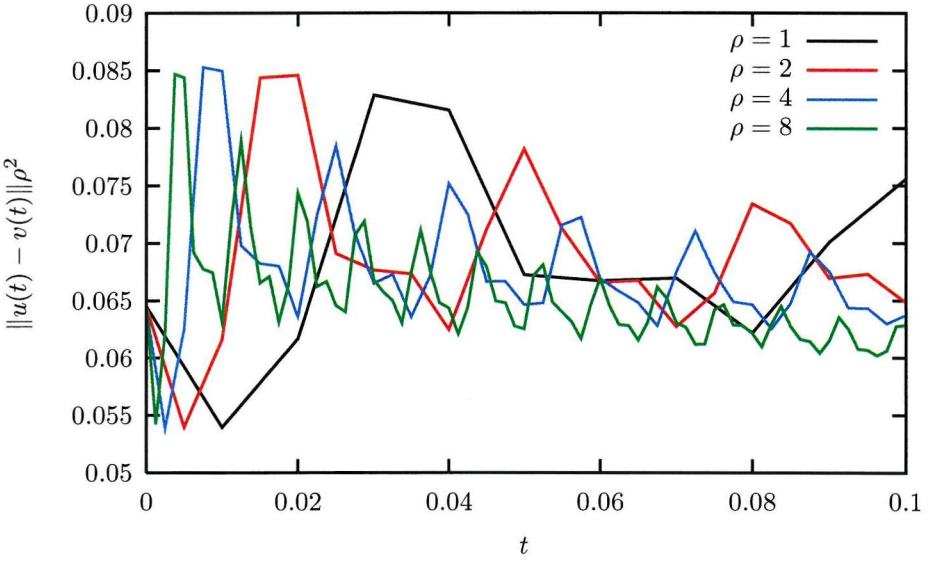

Figure 7.7: Convergence test, NOR-B, Minkowski

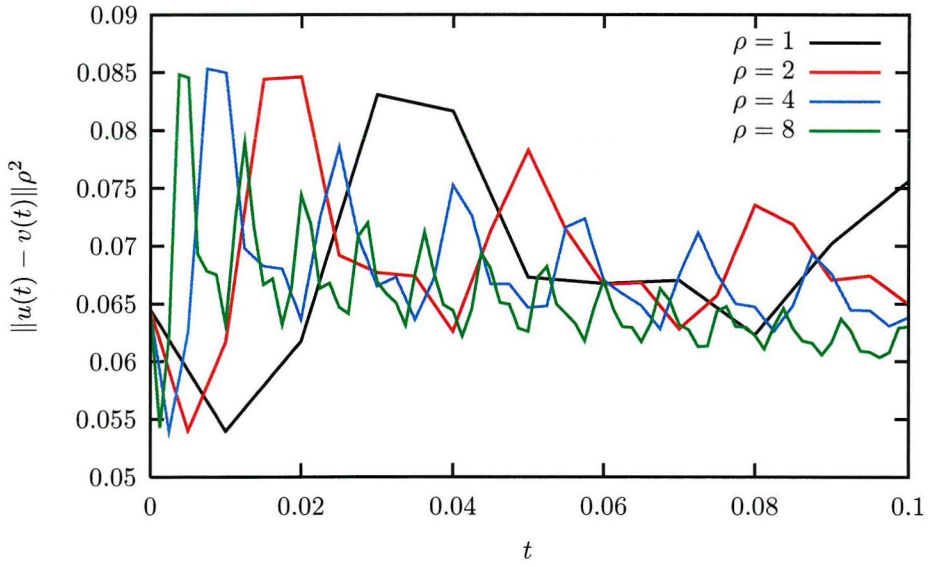

Figure 7.8: Convergence test, NOR-B, gauge wave

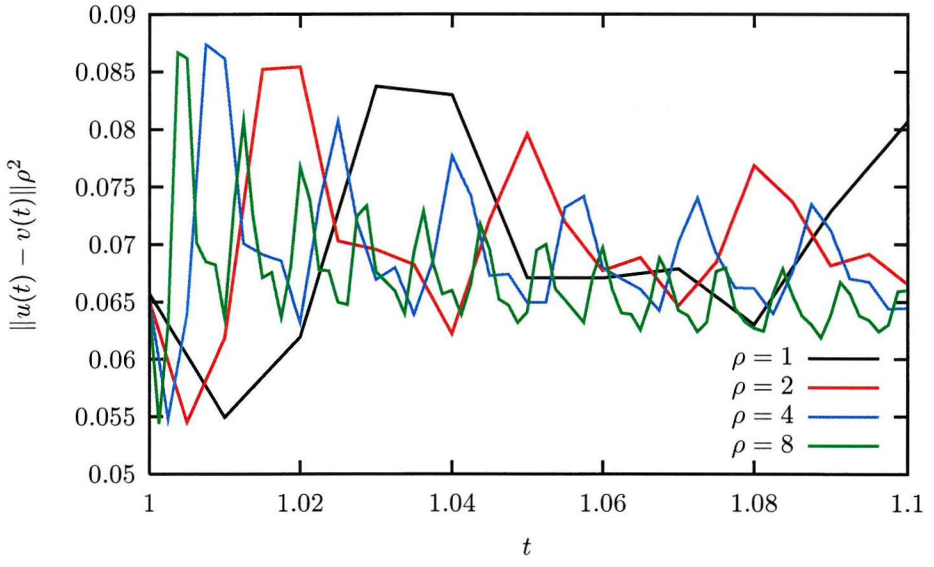

Figure 7.9: Convergence test, NOR-B, Gowdy

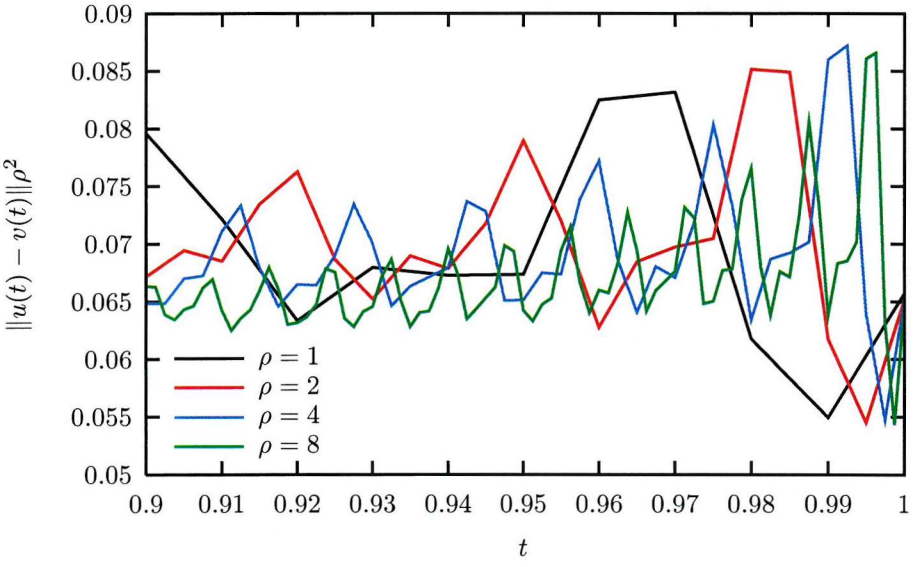

Figure 7.10: Convergence test, NOR-B, collapsing Gowdy

7.5.3 BSSN

Figure 7.11 shows that the BSSN system with no artificial dissipation is not convergent about Minkowski. Adding a small amount of artificial dissipation ($\sigma = 0.02$) appears to make the scheme convergent for all the tests (Figures 7.12–7.15).

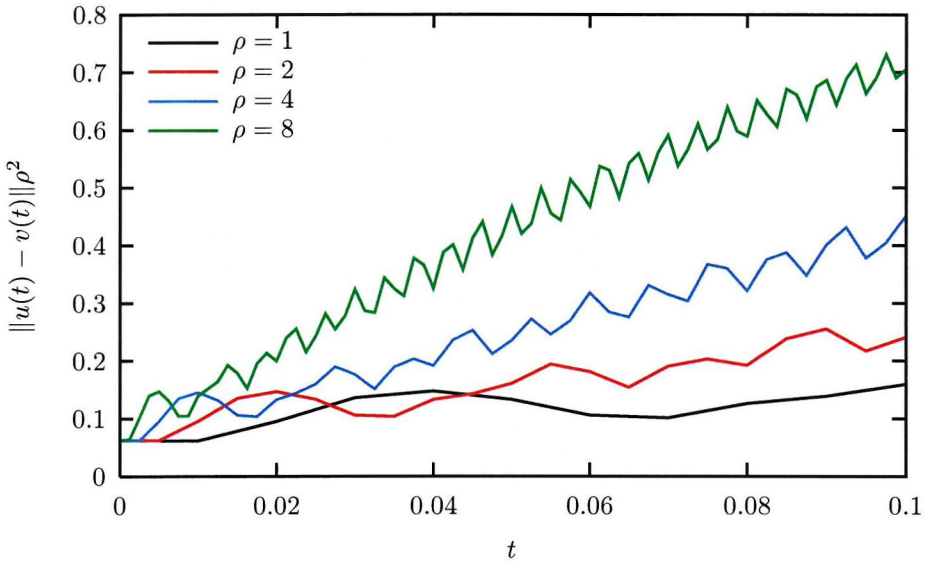

Figure 7.11: Convergence test, BSSN with $\sigma = 0$, Minkowski

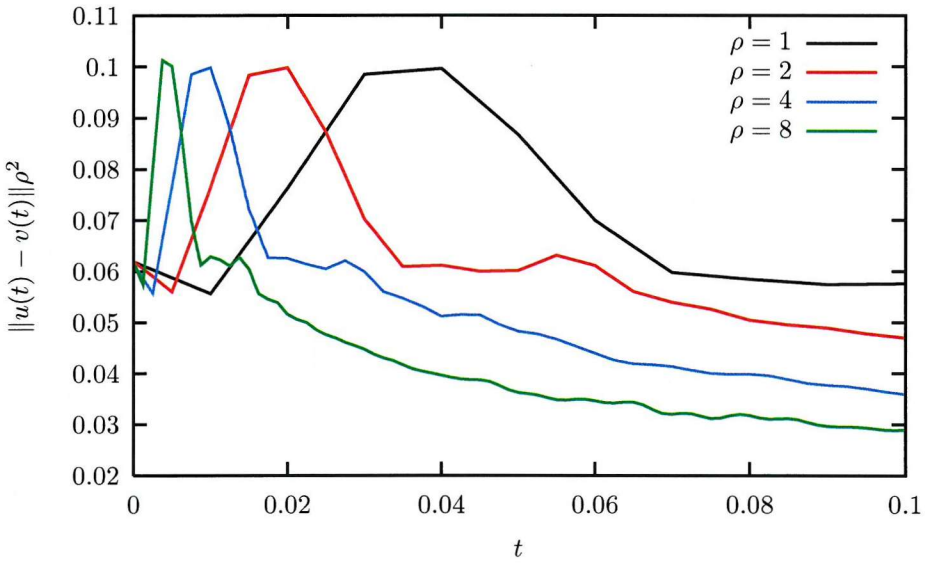

Figure 7.12: Convergence test, BSSN, Minkowski

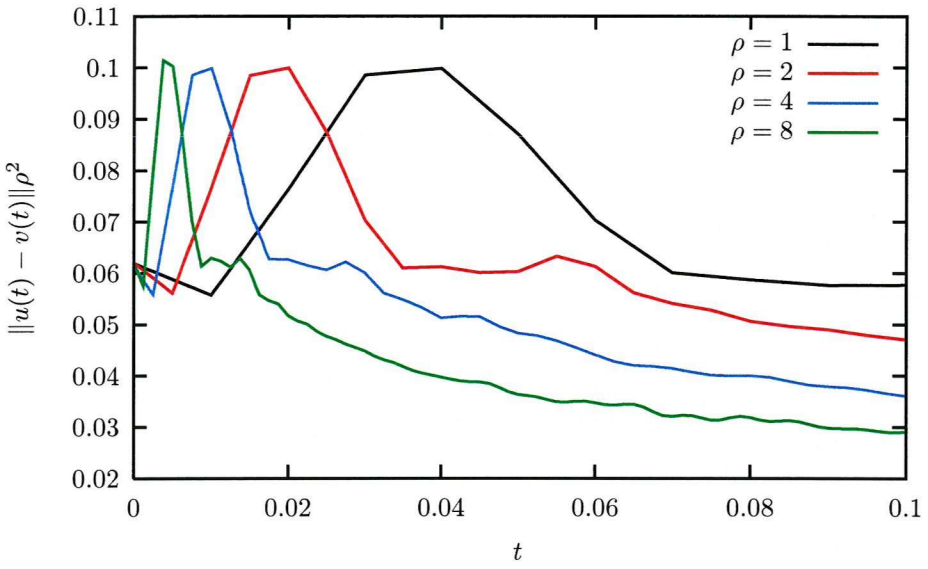

Figure 7.13: Convergence test, BSSN, gauge wave

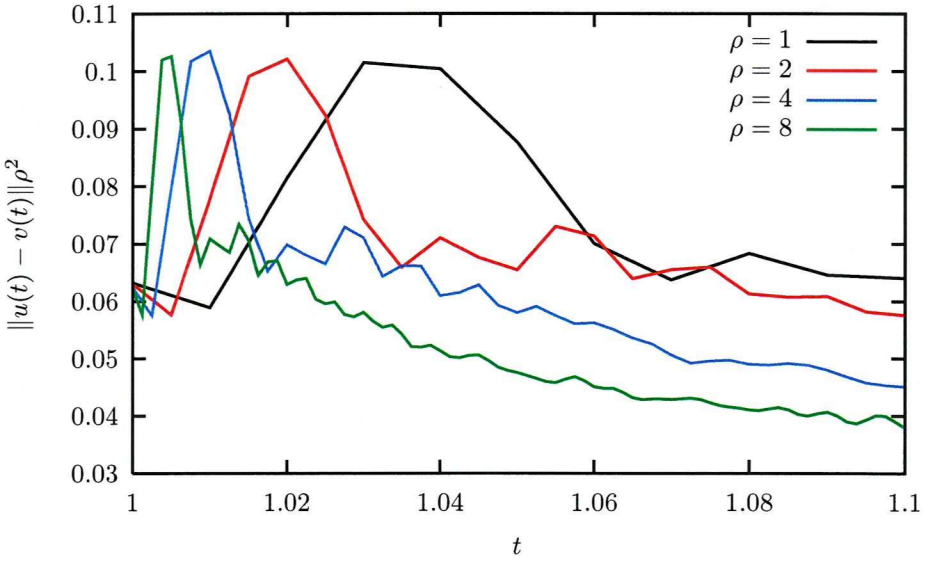

Figure 7.14: Convergence test, BSSN, Gowdy

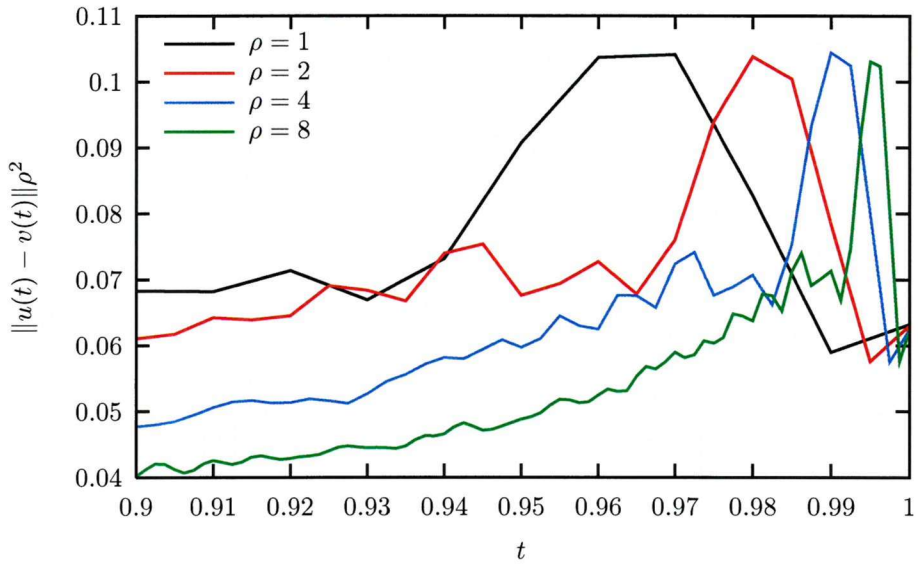

Figure 7.15: Convergence test, BSSN, collapsing Gowdy

7.5.4 ST

The ST continuum system is symmetrizable hyperbolic, so the results of Section 5.6 imply that the standard discretizations of the linear system should be stable and hence convergent. We expect this to apply to the nonlinear case as well, and Figures 7.16–7.19 suggest that this is the case.

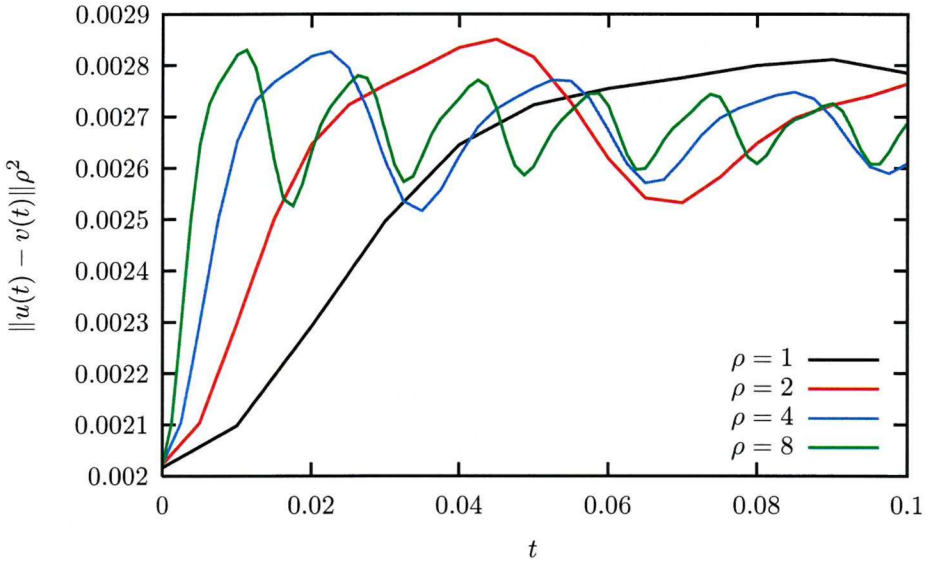

Figure 7.16: Convergence test, ST, Minkowski

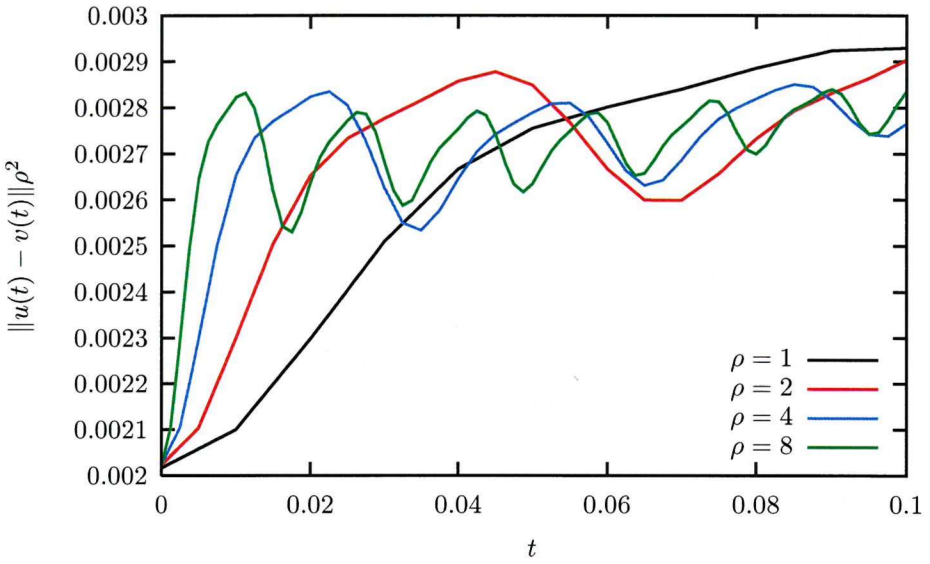

Figure 7.17: Convergence test, ST, gauge wave

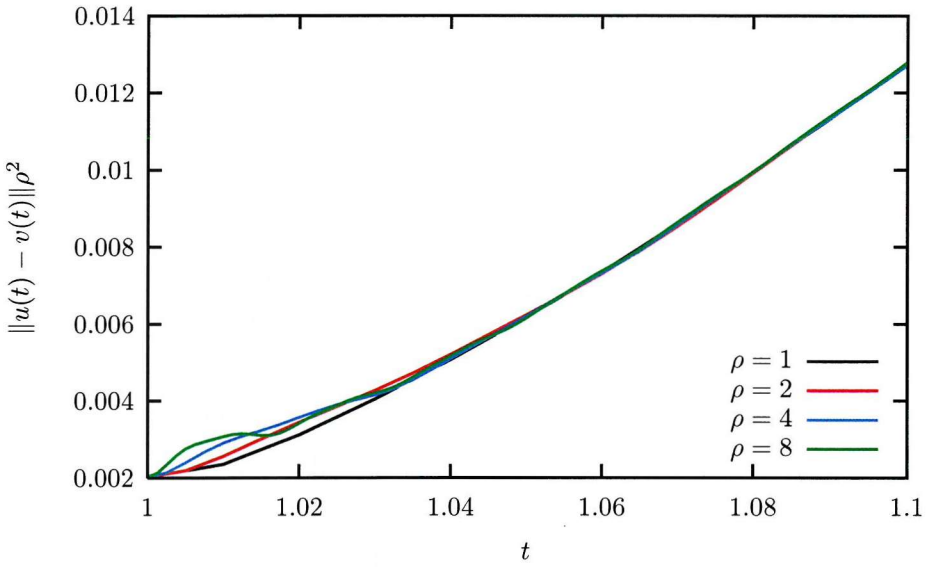

Figure 7.18: Convergence test, ST, Gowdy

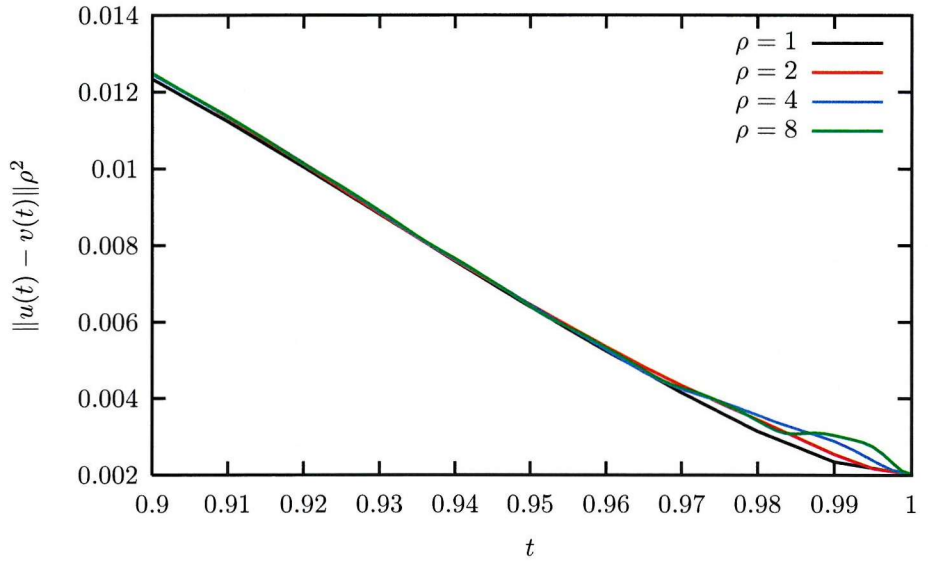

Figure 7.19: Convergence test, ST, collapsing Gowdy

7.5.5 Convergence test summary

The experimental convergence tests indicate that NOR-A, NOR-B and ST are convergent about all the test solutions. BSSN requires a small amount of artificial dissipation, but when this is used, it is also convergent for all the tests.

7.6 Accuracy of first and second order systems

We aim to investigate whether there is an advantage in using second order in space systems as opposed to first order in space systems for solving Einstein's equations.

7.6.1 The example of the wave equation

Consider a semidiscrete scheme

$$\partial_t v_j(t) = P v_j(t) \tag{7.19}$$

where P is some finite differencing operator. Taking a discrete Fourier transform, we obtain

$$\partial_t \hat{v}(t,\xi) = \hat{P}\hat{v}(t,\xi) \tag{7.20}$$

Formally, this can be solved by integration:

$$\hat{v}(t,\xi) = e^{\hat{P}(\xi)t}\hat{v}(0,\xi) \tag{7.21}$$

and transformed back to physical space

$$v_j(t) = \frac{1}{N}\sum_{\xi} e^{ij\xi} e^{\hat{P}(\xi)t}\hat{v}(0,\xi) \tag{7.22}$$

Noting that $x_j = hj$, we obtain

$$v_j(t) = \frac{1}{N}\sum_{\xi} \exp\left[\frac{i\langle x_j,\xi\rangle}{h} I + \hat{P}(\xi)t\right]\hat{v}(0,\xi) \tag{7.23}$$

Assuming that $\hat{P}$ can be diagonalized with eigenvalues $\lambda(\xi)$, we can consider the solution to be a sum of travelling waves $e^{ik\cdot x+\omega t}$ with wavenumber $k = \xi/h$ and $\omega = \lambda(\xi)/i$. The phase velocity of these waves is

$$v_p(\xi) = \frac{\omega}{k} = \frac{h}{i}\frac{\lambda(\xi)}{\xi} \tag{7.24}$$

We can now compare the wave equation in first order in time, second order in space, form with its fully first order reduction.

For the wave equation in first order in time and second order in space form

$$\partial_t \phi = \pi \qquad \partial_t \pi \;=\; \partial_x^2 \phi \tag{7.25}$$

we use the standard discretization and in Fourier space obtain

$$\partial_t \hat{\phi} = \hat{\pi} \qquad \partial_t \hat{\pi} = \frac{4}{h^2} \sin^2\left(\frac{\xi}{2}\right) \hat{\phi} \tag{7.26}$$

The eigenvalues of the semidiscrete symbol are

$$\lambda^{\pm} = \frac{2i}{h} \sin\left(\frac{\xi}{2}\right) \tag{7.27}$$

leading to phase velocities of

$$v_p^{\pm} = \pm\left(1 - \frac{\xi^2}{24} + O(\xi^4)\right) \tag{7.28}$$

The fully first order continuum reduction of (7.25) is

$$\partial_t \phi = \pi \qquad \partial_t \pi = \partial_x \psi \qquad \partial_t \psi = \partial_x \pi \tag{7.29}$$

and the standard discretization of this system leads to

$$\partial_t \hat{\phi} = \hat{\pi} \qquad \partial_t \hat{\pi} = \frac{i}{h} \sin \xi \hat{\psi} \qquad \partial_t \hat{\psi} = \frac{i}{h} \sin \xi \hat{\pi} \tag{7.30}$$

The eigenvalues are

$$\lambda^0 = 0 \qquad \lambda^{\pm} = \pm \frac{i}{h} \sin \xi \tag{7.31}$$

leading to phase velocities

$$v^0 = 0 \qquad v_p^{\pm} = \pm\left(1 - \frac{\xi^2}{6} + O(\xi^4)\right) \tag{7.32}$$

The continuum speed of propagation is unity, so to leading order in ξ the error in the propagation speed for the discretization of the fully first order reduction is four times that of the second order in space system. Note that this expansion in ξ assumes that the number of grid points is large enough that the solution is well-represented by the lowest frequencies.

In the following sections, we perform numerical experiments with the Apples with Apples test suite to determine if these results apply to the full nonlinear Einstein equations.

7.6.2 Testing details

We use the lowest Apples with Apples resolution ($N = 50$) and choose to evolve a truly one dimensional scheme on a one dimensional grid. From the point of view of the finite difference equations, this is equivalent to using a 3D scheme on a 3D grid since the initial data only has variation in one direction. However, when round-off errors are taken into account, the numerical solutions will be different. Since the exact solution is one dimensional, we choose to consider the accuracy of the one dimensional finite difference equations, rather than attempting to analyse the effects of round-off error. For the accuracy tests, we do not add random noise to the initial data.

7.6.3 Gauge wave

We evolve the gauge wave and monitor the metric components. Figure 7.20 shows the γ_{xx} component of the solution after ten crossing times. NOR-A, NOR-B and BSSN using the standard discretization show a large error which is approximately constant in space. Further, the BSSN wave profile is not reproduced accurately. Using the D_0^2 discretization for the second order in space formulations as suggested in [8] eliminates the constant in space error, and these and the ST formulation are much more accurate.

The maximum of the exact γ_{xx} is 1.1 for all time. We plot the numerical values in figure 7.21. We see that NOR-A with the D_0^2 discretization and ST both preserve the maximum. The standard discretization of NOR-A and NOR-B and both discretizations of BSSN quickly develop large errors in the maximum. NOR-A shows a worse than exponential growth in the maximum.

Figure 7.22 shows the x coordinate of the maximum of γ_{xx} every crossing time as a function of time, plotted every crossing time, for each system. For the exact solution, the maximum should remain at the same coordinate. The gradients of these lines are the errors in the speeds at which the maximum is propagated. The phase error is linear in time for all three systems, indicating that the numerical speeds of propagation have constant errors.

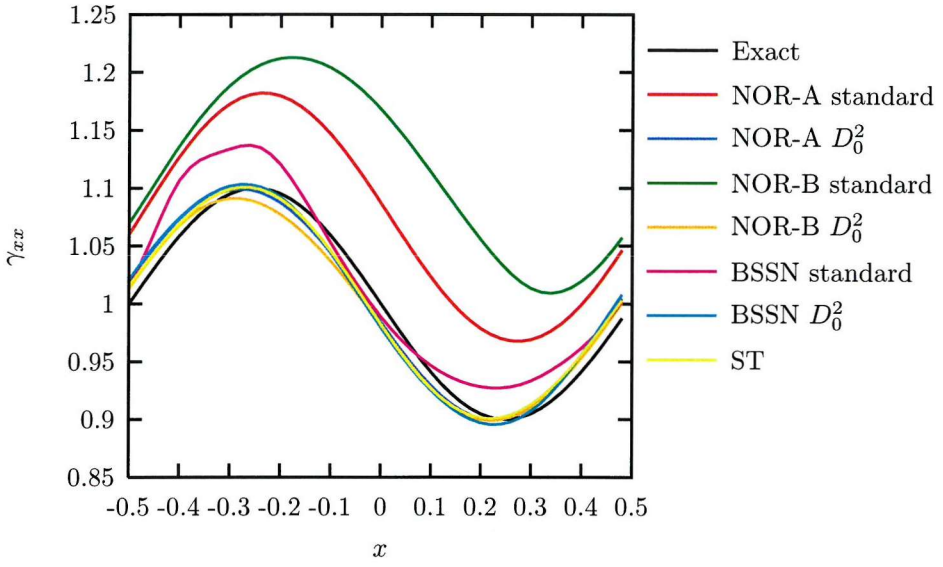

Figure 7.20: Gauge wave γ_{xx} profiles at $t = 10$. NOR-A, NOR-B and BSSN using the standard discretization are clearly distinguishable from the other lines.

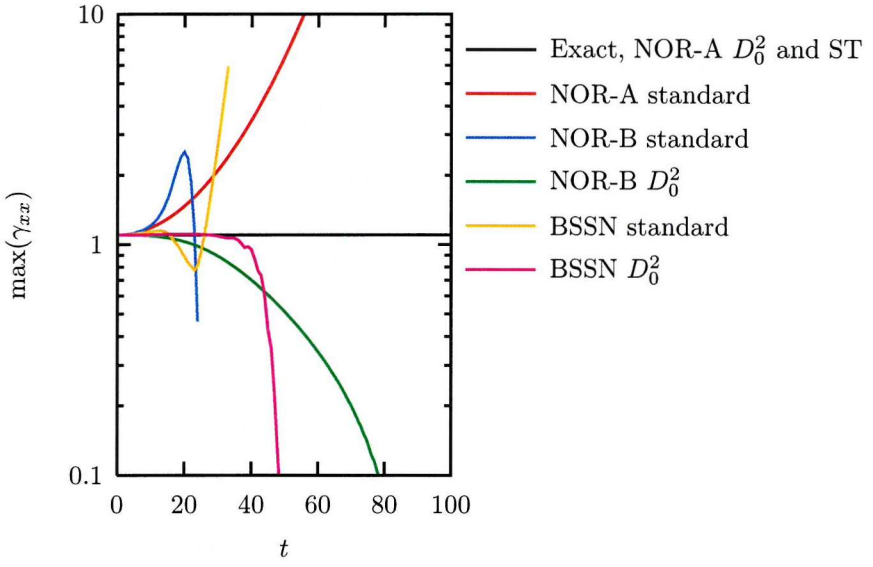

Figure 7.21: Gauge wave γ_{xx} maximum

The phase error lines for NOR-B and BSSN have been truncated in time once the solution no longer resembles a travelling wave.

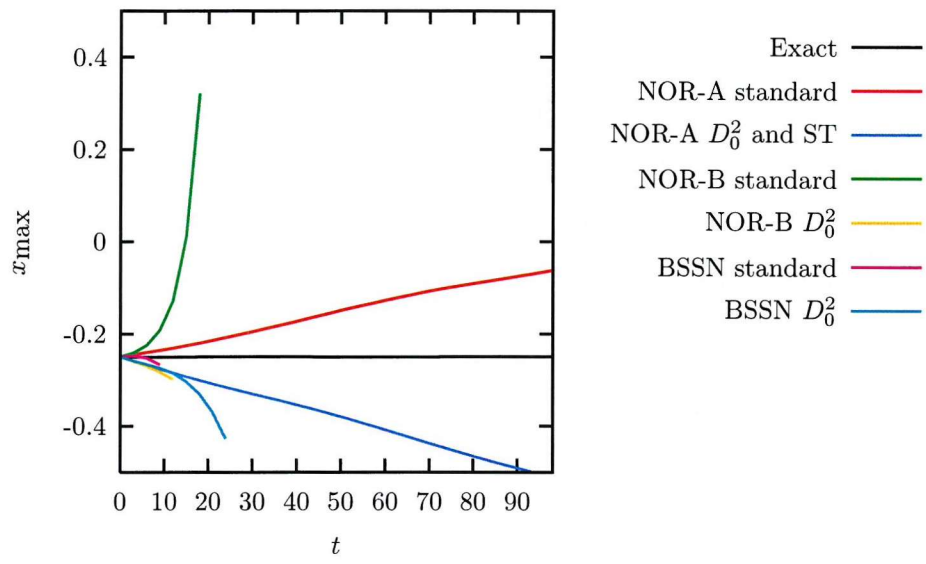

Figure 7.22: Gauge wave phase error comparison. The x coordinate of the maximum is plotted against time every crossing time.

7.6.4 Linear wave

The errors in the linear wave test are much smaller. Figure 7.23 shows that NOR and BSSN using the standard discretization have very similar phase errors, whereas ST has a much larger phase error.

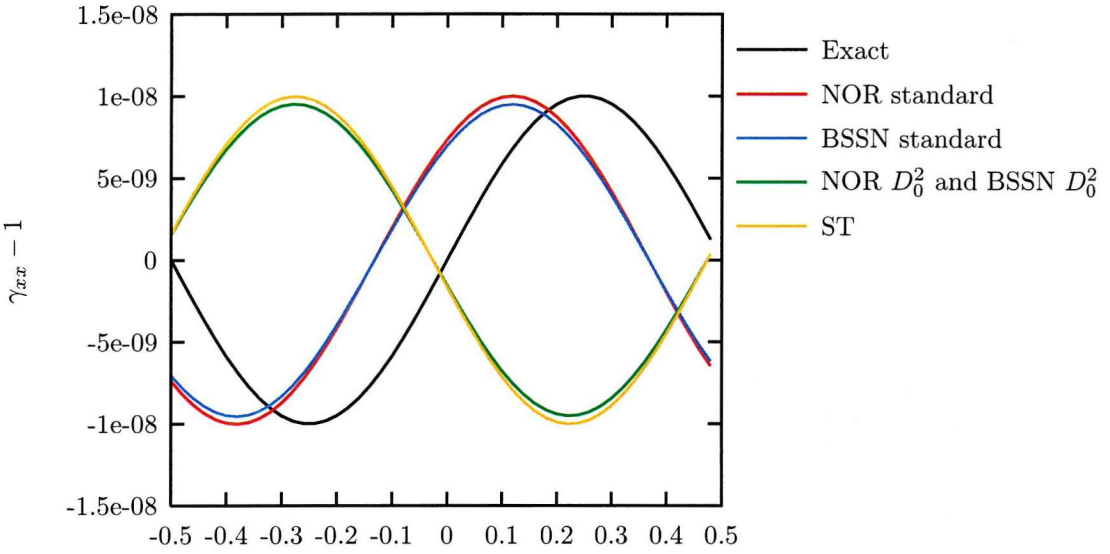

Figure 7.23: Linear wave profile $t = 200$

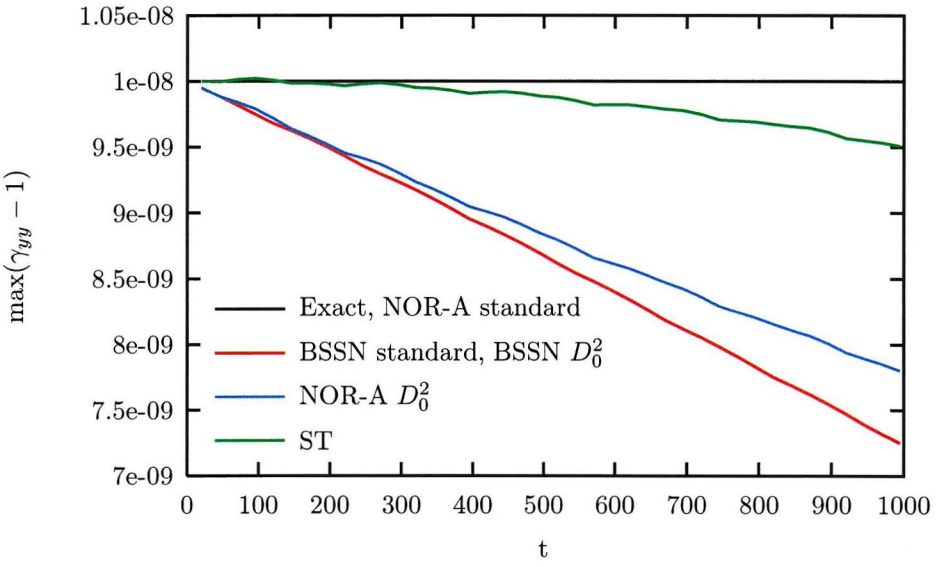

Figure 7.24: Linear wave amplitude comparison

The error in the amplitude of the wave is more pronounced in BSSN and NOR-A with the D_0^2 discretization; for ST and NOR-A standard, the amplitude is maintained very accurately (Figure 7.24). This is probably because BSSN and NOR-A with the D_0^2 discretization require artificial dissipation which damps the amplitude of the wave.

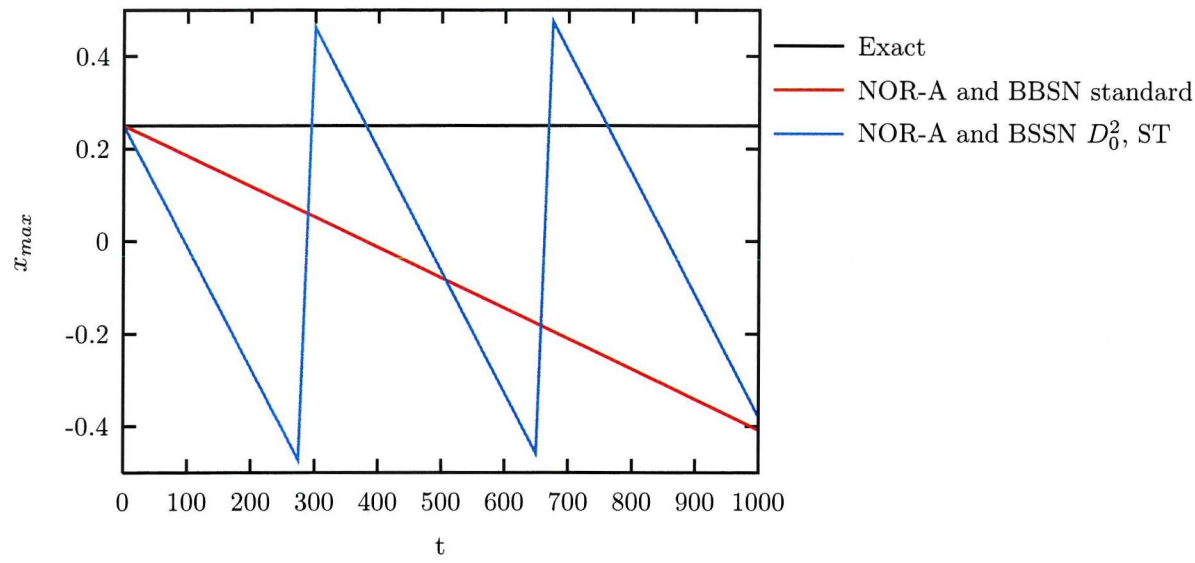

Figure 7.25: Linear wave phase error comparison, NOR and BSSN using standard discretization. The x coordinate of the maximum is plotted against time every crossing time. For the exact solution, the maximum should remain at the same coordinate.

Figure 7.25 shows the x coordinate of the maximum of γ_{yy} every crossing time as a function of time for each system. For the exact solution, the maximum should remain at the same coordinate. The phase error is linear in time for all three systems, indicating that the numerical speed of propagation has a constant error in it. NOR and BSSN propagate the solution at the same speed, though the error in the speed for ST is much larger.

7.6.5 Collapsing Gowdy

We evolve the collapsing Gowdy spacetime and plot the relative error in γ_{zz} as a function of time for the different formulations in figure 7.26. Note that data is given at $T = 2$ and the scheme is evolved backwards to $T = 0$. There is a singularity at $T = 0$, and γ_{zz} in the exact solution is singular there. During the approach to the singularity, ST has a relative error of less than 3%. BSSN and NOR-A have errors of less than 15% but NOR-B develops an error of order 100%. It seems that only on the final time step do the evolution variables attain infinite values; the solution is well-behaved until that point.

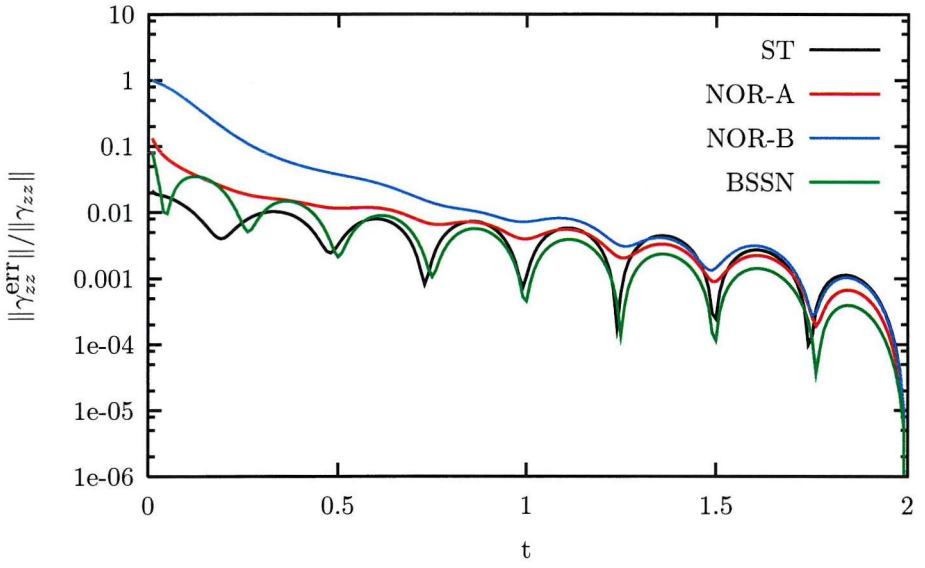

Figure 7.26: Relative error in γ_{zz} for the different formulations for the collapsing Gowdy spacetime

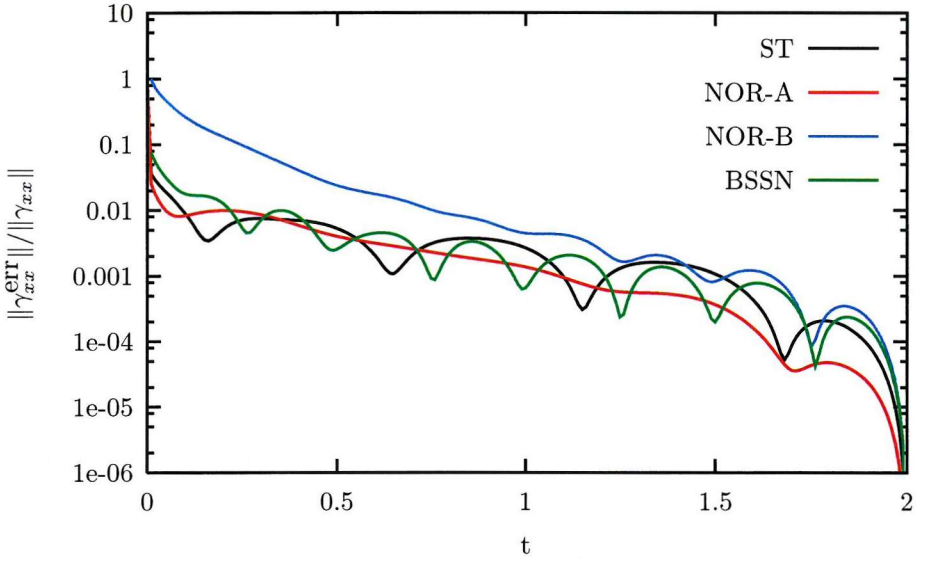

Figure 7.27: Relative error in γ_{xx} for the different formulations for the collapsing Gowdy spacetime

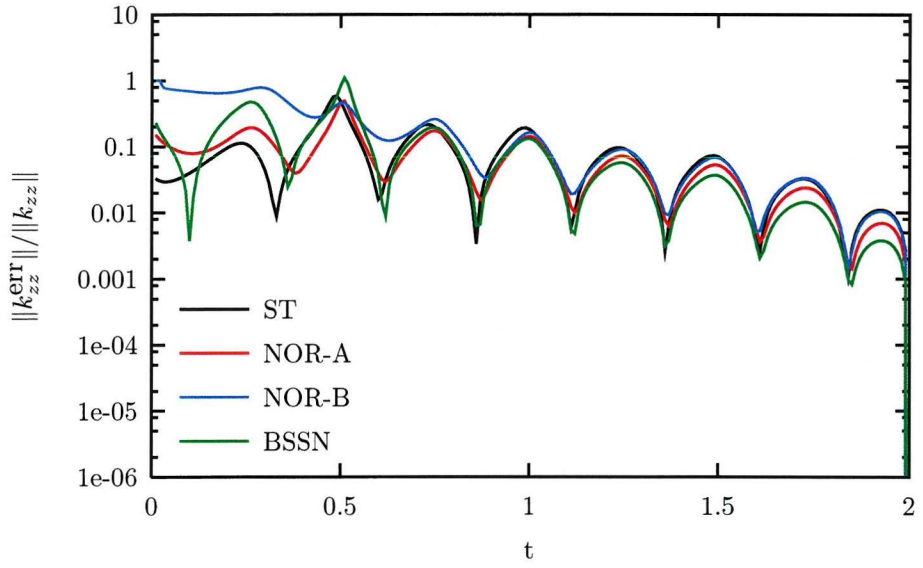

Figure 7.28: Relative error in K_{zz} for the different formulations for the collapsing Gowdy spacetime

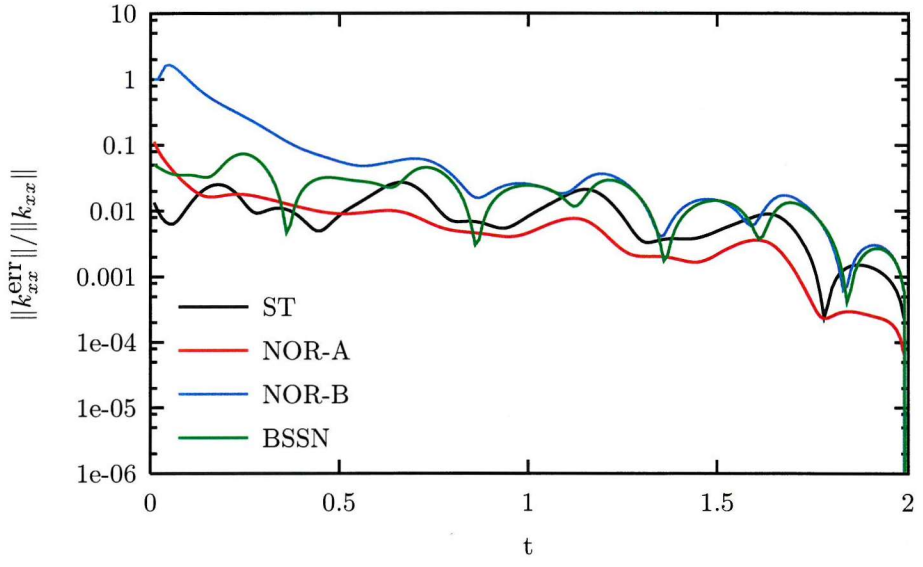

Figure 7.29: Relative error in K_{xx} for the different formulations for the collapsing Gowdy spacetime

7.6.6 Accuracy comparison summary

Errors in the propagation speed

For the wave testbeds, the errors in the propagation speeds are shown in Table 7.1.

	Gauge wave	Linear wave
NOR standard	1.57×10^{-3}	-6.58×10^{-4}
NOR D_0^2	-2.92×10^{-3}	-2.64×10^{-3}
BSSN standard		-6.50×10^{-4}
BSSN D_0^2		-2.63×10^{-3}
ST	-2.17×10^{-3}	-2.64×10^{-3}

Table 7.1: Errors in wave propagation speeds for the Einstein equations for various formulations

As discussed in Section 7.6.1, we predict that second order in space systems using the standard discretization may have propagation speed errors which are approximately four times smaller than fully first order systems. For the linear wave, the standard discretizations of NOR and BSSN have speed errors $\sim$4 times smaller than ST, in accordance with the prediction. The reduction in propagation speed error for the gauge wave when using second order rather than fully first order systems is less; the standard discretization of NOR has a propagation speed error which is $\sim$1.4 times smaller than the fully first order ST formulation.

For the linear wave, the D_0^2 discretizations of NOR-A and BSSN have speed errors approximately the same as ST, whereas for the gauge wave, the D_0^2 discretization of NOR-A has an error $\sim$1.35 times larger.

The prediction is verified for the linear wave, but it cannot be extended to the gauge wave.

Approach to singularity and the strong field regime

For collapsing Gowdy, only NOR-B shows a substantial difference in the behaviour of the error, where the relative errors in γ_{zz} and γ_{xx} are an order of magnitude larger at the approach to the singularity than for the other formulations.

NOR and BSSN

In all the tests performed here, BSSN behaves the same or worse than NOR-A. Additionally, it seems that BSSN requires artificial dissipation for convergence, whereas NOR-A and NOR-B do not. Real-world simulations involve artificial boundaries and more complicated solutions, and it is possible that BSSN has advantages in those situations. NOR-B has errors an order of magnitude larger than NOR-A in the collapsing Gowdy test, so there is a slight indication that NOR-A may be more suitable than NOR-B for spacetimes involving singularities.

The D_0^2 discretization for second order in space systems

Using the D_0^2 discretization as suggested for second order in space systems as suggested in [8] removes the major source of error in the gauge wave simulations for NOR and BSSN. However, using this discretization means that one of the benefits of using a second order in space system (reduced errors in the propagation speed) are lost. The fully first order ST system behaves the same or better than the D_0^2 discretizations of the second order in space systems. It should also be noted (Section 5.7.2) that using this discretization may lead to stability problems, and it is unknown whether the use of artificial dissipation can fully rectify these problems.

Overall conclusions

It has been shown that the second order in space NOR and BSSN systems give errors in the propagation speed of a linear gravitational wave which are four times smaller than the errors from the fully first order ST system, in agreement with the result for the wave equation. However, the same cannot be said of the gauge wave, where the error is only 1.4 times smaller. For the gauge wave test, the second order in space NOR and BSSN systems with the standard second order accurate discretization give rise to a mode which grows worse than exponentially with time. On the basis of this test, we must rule out these systems. This mode is not present when the D_0^2 discretization is used, but the advantages of the second order formulation in terms of reduced propagation speed are also lost. The NOR-A system is at least as accurate, and in some cases is more accurate, than BSSN in all the tests.

We conclude that *on the basis of these tests* the fully first order ST system is the most promising, and that when a second order in space formulation is used, NOR is more accurate than BSSN.

Chapter 8

Conclusions

In this work, we have shown how stability can be defined for discretizations of second order in space linear systems. The textbook definition of stability is usually given in the discrete $l_{2,h}$ norm. For systems which are first order in time but second order in space, and have finite speeds of propagation, it is necessary to use a norm containing difference operators. Using these norms for stability, such systems can be shown to converge in the $l_{2,h}$ norm as required. This use of norms containing difference operators is analogous to the use of norms containing derivatives when proving well-posedness for continuum first order in time and second order in space systems.

We consider a general form for a first order in time and second order in space system and show how to perform a *discrete reduction to first order in Fourier space.* Stability conditions for the first order system are then translated into direct requirements on the *second order* system, meaning that it is only necessary to check these conditions on a case-by-case basis, and it is not necessary to perform the reduction to first order for every system.

This technique is applied to discretizations of the ADM and NOR formulations of the Einstein equations, and Courant limits for these systems are derived. The ADM equations are shown to be unstable as expected. The linearized NOR system is shown to be conditionally stable in an appropriate norm.

The *Kranc* computer algebra software package has been developed to study complex nonlinear 3D systems such as the Einstein equations as time evolution problems. This software automatically generates computer code for solving the finite difference equations from an abstract mathematical description. This greatly reduces the amount of time needed to implement these problems numerically. In comparison to writing the code by hand, the

task of implementing different formulations of the Einstein equations is vastly simplified by using automated code generation. The obvious limitation of the Kranc system is that it does not include an easy way to specify boundary conditions at artificial boundaries. For first order in space systems, there is well-developed theory in this area, but the theory for second order in space systems is more limited. Future work will include enhancements to Kranc to allow the user to specify boundary conditions, as well as the structural changes necessary to support the Cactus-based mesh refinement system *Carpet*. There is currently much work being done on using multiple computational grids to cover more than one coordinate patch of the spacetime [63, 44]. Kranc can be extended to support these systems, which will eventually allow simulation of 3D spacetimes with smooth inner and outer boundaries, a capability which will hopefully lead to improvements in accuracy and stability of astrophysical simulations.

The Kranc package has been used to implement the NOR, BSSN and ST formulations, and these formulations have been experimentally tested for convergence using a set of standard numerical relativity tests. The standard discretization of each of these formulations is determined to be convergent for all the tests, though this implementation of BSSN requires the use of a small amount of artificial dissipation. A quantitative comparison of the accuracy of the different formulations in the standard tests is performed. For the wave equation, the numerical error in the speed of propagation of the solution is expected to be four times lower for a second order in space system than for a fully first order system. This result is reproduced for a linear gravitational wave, but the same is not true for a gauge wave (Minkowski spacetime in sinusoidally perturbed coordinates). Also, the NOR and BSSN systems exhibit numerical errors which grow worse than exponentially with time, whereas the ST system does not. For these reasons, we conclude that based on these simple tests, the ST system is more suitable for numerical evolutions than the NOR or BSSN systems. We note however that realistic astrophysical simulations require the use of sophisticated boundary treatments which are not modelled in these tests, and in these cases, it is possible that the superiority of the ST formulation may be lost.

Tests of this type can be extended in various ways. Fourth order accurate finite differencing has been implemented in Kranc, and it would be interesting to see to what extent this improvement in accuracy affects the conclusions concerning first and second order systems. The main problem noted for second order systems was the exponentially growing mode in the gauge wave test. The origin of this mode has been discussed in [8], and various remedies are discussed. A comparison of the use of these remedies with the use of a first order system

would give more indication as to whether second order systems are to be preferred.

Appendix A

Some results from linear algebra

A.1 Vector norms

A mapping $S \to \mathbb{R}, v \mapsto |v|$ on a vector space S is called a *norm* if it satisfies the following properties:

- $|v| = 0$ iff $v = 0$
- $|kv| = |k||v|$ for $k \in \mathbb{C}$, where $|k|$ is the absolute value of the complex number k
- $|u + v| \leq |u| + |v|$

For example, the l_2 or *Euclidean* norm defined on $\mathbb{C}^n$ by

$$|v| \equiv (v^* v)^{1/2} = \left(\sum_{i=1}^{n} |v_i|^2 \right)^{1/2} \tag{A.1}$$

is usually called "the" vector norm.

Two norms $|\cdot|$ and $|\cdot|'$ on a space S are said to be *equivalent* if $\exists K$ such that

$$K^{-1}|v|' \leq |v| \leq K|v|' \tag{A.2}$$

for all $v \in S$. This relation between $|\cdot|$ and $|\cdot|'$ is an equivalence relation. For finite dimensional linear spaces, all norms are equivalent. Specifically, all norms on $\mathbb{C}^n$ are equivalent to the Euclidean norm.

A.2 Matrix inequalities

The following notation for inequalities of matrices will be used:

$$A \leq B \equiv \forall x, x^* A x \leq x^* B x \tag{A.3}$$

A.3 Matrix norms

The set of $n \times n$ complex matrices forms a vector space, $\mathbb{C}^{n,n}$. A norm on this space is called a *matrix norm* if, in addition to the requirements for being a norm on the vector space, it satisfies the property

$$|AB| \leq |A||B| \tag{A.4}$$

A norm on the vector space $\mathbb{C}^n$ can induce a matrix norm on the space $\mathbb{C}^{n,n}$ by the following relation:

$$|A| \equiv \sup_{|v|=1} |Av| \tag{A.5}$$

Specifically, the matrix norm induced by the vector l_2 norm is usually called "the" matrix norm.

Matrix norms are unaffected by unitary transformations. For U a unitary matrix,

$$|U^* A U| = |A| \tag{A.6}$$

A.4 Calculating matrix norms

The norm of a matrix is given by

$$|A| \equiv \sup_{|v|=1} |Av| \tag{A.7}$$

For any matrix, (A.5) can be calculated by

$$|A| = \sigma(A^* A)^{1/2} \tag{A.8}$$

where

$$\sigma(M) \equiv \sup_{\nu} |m_\nu| \tag{A.9}$$

$$\tag{A.10}$$

where $\{m_\nu\}$ are the eigenvalues of M. $\sigma(M)$ is called the *spectral radius* of M.

If the matrix A is *normal*, i.e. $[A, A^*] = 0$, then we have

$$|A| = \sigma(A) \tag{A.11}$$

Suppose that $Q \in \mathbb{C}^{n,n}$ is a polynomial in a normal matrix P:

$$Q = \sum_{s=0}^{p} a_s P^s \tag{A.12}$$

Then Q is also normal. Further, by writing

$$Q = U^* \Lambda U \tag{A.13}$$

$$\Lambda = \mathrm{diag}(q_1, \ldots, q_m) \tag{A.14}$$

it can be seen that

$$|Q| = \sup_{\nu} \left| \sum_{s=0}^{p} a_s {p_\nu}^s \right| \tag{A.15}$$

A.5 Fractional powers of Hermitian positive definite matrices

Consider a positive definite Hermitian matrix H which is diagonalized by a unitary matrix U and has eigenvalues λ_i. Write

$$H = U^* \Lambda U \tag{A.16}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$. Then a fractional power of H is defined by

$$H^\alpha \equiv U^* \Lambda^\alpha U \tag{A.17}$$

where the power of the diagonal matrix is defined component-wise.

A.6 Energy norms

Writing the l_2 norm on $\mathbb{C}^n$ as $|\cdot|$, and given a positive definite Hermitian matrix H, a new norm $|\cdot|_H$ can be defined:

$$|v|_H \equiv (v^* H v)^{1/2} = |H^{1/2} v| \tag{A.18}$$

called an *energy norm*. This vector norm can be used to induce a matrix norm

$$\begin{aligned} |A|_H &= \sup_{|v|_H=1} |Av|_H && \text{(A.19)} \\ &= \sup_{|H^{1/2}v|=1} |H^{1/2} A v| && \text{(A.20)} \\ &= \sup_{|H^{1/2}v|=1} |H^{1/2} A H^{-1/2} H^{1/2} v| && \text{(A.21)} \\ &= \sup_{|v|=1} |H^{1/2} A H^{-1/2} v| && \text{(A.22)} \\ &= |H^{1/2} A H^{-1/2}| && \text{(A.23)} \end{aligned}$$

where in the penultimate line we have relabelled $H^{1/2}v$ as v.

A.7 Relations between norms equivalent to the identity

Suppose $|\cdot|_H$ is an energy norm equivalent to the identity; i.e.

$$K^{-1} I \leq H \leq K I \tag{A.24}$$

then one can show

$$\begin{aligned} |v|_H = (v^* H v)^{1/2} &\leq K^{1/2} |v| && \text{(A.25)} \\ &\geq K^{-1/2} |v| && \text{(A.26)} \end{aligned}$$

and for a matrix $A \in \mathbb{C}^{n,n}$,

$$
\begin{aligned}
|A|_H &= \sup_{|v|_H=1} |Av|_H = \sup_v \frac{|Av|_H}{|v|_H} && \text{(A.27)}\\
&\leq \sup_v \frac{K^{1/2}|Av|}{K^{-1/2}|v|} && \text{(A.28)}\\
&\leq K \sup_v \frac{|Av|}{|v|} && \text{(A.29)}\\
&\leq K|A| && \text{(A.30)}
\end{aligned}
$$

Similarly,

$$
|A|_H \geq K^{-1}|A| \qquad \text{(A.31)}
$$

Summarizing:

$$
\begin{aligned}
K^{-1}I \leq \quad H \quad &\leq KI && \text{(A.32)}\\
K^{-1/2}|v| \leq \quad |v|_H \quad &\leq K^{1/2}|v| && \text{(A.33)}\\
K^{-1}|A| \leq \quad |A|_H \quad &\leq K|A| && \text{(A.34)}
\end{aligned}
$$

Appendix B

Miscellaneous

B.1 Multi-indices

Sometimes it is necessary to refer to an arbitrary number of indices, for example when an expression involves many spatial dimensions. A multi-index is an n-tuple of integers j_i where $i = 1, \ldots, n$. An expression such as

$$u_{j_1 j_2 \ldots j_n} \tag{B.1}$$

can be compactly written

$$u_j \tag{B.2}$$

The *order* of a multi-index j is defined to be

$$|j| \equiv \sum_{i=1}^{n} j_n \tag{B.3}$$

Appendix C

Discrete Fourier Transform

C.1 Definition

Let $u = (u_0, \ldots, u_{N-1}) \in \mathbb{C}^N$. The *discrete Fourier transform* of u is $\hat{u} = (\hat{u}_0, \ldots, \hat{u}_{N-1}) \in \mathbb{C}^N$ defined by

$$\hat{u}_k = \sum_{j=0}^{N-1} u_j e^{-2\pi ijk/N} \tag{C.1}$$

Forming the expression

$$\begin{aligned}
\frac{1}{N}\sum_{k=0}^{N-1} \hat{u}_k e^{2\pi ijk/N} &= \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{j'=0}^{N-1} u_{j'} e^{-2\pi ij'k/N}\right) e^{2\pi ijk/N} && \text{(C.2)}\\
&= \sum_{j'=0}^{N-1}\left(\frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi i(j-j')k/N}\right) u_{j'} && \text{(C.3)}\\
&= \sum_{j'=0}^{N-1} \delta_{jj'} u_{j'} && \text{(C.4)}\\
&= u_j && \text{(C.5)}
\end{aligned}$$

it can be seen that given $\hat{u}$, u can be recovered with the inversion formula

$$u_j = \frac{1}{N}\sum_{k=0}^{N-1} \hat{u}_k e^{2\pi ijk/N} \tag{C.6}$$

We now consider two variations in the notation used for the discrete Fourier transform.

C.2 Wavenumber notation

Suppose first that u is a grid function with N points over a coordinate domain $[0, L)$, and these points have coordinates $x_j = hj$ with $j = 0, \ldots, N-1$ and $h = L/N$. Introducing the notation $\omega_k \equiv 2\pi k/L$, u can be written

$$u_j = \sum_{k=0}^{N-1} \hat{u}_k e^{i\omega_k x_j} \tag{C.7}$$

This can be interpreted as u being the sum of N modes at spatial frequencies (wave numbers) $0, 2\pi/L, \ldots, 2\pi(N-1)/L$ with amplitudes $\hat{u}_0, \ldots, \hat{u}_{N-1}$. The function

$$u(x) \equiv \sum_{k=0}^{N-1} \hat{u}_k e^{i\omega_k x} \tag{C.8}$$

is called the *trigonometric interpolant* of u. It satisfies the property

$$u(x_j) = u_j \tag{C.9}$$

C.3 Grid independent frequency range

Alternatively, it can be convenient to work with a spatial frequency whose range does not depend on the number of grid points or the coordinate size of the grid. The range chosen is $(-\pi, \pi]$. Define $\xi_k = 2\pi k/N$ to obtain

$$u_j = \frac{1}{N} \sum_{k=0}^{N-1} \hat{u}_k e^{ij\xi_k} \tag{C.10}$$

If $\hat{u}$ is extended using (C.1) to all $k \in \mathbb{Z}$, then $\hat{u}_k = \hat{u}_{k+N}$, i.e. $\hat{u}$ is periodic with period N, and the range of summation in (C.10) can be changed to obtain

$$u_j = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} \hat{u}_k e^{ij\xi_k} \tag{C.11}$$

(We have assumed N to be even. For N odd, the range could be chosen to be $-(N+1)/2 \ldots (N-1)/2$.) Hence $\xi_k = -\pi + 2\pi/N, -\pi + 4\pi/N, \ldots, \pi$. The notation

$$\hat{u}(\xi_k) \equiv \hat{u}_k \tag{C.12}$$

is sometimes used. Further, if $u_j^* = u_j$, then

$$\hat{u}_{-k} = \hat{u}_k^* \tag{C.13}$$

To summarize, in the remainder of this work, a grid function and its discrete Fourier transform will be related as follows:

$$u_j = \frac{1}{N} \sum_{\xi} \hat{u}(\xi) e^{ij\xi} \qquad \hat{u}(\xi) = \sum_{j} u_j e^{-ij\xi} \tag{C.14}$$

where j and ξ take the N values

$$j = 0, \ldots, N-1 \qquad \xi = -\pi + 2\pi/N, -\pi + 4\pi/N, \ldots, \pi \tag{C.15}$$

C.4 Extension to more than one spatial dimension

These results are generalized to s spatial dimensions via

$$u_j = \frac{1}{N} \sum_{\xi} \hat{u}(\xi) e^{i\langle j,\xi\rangle} \qquad \hat{u}(\xi) = \sum_{j} u_j e^{-i\langle j,\xi\rangle} \tag{C.16}$$

where $N = N_1 N_2 \ldots N_s$, j and ξ are now multi-indices (see Appendix B).

$$j = j_1 j_2 \ldots j_s \tag{C.17}$$

$$\xi = \xi_i \xi_2 \ldots \xi_s \tag{C.18}$$

The components of these take the values

$$j_r = 0, \ldots, N_r - 1 \tag{C.19}$$

$$\xi_r = -\pi + 2\pi/N_r, -\pi + 4\pi/N_r, \ldots, \pi \tag{C.20}$$

The inner product is defined as

$$\langle j, \xi \rangle \equiv \sum_{r=0}^{N-1} j_r \xi_r \tag{C.21}$$

and the sums are

$$\sum_{j} \equiv \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \cdots \sum_{j_s=0}^{N_s-1} \tag{C.22}$$

The sum over ξ is

$$\sum_{\xi} \equiv \sum_{\xi_1=-\pi+2\pi/N_2}^{\pi} \sum_{\xi_2=-\pi+2\pi/N_2}^{\pi} \cdots \sum_{\xi_s=-\pi+2\pi/N_s}^{\pi} \tag{C.23}$$

C.5 Infinite non-periodic grid

Finally, when u is a grid function on an infinitely extended uniformly spaced grid, $\hat{u}(\xi) \in [-\pi, \pi)$ can be considered as a continuous periodic function, and u as its Fourier series.

Appendix D

Kranc reference

D.1 Data structure specifications

Here we describe in detail the data structures which are used when calling the KrancThorns functions.

D.1.1 Calculation

Key	Type	Description
Equations	list of lists	`{loop1, loop2}` – Each loop is a list of rules of the form *variable* `->` *expression* where *variable* is to be set from *expression*
Shorthands (optional)	list of symbols	Variables which are to be considered as 'shorthands' for the purposes of this calculation
Name (optional)	string	A name for the calculation
Before (optional)	list of strings	Function names before which the calculation should be scheduled.
After (optional)	list of strings	Function names after which the calculation should be scheduled.

D.1.2 GroupCalculation

A GroupCalculation structure is a list of two elements; the first is the name (a string) of a Cactus group and the second is the Calculation to update the variables in that group.

D.1.3 GroupDefinition

A GroupDefinition structure is a list of two elements. The first is the name (string) of a Cactus group and the second is the list of variables (symbols) belonging to that group. The group name can be prefixed with the name of the Cactus implementation that provides the group followed by two colons (e.g. "ADMBase::metric"). If this is not done, then the KrancThorns functions will attempt to guess the implementation name, usually using the name of the thorn being created.

D.2 KrancThorns function reference

Here we document the arguments which can be specified for the functions CreateBaseThorn, CreateMoLThorn, CreateSetterThorn, CreateTranslatorThorn and CreateEvaluatorThorn.

Note that we use Mathematica syntax for function-specific section headers. Underscores denote function arguments, and OptArguments stands for optional arguments, also referred to as named arguments below. These are given in the form `myFunction[..., argumentName -> argumentValue]`.

D.2.1 Common Named Arguments

The following named arguments can be used in any of the Create*Thorn functions:

Argument	Type	Description	Default
SystemName	string	A name for the evolution system implemented by this arrangement. This will be used for the name of the arrangement directory	"MyPDESystem"
SystemParentDirectory	string	The directory in which to create the arrangement directory	"."
ThornName	string	The name to give this thorn	SystemName + thorn type
Implementation	string	The name of the Cactus implementation that this thorn defines	ThornName
SystemDescription	string	A short description of the system implemented by this arrangement	SystemName
DeBug	Boolean	Whether or not to print debugging information during thorn generation	False

D.2.2 Arguments relating to parameters

The following table describes named arguments that can be specified for any of the thorns except CreateBaseThorn. CreateBaseThorn is special because it can be used to define parameters which are inherited by each thorn in the arrangement, so the arguments it can be given are slightly different.

Argument	Type	Description	Default
RealBaseParameters	list of strings	Real parameters used in this thorn but defined in the base thorn	{}
IntBaseParameters	list of strings	Integer parameters used in this thorn but defined in the base thorn	{}
RealParameters	list of strings	Real parameters defined in this thorn	{}
IntParameters	list of strings	Integer parameters defined in this thorn	{}

D.2.3 CreateBaseThorn[groups_, evolvedGroupNames_, primitiveGroupNames_, OptArguments___]

Positional arguments

Argument	Type	Description
groups	list of GroupDefinition structures	Definitions of any groups referred to in the other arguments. Can supply extra definitions for other groups which will be safely ignored.
evolvedGroupNames	list of strings	Names of groups containing grid functions which will be evolved by MoL in any of the thorns in the arrangement.
primitiveGroupNames	list of strings	Names of groups containing grid functions which will be referred to during calculation of the MoL right hand sides in any of the thorns in the arrangement.

Named arguments

Argument	Type	Description	Default
RealBaseParameters	list of strings	Real parameters defined in this thorn and inherited by all the thorns in the arrangement	{}
IntBaseParameters	list of strings	Integer parameters defined in this thorn and inherited by all the thorns in the arrangement	{}

D.2.4 CreateEvaluatorThorn[groupCalculations_, groups_, OptArguments___]

Positional arguments

Argument	Type	Description
groupCalculations	list of GroupCalculation structures	The GroupCalculations to evaluate in order to set the variables in each group
groups	list of GroupDefinition structures	Definitions for each of the groups referred to in this thorn. Can supply extra definitions for other groups which will be safely ignored.

D.2.5 CreateMoLThorn[calculation_, groups_, OptArguments___]

Positional Arguments

Argument	Type	Description
calculation	Calculation	The calculation for setting the right hand side variables for MoL. The equations should be of the form `dot[`*gf*`] ->` *expression* for evolution equations, and *shorthand* `->` *expression* for shorthand definitions, which can be freely mixed in to the list.
groups	list of GroupDefinition structures	Definitions for each of the groups referred to in this thorn. Can supply extra definitions for other groups which will be safely ignored.

Named Arguments

Argument	Type	Description	Default
PrimitiveGroups	list of strings	These are the groups containing the grid functions which are referred to but not evolved by this evolution thorn	{}

D.2.6 CreateSetterThorn[calculation_, OptArguments___]

Positional Arguments

Argument	Type	Description
calculation	Calculation	The calculation to be performed
groups	list of GroupDefinition structures	Definitions for each of the groups referred to in this thorn. Can supply extra definitions for other groups which will be safely ignored.

Named Arguments

Argument	Type	Description	Default
SetTime (optional)	string	"initial_and_poststep", "initial_only" or "poststep_only"	"initial_and_poststep"

D.2.7 CreateTranslatorThorn[groups_, OptArguments___]

Positional Arguments

Argument	Type	Description
groups	list of GroupDefinition structures	Definitions for each of the groups referred to in this thorn. Can supply extra definitions for other groups which will be safely ignored.

Named Arguments

Argument	Type	Description
TranslatorInCalculation	Calculation	The calculation to set the evolved variables from some other source
TranslatorOutCalculation	Calculation	The calculation to convert the evolved variables back into some other set of variables

Bibliography

[1] M. Alcubierre *et al.*, Phys. Rev. D **62**, 044034 (2000).

[2] M. Alcubierre, *The status of numerical relativity*, Report on plenary talk at the 17th International Conference on General Relativity and Gravitation (GR17), held at Dublin, Ireland, July 2004, gr-qc/0412019.

[3] M. Alcubierre *et al.*, Class. Quantum Grav. **21**, 589 (2004).

[4] M. Alcubierre, B. Brügmann, P. Diener, F. S. Guzmán, I. Hawke, S. Hawley, F. Hermann, M. Koppitz, D. Pollney, E. Seidel, J. Thornburg, Phys. Rev. D **72** 044004 (2005).

[5] G. Allen *et. al.*, *Cluster Computing*, **4** 179 (2001).

[6] A. Anderson and J. W. York, Jr., Phys. Rev. Lett. **82**, 4384 (1999).

[7] R. Arnowitt, S. Deser, and C. Misner, in *Gravitation: An Introduction to Current Research,* edited by L. Witten (Wiley, New York, 1962).

[8] M. Babiuc, B. Szilágyi, and J. Winicour, *Some mathematical problems in numerical relativity*, 21st April 2004, gr-qc/0404092.

[9] J. Baker, M. Campanelli, C. O. Lousto and R. Takahasi, Phys. Rev. D **65** 124012 (2002)

[10] J. Baker, B. Brügmann, M. Campanelli, C. O. Lousto and R. Takhashi, Phys. Rev. Lett. **87**, 121103 (2001)

[11] T. Baumgarte and S. Shapiro, Phys. Rev. D **59**, 024007 (1999).

[12] T. Baumgarte and S. Shapiro, Phys. Rept. **376** 41–131 (2003).

[13] H. Beyer and O. Sarbach, Phys. Rev. D **70**, 104004 (2004).

[14] L. Blanchet, T. Damour, B. R. Iyer, C. M. Will and A. G. Wiseman, Phys. Rev. Lett. **74**, 3515 (1995)

[15] C. Bona, J. Masso, E. Seidel and J. Stela, Phys. Rev. Lett. **75** 600–603 (1995).

[16] C. Bona, T. Ledvinka, C. Palenzuela, and M. Žáček, Phys. Rev. D **69**, 064036 (2004).

[17] B. Brügmann, W. Tichy, N. Jansen, Phys. Rev. Lett. **92** 211101 (2004).

[18] Y. Bruhat, in *Gravitation, and Introduction to Current Research*, edited by L. Witten (John Wiley, New York, 1962).

[19] Cactus development team, http://www.cactuscode.org.

[20] G. Calabrese, I. Hinder and S. Husa, *Numerical stability for finite difference approximations of Einstein's equations*, 13th March 2005, gr-qc/0503056.

[21] G. Calabrese, J. Pullin, O. Sarbach and M. Tiglio, Phys. Rev. D **66**, 064011 (2002).

[22] Y. Choquet-Bruhat, Acta Math., 88, 141–225, (1952).

[23] G. B. Cook, Living Rev. Rev. , **5**, 1, (2000)

[24] C. Cutler, *et. al.*, Phys. Rev. Lett. **70**, 2984

[25] T. Damour, P. Jaranowski and G. Schäfer, Phys. Rev. D **66** 024007 (2002)

[26] P. Diener, Class. Quantum Grav. **20** 4901-4917 (2003).

[27] H. Friedrich and G. Nagy, Comm. Math. Phys. **201** 619–655 (1999).

[28] S. Frittelli and R. Gomez, J. Math. Phys. **41**, 5535 (2000).

[29] S. Frittelli and O. Reula, Phys. Rev. Lett. **76**, 4667 (1996).

[30] T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel and J. Shalf, *The Cactus Framework and Toolkit: Design and Applications*, in *Vector and Parallel Processing - VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science*, Springer, Berlin, 2003.

[31] C. Gundlach, J. M. Martín-García, Phys. Rev. D **70**, 044031 (2004).

[32] C. Gundlach, J. M. Martín-García, Phys. Rev. D **70**, 044032 (2004).

[33] C. Gundlach, J. M. Martín-Garcia, *Well-posedness of the NOR/BSSN formulation of the Einstein equations with dynamical lapse and shift conditions*, in preparation.

[34] B. Gustafsson, H. Kreiss, and J. Oliger, *Time dependent problems and difference methods* (John Wiley & Sons, New York, 1995).

[35] S. Hawking and G. Ellis, *The large scale structure of space-time*, Cambridge University Press, 1973.

[36] S. D. Hern, Ph.D. Thesis, University of Cambridge, 1999, gr-qc/0004036.

[37] S. Husa, I. Hinder, C. Lechner, *Kranc: a Mathematica application to generate numerical codes for tensorial evolution equations*, 6th April 2004, gr-qc/0404023.

[38] L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, Phys. Rev. D **64**, 064017 (2001).

[39] H. O. Kreiss, J. Lorenz, *Initial-Boundary Value Problems and the Navier-Stokes Equations* (Academic Press, Boston, 1989).

[40] H.-O. Kreiss and O. E. Ortiz, in Lecture Notes in Physics **604** (Springer, New York, 2002).

[41] H. Kreiss, N. Petersson, and J. Yström, SIAM J. Numer. Anal. **40**, 1940–1967 (2002).

[42] H.-O. Kreiss and G. Scherer, SIAM J. Numer. Anal., **29**, No. 3, 640–646 (1992).

[43] L. Lehner, Class. Quantum Grav. **18**, R25 (2001).

[44] L. Lehner, O. Reula, M. Tiglio, *Multi-block simulations in general relativity: high order discretizations, numerical stability, and applications*, 1st July 2005, gr-qc/0507004.

[45] L. Lindblom, M. A. Scheel, L. E. Kidder, H. P. Pfeiffer, D. Shoemaker, S. A. Teukolsky, Phys. Rev. D **69**, 124025 (2004).

[46] M. Miller, *On the Numerical Stability of the Einstein Equations*, 8th August 2000, gr-qc/0008017.

[47] G. Nagy, O. Ortiz, and O. Reula, Phys. Rev. D **70**, 044012 (2004).

[48] K. C. B. New, Living Rev. Rel. **6** 2 (2003)

[49] R. Penrose and W. Rindler, *Spinors & Space-time: Two-spinor Calculus & Relativistic Fields*, Cambridge University Press 1984.

[50] F. Pretorius, Class. Quant. Grav. **22** 425-452 (2005)

[51] F. Pretorius, *Evolution of Binary Black Hole Spacetimes*, 4th July 2005, gr-qc/0507014.

[52] R. H. Price and J. Pullin, Phys. Rev. Lett. **72**, 3297 (1994)

[53] O. Reula, Living Rev. Rel. **1**, 3

[54] O. Sarbach, G. Calabrese, J. Pullin, and M. Tiglio, Phys. Rev. D **66**, 064002 (2002).

[55] O. Sarbach and M. Tiglio, Phys. Rev. D **66**, 064023 (2002).

[56] E. Schnetter, Class. Quant. Grav. **21** 1465-1488 (2004)

[57] M. Shibata and T. Nakamura, Phys. Rev. D **52**, 5428 (1995).

[58] B. Szilágyi, H.-O. Kreiss, and J. Winicour, Phys. Rev. D **71**, 104035 (2005).

[59] B. Talbot, S. Zhou, G. Higgins, *Review of the Cactus framework*, `http://ct.gsfc.nasa.gov/esmf_tasc/Files/Cactus_b.html`

[60] S. A. Teukolsky, Phys. Rev. D **61**, 087501 (2000).

[61] J. W. Thomas, *Numerical Partial Differential Equations* (Springer-Verlag, New York, 1995).

[62] J. Thornburg, Class. Quant. Grav. **21(2)**, 743–766 (2004).

[63] J. Thornburg, Class. Quant. Grav. **21**, 3665–3692 (2004).

[64] R. Wald, *General Relativity*, University of Chicago Press 1984.

[65] J. W. York, Jr., *Kinematics and Dynamics of General Relativity*, in L. L. Smarr (Ed.), *Sources of Gravitational Radiation* (Cambridge University Press, Cambridge, 1979.