

UNIVERSITY OF SOUTHAMPTON

FACULTY OF SOCIAL AND HUMAN SCIENCES

Mathematics

Numerical simulations of neutron star crusts

by

Stephanie Jeanne Erickson

Thesis for the degree of Doctor of Philosophy

January 2015

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SOCIAL AND HUMAN SCIENCES
Mathematics

Doctor of Philosophy

NUMERICAL SIMULATIONS OF NEUTRON STAR CRUSTS

by Stephanie Jeanne Erickson

A neutron star has a solid crust and a fluid core. There are various different mechanisms that can break the crust, including magnetic field decay, spindown of a rotating star, and tidal forces due to a companion star in a binary merger. Although the crust makes up only a small portion of the mass of the star, its oscillations frequencies are different from the fundamental fluid modes, meaning that behavior originating in the crust can be distinguished from fluid behavior—in other words, starquakes can produce observable effects. They have been suggested as a possible mechanism behind quasiperiodic oscillations after magnetar flares, flares and outbursts in anomalous x-ray pulsars and soft gamma-ray repeaters, precursors to short-hard gamma-ray bursts, and pulsar glitches. Consequently, we want to investigate the dynamics of starquakes in neutron stars.

The goal of my PhD was to develop a toy model, which demonstrates methods that are necessary to perform realistic simulations of neutron stars. First, we need a mechanism to simulate the solid crust: for this, we implement a conservation-law formulation of non-linear elasticity in general relativity. The crust then must be coupled to a fluid core; in particular, the crust-core transition is much smaller than the size of the neutron star. This means that it is best modeled as a sharp transition; we use a ghost-fluid based method here. We use an atmosphere to handle the surface of the star where density, pressure, and internal energy all go to zero. Once these components are combined to produce a toy model, we shatter the crust and analyze the outcome. Although the results may not be physically significant, this is the first time that these technical aspects have been combined into a single simulation. This represents a significant step forward, showing that realistic simulations of starquakes *are* technically feasible.

Contents

Declaration of Authorship	xiii
Acknowledgements	xv
1 Introduction and motivation	1
1.1 Physics background	1
1.1.1 Neutron stars	1
1.1.2 The neutron star crust	2
1.1.3 Crust breaking	3
1.1.4 Magnetic field decay or reconnection	5
1.1.5 Pulsar glitch possibilities	7
1.1.6 Resonant shattering during binary inspirals	9
1.2 Motivation	10
1.3 Overview	10
2 Numerics background	13
2.1 Finite difference versus finite volume methods	13
2.1.1 Finite differencing	13
2.1.2 Finite volume methods	14
2.2 Method of lines	15
2.3 Time evolution	16
2.4 REA algorithm	16
2.5 Reconstruction methods	17
2.6 ENO/WENO reconstruction	19
2.7 Godunov-type methods	20
2.8 HLL Riemann solver	23
2.9 Hamilton-Jacobi equations	24
2.10 Lax-Friedrichs scheme	25
2.11 Fast-marching method	26
3 Elasticity	29
3.1 Elasticity Background	29
3.2 Kinematics	31
3.2.1 Evolution equations	31
3.2.2 Physical meaning of $\psi^A{}_i$ evolution equations	35

3.2.3	Relationship between particle number density and ψ^A_i	41
3.3	Dynamics	44
3.3.1	Derivation of stress-energy tensor	44
3.3.2	Evolution equations	52
3.3.3	Special source terms	55
3.4	Hyperbolicity	56
3.4.1	Second order system	57
3.4.2	Unsheared state	61
3.4.3	First order system	62
	Showing symmetric hyperbolicity for the first-order system	64
	Equivalence of solutions to $(\mathcal{E}^a_A, \mathcal{A}_A)$ and (E^a)	65
	Positive definiteness of the energy norm for the first- order system	66
3.4.4	Characteristics of the first-order system	68
3.4.5	Characteristics of the second-order system in the unsheared state	70
3.4.6	Characteristics of the second-order system in general	72
3.4.7	Hyperbolicity without constraint addition	73
3.5	Numerical elasticity code	74
3.5.1	Description of the <code>ElasticEvolution</code> code	74
3.5.2	n_D versus n_ψ	75
3.5.3	ψ versus F	76
3.5.4	Newtonian Riemann tests	76
3.5.5	Newtonian limit vs Newtonian code	79
3.5.6	Relativistic Riemann tests	81
3.5.7	A method for Riemann tests on a 2D grid	81
3.5.8	Two-dimensional Riemann tests	83
3.5.9	Two-dimensional rotor tests	85
3.5.10	Considerations for a two-dimensional cylindrical grid	88
	Source terms	88
	Transforming initial data and output	88
	Limitations on the initial data	89
	Boundaries	89
3.5.11	Tests in two-dimensional curvilinear coordinates	90
3.5.12	Initial data for numerical tests	92
	BDRT1	93
	BDRT1 transformed	93
	4-wave relativistic solution	94
	Rotor tests	94
4	Interfaces	97
4.1	Level-set methods	97
4.2	Ghost fluid method	99

4.2.1	Fluid-fluid interface	101
4.2.2	Fluid-fluid interface (using the ψ^A_i formulation)	101
4.2.3	Solid-fluid interface (setting the fluid)	102
4.2.4	Solid-fluid interface (setting the solid)	102
4.2.5	Solid-solid interface (slip conditions)	103
4.2.6	Solid-solid interface (stick conditions)	103
4.2.7	Solid-vacuum interface	103
4.2.8	Fluid-vacuum interface	104
	Neutron-star surface	104
4.2.9	Atmosphere interfaces	105
4.2.10	Isobaric fix	105
4.2.11	GFM in general relativity	105
4.3	Other interface methods	108
4.4	Results	108
4.4.1	Newtonian fluid-fluid test: Test B	109
4.4.2	Special relativistic fluid-fluid test: separation	110
4.4.3	Solid-fluid test: BOD1	112
4.4.4	Solid-fluid test: BOD2	114
4.4.5	Solid-solid test: slip	115
4.4.6	Solid-solid test: stick	117
4.4.7	Solid-vacuum test	117
4.4.8	Solid-atmosphere test	122
4.4.9	2D Riemann test	122
4.4.10	Conservation errors	124
5	Shattering	129
5.1	Earthquakes	129
5.2	Shattering	130
5.3	Results	132
5.3.1	Homogeneous initial data	132
5.3.2	Starquake toy model	140
6	Conclusions	149
6.1	Summary	149
6.2	Future work	151
A	Introduction to hyperbolicity	153
A.1	An example	153
A.2	Definition of well-posedness	155
A.3	The first-order, constant-coefficient, 1D system	158
A.4	The symmetrizer in a constant coefficient system	162
A.5	The constant-coefficient, first-order, multi-dimensional system	167
A.6	Hyperbolicity in differential geometry notation	170
B	Additional information about the elasticity formulation	175

B.1	3+1 split of spacetime	175
B.2	The Newtonian limit	177
B.3	Shear invariants and the shear scalar	178
C	Equations of state	181
C.1	A toy relativistic EOS	181
C.2	Cranfield EOS	182
C.3	JWL	183
D	Estimating error	185
D.1	Richardson scaling	186
E	Relativistic conservation-error calculation bug	189
F	Tensor properties of g^{AB} on matter space	191
	References	193

List of Figures

1.1	The structure of the neutron star crust	3
2.1	Oscillations induced by a simple one-sided slope function	17
3.1	Sketch of two manifolds and the map between them	32
3.2	Sketch of the 1-form components of ψ^A_i in spacetime	35
3.3	Matter-space coordinates superimposed onto spacetime	36
3.4	Illustrations of crystal ‘surgery’ that is <i>not</i> allowed by the jump conditions for the constraints	38
3.5	Discontinuities that <i>are</i> allowed by the jump conditions from the evolution equations	39
3.6	The wave structure of the BDRT1 Newtonian Riemann test	77
3.7	Low-resolution results of the BDRT1 Riemann test	78
3.8	High-resolution results of the BDRT1 Riemann test	79
3.9	Newtonian limit of the relativistic code, shown with the BDRT1 Riemann test	80
3.10	Wave structure of a special relativistic 4-wave Riemann test	82
3.11	Low-resolution results of the relativistic 4-wave Riemann test	83
3.12	High-resolution results of the relativistic 4-wave Riemann test	84
3.13	Sketch of method for using 1D Riemann tests to test the 2D code .	85
3.14	Results for the BDRT1 Riemann test, evolved on a 2D grid	86
3.15	Results of a Newtonian rotor test	87
3.16	Results of a relativistic rotor test	88
3.17	Results of two tests evolved on a 2D cylindrical grid	90
3.18	1D slice of the BDRT1 Riemann test evolved on a 2D cylindrical grid	91
4.1	Sketch of the original ghost fluid method	101
4.2	Sketch of the interface in spacetime	106
4.3	Results of a fluid-fluid interface Riemann problem	109
4.4	Results of a relativistic fluid-fluid interface Riemann problem . . .	111
4.5	Results of a solid-fluid interface Riemann problem	113
4.6	Results of a solid-solid interface Riemann problem with slip boundary conditions	116
4.7	Results of a solid-solid interface Riemann problem with stick boundary conditions	118
4.8	Results of a solid-vacuum interface Riemann problem	119

4.9	Results of the solid-vacuum interface Riemann problem, evolved with an atmosphere in the place of the vacuum	120
4.10	Error produced by the atmosphere, as a function of the atmospheric pressure	121
4.11	Results of a 2D evolution of the solid-solid slip test	123
4.12	Cumulative error in conservation over time while using the ghost fluid method	126
4.13	Dependence of cumulative error in conservation and period on resolution	127
5.1	Results of shattering in a homogeneously stressed solid	133
5.2	1D slice of the results of the homogeneously stressed shattering simulation	134
5.3	Convergence of the error in the homogeneously stressed shattering simulation	135
5.4	1D homogeneously stressed shattering simulation	137
5.5	Close up of the right-traveling shear wave in the 1D shattering simulation	138
5.6	Convergence of the smooth regions of the 1D homogeneously stressed shattering simulation	139
5.7	Sketch showing the toy model surface with an atmosphere	142
5.8	Initial data for the toy model shattering simulation	144
5.9	Results of the toy model shattering simulation at $t = 0.3$	145
5.10	Results of the toy model shattering simulation at $t = 0.6$	146
5.11	Results of the toy model shattering simulation at $t = 1.1$	146
5.12	Convergence of the error in the toy model shattering simulation . .	148

List of Tables

4.1 Summary of interface treatment for different interface types	100
--	-----

Declaration of Authorship

I, Stephanie Jeanne Erickson , declare that the thesis entitled *Numerical simulations of neutron star crusts* and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as: [1]

Signed:.....

Date:.....

Acknowledgements

I would like to acknowledge the help of Carsten Gundlach; without his guidance, this thesis would not exist. I thank him for his patience, advice, discourse, and explanations. He has encouraged me to take the time to fully understand difficult concepts, and this has deepened my understanding of relativity, elasticity, and numerical methods.

Ian Hawke has also been integral to the success of this work. Through Ian's guidance, I have learned a great deal about numerics: from the implementation of numerical methods to the analysis of their results. His thoughtful criticism has also pushed me to become a better scientific writer and communicator.

Carsten and Ian have also made important contributions toward proofreading this thesis, and I am grateful for the time and effort that they have spent on this task.

In addition, I would like to thank John Muddle for his work on the `MultiModel` code. He has contributed to many discussions about the functioning of the code and identifying possible bugs.

I would also like to acknowledge the relativity group in Southampton as a whole; the community has been very welcoming and supportive. In particular, the PhD students have been an important social support network for me. I specifically should thank Christian Krüger for listening patiently to all of my rants, Lucy Keer for accompanying me on many stress-relieving walks in the countryside, and Stuart Wells for keeping me company when I needed to procrastinate for just a few more minutes: the best breakthroughs often come during times of rest.

My parents, John and Jill Erickson, have always supported my efforts in academic undertakings. An urge to build things, inspired by my father, and a good measure of determination, cultivated by my mother, are hopefully both reflected in this work. I also want to thank my sister, Heidi Erickson, who was always forced to be the “student” when we played as children. For the reader’s sake, I hope my explanation skills have improved since then.

Chapter 1

Introduction and motivation

1.1 Physics background

The following sections summarize some information about neutron stars, the neutron star crust, how it breaks, and some observed phenomena that could be linked to starquakes in neutron stars.

1.1.1 Neutron stars

Compact objects are stars at the end stage of stellar evolution; they no longer burn nuclear fuel, and as a result, can no longer be supported by thermal pressure. Because of this, they are extremely small and dense.

Neutron stars are an intermediate form of compact object (between white dwarf stars and black holes). They have too much mass to be held up by electron degeneracy pressure (as white dwarf stars are), but not enough mass to cause them to collapse completely. Instead, neutron stars are supported by the pressure of degenerate neutrons. Because neutron stars populate this intermediate space, the theory to describe them must incorporate a wide range of interesting theoretical physics: to fully understand them, we must combine our understanding of general relativity and nuclear physics.

Neutron stars were initially proposed by Baade and Zwicky 1934 [2] who posited that “a supernova represents the transition of an ordinary star into a neutron star, consisting mainly of neutrons,” as a passing comment in a paper focused on the analysis of supernovae.

In 1939, Tolman [3] and Oppenheimer and Volkoff [4] presented the first neutron star model. A static solution for Einstein’s field equations was found using an ideal gas equation of state. Assuming that the neutron star consisted of an

ideal gas of free neutrons at high density, a maximum mass of $0.7M_{\odot}$ with radius, $R = 9.6$ km was derived.

Most of the early work on neutron stars was motivated by the idea that main sequence stars could contain neutron cores; when it became clear that this was not the case, the theoretical community largely lost interest in the stars. However, this changed with the discovery of the first pulsar in 1968 [5]. Gold 1968 [6] proposed that the most likely source for these regular radio pulses were rotating neutron stars, where radiation is beamed along some axis other than the axis of rotation, so that pulses occur when the beam sweeps across the Earth. Observations of pulsars generated new theoretical interest in neutron stars.

In 1931, Chandrasekhar [7] calculated the maximum mass of a white dwarf to be $\sim 1.4M_{\odot}$; above this limit, stars must collapse further to become neutron stars or black holes. The upper limit on the neutron star mass is more uncertain, but it is expected to be $\sim 3M_{\odot}$. Observed masses of neutron stars range from $1 - 2.5M_{\odot}$ [8]. The typical neutron star has a radius of ~ 10 km. A good overview of the physics of neutron stars (as well as other compact objects) is given in [9].

The structure of a neutron star is roughly as follows. It has a outer envelope of normal nuclear matter at extreme temperatures and magnetic fields ($\rho \gtrsim 10^6$ g cm $^{-3}$), a solid outer crust consisting of a Coulomb lattice of heavy nuclei in a relativistic degenerate electron gas (10^6 g cm $^{-3} \lesssim \rho \lesssim 4 \times 10^{11}$ g cm $^{-3}$), an inner crust where the solid lattice is also permeated by a gas of free neutrons (4×10^{11} g cm $^{-3} \lesssim \rho \lesssim 2.8 \times 10^{14}$ g cm $^{-3}$), a core containing a gas of superfluid neutrons and Type II superconducting protons ($\rho \lesssim 2.8 \times 10^{14}$ g cm $^{-3}$), and possibly a further phase transition to more exotic matter towards the center of the star. In what follows, we focus on the neutron star crust.

1.1.2 The neutron star crust

Nuclear saturation density, or the density where each nucleon is at its minimum energy, is around $\rho_0 = 2.8 \times 10^{14}$ g cm $^{-3}$; however, we know that neutron stars could have central densities as high as $\sim 5\rho_0 - 10\rho_0$. Although theories exist to describe material at such high densities, this is far beyond the range that is achievable on Earth. However, most of the neutron star crust is in the region where $\rho < \rho_0$, so we know considerably more about this region, and methods developed for terrestrial materials can be applied [10].

Fig. 1.1 shows a sketch of the structure of the neutron star crust in ground state. The outer-most layer of the star is the envelope, which consists of mostly iron atoms. As the pressure and density increase, the atoms become ionized, with

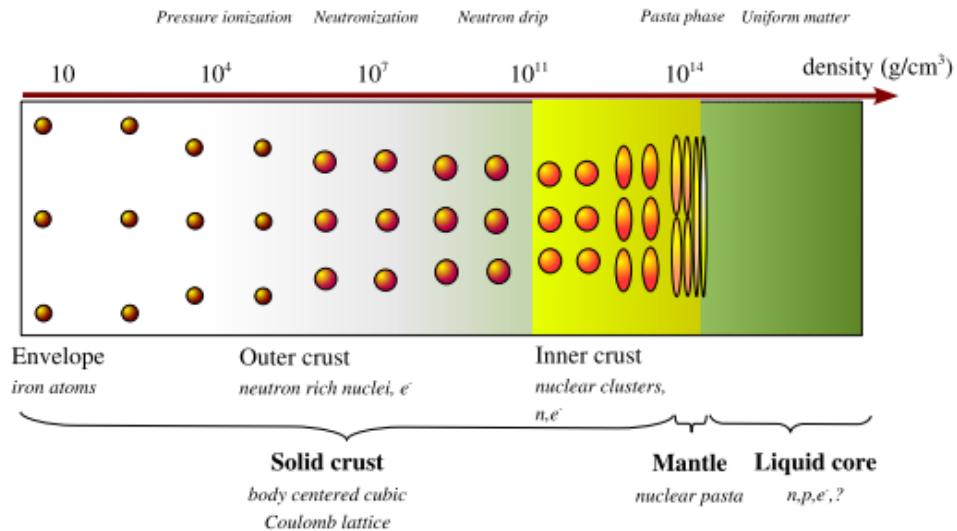


Figure 1.1: A sketch showing the structure of the neutron star crust in its ground state. *This image comes directly from [10].*

atoms consisting of only nuclei at densities of around $\rho \sim 10^4$ g cm⁻³. As density continues to increase to above around $\rho \sim 10^7$ g cm⁻³, electrons and protons begin to combine to become neutrons, making the material much more neutron-rich. As the density rises above $\rho_{ND} \approx 4 \times 10^{11}$ g cm⁻³, neutrons begin to “drip” out of nuclei to become unbound, and form a gas of free neutrons around the nuclei. As the density continues to increase, some calculations predict “pasta” phases consisting of non-spherical nuclei in the shape of elongated spheres, cylinders or slabs. Once the density goes above around 10^{14} g cm⁻³, nuclei can no longer exist, and there is a uniform plasma of nearly pure neutron matter [10].

There is a distinction between the outer crust, where the density is below $\rho < \rho_{ND}$, and the inner crust at higher densities. The outer crust consists of normal nuclear matter—albeit in extreme conditions—and can therefore be modeled using extrapolated experimentally measured values. However, there is much less certainty as the density is increased to $\rho > \rho_{ND}$, and we must rely solely on theoretical calculations.

1.1.3 Crust breaking

There are various different mechanisms that could break the neutron star crust, including magnetic field decay, spin down of a rotation neutron star, or tidal forcing due to a companion star in a binary merger system. Some of the possible mechanisms are discussed in the following sections. However, it is important to first discuss *how* crust breaking is likely to occur.

Horowitz and Kadau 2009 [11] perform molecular dynamics simulations of the neutron star crust material in order to estimate the breaking strain. In the neutron star crust, the atoms are completely pressure-ionized, and are immersed in a degenerate relativistic electron gas. The Coulomb interactions of the individual ions are screened, but only at large distances due to the electron gas. To describe the interaction between two particles, a screened Coulomb or Yukawa potential is used. Then the total potential energy is calculated summing over all particle pairs that are not screened from one another.

Shearing of the material is then simulated. The strain is applied by deforming a periodic boundary, and is applied linearly with time. The resulting shear stress is then calculated, and a stress-strain curve is produced. The timescales simulated are much shorter than astrophysical timescales, and a model-dependent extrapolation is needed to apply these results to relevant timescales, as mentioned below [11].

The breaking strain is measured as the peak of the stress-strain relation. Stress increases with strain until the material breaks, or *yields*, at a maximum breaking stress—further strain causes a decrease in the stress—and the associated strain is the breaking strain. Perfect crystals, which exhibit no defects or granularity, show a breaking strain well above 0.1, and breaking occurs in a rather abrupt fashion, with very little plasticity present; in other words, the stress-strain relation is linear nearly until the material yields. Although it is not possible to simulate the large time and length scales of the physical system, in the small range of variation that *is* possible, no size effects were present in single crystal simulations.

Simulations with multiple grains in the crystal were also performed. As it fails, the material deforms plastically along grain boundaries in a collective manner, rather than exhibiting dislocations or localized events. Because the failure is more collective and less localized, the crystal is stronger, and breaks at larger strains than terrestrial materials.

This is basically because the density and pressure are so high that electrons are contained in an electron gas, rather than associated with particular atoms, so variations in electron density cannot cause localized structure to allow for local defects. These molecular dynamics simulations agree with the earlier predictions of Jones 2003 [12], which showed that it is not possible to have voids or fractures in the material: essentially, this can be thought of as localized defects immediately healing due to the high pressure. In fact, when simulations are initialized with a void in an otherwise perfect crystal, this quickly heals and disappears, reducing the breaking stress by only 25% [11].

Chugunov and Horowitz 2010 [13] then look at how the breaking stress of the neutron star crust depends on both the temperature (via the coupling parameter), and the strain rate. Using molecular dynamics simulations, the parameters for the Zhurkov model of strength are estimated and used to determine what the breaking stress should be for timescales of ~ 1 s – 1 year; these are the timescales relevant for astrophysical applications, but they are much longer than is possible to model using molecular dynamics (typically molecular dynamics simulations have a timescale of ~ 1 ms, so this large extrapolation may be a cause for concern).

The results show that the breaking stress depends significantly on the temperature, with higher temperatures leading to lower breaking stresses. At higher temperatures, the breaking stress varies more for different strain timescales, with longer timescales producing lower breaking stresses. It is worth noting, though, that to get results at this time scale, the molecular dynamics results have to be extrapolated over more than 10 orders of magnitude; hence, they should be used with caution [13].

1.1.4 Magnetic field decay or reconnection

Both anomalous X-ray pulsars (AXPs) and soft gamma-ray repeaters (SGRs) are thought to be best described as isolated neutron stars with very high magnetic fields ($B \sim 10^{14} - 10^{15}$ G); however, they behave very differently, with SGRs exhibiting giant flares, and AXPs showing X-ray outbursts where the burst location correlates with the pulsar phase. Perna and Pons 2011 [14] attempt to explain some of these discrepancies by examining the model for these events.

In both cases, after birth, the magnetic field of the magnetar decays due to some dissipative process (Ohmic dissipation, ambipolar diffusion, Hall drift). At the star’s birth, the magnetic stresses are balanced by the elastic stresses in the crust, but as the magnetic field evolves, these stresses can go out of balance. Eventually the elastic stress can exceed its breaking stress, and the crust will break. This release of energy could result in outbursts or flares. Clearly this mechanism relies on the magnetic field, but it *is not* directly correlated to field strength. Namely, there are some low magnetic field stars that produce X-ray bursts, and some high-magnetic field stars that do not; why should this be the case if it is the magnetic field that is important?

Perna and Pons numerically follow the magnetic field evolution of the crust of a star as it ages. The elastic stress induced by the magnetic stress is calculated; if the breaking stress is exceeded, then the crust is said to have broken in that region. In this way, it is possible to map out the areas of the star where the crust will

fracture, as well as to estimate the energy released by the burst and the frequency with which bursts will occur over time.

The authors identify three different ages of stars that seem to correspond to different observed behaviors. “Young” stars, of around 400 – 1600 years behave in a more “SGR-like” manner; the energies released are higher, as is the event rate. As the stars age, the energy released by each burst begins to form a bimodal distribution, with a second peak about three orders of magnitude lower; the bursts also occur more often at smaller polar angles, exhibiting the correlation between burst location and pulsar phase characteristic of AXPs. [14]

This model suggests that starquakes would be the source of the flares of SGRs; basically, the magnetic field decays, causing the stress in the elastic crust to exceed the breaking stress, which causes the crust to break. The seismic waves in the crust couple to the magnetosphere, and a fireball forms above the surface. Levin and Lyutikov 2012 [15] suggest another mechanism for the production of SGRs. Instead of the starquakes causing the burst, their proposed mechanism is essentially the other way around. As the magnetic field evolves and untwists, it eventually reaches a stability threshold that requires it to reconnect in a process similar to that which produces coronal mass ejections from our sun (solar flares). This sudden relaxation and change in magnetic field topology results in a flare; it will also result in large-amplitude torsional oscillations, which would likely cause the crust to break.

By studying the dynamics of how thin cracks form in the crust, Levin and Lyutikov 2012 [15] reveal that the energy released from a thin crack is strongly suppressed, because the magnetic field does not allow much slippage between the two sides of the crack; even if cracks form, they are held in place by the magnetic field lines. The energy travels through the crust as seismic oscillations. Even for an optimistically large crack area, the energy released is $\sim 10^{12}$ times less than a typical weak SGR flare.

In any case, whether the seismic oscillations cause the flare or the flare causes the seismic oscillations, starquakes will still be important for the overall dynamics of the magnetar system.

In the NS environment, crack formation is suppressed in the first place because of high pressure [12]; even if cracks *did* form, we now know that strong magnetic fields would prevent slippage along those cracks, and the subsequent release of energy. However it *is* believed that some magnetar events *are* linked to crustal motion: the twist in the magnetosphere decays too quickly to be replenished by ambipolar diffusion or Hall drift. Because of this, Beloborodov and Levin 2014 [16] investigate the possible involvement of plastic flow in these events. There, a new instability is described, which leads to a thermoplastic wave (TPW), similar in

quality to a deflagration front seen in terrestrial combustion simulations. Basically, over a critical stress, the NS crust begins to deform plastically; this leads to heating of that region of the crust. The heating, in turn, lowers the critical stress of the material, which speeds the plastic deformation.

Although the TPW is likely to operate in the neutron star environment, it cannot explain all magnetar events. In particular, it cannot explain short duration bursts $\sim 0.1\text{--}0.3$ s [16]. These could be explained by magnetic reconnection events in the magnetosphere, as mentioned above, or by some faster failure mechanism in the crust.

A related phenomenon is that of quasiperiodic oscillations (QPOs) after X-ray flares from magnetars. Magnetar flares are most likely powered by reconfigurations of the decaying magnetic field; since the field is pinned to the crust, which is highly conductive, these events cause the crust to break and generate global seismic oscillations [17]; this is similar to the mechanism mentioned previously for SGRs [15].

These QPOs reflect crustal frequencies, and can be measured fairly accurately. It turns out that the frequency of shear oscillations in the crust depends sensitively on the nuclear symmetry energy, meaning that QPOs can constrain this property, which is a significant uncertainty in the current description of the neutron star crust. Basically, the speed of shear waves in the crust is strongly affected by the nuclear symmetry energy, and this, in turn, affects the fundamental-mode frequency and first radial overtone [18].

1.1.5 Pulsar glitch possibilities

Baym *et al* 1969 [19] suggest a model for the sudden observed increase in pulse frequency of some pulsars, called “glitches.” Generally, the systems spin up, and then slowly relax back to their original frequencies. Several hundred glitches have since been observed; their magnitude being around $\Delta\Omega_c/\Omega_c \sim 10^{-9} - 10^{-6}$, where Ω_c is the observed rotation frequency [20].

The scenario suggested by Baym *et al* is as follows. As the star forms, it is rotation, and therefore slightly oblate; the crust freezes in this configuration. As the spin frequency of the star decreases, the centrifugal force decreases, putting the system out of balance and inducing elastic stresses in the crust of the star. Eventually, when the breaking stress of the crust is exceeded, the crust will break, and the shape of the star will change; because the star will become more spherical, its moment of inertia will decrease. The timescale of the cracking is much shorter

than the timescale of spin down, so we can apply conservation of angular momentum: the sudden decrease in moment of inertia must induce a sudden increase in the angular velocity of the star. This is what is observed as the glitch [19].

The initial cracking and spin-up occur on a timescale of around $\sim 0.01 - 0.1$ s (the time it takes a shear wave to propagate through one stellar radius). Then this spin up is communicated to the charged particles in the interior of the star via the magnetic field within around ~ 100 s. Eventually, the neutron superfluid responds ($t \sim 1$ year) due to electron scattering off of the vortex lines of the superfluid—this timescale of relaxation after a glitch is used as evidence for the presence of superfluidity in neutron stars [19].

Another model for pulsar glitches was proposed by Anderson and Itoh 1975 [21]. In this model, glitches are due to the unpinning of vortices from the crust due to some instability or other unpinning mechanism. Basically, a superfluid rotates by forming a dense array of vortices; these induce global rotation of the system. If these vortices are pinned to the crustal lattice, then a rotational lag begins to build up between the crust and the superfluid as the crust slows down due to electromagnetic braking; the superfluid cannot change its rotation frequency, because the locations of its vortices are fixed by the pinning, and it can only change its global rotation by moving these vortices around. Eventually, some unknown mechanism breaks the pinning, and the vortices move, allowing some angular momentum to be transferred from the superfluid to the crust, which increases the crust's rotational frequency (and therefore, the observed pulsar frequency).

Glampedakis and Andersson 2009 [20] suggest a possible mechanism behind the unpinning of the vortices from the crust, showing the presence of a new instability, which acts on the inertial modes of a rotating superfluid star, and begins to have an effect above a certain critical rotational lag between the two rotating components: a superfluid neutron component and a combination of all of the charged particles in the star. The instability has a short enough growth timescale to overcome the relevant viscous damping timescale. The instability has roughly the right features for a glitch trigger mechanism. The maximum expected glitch size matches well with observations, as does the idea that systems should evolve into the instability region before they glitch. The time between successive glitches is estimated for various pulsars and compared to the observations.

Alternatively, Link 2012 [22] suggests that vortices could undergo vortex creep via thermal activation or quantum tunneling, which could cause the vortex lattice to become tangled. This would cause the superfluid to become turbulent, causing an instability—this is another possible mechanism behind unpinning.

One missing link is how these mechanisms actually break the pinning of the vortices to the neutron star crust. When the system evolves into the instability region, a range of unstable modes will grow on a timescale of a few rotation periods of the star, causing large-amplitude oscillations. It seems likely that this could cause the vortex pinning to break, but it is not clear exactly *how* this would come about [20]. Naively, our idea is that this could be due to breaking or shattering in the crust; another possibility is that starquakes would not be the mechanism that links these, but could occur as a side effect.

1.1.6 Resonant shattering during binary inspirals

An argument *against* the inclusion of an elastic crust in neutron star models might rely on the fact that the crust only makes up a small portion of the total mass of the star. One could also point out that the ratio between the shear modulus and the compression modulus is much smaller than one [10] in the outer crust, so the effect of the shear stress will be small compared to the effect of pressure in those regions; one might say that it could be just as effective to simply treat the material as a fluid.

However, the interaction between the crust and the core can introduce modes that are qualitatively different from pure fluid modes. Because the stresses are weak, crustal and interfacial modes have lower frequencies than purely fluid modes, meaning that these modes could be excited when purely fluid modes are not.

A possible mechanism behind the precursors to short-hard gamma-ray bursts (SGRBs) has been proposed by Tsang *et al* 2012 [23] and relies precisely on this. For a handful of SGRBs, a flare was observed preceding the main GRB event by around $1 - 10$ s. Crust cracking was suggested as a possible explanation behind these precursors, but the timescale implied cracking due to direct tidal deformation did not fit the observations.

Instead, Tsang *et al* 2012 [23] suggest that modes could be excited due to resonance with periodic tidal deformation of the system; tidal deformation occurs in each star due to its companion, and this will naturally vary with the orbital frequency of inspiral. A mode with amplitude concentrated at the crust-core interface is identified; this could be the source of crust breaking, and therefore the precursor flares [23].

Tsang also shows that this type of resonant shattering of the crust could occur when compact objects undergo close passes via eccentric or hyperbolic encounters. Although the event rate for these situations is low, the resulting flares could provide an important electromagnetic counterpart for the expected gravitational

wave signals from these systems, meaning that triggered searches could be used [24].

1.2 Motivation

It is clear that there are some astrophysical scenarios that will cause the crust of a neutron star to break. In fact, starquakes have been proposed as a possible mechanism behind a number of observed phenomena, as described in the previous sections. Because of this, we would like to investigate how crust breaking affects the dynamics of neutron star binary systems as well as isolated neutron stars. To do this, a number of technological advances are necessary.

Because of this, the goal of my PhD has been to develop a toy model that demonstrates a number of numerical methods that we will need in order to do more realistic, fully-relativistic neutron star simulations (i.e. binary neutron star mergers, etc). The toy model includes a solid crust coupled to a fluid core in a background Newtonian gravitational potential. A region of the crust is then shattered, and the system is allowed to evolve.

Obviously, to do this type of simulation it is first necessary to have some method for simulating a solid crust in general relativity; for this a conservation-law formulation for elasticity was developed and is described in Gundlach *et al* 2012 [1]. Next, the solid crust must be coupled to a fluid core. In particular, the crust-core transition is much smaller than the overall size of the neutron star, so this should be modeled as a sharp interface; this means that we will need some method for numerically evolving two different materials with a dynamical boundary between them. At the surface of the star, the density, pressure, and internal energy will all go to zero; we will need to take this into consideration to avoid numerical problems. Once we have all of these aspects together, we will also need some way to break the crust in the numerical code.

As described in the next section, each of these technical challenges is addressed in the following chapters of this thesis, culminating in a 2D toy model, which combines all of the methods into a single simulation.

1.3 Overview

For the sake of clarity, it is important to note that there are two separate codes that are used in this work. The first code, `ElasticEvolution`, was written by me and Ian Hawke, and is the code that was used in [1] to demonstrate the general relativistic elasticity formulation presented there. The code is general relativistic

with a general background metric. It can run 1D and 2D simulations with 3D variables. `ElasticEvolution` uses a choice of slope limiting techniques to reconstruct the primitive variables; however, all the results shown here use van Leer's MC limiter [25]. The numerical fluxes are calculated using the HLL approximate Riemann solver [26]. Quantities that are advected in `ElasticEvolution` are done so using upwind differencing.

Preliminary work on 1D material interfaces was done in the `ElasticEvolution` code, but eventually it became clear that 2D implementation within this code would take a significant time investment. For this reason, we moved to using the `MultiModel` code, originally implemented by Ian Hawke and John Muddle, to make use of the 2D infrastructure already in place in that code, adding an elasticity model, as well as methods to handle different types of material interfaces. `MultiModel` also runs 1D and 2D simulations using 3D variables, and makes use of the HLL approximate Riemann solver. However, instead of slope limiter reconstruction methods, `MultiModel` uses WENO reconstructions [27]; for all results shown here, the primitive variables are reconstructed. The code is also *not* general relativistic; all of the results shown here from the `MultiModel` code will be Newtonian (again, some special relativistic models have been implemented in the `MultiModel` code, but the metric is hard-coded, and no results are shown here).

For advection, `MultiModel` uses the Lax-Friedrichs method for Hamilton-Jacobi equations with WENO reconstruction of the derivatives. For reinitialization of the level-set function, as well as extrapolation as prescribed by the ghost-fluid method, the fast-marching method is used.

Both codes use the method of lines to convert the set of partial differential equations to a set of ordinary differential equations and solve the time evolution part using the Runge-Kutta method. Typically, for the results shown here, the third-order Runge-Kutta method is used.

In Chapter 2, we discuss some general numerics background and some numerical methods that we use to implement the elasticity formalism and interface treatment in the codes described later. In this chapter we discuss the REA algorithm, reconstruction methods, ENO and WENO schemes, Godunov-type methods for evolving conservation-law systems, and the HLL approximate Riemann solver. Much of this chapter is a summary of content from [28].

The next chapter, Chapter 3, is based on Gundlach *et al*, 2012 [1]. We cover the kinematic and dynamic equations describing elasticity, deriving a conservation-law formulation of the equations, and then showing that, with appropriate constraint additions, the system is symmetric hyperbolic. We then go on to discuss the numerical code written based on this formulation, and various numerical tests that

show the functioning of the code. In the paper [1], the content in the kinematics, dynamics, and hyperbolicity sections is the work of Carsten Gundlach; I have rewritten this material to get a better understanding of these sections, but the content is largely the same. The numerics section of the paper was written by Ian Hawke and myself, and is included as is.

Chapter 4 discusses methods used for interfaces between different materials. It includes a discussion of level-set functions and the ghost-fluid method (GFM) for fluids. The GFM is then adapted to other types of material interfaces, and a description of the method in general relativity is included. We also look at some numerical tests of the implementation of these methods.

Chapter 5 discusses the mechanism for shattering in our code. It shows some results for tests of numerical shattering mechanisms in 2D simulations, as well as a description of the final 2D toy model and a description of the results of this simulation.

Chapter 6 summarizes conclusions of this work and discusses possible next steps.

There are also two appendices included for the sake of completeness. The first, Appendix A, includes a summary of what hyperbolicity is and why it is important; this section closely follows the lecture notes of Kreiss and Busenhart [29], with the last section on hyperbolicity in differential geometry notation covering material from Beig and Schmidt [30], and Anile [31].

Appendix B covers additional information relevant to the elasticity formulation covered in [1]. Appendix C discusses the various equations of state used in this work, while Appendix D clarifies the method used for calculating the error for various convergence tests. Lastly, Appendix E discusses some details of a minor unresolved bug in the calculation of the error in conservation for relativistic tests.

Chapter 2

Numerics background

2.1 Finite difference versus finite volume methods

2.1.1 Finite differencing

Say we have a system of conservation laws as follows:

$$q_{,t} + f(q)^l_{,m} = 0, \quad (2.1)$$

where q is generally a vector of conserved quantities, $f(q)^j$ are the corresponding fluxes in each direction, l and m are spatial tensor indices, and the comma represents a partial derivative (i.e. $q_{,t} := \frac{\partial q}{\partial t}$). In order to numerically evolve this system, we need to discretize it; we could imagine doing this by simply approximating the derivatives using finite-difference approximations. If we use forward-differencing in time, centered differencing in space, and only consider variation in one spatial dimension, we would get the following:

$$q_i^{n+1} = q_i^n + \frac{\Delta t}{2\Delta x} (f(q_{i+1}) - f(q_{i-1})) \quad (2.2)$$

where the spatial discretization is indicated by the subscript i ,¹ and temporal discretization is indicated by the superscript n . The discretization is most naturally thought of as a sampling of the continuum solution at specific points in space and time.

In some situations, this discretization would be suitable. However, conservation laws generally arise most naturally from physical laws in their integral form.

¹Note that i , j , and k are used to indicate spatial discretization in the x , y , and z directions, respectively, while l and m are spatial tensor indices.

When we translate to the differential form, we assume that the relevant derivatives exist; in other words, we assume that the conserved quantities do not exhibit any discontinuities. The problem is that this is not a requirement in the integral form of the conservation law and is also not a requirement in nature. In fact, in many cases, the physically correct behavior for non-linear conservation law systems is for discontinuities (called *shocks* or *weak solutions*, referring to the fact that they are solutions of the integral but not differential form of the governing equations) to arise from smooth initial data.

Because finite differencing methods are derived from discretizing the differential form of the equations, they will inherently break down at discontinuities where the derivatives do not exist. Because of this, we would like to use methods that are more suited to discontinuous behavior.

2.1.2 Finite volume methods

We know that discontinuous solutions *are* allowed by the integral form of conservation laws, so that is a natural place to start. If we integrate the above conservation law over a particular volume, \mathcal{V} , it becomes

$$\int_{\mathcal{V}} q_{,t} dV + \int_{\mathcal{V}} f(q)_{,m}^l dV = 0, \quad (2.3)$$

which can be rewritten as

$$\left(\int_{\mathcal{V}} q dV \right)_{,t} + \oint_{\mathcal{S}} f(q)^l n_l dS = 0, \quad (2.4)$$

by reordering operations in the first term and applying the divergence theorem in the second. \mathcal{S} is the surface of the volume \mathcal{V} , and has surface element dS and unit normal n_l . This is the form of the equation that is usually easiest to intuitively relate to the underlying physical law: the change in the amount of a quantity in a given volume over time is given by the amount of that quantity flowing through the surfaces of that volume.

Now, suppose that instead of sampling the continuum solution at certain points, we split the space up into a finite number of cells, and we think of the discrete solution as the *cell average* of the continuum solution within each cell. We use the compound index $I = (i, j, k)$, where integer indices, i , refer to cell centers, and half-integer indices (i.e. $i + \frac{1}{2}$) to refer to cell edges. To keep the distinction between sampling and cell averages clear, the cell average in cell I will

be called \bar{q}_I . Then the cell average at any time can be expressed as follows:

$$\bar{q}_I = \frac{1}{v_I} \int_{v_I} q dV, \quad (2.5)$$

where v_I is the volume of cell I , and could be expressed as

$$v_I = (x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})(z_{k+\frac{1}{2}} - z_{k-\frac{1}{2}}) = \Delta x \Delta y \Delta z. \quad (2.6)$$

From this, we can write the following exact equation for the time derivative of the cell average in a particular cell:

$$\frac{d\bar{q}_I}{dt} + \frac{1}{v_I} \oint_{S_I} f(q)^j n_j dS = 0, \quad (2.7)$$

where S_I is the surface of the cell. If we consider this in one dimension, we get

$$\frac{d\bar{q}_i}{dt} + \frac{1}{\Delta x} \left(f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}} \right) = 0, \quad (2.8)$$

where $f_{i\pm\frac{1}{2}} = f(q(x_{i\pm\frac{1}{2}}, t))$, which is exact and already discretized in space. To evolve this numerically, we need to choose some discretization in time, and also to choose some appropriate approximation to $f_{i\pm\frac{1}{2}}$ at the cell edges.

It is easy to see that, for certain choices of approximations to $f_{i\pm\frac{1}{2}}$, our finite volume method would be identical to a finite differencing method. However, if we think carefully about how to approximate the flux, or more generally the whole flux term, then we can get more accurate treatment of discontinuous behavior than we get using finite differencing methods.

2.2 Method of lines

We consider our discrete solution, $Q_i(t)$, to be an approximation to the cell averages $Q_i(t) \approx \bar{q}_i(t)$. If we choose $F_{i\pm\frac{1}{2}}$ to be some approximation of $f(q(x_{i\pm\frac{1}{2}}, t))$, which will depend on the values of $Q_i(t)$ over several cells, then, in 1D, we can write

$$\frac{dQ_i}{dt} = -\frac{1}{\Delta x} \left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} \right), \quad (2.9)$$

which is a discrete system of ordinary differential equations for $Q_i(t)$. In this way, we have separated the spatial and temporal parts of our original system of partial differential equations; this process is called the *method of lines*.

One major advantage to the method of lines is that the accuracy in time and space can be addressed independently. Spatial accuracy is increased by improving

the numerical approximation of the fluxes at the cell boundaries, while temporal accuracy can be improved by using higher-order time-stepping methods.

2.3 Time evolution

One way to solve an ordinary differential equation (as we must do in order to evolve our system forward in time) is to use the Runge-Kutta method. Runge-Kutta methods use a trial step at the midpoint of the integration interval to cancel out lower-order error terms. Typically, the results shown in this work are produced using a third-order Runge-Kutta integrator, although first⁻² through fourth-order Runge-Kutta schemes are implemented in both the `ElasticEvolution` and `MultiModel` codes.

2.4 REA algorithm

REA stands for *reconstruct, evolve, average*; these are the three main steps to the algorithm. At the beginning of each time step, the information provided is the discrete solution at the previous time step; we assume that each point in the discrete solution represents the cell average of the continuous solution in that region.

The first step is to *reconstruct* a continuous solution from the discrete solution. The reconstruction should be consistent with the assumption that the discrete solution from the previous time step was a cell average of some continuous solution: the reconstruction that we choose must average to the discrete solution. One simple reconstruction uses a piecewise constant function, where the reconstructed solution at any point within a cell is simply the value of the discrete solution at the central point of the cell. It is trivial to show that taking cell averages of this reconstruction will give the original discrete solution. To get better accuracy, we use better reconstruction methods; one example is a piecewise linear reconstruction, where a neighboring point is used to define a slope. The reconstruction then consists of line segments rather than constant states, and, in smooth regions, the result is a more accurate approximation of the solution in the continuum.

The second step is to *evolve* the reconstruction. This could be done via a number of different methods; for nonlinear conservation laws, we will typically solve a Riemann problem at each boundary between two cells, either exactly or

²We note that first-order Runge-Kutta is equivalent to forward-in-time differencing.

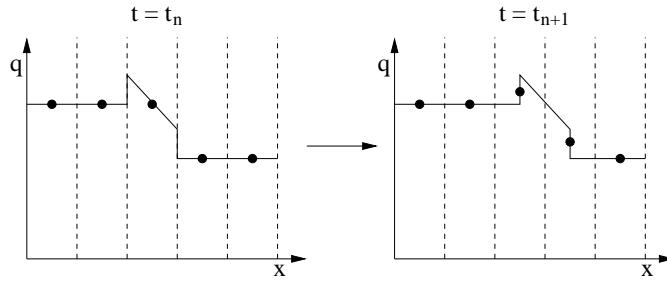


Figure 2.1: Oscillations induced by a simple one-sided slope function in scalar advection of Riemann initial data. The discrete solution, Q_i^n , is shown as dots in the diagram on the left; its reconstruction, \tilde{q} , is represented on that figure as line segments. The reconstruction is evolved to the later time, $t = t_{n+1}$, and a new discrete solution, Q_i^{n+1} , is constructed from the cell averages, shown as dots in the right-hand side of the figure. This type of oscillation can be avoided by using a slope-limiter method. *This figure is based on a similar figure from Chapter 6 in [28].*

approximately. Evolution gives us a solution at the next time step, which is known everywhere.

Once we have a new continuous solution, we must *average* over each cell and construct a discrete solution from these cell averages. We can then repeat the process for the next time step.

2.5 Reconstruction methods

The simplest reconstruction method that will give us higher-than-first-order accuracy is a piecewise-linear reconstruction. This is relatively straightforward; we create a reconstruction, $\tilde{q}(x, t_n)$, from the discrete solution, Q_i^n , as follows:

$$\tilde{q}(x, t_n) = Q_i^n + \sigma_i^n (x - x_i) \quad \text{for } x_{i-\frac{1}{2}} \leq x < x_{i+\frac{1}{2}}, \quad (2.10)$$

where σ_i^n is the slope of the line segment, x_i is the cell center, and $x_{i\pm\frac{1}{2}}$ are the left and right edges of the cell. Now all that remains is to decide how to choose the slope, σ_i^n . Note that choosing the slope, $\sigma_i^n = 0$, will give us a piecewise-constant reconstruction, and therefore, a first-order method.

One simple idea for determining the slopes is simply to take the differences between adjacent cell centers. Three possible slopes that are determined in this way are as follows:

$$\text{Centered Slope} : \sigma_i^n = \frac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x} \quad (2.11)$$

$$\text{Upwind Slope} : \sigma_i^n = \frac{Q_i^n - Q_{i-1}^n}{\Delta x} \quad (2.12)$$

$$\text{Downwind Slope} : \sigma_i^n = \frac{Q_{i+1}^n - Q_i^n}{\Delta x} \quad (2.13)$$

$$(2.14)$$

These slopes work well in some cases, but they assume that the continuous solutions are smooth. When the solutions become discontinuous, the slopes listed above can introduce unphysical Gibbs oscillations. Fig. 2.1 shows how these oscillations can come about [28].

To avoid these oscillations, several methods for choosing slopes have been developed specifically for discontinuous solutions. In general, these are called *slope-limiter* methods, because they limit the slope in some way that eliminates the introduction of oscillations. The first is called the *minmod* slope-limiter. It is as follows:

$$\sigma_i^n = \text{minmod} \left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x}, \frac{Q_{i+1}^n - Q_i^n}{\Delta x} \right), \quad (2.15)$$

where

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases}. \quad (2.16)$$

Basically, this chooses the smaller of the two slopes as long as they have the same sign; if the two slopes have different signs, then the slope is set to zero. Using this choice of slopes prevents the type of oscillations shown in Fig. 2.1. At the central point, a slope of 0 would be used for the reconstruction rather than the negative slope pictured; this would prevent the central cell average from increasing unphysically during the evolution stage.

Another choice is the *superbee* limiter, proposed by Roe [32],

$$\sigma_i^n = \text{maxmod} \left(\sigma_i^{(1)}, \sigma_i^{(2)} \right), \quad (2.17)$$

where $\sigma_i^{(1)}$ and $\sigma_i^{(2)}$ are as follows:

$$\sigma_i^{(1)} = \text{minmod} \left(2 \left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x} \right), \frac{Q_{i+1}^n - Q_i^n}{\Delta x} \right), \quad (2.18)$$

and

$$\sigma_i^{(2)} = \text{minmod} \left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x}, 2 \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x} \right) \right). \quad (2.19)$$

In this choice of slope-limiter, each one-sided slope is compared against twice the value of the other one-sided slope; maxmod chooses the larger of these two options. This limiter method is particularly good at showing sharp discontinuities; the downside is that it can sometimes also sharpen smooth transitions [28].

A third type of slope-limiter is the *monotonized central-difference* limiter, proposed by van Leer [25], also known as the *MC* limiter. It is chosen as follows:

$$\sigma_i^n = \text{minmod} \left(\frac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x}, 2 \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x} \right), 2 \left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x} \right) \right). \quad (2.20)$$

The MC limiter is very versatile, producing sensible solutions for a wide variety of problems [28]. Although the minmod limiter is also implemented in our code, we most often use the MC limiter, because of its versatility.

It is also possible to produce higher-order methods by choosing higher-order reconstruction methods. For example, instead of using a piecewise-constant reconstruction, one might instead use a piecewise-polynomial reconstruction.

2.6 ENO/WENO reconstruction

To get higher-order accuracy, we will need to use higher-order reconstruction methods, so that the reconstructed continuous solution more closely approximates the physical solution. This allows us to get high-order convergence, but only as long as the physical solution is sufficiently differentiable; if the physical solution is discontinuous, we get the type of oscillations seen Fig. 2.1. To solve this problem, essentially non-oscillatory (ENO) schemes attempt to achieve high-order accuracy in smooth regions while still handling discontinuities by generating piecewise polynomial reconstructions that keep the total variation bounded [33]. This is done by choosing the smoothest possible stencil for the order polynomial desired.

Weighted ENO (WENO) schemes are similar to ENO schemes, but instead of using the smoothest possible stencil, a convex combination of all of the different possible stencils is used [27]. WENO schemes allow the user to get higher order accuracy for the same set of stencils. Additionally, very small changes in the solution near zeros can cause ENO schemes to swap from one stencil to another,

whereas WENO schemes smoothly transition between stencils. A useful review of ENO and WENO schemes is given in [34].

2.7 Godunov-type methods

Godunov's method is a specific method using the REA algorithm. It uses a piecewise-constant reconstruction method. However, the main idea behind Godunov's method is that the boundaries between cells are small-scale Riemann problems, consisting of piecewise-constant initial data with a single discontinuity. As long as the increment, Δt , between time steps is short enough that the waves do not begin to interact, we can simply evolve the system by solving these Riemann problems. A similar evolution approach can be used with other reconstruction methods; these will technically *not* be Godunov's method, but are typically referred to as *Godunov-type methods*.

There are two ways to adjust the cell averages during evolution: one way is to calculate the flux at the cell boundary, and to use this in the flux-differencing equation, and the other is to use the wave decomposition of the solution to the Riemann problem to determine the fractional change in the cell average over a time step. Because these two methods are equivalent for linear systems, the wave method will first be demonstrated via a linear system, and then this will be related to the flux-differencing approach for that linear system. Then I will discuss how this is generalized to non-linear systems.

To evolve a constant-coefficient linear system, we solve the Riemann problem at each cell interface; from this we are able to decompose the solution into waves moving away from the interface at different speeds. Godunov's method seeks to use these waves to calculate new cell averages. A linear system in one dimension can be written as

$$q_t + Aq_x = 0, \quad (2.21)$$

because the flux is simply $f(q) = Aq$, where A is a matrix of constant coefficients. The wave speeds will be the eigenvalues of A , denoted λ^p , and the eigenvectors, r^p , will be proportional to the wave-jumps, so that we can decompose the solution into a set of waves:

$$Q_i - Q_{i-1} = \sum_{p=1}^m \alpha_{i-\frac{1}{2}}^p r^p \equiv \sum_{p=1}^m \mathcal{W}_{i-\frac{1}{2}}^p. \quad (2.22)$$

We will use the example of a linear system of m equations [28], assuming that $\lambda^1 \dots < 0 < \dots \lambda^m$.³ The Riemann problem for a system of m equations will generally produce m discontinuities at $x_{i-\frac{1}{2}} + \lambda^1 \Delta t, \dots, x_{i-\frac{1}{2}} + \lambda^m \Delta t$, where waves with negative speeds propagate into the cell centered at x_{i-1} , and waves with positive speeds propagate into the cell centered at x_i . Each wave contributes a jump in the value of $\tilde{q}^n(x, t_{n+1})$, the reconstructed solution evolved forward in time. However, because the waves move at different speeds, they only contribute this jump over a fraction of the grid, i.e. the wave, $\mathcal{W}_{i-\frac{1}{2}}^2$, has traveled a distance $\lambda^2 \Delta t$ in the time Δt , and therefore, has modified the value of \tilde{q} over a fraction of the grid, $\frac{\lambda^2 \Delta t}{\Delta x}$. This changes the value of the grid average by $-\frac{\lambda^2 \Delta t}{\Delta x} \mathcal{W}_{i-\frac{1}{2}}^2$. From this we can infer the general form of the update of the cell averages for Godunov's method:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left[\sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-\frac{1}{2}}^p + \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i+\frac{1}{2}}^p \right], \quad (2.23)$$

where the positive and negative characteristic speeds are calculated as follows:

$$\lambda^+ = \max(\lambda, 0) \quad \text{and} \quad \lambda^- = \min(\lambda, 0). \quad (2.24)$$

Basically, we have split the system up into left-going and right-going waves, and determined which should affect each cell average; we get the right-going waves from the Riemann problem at the left edge of the cell and the left-going waves from the Riemann problem at the right edge of the cell.

This can also be written in a short-hand form, as follows:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(A^+ \Delta Q_{i-\frac{1}{2}} + A^- \Delta Q_{i+\frac{1}{2}} \right), \quad (2.25)$$

where

$$A^\pm \Delta Q_{i \pm \frac{1}{2}} = \sum_{p=1}^m (\lambda^p)^\pm \mathcal{W}_{i \pm \frac{1}{2}}^p. \quad (2.26)$$

In the linear case, A^\pm can be interpreted as the parts of the matrix of constant coefficients corresponding to the positive and negative characteristic speeds; however, in a nonlinear system, we will not have such a simple correspondence between the Jacobian matrix and the wave-structure of the solution.

In the constant-coefficient linear case, we can find a relation between the numerical flux function, $F_{i-\frac{1}{2}}^n$, and this wave-propagation form of Godunov's method. We start by recognizing that the value of the evolved reconstructed solution, \tilde{q} ,

³Generally, we could have a situation where all wave speeds are positive (or negative), but we choose the above as an illustrative example.

along $x = x_{i-\frac{1}{2}}$, which we refer to as $Q_{i-\frac{1}{2}}^\downarrow$, is

$$Q_{i-\frac{1}{2}}^\downarrow = Q_{i-1} + \sum_{p:\lambda^p < 0} \mathcal{W}_{i-\frac{1}{2}}^p = Q_i - \sum_{p:\lambda^p > 0} \mathcal{W}_{i-\frac{1}{2}}^p, \quad (2.27)$$

where we step through the negative waves from the left for the first equality, and the positive waves from the right for the second equality to get to the central value at $x = x_{i-\frac{1}{2}}$. Since the flux at $Q_{i-\frac{1}{2}}$ is given by

$$f(Q_{i-\frac{1}{2}}^\downarrow) = A Q_{i-\frac{1}{2}}^\downarrow, \quad (2.28)$$

where $f(Q_{i-\frac{1}{2}}^\downarrow)$ the value of the flux function when $q = Q_{i-\frac{1}{2}}^\downarrow$, we can see that the numerical flux will be

$$F_{i-\frac{1}{2}}^n = A Q_{i-1} + \sum_{p:\lambda^p < 0} A \mathcal{W}_{i-\frac{1}{2}}^p. \quad (2.29)$$

And since $\mathcal{W}_{i-\frac{1}{2}}^p$ is an eigenvector of A with eigenvalue λ^p , we can write

$$F_{i-\frac{1}{2}}^n = f(Q_{i-1}) + \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i-\frac{1}{2}}^p \equiv f(Q_{i-1}) + A^- \Delta Q_{i-\frac{1}{2}}, \quad (2.30)$$

or similarly,

$$F_{i-\frac{1}{2}}^n = f(Q_i) - \sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-\frac{1}{2}}^p \equiv f(Q_i) - A^+ \Delta Q_{i-\frac{1}{2}}. \quad (2.31)$$

If you plug these into the flux-differencing formula,

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(F_{i+\frac{1}{2}}^n - F_{i-\frac{1}{2}}^n \right), \quad (2.32)$$

then you recover the wave-propagation form of Godunov's method, Eq. 2.25.

For nonlinear problems, we generalize this procedure by solving nonlinear Riemann problems at each cell interface (either exactly or approximately); again we find the waves and characteristic speeds or the numerical flux function, using either an exact or an approximate Riemann solver, and use these to evolve the system. In the non-linear system, we can use the Jacobian of the flux vector in the place of A , and we can decompose the problem into waves and characteristic speeds to do the update—this is called *flux-vector splitting*, and in some sense, we are finding the fluxes *first* and then reconstructing these to the cell edges. The other option is to first reconstruct the conserved or characteristic variables to the cell edges, and

then to calculate numerical fluxes from these by solving an exact or approximate Riemann problem. As shown above, these two methods are equivalent for a linear system with piecewise constant reconstruction, but this will not generally be the case.

2.8 HLL Riemann solver

While effective HRSC methods for systems of nonlinear conservation laws often require the solution of a Riemann problem over the boundaries of cells, it is often not practical to solve the problem exactly, because of the computation time involved. Often, approximate Riemann solvers are used instead; with a well-designed approximate solver, the code is still able to accurately capture the discontinuous behavior of the conservation law, but the vast computation time needed for exact solutions is drastically decreased.

One simple approximate Riemann solver, originally proposed by Harten, Lax, and van Leer [26], is known as the HLL solver. Basically, the HLL solver assumes that the solution to the Riemann problem consists of two discontinuities separated by a constant state. The two waves travel at speeds λ_R and λ_L , where these speeds represent the maximum and minimum characteristic speeds, respectively (waves do not travel at speeds outside of this range). These can either be chosen to be physical limitations of the system (light speed in a relativistic problem), or they can be calculated by solving the eigenproblem. The solution is as follows:

$$Q(x, t) = \begin{cases} Q_L & \text{if } x/t \leq \lambda_L \\ Q^* & \text{if } \lambda_L \leq x/t \leq \lambda_R \\ Q_R & \text{if } x/t \geq \lambda_R \end{cases} \quad (2.33)$$

where Q^* is the value of the intermediate state. The value of Q^* can be calculated as follows. We know, from the Rankine-Hugoniot conditions, that the numerical flux for the central state, F^{HLL} , should have the following relationship with the left and right states as well as their flux functions:

$$f(Q_L) - F^{\text{HLL}} = \lambda_L (Q_L - Q^*) \quad (2.34)$$

and

$$F^{\text{HLL}} - f(Q_R) = \lambda_R (Q^* - Q_R). \quad (2.35)$$

We can solve these equations for the intermediate state Q^* , as well as the numerical flux, F^{HLL} . Note that we refer to this flux as F^{HLL} , *not* $f(Q^*)$; because we are

requiring conservation, it will *not* be equivalent to applying the flux function to the intermediate state. Using the above equations, we find

$$Q^* = \frac{f(Q_L) - f(Q_R) - \lambda_L Q_L + \lambda_R Q_R}{\lambda_R - \lambda_L} \quad (2.36)$$

and

$$F^{\text{HLL}} = \frac{\lambda_R f(Q_L) - \lambda_L f(Q_R) + \lambda_L \lambda_R (Q_R - Q_L)}{\lambda_R - \lambda_L}. \quad (2.37)$$

For simplicity, we may want to use only one maximum characteristic speed magnitude, λ . For example, we may not want to calculate the maximum characteristic speed, but instead, set it to some maximum physical value ($\lambda = 1.0$ in relativistic systems). If this is the case, then we can assume that the maximum and minimum characteristic speeds are $\lambda_R = \lambda$ and $\lambda_L = -\lambda$. For this case, the HLL flux becomes

$$F^{\text{HLL}} = \frac{f(Q_L) + f(Q_R) - \lambda(Q_R - Q_L)}{2}. \quad (2.38)$$

This is the form of the HLL flux that we use in the code. Note, however, that this equation appears to conflict with the HLL-flux equation mentioned in [1]; this is because, in that equation, \mathbf{q}_{i-1}^R is the right edge of the cell to the *left* of the interface, meaning that it corresponds to Q_L in the above equation.

2.9 Hamilton-Jacobi equations

Hamilton-Jacobi equations are equations of the form

$$\phi_{,t} + H(\phi_{,j}) = 0, \quad (2.39)$$

where H can be a function of both space and time. Note that, if the quantity ϕ is advected with velocity, v^j , across the grid, then we have

$$\phi_{,t} + v^j \phi_{,j} = 0. \quad (2.40)$$

This is just a Hamilton-Jacobi equation with $H(\phi_{,j}) = v^j \phi_{,j}$. Because of this, we can use methods developed to evolve Hamilton-Jacobi equations to advect quantities in our code. One limitation of this is that the derivatives, $\phi_{,j}$, must exist, meaning that ϕ should be smooth in order for our treatment to be accurate.

To discretize these types of equations in space, we must find a numerical approximation of $H(\phi_{,j})$:

$$\hat{H} = \hat{H}(\phi_{,x}^-, \phi_{,x}^+; \phi_{,y}^-, \phi_{,y}^+; \phi_{,z}^-, \phi_{,z}^+), \quad (2.41)$$

where $\phi_{,x}^\pm$, for example, are approximations to the spatial derivatives to the left and right edges of the cell. The simplest approximations are forward and backward differences:

$$\phi_{,x}^+ = \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (2.42)$$

and

$$\phi_{,x}^- = \frac{\phi_i - \phi_{i-1}}{\Delta x}. \quad (2.43)$$

However, higher order estimates of the derivatives can be made using ENO or WENO schemes. This is similar to the reconstruction methods described in Section 2.6. When the ENO/WENO scheme is used to reconstruct variables, a polynomial expression for the variable is constructed from surrounding values, and then the value of that expression at the cell edges is computed. Which surrounding values are used depends on which scheme is used (ENO or WENO) and how smooth the numerical solution is in that region. However, in order to numerically estimate \hat{H} , we need estimates of the *derivatives* of ϕ , not the value of ϕ itself. To get these, we can calculate the same polynomial expression as we would in the usual ENO/WENO reconstruction scheme, and simply take its derivative analytically. From this expression, we can estimate the value of the derivative at the cell edges.

2.10 Lax-Friedrichs scheme

One method for calculating the approximation \hat{H} is called the *Lax-Friedrichs* scheme. It is as follows (in 2D):

$$\hat{H} = H \left(\frac{\phi_{,x}^- + \phi_{,x}^+}{2}, \frac{\phi_{,y}^- + \phi_{,y}^+}{2} \right) - \alpha^x \left(\frac{\phi_{,x}^+ - \phi_{,x}^-}{2} \right) - \alpha^y \left(\frac{\phi_{,y}^+ - \phi_{,y}^-}{2} \right), \quad (2.44)$$

where α^x and α^y are dissipation coefficients controlling the amount of numerical viscosity. The dissipation coefficients are chosen based on partial derivatives of H :

$$\alpha^x = \max \left| \frac{\partial H(\phi_{,x}, \phi_{,y})}{\partial(\phi_{,x})} \right| \quad (2.45)$$

and

$$\alpha^y = \max \left| \frac{\partial H(\phi_{,x}, \phi_{,y})}{\partial(\phi_{,y})} \right| \quad (2.46)$$

For advection equations we simply get $\alpha^x = \max |v_x|$ and $\alpha^y = \max |v_y|$.

The stencils over which we find the maximum for the values of α^x and α^y determine the amount of dissipation due to numerical viscosity. In the original

Lax-Friedrichs scheme, the maximum is taken over the entire grid (or rather, we find the maximum and minimum values of $\phi_{,x}$ and $\phi_{,y}$ over the grid, and then find the maximum value of the partial derivatives of H over the resulting intervals, $I^x = [\phi_{,x}^{\min}, \phi_{,x}^{\max}]$ and $I^y = [\phi_{,y}^{\min}, \phi_{,y}^{\max}]$). This scheme can be improved by limiting the stencil over which the dissipation coefficients are determined, but we find the original Lax-Friedrichs scheme to be sufficient for our purposes.

2.11 Fast-marching method

To do efficient extrapolation (both linear and zeroth order) in the `MultiModel` code, we use the *fast-marching method*. The method was originally developed to solve boundary value problems of the Eikonal equation, where a closed curve travels outward in a direction normal to the curve; a key feature is that information propagates in one direction (away from the curve). We can think of extrapolation as evolving some Eikonal equation in fictitious time: values on the curve are propagated outward to the remainder of the grid.

We will not go into detail about the theory behind the algorithm here; there is a good description in [35]. We will just present a brief overview of the main points of implementation of the algorithm.

To start, we choose the values of the variable in the cells from which we would like to extrapolate. Typically, in this work, the fast-marching method is used for extrapolation away from the interface between two materials—material interfaces will be discussed further in Chapter 4—so for us, this means choosing the values of the variable in a thin band immediately adjacent to the interface. These cells are considered part of the *accepted* region, as they already have their value for the variable in question.

Next, cells adjacent to this initial band are reserved as possible candidates for further extrapolation; we will call these cells *candidates*. The distance of each of these from the interface is calculated and stored.

After this initial set up, the iterative part of the algorithm begins. It proceeds as follows:

1. The *candidate* with the minimum distance to the interface is removed from the *candidate* pool. Call this cell A.
2. The variable is extrapolated to cell A from its *accepted* neighbors using whichever method we choose.
3. Cell A becomes *accepted*.

4. All of cell A's *non-accepted* neighbors are added to the *candidate* pool, and their distances from the interface are calculated and stored.

This proceeds until the variable has been extrapolated to the entirety of the grid.

Chapter 3

Elasticity

3.1 Elasticity Background

Although the Newtonian theory of elasticity is well established¹, the problem of describing the behavior of elastic media in relativity was ignored for many years. The first real need for a description of elastic solids in relativity comes about with attempts to detect gravitational waves. Weber 1960 [38] proposes the use of solid bars, which would resonate when they encounter gravitational waves of specific frequencies; the idea is that the induced strain would cause a measurable voltage via the piezoelectric effect. For this, he needs a theory to describe how the gravitational waves interact with the solid bars. For his purpose, it is sufficient to consider the linearized Einstein equations in a Newtonian background. Dyson 1969 [39] uses a similar theory to examine the interaction between gravitational waves and the Earth, to determine whether it is possible to detect these waves via seismic oscillations.²

Around the same time, there are several attempts to understand elasticity in full general relativity, though none are completely successful. Synge 1959 [40] proposes an unsound theory, and introduces the idea that only *time rates of strain*, but not *absolute strains* can be examined in relativity. The (faulty) logic behind this is that the idea of a “natural” state of the body against which strain should be measured is not consistent with relativity; it does not make sense to think of “turning off” gravity for the entire body to examine the relaxed state. This idea seems to have thwarted several subsequent attempts to develop a relativistic elasticity theory, namely the work of Bennoun 1965 [41]. However, Hernandez 1970

¹See Landau and Lifshitz [36] and Marsden and Hughes [37] for a good overview of both linear and non-linear Newtonian elasticity.

²He finds that the seismic oscillations would be 10^5 times smaller than the prevailing noise levels for 1 Hz gravitational waves.

[42] points out that strain is a *microscopic* property, so we can simply consider the configuration with respect to the relaxed or “natural” state *locally*, by taking a small region to a large distance where it is free from stresses. It is true that there is no “natural” state for the *body* as a whole, but there *is* a natural state for the *material*. In order for relativity to influence our local understanding of strain, it would need to be dominant on a *microscopic* scale; however, while it dominates on a *macroscopic* scale in extreme astrophysical environments, it still remains negligible when compared to electromagnetic or strong forces between particles, and so does not affect local crystal (or fluid) properties.

Another partially successful attempt is by Rayner 1963 [43], who attempts to produce a *Hookean* (i.e. with a linear stress-strain relationship) theory of elasticity in relativity. This theory assumes the validity of the concept of absolute strain, but ultimately, his guess for the form of the equations of motion was not consistent with perfect elasticity. In the small-strain limit, the theory does not pose any problems, but it does not satisfy energy conservation for finite strains.

A major step towards a theory of relativistic elasticity is Oldroyd 1970 [44], where a mathematical formalism describing general continuous media in general relativity is presented. Convected coordinates (basically particle labels) are introduced. The important concept of *rheological invariance* is also introduced; namely that the material state of any particle in the medium should only depend on quantities at that particle and along its past world line. This means that the equation of state must be written in terms of only integrals, derivatives, and functions of quantities that are space-tensor functions (tensors on the convected material space) of only particles (i.e. coordinates in the material space) and their proper time.

Carter and Quintana 1972 [45] rewrite this mathematical formalism in clearer mathematical terms, emphasizing the existence of two manifolds (the 3D material manifold that keeps track of particles—in Oldroyd, these are convected coordinates—and the background spacetime manifold) and a map between them, which allows us to identify spacetime points with specific particles. They then apply this mathematical approach to relativistic elasticity. This is analogous to the approach taken in the Newtonian literature for non-linear elasticity [37].

As noted in Carter 1980 [46], this formulation of elasticity was developed earlier by Souriau 1965 [47]; however, it went largely unnoticed. The approach here is slightly more elegant than the Carter and Quintana approach: Carter and Quintana assume energy conservation in the local rest frame in order to derive equations with the appropriate number of degrees of freedom. As a consequence, their theory

can be written as a simple action principle. In Souriau 1965 [47], the variational principle is taken as an axiom, and, as a result, conservation of energy is derived.

It is also worth mentioning the work of Maugin (1971 [48] and 1978 [49]) and of Christodoulou 1998 [50], where a theory of electromagnetic elastic media is introduced and studied.

Kijowski and Magli (1992 [51] and 1998 [52]) have also reformulated relativistic elasticity to emphasize its gauge character. A nice, clear discussion of how this reformulation relates to the formulation discussed in Carter and Quintana is presented in [53]. They make the analogy between electromagnetism and the elasticity theory: essentially, the Carter and Quintana approach is like deriving the Maxwell equations *directly* from the Hilbert variational principle, without making use of electromagnetic potentials. The Kijowski and Magli approach is similar to the standard approach to electrodynamics, where physical quantities are written as first derivatives of potentials. The matter-space coordinate labels are the necessary gauge potentials in elasticity, and the material equations of motion are written in terms of these potentials. This approach may have conceptual advantages.

Beig and Schmidt 2003 [30] discuss the characteristic decomposition of relativistic elasticity. They present several existence and uniqueness theorems, including showing that the system can be written as a first-order symmetric hyperbolic system.

Karlovini and Samuelsson 2003 [54] gives a good summary of relativistic elasticity, following Carter and Quintana closely. They include new methods and results that help to elucidate and update the theory.

In the following sections, we closely follow three of these works. Firstly, we use the approach of Carter and Quintana 1972 [45] and Karlovini and Samuelsson 2003 [54] to develop the basis for our description of the kinematic and dynamical equations for relativistic elasticity. Section 3.4 follows the approach of Beig and Schmidt 2003 [30] closely to determine the conditions for hyperbolicity of our system, and analyze its characteristics.

3.2 Kinematics

3.2.1 Evolution equations

One major difference between an elastic solid and a fluid is the fact that we need to keep track of the particle positions relative to their preferred configuration. As the solid is deformed from its relaxed state, it behaves differently; we need some way of keeping track of this, and allowing this to affect the evolution of the system.

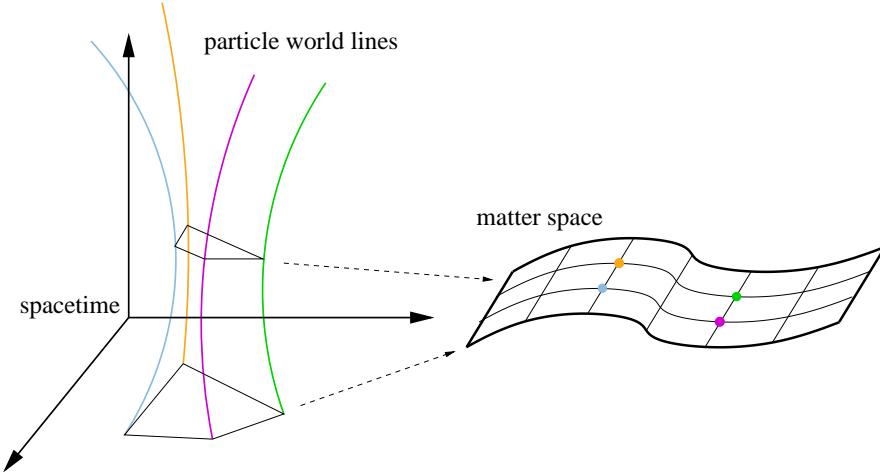


Figure 3.1: In elasticity, we must keep track of the particle positions in relation to their preferred relative positions. To do this, we make use of two manifolds and a map between them. The four-dimensional spacetime is the background with respect to which the matter moves and transforms, while the three-dimensional matter space keeps track of the preferred particle positions with respect to one another. Each point on matter space corresponds to a particle, and therefore, a world-line on spacetime. Information about deformations that occur can be obtained by comparing the matter-space and spacetime metrics, which is done by pushing forward one metric onto the other manifold using the gradient of the map (the configuration gradient).

We do this by following Carter and Quintana [45] and using two manifolds and a map between them. The two manifolds are called the *matter space*, which keeps track of the preferred particle positions relative to one another, and the *spacetime*, which is the background with respect to which the matter is moving and transforming. We call the map between these the *configuration*, because it contains information about how the particles are *configured* on spacetime. This is illustrated in Fig. 3.1. Formally, the configuration is described as follows:

$$\chi : M^4 \rightarrow X^3, \quad (3.1)$$

or in coordinates:

$$\chi^A : x^a \mapsto \xi^A = \chi^A(x^a). \quad (3.2)$$

Assuming that the map is continuous, we then introduce the derivative of this map with respect to spacetime coordinates, called the *configuration gradient*, as follows:

$$\psi : x^a \mapsto \psi^A_a = \frac{\partial \xi^A}{\partial x^a}. \quad (3.3)$$

This derivative, along with the metrics of the two manifolds, contains all the information about the deformations that occur, because they tell us how the preferred particle positions and the actual particle positions compare on spacetime. To do this, we can use the configuration gradient to push forward and pull back between the two manifolds: for example, the spacetime metric pushed forward onto matter space is

$$g^{AB} := \psi^A{}_a \psi^B{}_b g^{ab}. \quad (3.4)$$

We can also introduce a time foliation on spacetime,

$$M^4 \rightarrow X^3 \times \mathbb{R}, \quad (3.5)$$

so that the coordinates will be

$$x^a \mapsto (t, x^i), \quad (3.6)$$

and the configuration gradient becomes

$$\psi^A{}_a \mapsto (\psi^A{}_t, \psi^A{}_i) \quad (3.7)$$

We define the four velocity, u^a , such that it points along particle world lines; since the matter-space coordinates label particles, these particle labels should stay constant along particle world lines, and we can define the four velocity such that

$$u^a \psi^A{}_a := 0. \quad (3.8)$$

If we split the four velocity in the usual way, such that

$$u^a = (u^t, u^i) = \alpha^{-1} W(1, \hat{v}^i), \quad (3.9)$$

where

$$\hat{v}^i := \alpha v^i + \beta^i \quad (3.10)$$

and W is the Lorentz factor based on the three velocity,

$$W := (1 - v^i v_i)^{-1/2}, \quad (3.11)$$

then we can express the time component of the configuration gradient in terms of \hat{v}^i , and the spatial components of the configuration gradient as follows:

$$\psi^A{}_t = -\hat{v}^i \psi^A{}_i. \quad (3.12)$$

(A summary of the 3 + 1 split used in this work is given in Appendix B.1.)

Since the configuration gradient is a group of partial derivatives, we can use the fact that partial derivatives must commute to write

$$C^A_{ab} := \psi^A_{[a,b]} = 0; \quad (3.13)$$

we call this system C^A_{ab} for convenience. Next, we can split this system using the time foliation introduced earlier to get the following:

$$C^A_{ij} = \psi^A_{[i,j]} = 0 \quad (3.14)$$

and

$$E^A_i := C^A_{it} = \psi^A_{i,t} + (\hat{v}^j \psi^A_j)_{,i} = 0. \quad (3.15)$$

Eq. 3.14 is in the form of a constraint, while Eq. 3.15 is an evolution equation for ψ^A_i . In fact, both of these are already in conservation-law form; this is made clearer by writing them as

$$E^A_i = \psi^A_{i,t} + (\hat{v}^j \psi^A_j \delta_i^k)_{,k} = 0, \quad (3.16)$$

and

$$C^A_{ij} = (\psi^A_{[i} \delta^l_{j]})_{,l} = 0. \quad (3.17)$$

However, although the evolution equation is already in conservation-law form, it turns out that in order to make our system of equations strongly hyperbolic, we will need to instead use the following system (compare o Eq. 3.222):

$$2\alpha W^{-1} u^a C^A_{ia} = 2\alpha W^{-1} u^a \psi^A_{[i,a]} = 0, \quad (3.18)$$

which can be written in balance law form as

$$\psi^A_{i,t} + (\hat{v}^j \psi^A_j)_{,i} = -2\hat{v}^j \psi^A_{[i,j]}. \quad (3.19)$$

This will be discussed further in Section 3.4. As you can see, Eq. 3.19 is simply the evolution equation, Eq. 3.15, with an added source term that is proportional to the constraint, Eq. 3.14. As long as the constraint is satisfied, this balance law is equivalent to the conservation law, Eq. 3.15.

Although, formally the balance law form of the evolution equation for the configuration gradient (Eq. 3.19) is needed for hyperbolicity, in practice, the conservation law form (Eq. 3.15) is implemented in the code. As discussed later in Section 3.5.8, the constraints (Eq. 3.14) are trivial if all variables depend on only

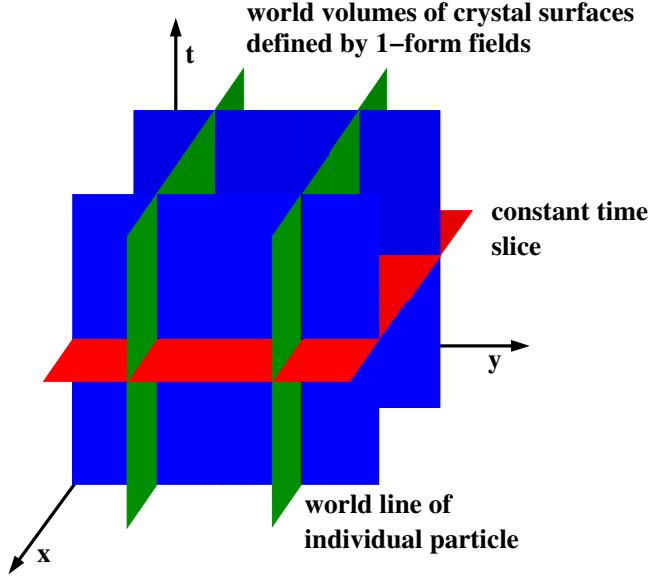


Figure 3.2: Each 1-form component of ψ^A_i represents a set of crystal surfaces over all time (world volumes); the intersection between the set of these and a constant time slice is the instantaneous crystal lattice at that specific time. The intersections between the three sets of crystal world volumes give the world lines of the individual particles.

one dimension. In 2D, where the constraints are non-trivial, the system without the constraint addition is no longer strongly hyperbolic; we expect this constraint addition to be necessary for *stability*, as it is in the Newtonian literature. Although we have seen no stability issues to date despite neglecting this constraint addition, we note that it may be necessary in the future. Despite the importance for stability, there seems to be no agreement as to how important constraint violations are to the *accuracy* of solutions. In MHD, the $\nabla \cdot \mathbf{B} = 0$ constraint is crucial for accuracy; similarly, one might imagine that some methods that reduce these constraint violations—for example, adding a parabolic damping term [55], which is analogous to the Powell method in MHD [56, 57]—could be necessary in the future.

It will also be convenient to write this equation as an advection equation:

$$\psi^A_{i,t} + \hat{v}^j \psi^A_{i,j} = -\psi^A_j \hat{v}^j. \quad (3.20)$$

As an advection equation, this is hyperbolic in ψ^A_i for given \hat{v}^i .

3.2.2 Physical meaning of ψ^A_i evolution equations

The configuration gradient, ψ^A_i , can be thought of as being three 1-forms in spacetime with labels, $A = 1, 2, 3$. Each 1-form field defines a set of surfaces

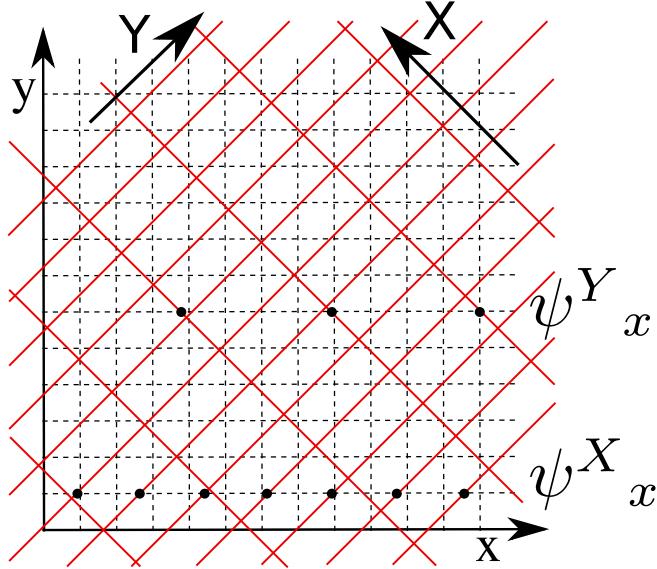


Figure 3.3: Matter-space coordinates superimposed onto the spacetime coordinates using the map, χ^A . To integrate ψ^X_x , we must count up lines of constant X crossed when moving in the x direction; for ψ^Y_x , we count up lines of constant Y .

which physically represent the world volumes of the crystal surfaces, or the planes of the crystal at all times. The intersection between these 1-forms and a time slice gives the instantaneous crystal surfaces. The intersections between the three crystal surfaces give the world lines of individual particles. When the intersection between two of the crystal surfaces and a time slice is found, this gives the crystal axis in the third direction. This is illustrated in 3D (with two spatial dimensions and time) in Fig. 3.2.

Because of this, the evolution equation for ψ^A_i in some way represents the conservation of crystal axes; each 1-form represents a set of crystal surfaces, and these are evolved using the conservation law discussed earlier. The flux, which is $-\psi^A_t$, tells whether the particle label at any given grid point should change, thus making it a natural flux for the conservation of crystal surfaces.

We should examine exactly what we mean by conservation of crystal structure. It is helpful to draw the matter-space coordinates superimposed onto the spacetime coordinates; of course, this can only be done using the map. Fig. 3.3 shows this situation, where the spacetime coordinates are (x, y) , and the matter-space coordinates are (X, Y) . This is equivalent to looking at a constant time slice of Fig. 3.2: the red lines are the intersections between the world volumes of the crystal surfaces with the constant time slice, giving the instantaneous crystal surfaces at that particular time.

For conservation laws, the actual conserved quantity is the variable in the evolution equations integrated over some volume; for example, the equation that physically represents conservation of mass is an evolution equation for the density when we look at it in differential form. To understand the physical meaning of the evolution equations for ψ^A_i , we should first determine what the conserved quantity is by integrating the configuration gradient, ψ^A_i , over some volume. Essentially, what this comes down to is counting up the lines of constant matter-space coordinate that we cross as we move in a particular spacetime direction. This is illustrated in Fig. 3.3. If we integrate ψ^X_x , we count up the number of constant X contours that we cross as we move along the x -direction. Likewise, integrating ψ^Y_x gives us the number of constant Y lines crossed moving in the same direction. So, roughly, we can think of ψ^A_i as a sort of “density” of crystal axes in a particular direction; all of this is really just a physical interpretation of the meaning of a gradient.

Now that we have some physical intuition about the meaning of ψ^A_i , we can use this to interpret the evolution equations and constraints. To do this, it is best to look at the jump conditions.

To examine the jump conditions for ψ^A_i , we consider a shock with a normal covector n_i and velocity s^i . Then $s := s^i n_i$ is the normal shock speed. We find the Rankine-Hugoniot conditions from the evolution equation and constraint for ψ^A_i by first rotating into a frame where one coordinate direction is perpendicular to the shock (so we only have variation in 1D), and then integrating across the shock. They are as follows:

$$[[\psi^A_k \delta^k_{[i} \delta^l_{j]}]] n_l = [[\psi^A_{[i} n_{j]}]] = 0 \quad (3.21)$$

and

$$-s [[\psi^A_i]] + [[\hat{v}^j \psi^A_j \delta^k_i]] n_k = 0, \quad (3.22)$$

where the double square brackets indicate the difference in the quantity between the left and right states.

We start by looking at the constraint jump conditions; it is easiest to consider what these *do not* allow. Consider the situation shown in Figs. 3.4a and 3.4b, where the direction normal to the shock is the horizontal, or x , direction. From the constraint jump conditions, we see that, since $n_y = 0$, and n_x is constant across the shock, we must have $[[\psi^A_y]] = 0$. In other words, if we move in the y direction, the number of crystal axes that we cross (either X or Y) should be the same for the left and right states. Figs. 3.4a and 3.4b show two scenarios that are disallowed by this constraint. In Fig. 3.4a, the number of Y lines crossed

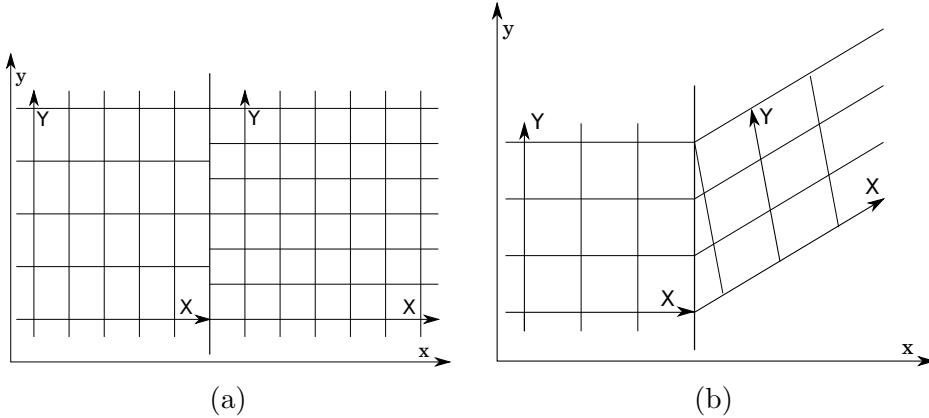


Figure 3.4: The first jump condition, Eq. 3.27, prevents the above examples of ‘surgery’ of the material. If we choose space and matter space coordinates such that the shock is along the y -axis and $\psi^A_i = \delta^A_i$ on the left-hand side, with the sketched coordinates being the matter-space coordinates, and the space coordinates being evenly spaced and square, then 3.4a is prevented by $[[\psi^Y_y]] = 0$, and 3.4b is prevented by $[[\psi^X_y]] = 0$. *This figure is a modified version of Figure 1 in [1].*

as you move along y varies between the left and right states; in other words, we have $[[\psi^Y_y]] \neq 0$. In Fig. 3.4b, the number of X lines varies, as in the left state, no constant X lines are crossed, while they are crossed in the right state, which means that we have $[[\psi^X_y]] \neq 0$.

Another way to see this is to decompose the jump conditions into their components normal and parallel to the shock. This will also help to illuminate the jump conditions for the evolution equations for ψ^A_i . To do this decomposition, we introduce the normal vector such that

$$n^i n_i = 1, \quad (3.23)$$

and the projector normal to the normal covector (tangent to the shock surface),

$$\parallel^i{}_j := \delta^i{}_j - n^i n_j, \quad (3.24)$$

which has the expected properties of a projector, i.e. $\parallel^i{}_j n^j = 0$ and $\parallel^i{}_j \parallel^j{}_k = \parallel^i{}_k$.

Next we define

$$\hat{v}^n := \hat{v}^i n_i \quad (3.25)$$

and

$$\hat{v}^{\parallel i} := \parallel^i{}_j \hat{v}^j. \quad (3.26)$$

We split the other tensors into their normal and tangential parts in a similar way.

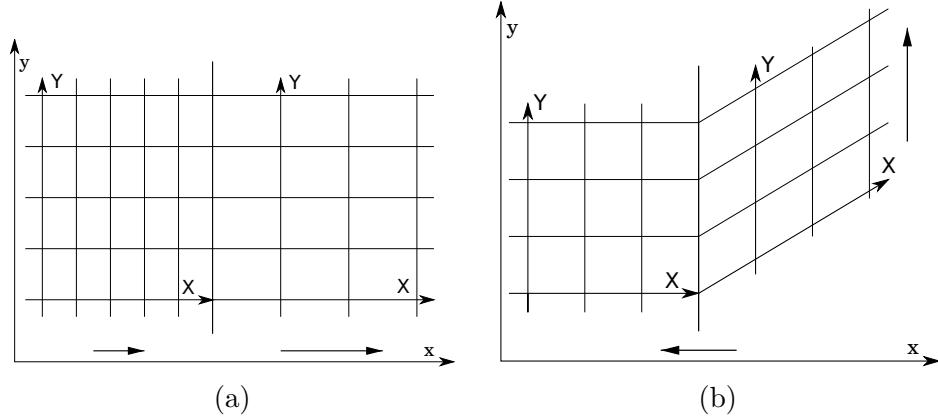


Figure 3.5: These types of discontinuities *are* allowed by the jump conditions. 3.5a shows a pure density shock, where v^y is continuous. It is shown in the rest frame of the shock, so $s = 0$. The density and velocity on the left and right are therefore related by $\llbracket \psi^X_x v^x \rrbracket = 0$; in the higher density region, v^x is smaller than in the lower density region. 3.5b shows a pure traveling kink; for simplicity, we have chosen $v^x = 0$, ψ^X_x is constant, and we also have a discontinuity in v^y . We see that the density is continuous across the discontinuity, but we have a discontinuity in ψ^A_i , and therefore in the crystal axes. *This figure is a modified version of Figure 2 in [1].*

We now see that, for the constraints, we have

$$\llbracket \psi^A_i ||^i_k n_j - \psi^A_j ||^i_k n_i \rrbracket = \llbracket \psi^A_{||k} \rrbracket = 0. \quad (3.27)$$

Now it is easy to see that when we have n_i completely in the x direction, we must have $\llbracket \psi^A_y \rrbracket = 0$.

Next we can look at the jump conditions for the evolution equations. Projecting as above, we get

$$\llbracket \psi^A_n (\hat{v}^n - s) \rrbracket + \psi^A_{||i} \llbracket \hat{v}^{||i} \rrbracket = 0, \quad (3.28)$$

where we have plugged in $\psi^A_i \hat{v}^i = \psi^A_n \hat{v}^n + \psi^A_{||j} \hat{v}^{||j}$ and rearranged.

From this, we can see that there are two basic shocks that are allowed in this system. The first is a density shock; this is illustrated in Fig. 3.5a, in the frame of reference where the shock is stationary. In this case, $s = 0$ and $\hat{v}^{||i} = 0$, so we must have $\llbracket \psi^A_n \hat{v}^n \rrbracket = 0$. This tells us that the number of crystal axes counted up along a certain direction can only change if the velocity in that direction also changes. Intuitively, this is what you would expect for conservation of crystal axes: there must be the same number of crystal axes flowing into the shock as there are flowing out over some period of time. Specifically, this is illustrating conservation of crystal axes, but if we combine the equations for the crystal axes in all directions with advection of the matter-space metric, which keeps track of the particles, this

situation is equivalent to a density shock. The relationship between the particle number density evolution equation and the evolution equations for ψ^A_i and k_{AB} is derived explicitly in Section 3.2.3.

Another possible scenario is a traveling kink, illustrated in Fig. 3.5b. In this situation, the left region is static, and the right is moving upward at some constant velocity, so we have $\hat{v}^n = 0$, and $\hat{v}^{\parallel i}$ and s are non-zero. From the jump conditions, we see that we are only allowed to have a non-zero velocity parallel to the shock surface if we also have some non-zero shock speed. The result is that the crystal axes stay connected, and are dragged upward; you can think of this as being like someone lifting a tablecloth up off of a table. In this situation, the discontinuity between the moving and static segments must move to the left; in the tablecloth analogy, the kink between the lifted and static portions is moving towards the non-lifted area.

In hydrodynamics, we have both a contact discontinuity where entropy jumps, as well as one where tangential velocity can jump. In elasticity, we are left with only an entropy-jump contact discontinuity, because of the kinematic jump conditions discussed above.

One might expect that it would be possible to recover fluid behavior using the formulation described here simply by setting the shear modulus in the equation of state to zero; however, the above jump condition makes this impossible.

In fact, the tangential-velocity-jump contact discontinuity requires a discontinuous map, χ , between spacetime and matter space. Consider a situation where the map is initially continuous and we have an initial discontinuity only in the tangential velocity so that $v^y = -v$ where $x < 0$, and $v^y = v$ where $x \geq 0$. Assume that initially $X = x$ and $Y = y$ everywhere; this defines a continuous map with a configuration gradient, ψ^A_i , equal to the identity matrix. Physically, in a fluid, this tangential-velocity discontinuity should stay stationary in time; if we assume this behavior for the current situation, then after some time Δt , the discontinuity is still located at $x = 0$, but the matter to the left-hand side has moved downward, and the matter to the right-hand side has moved upward. Therefore, we have a new map with new expressions for the matter-space coordinates in terms of the spacetime coordinates: $Y = y + v\Delta t$ where $x < 0$ and $Y = y - v\Delta t$ where $x \geq 0$. Since the Y coordinate is now a discontinuous function of y and x , the map must also be discontinuous.

Because, in the process of deriving the elasticity formulation, we assume that the map is continuous, which is required in elastic materials (where we expect crystal axes to be continuous), but not in fluids, we acquire jump conditions that prevent tangential-velocity contact discontinuities. Consequently, even if the shear

modulus is set to zero, shear waves will still be present when using the elasticity formulation, and we are left with only an entropy-jump contact discontinuity.

3.2.3 Relationship between particle number density and ψ^A_i

It is intuitively clear that an evolution equation for crystal axes should automatically contain an evolution equation for the particle number density of the material. We can see that this is indeed the case by taking the following steps.

First, we introduce a volume form on matter space, n_{ABC} . Then, integration of this volume form over a particular volume represents the number of particles contained within that volume.

In addition to the volume form, we will also need to define a conformal metric in order to define angles. These angles are essential in elasticity, because we will need to compare these to the angles on spacetime to define deformations of the material.

Together, the volume form and the conformal metric define a full Riemannian metric, which we will call k_{AB} , and which is compatible with n_{ABC} such that,

$$n_{ABC} = \sqrt{k_\xi} \delta_{ABC}, \quad (3.29)$$

where k_ξ is the usual matrix determinant:

$$k_\xi := \frac{1}{3!} \delta^{ABC} \delta^{DEF} k_{AD} k_{BE} k_{CF}, \quad (3.30)$$

and δ_{ABC} is the totally antisymmetric tensor with an independent component of 1 in a particular coordinate system on matter space. We label these coordinates ξ^A , and because this determinant is coordinate dependent, we label it with the subscript, ξ , indicating that it is associated with the particular choice of coordinates ξ^A . The unsheared state occurs when the matter-space metric is proportional to the spacetime metric pushed forward onto matter space: specifically, when

$$k_{AB} = n^{2/3} g_{AB}, \quad (3.31)$$

where k_{AB} is the matter-space metric, and n is the particle number density.

We next consider a pull-back of volume form onto spacetime:

$$n_{abc} := \psi^A_a \psi^B_b \psi^C_c n_{ABC}. \quad (3.32)$$

Spacetime also has a volume form compatible with the metric, which we write as

$$\epsilon_{abcd} = \sqrt{g_x} \delta_{abcd} \quad (3.33)$$

where g_x is the determinant of the metric,

$$g_x := -\frac{1}{4!} \delta^{abcd} \delta^{efgh} g_{ae} g_{bf} g_{cg} g_{dh}, \quad (3.34)$$

and δ^{abcd} is associated with a particular coordinate system, and therefore, so is g_x .

Next, we define the particle-number current to be

$$j^a := \frac{1}{3!} \epsilon^{abcd} n_{bcd}. \quad (3.35)$$

This vector is time-like and conserved:

$$\nabla_a j^a = \epsilon^{abcd} \nabla_a n_{bcd} = 0, \quad (3.36)$$

where the second equality comes about because we can write

$$\epsilon^{abcd} \nabla_{[a} n_{bcd]} = \epsilon^{abcd} n_{[bcd,a]} = \epsilon^{ABCD} n_{[BCD,A]}. \quad (3.37)$$

The last equality depends on the fact that $\psi^A{}_{a,b}$ is symmetric in a and b , but is contracted with the totally antisymmetric tensor ϵ^{abcd} . Finally, $n_{[BCD,A]}$ is a 4-form on 3D space, meaning it must be 0.

We then define the split,

$$j^a =: n u^a. \quad (3.38)$$

Then the conservation of the particle-number current becomes

$$(\sqrt{\gamma_x} W n)_{,t} + (\sqrt{\gamma_x} W n \hat{v}^i)_{,i} = 0, \quad (3.39)$$

where γ_x is the determinant of the spatial part of the spacetime metric in the $3+1$ split; its definition is analogous to the definition for g_x . Therefore, we note that $g_x = \alpha^2 \gamma_x$.

We can also write

$$n = -u_a j^a. \quad (3.40)$$

We recall that

$$u_a = W(-\alpha + v_j \beta^j, v_i) \quad (3.41)$$

and

$$\epsilon^{0ijk} = \alpha^{-1} \epsilon^{ijk}. \quad (3.42)$$

If we plug all these in, and also note that $n_{ij0} = -\hat{v}^k n_{ijk}$ (from writing n_{abc} in terms of $\psi^A{}_a$ and n_{ABC}), we can work out that

$$n = \frac{1}{3!} W^{-1} \epsilon^{ijk} n_{ijk}. \quad (3.43)$$

Because we know that $\epsilon^{ijk} = \frac{1}{\sqrt{\gamma_x}} \delta^{ijk}$, and $n_{ABC} = \sqrt{k_\xi} \delta_{ABC}$, we can write n as

$$n = \frac{\sqrt{k_\xi} \psi_{x\xi}}{W \sqrt{\gamma_x}} \quad (3.44)$$

where

$$\psi_{x\xi} := \frac{1}{3!} \delta^{ijk} \delta_{ABC} \psi^A{}_i \psi^B{}_j \psi^C{}_k. \quad (3.45)$$

From here we can show that $\nabla_a j^a = 0$ is a linear combination of the evolution equations for $\psi^A{}_i$ with the hyperbolicity fix. To do this, we first contract the advection version of the evolution equation for $\psi^A{}_i$ with the matrix inverse, $F^i{}_A$, and use the matrix identity ($\delta(\ln \psi_{x\xi}) = F^i{}_A \delta(\psi^A{}_i)$) to get the following:

$$(\ln \psi_{x\xi})_t + \hat{v}^i (\ln \psi_{x\xi})_{,i} + \hat{v}_{,i}^i = 0. \quad (3.46)$$

Now to manipulate $\nabla_a j^a = 0$, we recall that

$$u^a = \alpha^{-1} W(1, \hat{v}^i) \quad \text{and} \quad j^a = n u^a, \quad (3.47)$$

so that

$$j^a = \frac{\sqrt{k_\xi} \psi_{x\xi}}{\sqrt{g_x}} (1, \hat{v}^i). \quad (3.48)$$

Then $\nabla_a j^a = 0$ becomes

$$(\sqrt{k_\xi} \psi_{x\xi})_t + (\sqrt{k_\xi} \psi_{x\xi} \hat{v}^i)_{,i} = 0. \quad (3.49)$$

We can now define the pull-back of the matter-space metric onto spacetime:

$$k_{ab} := \psi^A{}_a \psi^B{}_b k_{AB}. \quad (3.50)$$

From this definition, we have

$$u^a k_{ab} = 0. \quad (3.51)$$

From our definition of j^a , we can see that

$$j^a k_{AB,a} = 0, \quad (3.52)$$

which implies that

$$u^a k_{AB,a} = 0. \quad (3.53)$$

This is what we would expect, since we can view the components of k_{AB} as scalars on spacetime that should be carried along with the particle flow. In fact, the above implies that

$$\mathcal{L}_u k_{ab} = 0, \quad (3.54)$$

meaning that the matter-space metric is Lie-dragged along with the flow. In the 3+1 split, the Eq. 3.53 becomes

$$k_{AB,t} + \hat{v}^i k_{AB,i} = 0 \quad (3.55)$$

and we use this to evolve the matter space metric.

Since the determinant of the matter-space metric, k_ξ , is just a function of scalars on spacetime (the components of k_{AB}), it will be evolved in the same way as those scalars. Thus, the advection equation for k_{AB} implies

$$(k_\xi)_{,t} + \hat{v}^i (k_\xi)_{,i} = 0, \quad (3.56)$$

and means that Eq. 3.49 is equivalent to Eq. 3.46. This can be shown by expanding Eq. 3.49 using the product rule: we then get a term proportional to the advection equation for k_ξ , and one proportional to Eq. 3.46. Therefore, we have shown that the evolution equation for ψ^A_i with the hyperbolicity fix carries all of the necessary information about conservation of particle number. In fact, in the `ElasticEvolution` code, we evolve both the conservation-of-particle-number equation *and* the evolution equation for ψ^A_i ; this gives us two independent sources for the particle number, which we call n_D and n_ψ .

3.3 Dynamics

3.3.1 Derivation of stress-energy tensor

To derive the stress-energy tensor for elasticity, we start with the action:

$$S := \int e(g^{ab}, \psi^A_a, k_{AB}, \dots, s) g_x^{1/2} d^4x, \quad (3.57)$$

where the dots represent any other matter-space tensors. We assume that there is no heat flow between particles, so entropy is constant along particle world lines aside from at shocks, and we can consider it to be a scalar on matter space. We call the Lagrangian density here e , because it turns out later that it is the total energy density when our resulting equations are solutions to the Euler equations.

Next, we vary the metric to get the standard definition of the stress energy tensor:

$$\delta S =: \frac{1}{2} \int T_{ab} \delta g^{ab} g_x^{1/2} d^4x. \quad (3.58)$$

From this definition, the stress energy tensor ends up being the following:

$$T_{ab} = 2 \frac{\partial e}{\partial g^{ab}} - e g_{ab}. \quad (3.59)$$

This is derived as follows:

$$\delta(\sqrt{g_x} e) = \sqrt{g_x} \frac{\partial e}{\partial g^{ab}} \delta g^{ab} + e \delta(\sqrt{g_x}) \quad (3.60)$$

and then

$$\delta(\sqrt{g_x}) = \frac{1}{2} g_x^{-1/2} \delta g_x \quad (3.61)$$

$$= \frac{1}{2} g_x^{1/2} \frac{1}{g_x} \delta g_x \quad (3.62)$$

$$= \frac{1}{2} g_x^{1/2} \delta(\ln g_x) \quad (3.63)$$

$$= \frac{1}{2} g_x^{1/2} g^{ab} \delta g_{ab} \quad (3.64)$$

$$= -\frac{1}{2} g_x^{1/2} g_{ab} \delta g^{ab}. \quad (3.65)$$

We define the projector onto the tangent space normal to the 4-velocity,

$$h_{ab} := u_a u_b + g_{ab}, \quad (3.66)$$

and then, we can then write the stress-energy tensor as

$$T_{ab} = e u_a u_b + p_{ab}, \quad (3.67)$$

where

$$p_{ab} := 2 \frac{\partial e}{\partial g^{ab}} - e h_{ab}. \quad (3.68)$$

Note that p_{ab} is symmetric by definition. Physically, we will discover later that it encodes the stresses: namely the isotropic part is the pressure, and, in elasticity,

we additionally have anisotropic stresses.

Next we define the push-forward of the spacetime metric onto matter space:

$$g^{AB} := \psi^A{}_a \psi^B{}_b g^{ab}. \quad (3.69)$$

Its matrix inverse is defined such that

$$g^{AC} g_{CB} := \delta^A{}_B. \quad (3.70)$$

We now have two Riemannian metrics on matter space: k_{AB} and g_{AB} . g_{AB} transforms like a tensor on matter space,³ but it depends on deformations in the matter, so it is time-dependent. k_{AB} is a genuine tensor on matter space so we call this the *matter-space metric*. However, we use g_{AB} and g^{AB} to move indices on matter space, so $k^{AB} := g^{AC} g^{BD} k_{CD}$ and k^{AB} is *not* the matrix inverse of k_{AB} .

We also define

$$\psi_A{}^a := \psi^B{}_b g^{ab} g_{AB}. \quad (3.71)$$

This is the inverse of $\psi^A{}_a$ in the sense that

$$\psi^A{}_a \psi_B{}^a = \delta^A{}_B \quad (3.72)$$

and

$$\psi^A{}_a \psi_A{}^b = h_a{}^b. \quad (3.73)$$

The first equality is easy to see from the definitions of $\psi_A{}^a$ and g_{AB} . The second can be shown by demonstrating that the construction has the properties of being normal to u^a and u_b and satisfying $h_a{}^b h_b{}^c = h_a{}^c$ (and also that the trace is 3). This shows that the construction is indeed a projector, and that it is the specific projector onto a 3D surface normal to u^a .

To achieve covariance on both matter space and spacetime, e must be a scalar quantity on both of these manifolds. Eventually, this means that we will need e to depend only on scalars on both matter space and spacetime. We start by writing e as a function of tensors on matter space only; since the components of tensors on matter space are scalars on spacetime, this means that we have e as a function of scalars on spacetime. Later, we will have to rewrite this as a function of scalars on matter space, constructed from the tensors here. As a function of tensors on matter space, we must have $e(\psi^A{}_a, g^{ab}) = e(g^{AB})$: this is the only way that $\psi^A{}_a$ and g^{ab} can be contracted to form a tensor on matter space.

³This is shown in Appendix F.

Now we must write our expression for the stress energy tensor in terms of g^{AB} instead of g^{ab} . We start by writing the following:

$$\frac{\partial e}{\partial g^{ab}} = \frac{\partial e}{\partial g^{AB}} \psi^A{}_a \psi^B{}_b, \quad (3.74)$$

using the chain rule. Then

$$p_{ab} u^a = 0 \quad (3.75)$$

and

$$u_a h_{bc} T^{ab} = 0. \quad (3.76)$$

This contraction of T^{ab} gives the normal-tangential components in the frame comoving with the material: these components represent the energy flux relative to the matter, i.e. heat flux or other energy dissipation. This means that there is no energy flux relative to the matter, and we have non-dissipative (ideal) elasticity. Roughly, this lack of generality is due to the fact that our calculations are all tied to specific particles on a matter space; we have not included any transfer of energy *between* the particles. Our matter space ensures that all matter variables stay associated with specific particles, meaning that there is no interaction between the particles, and physical quantities cannot move along at any other rate. To produce a more general stress-energy tensor, we would need a way of modeling the flow of variables *relative* to the matter; one example of this is the approach presented in Andersson and Comer's review from 2006 [58], where heat flow could be modeled by taking the matter particles and entropy to be two separate fluids (so entropy is a second fluid with its own associated matter space).

We call p_{ab} the pressure, or stress, tensor. We can now see that the Lagrangian density, e , is the total energy density when our equations are solutions to the Euler-Lagrange equations. This is precisely because of Eq. 3.75: we find that the contraction

$$u^a u^b T_{ab} = e, \quad (3.77)$$

so we can see that e is the total energy density, measured in the frame comoving with the material.

Note that

$$n^2 = \frac{1}{3!} n^{abc} n_{abc} = \frac{1}{3!} g^{ad} g^{be} g^{cf} n_{abc} n_{def}. \quad (3.78)$$

From its definition, n_{abc} does not depend on g^{ab} , but only on $\psi^A{}_a$, so we can write the following derivative:

$$\frac{\partial n}{\partial g^{ab}} = \frac{1}{2} n h_{ab}. \quad (3.79)$$

This is done by first finding the derivative

$$\frac{\partial n^2}{\partial g^{pq}} = \frac{1}{2} g^{be} g^{cf} n_{pbc} n_{qef}, \quad (3.80)$$

where we note that

$$\frac{\partial g^{ad}}{\partial g^{pq}} = \delta^a{}_p \delta^d{}_q, \quad (3.81)$$

and multiplying by a factor of 3 comes from the fact that we can swap and rename the indices, so that all of the terms from the derivative are the same. We can then write

$$\frac{\partial n^2}{\partial g^{pq}} = \frac{1}{2} g^{BE} g^{CF} n_{PBC} n_{QEF} \psi^P{}_p \psi^Q{}_q. \quad (3.82)$$

Since

$$n^2 = \frac{1}{3!} g^{AD} g^{BE} g^{CF} n_{ABC} n_{DEF}, \quad (3.83)$$

and $g^{AB} g_{AB} = 3$, we can say that

$$\frac{1}{2} g^{BE} g^{CF} n_{ABC} n_{DEF} = n^2 g_{AD}. \quad (3.84)$$

(This also depends on the fact that the left-hand side is symmetric.) From here, we can write

$$\frac{\partial n^2}{\partial g^{pq}} = n^2 g_{PQ} \psi^P{}_p \psi^Q{}_q = n^2 h_{pq}. \quad (3.85)$$

Then we use the chain rule to find

$$\frac{\partial n}{\partial g^{ab}} = \frac{\partial n}{\partial n^2} \frac{\partial n^2}{\partial g^{ab}} = \frac{1}{2} n h_{ab}. \quad (3.86)$$

Next we define

$$e =: n(1 + \epsilon) \quad (3.87)$$

so that we get

$$p_{ab} = 2n \frac{\partial \epsilon}{\partial g^{ab}}, \quad (3.88)$$

which can be written as

$$p_{ab} = n \tau_{AB} \psi^A{}_a \psi^B{}_b \quad (3.89)$$

where we define

$$\tau_{AB} := 2 \frac{\partial \epsilon}{\partial g^{AB}}. \quad (3.90)$$

We now restrict ourselves to considering isotropic matter. This is the special case when the internal energy, ϵ , depends on $\psi^A{}_a$, g^{ab} , s , and k_{AB} only—for an

anisotropic crystal structure, the internal energy would need to depend on other tensors that provide additional information about preferred frames, etc.

We require that e and ϵ transform as scalars on both matter space and spacetime. To achieve this property in e and ϵ , we need them to be functions of only double scalars (which are scalars on matter space and spacetime) that can be constructed from g^{AB} and k_{AB} . In order for e and ϵ to be scalars, they must depend on scalars; if they depended on the components of the tensors, g^{AB} and k_{AB} , then they would be gauge dependent. Therefore, we need to find these double scalars, to see how our equation of state should be structured.

We know that g^{AB} transforms as a (2,0)-tensor on matter space and a set of scalars on spacetime. We now define

$$k^A_B := g^{AC} k_{BC} = g^{ac} \psi^A_a \psi^C_c k_{BC}, \quad (3.91)$$

which is a set of scalars on spacetime and a (1,1)-tensor on matter space. Its eigenvalues are scalars on matter space. Since the components of k^A_B are scalars on spacetime, a combination of them will also be a scalar. This means that they are the required double scalars.

Next we split k^A_B into its determinant and the unit-determinant matrix:

$$\eta^A_B := k^{-1/3} k^A_B, \quad (3.92)$$

where the determinant is related to the particle number density as follows:

$$k := \frac{1}{3!} \delta_{ABC} \delta^{DEF} k^A_D k^B_E k^C_F \quad (3.93)$$

$$= \delta^D_{[A} \delta^E_{B} \delta^F_{C]} k^A_D k^B_E k^C_F \quad (3.94)$$

$$= g^{[A|D} k_{AD} g^{B|E} k_{BE} g^{C]F} k_{CF} \quad (3.95)$$

$$= \frac{1}{3!} g^{AD} g^{BE} g^{CF} n_{ABC} n_{DEF} = n^2. \quad (3.96)$$

From the third line to the fourth line, we use the fact that n_{ABC} is the volume form for k_{AB} . This can be seen as follows: the third line can be rewritten as

$$k = g^{AD} g^{BE} g^{CF} X_{ABCDEF}, \quad (3.97)$$

where X_{ABCDEF} is some expression that is as of yet to be determined. We know that, in order for this expression to be equal to Eq. 3.95, we must have

$$X_{ABCDEF} = X_{[ABC]DEF} = k_{[A|D} k_{B|E} k_{C]F}. \quad (3.98)$$

Since k_{AB} is symmetric, we now have

$$X_{ABCDEF} = X_{[ABC][DEF]}. \quad (3.99)$$

That means that we should be able to write the expression X_{ABCDEF} as follows:

$$X_{ABCDEF} = X_\xi \delta_{ABC} \delta_{DEF}, \quad (3.100)$$

where X_ξ is some constant factor, and δ_{ABC} is the unit totally antisymmetric object in the particular coordinates, ξ^A . If we plug in our known value for X_{ABCDEF} , contract with the upstairs unit totally antisymmetric object, δ^{ABC} , and divide by a factor of $3!$, then we get

$$\frac{1}{3!} \delta^{ABC} \delta^{DEF} k_{AD} k_{BE} k_{CF} = \frac{1}{3!} X_\xi \delta^{ABC} \delta_{ABC} \delta^{DEF} \delta_{DEF}, \quad (3.101)$$

which becomes

$$k_\xi = 3! X_\xi, \quad (3.102)$$

and so

$$X_{ABCDEF} = \frac{1}{3!} n_{ABC} n_{DEF}, \quad (3.103)$$

which gets us to Eq. 3.96.

Now we can think of ϵ as a function of n , η^A_B , and s . However, since ϵ should only depend on scalars, it can only depend on scalar invariants of η^A_B , so

$$\epsilon(k^A_B, s) = \epsilon(k, \eta^A_B, s) = \epsilon(n, I^1, I^2, s), \quad (3.104)$$

where we choose

$$I^1 := \eta^A_A = k^{-1/3} g^{AB} k_{AB}, \quad (3.105)$$

$$I^2 := \eta^A_B \eta^B_A = k^{-2/3} g^{AB} g^{CD} k_{AC} k_{BD}. \quad (3.106)$$

We can write

$$\frac{\partial k}{\partial g^{AB}} = k g_{AB}. \quad (3.107)$$

(The derivation for this is similar to that of Eq. 3.79.) From this, we can then derive the following derivative:

$$\frac{\partial n}{\partial g^{AB}} = \frac{1}{2} n g_{AB}. \quad (3.108)$$

With these derivatives, we can write τ_{AB} as follows:

$$\tau_{AB} = \frac{p}{n}g_{AB} + 2f_\alpha\pi_{AB}^\alpha, \quad (3.109)$$

where the index α tells which scalar invariant the variable indexed depends on, as defined below. (α is replaced below by 1 and 2.) We use a summation convention over the index, α , and we have the following definitions:

$$p := n^2 \frac{\partial \epsilon}{\partial n}, \quad (3.110)$$

$$f_{1,2} := \frac{\partial \epsilon}{\partial I^{1,2}}, \quad (3.111)$$

$$\pi_{AB}^1 := \frac{\partial I^1}{\partial g^{AB}} = \eta_{AB} - \frac{1}{3}g_{AB}I^1, \quad (3.112)$$

$$\pi_{AB}^2 := \frac{\partial I^2}{\partial g^{AB}} = 2(\eta_{AC}\eta_B^C - \frac{1}{3}g_{AB}I^2). \quad (3.113)$$

If we put this expression for τ_{AB} into our expression for p_{ab} , then we get

$$p_{ab} = ph_{ab} + \pi_{ab} \quad (3.114)$$

(recall from earlier that $h_{ab} = \psi^A{}_a\psi^B{}_b g_{AB}$), where

$$\pi_{ab} = \psi^A{}_a\psi^B{}_b\pi_{AB} \quad (3.115)$$

and

$$\pi_{AB} := 2nf_\alpha\pi_{AB}^\alpha. \quad (3.116)$$

We can then see that π_{ab} is trace free ($g^{ab}\pi_{ab} = 0$) and spatial ($\pi_{ab}u^a = 0$). In the fluid limit, when ϵ only depends on n and s , π_{ab} vanishes automatically, leaving only $p_{ab} = ph_{ab}$, as mentioned before. Since π_{ab} is trace free, we can also think of p as being the trace of the pressure tensor:

$$p = g^{ab}p_{ab}. \quad (3.117)$$

With temperature defined as follows:

$$T := \frac{\partial \epsilon}{\partial s}, \quad (3.118)$$

the first law of thermodynamics becomes

$$d\epsilon = Tds - pd\left(\frac{1}{n}\right) + f_1dI^1 + f_2dI^2. \quad (3.119)$$

This means that $f_{1,2}$ are generalized forces associated with the variables $I^{1,2}$ in the thermodynamical sense.

Finally, the stress-energy tensor can be written as

$$T^{ab} = eu^a u^b + p^{ab} = (e + p)u^a u^b + pg^{ab} + \pi^{ab}. \quad (3.120)$$

3.3.2 Evolution equations

Non-relativistic fluids are governed by three conservation laws: conservation of mass, momentum, and energy. In general relativity, things are more complex; there is no local conservation of momentum or energy. Instead, we have the following constraint on the stress-energy tensor:

$$\nabla_a T^{ab} = 0. \quad (3.121)$$

To get this in conservation-law form, we start by contracting with a basis covector and expanding using the product rule to get

$$\nabla_a (T^{ab} e_b) = T^{ab} \nabla_{(a} e_{b)}. \quad (3.122)$$

If we define the basis as follows:

$$e_a = \begin{cases} n_a \\ (\partial_j)_a \end{cases}. \quad (3.123)$$

where n_a is the unit normal to the time-constant hypersurfaces and $(\partial_j)_a$ are tangent to these surfaces, then we can split the above expression as follows:

$$\nabla_a [T^{ab} (\partial_j)_b] = T^{ab} \nabla_{(a} (\partial_j)_{b)}, \quad (3.124)$$

$$\nabla_a (-T^{ab} n_b) = -T^{ab} \nabla_{(a} n_{b)}. \quad (3.125)$$

In the $3 + 1$ split, this becomes the Valencia formulation [59]:

$$(\alpha \sqrt{\gamma_x} T^0{}_j)_{,t} + (\alpha \sqrt{\gamma_x} T^i{}_j)_{,i} = \frac{1}{2} \alpha \sqrt{\gamma_x} T^{ab} \partial_j g_{ab}, \quad (3.126)$$

$$(\alpha^2 \sqrt{\gamma_x} T^{00})_{,t} + (\alpha^2 \sqrt{\gamma_x} T^{0i})_{,i} = \alpha^2 \sqrt{\gamma_x} [T^{a0} \partial_a (\ln \alpha) - T^{ab} \Gamma^0{}_{ab}]. \quad (3.127)$$

In addition to these, we use a conservation law for particle number, which is analogous to conservation of mass in the non-relativistic case:

$$\nabla_a (n u^a) = 0. \quad (3.128)$$

With this, we can write the set as

$$(\sqrt{\gamma_x} \mathcal{U})_{,t} + (\alpha \sqrt{\gamma_x} \mathcal{F}^i)_{,i} = \mathcal{S}, \quad (3.129)$$

where $\mathcal{U} = (D, S_i, \tau)$ represents the conserved variables which are defined as follows:

$$D = \alpha n u^0 \quad (3.130)$$

$$S_i = \alpha T^0{}_i \quad (3.131)$$

$$\tau = \alpha^2 T^{00} - D. \quad (3.132)$$

The corresponding fluxes are given by

$$\mathcal{F}(D)^i = n u^i \quad (3.133)$$

$$\mathcal{F}(S_j)^i = T^i{}_j \quad (3.134)$$

$$\mathcal{F}(\tau)^i = \alpha T^{0i} - \mathcal{F}(D). \quad (3.135)$$

The source terms are

$$\mathcal{S}(D) = 0, \quad (3.136)$$

$$\mathcal{S}(S_j) = \frac{1}{2} \alpha \sqrt{\gamma_x} T^{ab} \partial_j g_{ab}, \quad (3.137)$$

$$\mathcal{S}(\tau) = \alpha^2 \sqrt{\gamma_x} [T^{a0} \partial_a (\ln \alpha) - T^{ab} \Gamma^0{}_{ab}]. \quad (3.138)$$

A perfect fluid is an idealized fluid that experiences no viscosity, shear stress, or heat conduction. For relativistic perfect fluids, the stress-energy tensor is

$$T^{ab} = (e + p) u^a u^b + p g^{ab}, \quad (3.139)$$

where e is the energy density, and p is the pressure. In an orthonormal frame comoving with the fluid, $T^{00} = e$, $T^{ii} = p$, and $T^{ab} = 0$ when $a \neq b$; this corresponds to a fluid with energy density, e , pressure, p , zero viscosity ($T^{ij} = 0$ for $i \neq j$), and zero heat flux ($T^{0i} = 0$). The T^{0i} components are usually interpreted as representing the flow of energy through the x^i surfaces. Since mass and internal energy will flow with the fluid, we expect these contributions to be 0 in a comoving frame; the only contribution that would be non-zero would be that of heat. However, in a perfect fluid, we expect no heat conduction, and therefore, this term is also zero. From this, the corresponding values of D , S_i , and τ , along with their fluxes, can be calculated.

The structure of the stress-energy tensor for a perfect fluid reflects the fact that there is no shear stress in this situation. For example, think of a perfect fluid that is divided into two regions, say corresponding to positive and negative x in a three-dimensional Cartesian grid. Now imagine that the negative- x region has a velocity in the positive- y direction, and the positive- x region has a velocity in the negative- y direction: both velocities are parallel to the interface between the two regions. In a perfect fluid, this does not induce any stresses at the interface; however, in an elastic material, this motion *would* induce stresses, and waves would propagate through the material in directions *other than* the original direction of motion. Because of the shear stresses present in elasticity, the T^{ij} components of the stress-energy tensor in the comoving frame, will need to be non-zero. This is achieved by using

$$T^{ab} = (e + p)u^a u^b + pg^{ab} + \pi^{ab}, \quad (3.140)$$

as derived in the previous section. The only difference from the perfect-fluid form is the addition of the anisotropic stress term, π^{ab} . This term is trace free ($g^{ab}\pi_{ab} = 0$) and spatial ($u^a\pi_{ab} = 0$). The additional term means that, other than the fact that we still have no heat flux, this form of the stress-energy tensor is completely general. Our expressions for D , S_i , and τ include both a term analogous to those found in relativistic hydrodynamics and a term derived from the anisotropic stress:

$$D = nW, \quad (3.141)$$

$$S_i = nhW^2 v_i + \pi_{ij} v^j, \quad (3.142)$$

$$\tau = n(hW^2 - W) - (p - \pi), \quad (3.143)$$

where we have defined the standard specific enthalpy

$$h := 1 + \epsilon + \frac{p}{n}. \quad (3.144)$$

The fluxes become

$$\mathcal{F}(D)^i = n\alpha^{-1}W\hat{v}^i, \quad (3.145)$$

$$\mathcal{F}(S_j)^i = nhW^2\alpha^{-1}\hat{v}^i v_j + p\delta^i{}_j + \pi^i{}_j, \quad (3.146)$$

$$\begin{aligned} \mathcal{F}(\tau)^i &= n(hW^2 - W)\alpha^{-1}\hat{v}^i \\ &\quad + (p - \pi)\alpha^{-1}\beta^i + \gamma^{ij}\pi_{jk}v^k. \end{aligned} \quad (3.147)$$

In the Newtonian limit, where $v^i/c \ll 1$, $p/c^2 \ll 1$, and $\epsilon/c^2 \ll 1$, these equations become the standard Euler equations for conservation of mass, momentum,

and internal energy. For more on the Newtonian limit of these equations, see Appendix B.2.

3.3.3 Special source terms

One particularly useful case where the source terms are non-zero is the case where we have a Newtonian gravitational potential. In this case, the metric becomes

$$ds^2 = -(1 + 2\phi)dt^2 + (1 - 2\phi)dx^i dx^j \gamma_{ij}. \quad (3.148)$$

From this, we can derive source terms for this potential as follows, assuming that γ_{ij} is Cartesian:

$$\mathcal{S}(S_j) = -\alpha\sqrt{\gamma_x}\partial_j\phi[T^{tt} + T^{xx} + T^{yy} + T^{zz}], \quad (3.149)$$

$$\mathcal{S}(\tau) = -\sqrt{\gamma_x}T^{ti}\partial_i\phi. \quad (3.150)$$

In the Newtonian limit, where $T^{tt} \approx \rho$, $T^{tt} \gg T^{ij}$, and $T^{ti} \approx \rho v^i$, and in Cartesian coordinates, where $\alpha = \sqrt{\gamma_x} = 1$, these source terms become

$$\mathcal{S}(S_j) = -\rho\partial_j\phi, \quad (3.151)$$

$$\mathcal{S}(\tau) = -\rho v^i\partial_i\phi. \quad (3.152)$$

This is what we expect; the source term for the momentum should be the force on the material, which is the gradient of the potential, and the source term for the energy should be the work done as we move through the potential.

As a test that the code and formulation can handle a general background metric, we will also consider the case of cylindrical coordinates. In this case, the metric is as follows:

$$ds^2 = -dt^2 + dr^2 + r^2d\phi^2 + dz^2. \quad (3.153)$$

This means that all of the Christoffel symbols are zero aside from the following:

$$\Gamma^r_{\phi\phi} = -r \quad (3.154)$$

and

$$\Gamma^\phi_{r\phi} = \Gamma^\phi_{\phi r} = \frac{1}{r}. \quad (3.155)$$

The source term for τ is 0, because all of the Christoffel symbols, Γ^0_{ab} , are 0, and because α is constant in these coordinates. We also see that the ϕ and z components of the source term for S_j are 0; the only non-zero component is the r

component:

$$\mathcal{S}(S_r) = r^2 T^{\phi\phi}. \quad (3.156)$$

Since the stress energy tensor for an elastic material is

$$T^{ab} = \rho u^a u^b + p g^{ab} + \pi^{ab}, \quad (3.157)$$

in this case, the $\phi\phi$ component becomes

$$T^{\phi\phi} = \rho h \hat{v}^\phi \hat{v}^\phi + \frac{p}{r^2} + \frac{1}{r^4} \pi_{\phi\phi}, \quad (3.158)$$

and the source term is

$$\mathcal{S}(S_r) = \rho h \hat{v}^\phi \hat{v}^\phi r^2 + p + \frac{\pi_{\phi\phi}}{r^2}. \quad (3.159)$$

We also must be sure to use the general form of the evolution equations. In cylindrical coordinates, $\sqrt{\gamma_x} = r$, instead of 1, so our evolution equations look like

$$(r\mathcal{U})_{,t} + (r\mathcal{F}^i)_{,i} = \mathcal{S}. \quad (3.160)$$

3.4 Hyperbolicity

To numerically evolve an initial value problem for a system of partial differential equations (PDE's), we first need to ensure that the system is *well-posed*; in other words, the solutions evolved in time should be bounded so that small errors in the initial data do not cause significantly different behavior later in the evolution. *Hyperbolicity* is a property of the system of equations that implies well-posedness, so we would like to show this property for the elasticity formulation to show that these evolution equations will be suitable for this type of numerical simulation.

A discussion of well-posedness and its relationship with hyperbolicity, as well as a definition for when a system of equations is hyperbolic are given in Appendix A. The relevant points to the following discussion are summarized here.

To show hyperbolicity in general relativity, using differential geometry notation, we must rewrite our system of equations so that it is in the following form:

$$P_{\alpha\beta}^c \omega^\beta,_c + \text{l.o.} = 0, \quad (3.161)$$

where "l.o." refers to lower order terms, the indices α label the equations, and the indices β label the variables. Here, $P_{\alpha\beta}^c$ is called the *principle symbol*. If α and β are indices of the same type, and $P_{\alpha\beta}^c$ is symmetric in α and β , then we can

construct the following *energy norm*:

$$E(\omega, \omega) := t_c P_{\alpha\beta}{}^c \omega^\alpha \omega^\beta, \quad (3.162)$$

where, in relativity, t_c is a time-like covector. If there exists some t_c for which E is positive definite, then the system of equations is *symmetric hyperbolic*. For this reason, the goal of the following sections will be to rewrite the elasticity formulation in the form of Eq. 3.161, show that the principle symbol is symmetric, and then derive an expression for the energy norm, E , to determine the conditions on its positive definiteness.

3.4.1 Second order system

We need to show that the kinematic equations (evolution equations for $\psi^A{}_i$) and the dynamic equations (equations from $\nabla_a T^{ab} = 0$) with constraints, form a symmetric hyperbolic system of evolution equations, regardless of whether or not the constraints are actually obeyed. We also will show that without the constraints, the system is strongly hyperbolic, but not symmetric hyperbolic. Background on hyperbolicity, as well as its treatment in differential geometry notation, is given in Appendix A.

We will start by deriving the second order equations. This is because the first order equations for $\psi^A{}_a$ are the second order equations for χ^A , so all we need to do to go from one to the other is replace $\chi^A{}_{,ab}$ with $\psi^A{}_{a,b}$; we can also add multiples of $\psi^A{}_{[a,b]}$ to the right-hand side (since we know that $\psi^a{}_{[a,b]} = 0$). This means that we have an infinite number of first-order systems that are equivalent to our second-order system; these are parameterized by the factor multiplied by our constraint, $\psi^A{}_{[a,b]}$. We now must select one that is symmetric hyperbolic.

We start by thinking of χ^A and g^{ab} as the independent variables, and $\psi^A{}_a = \chi^A{}_{,a}$ as a derived object. The matter space metric, k_{AB} , and the entropy, s , are not varied. With all of these considerations in mind, the action becomes

$$S := \int e(g^{ab}, \chi^A, \chi^A{}_{,a}, k_{AB}, \dots, s) \sqrt{g_x} d^4x; \quad (3.163)$$

again, the dots stand for any other matter-space tensors. With integration by parts, and neglecting boundary terms, we get

$$\delta S = \int \left(\frac{1}{2} T_{ab} \delta g^{ab} + \mathcal{E}_A \delta \chi^A \right) \sqrt{g_x} d^4x, \quad (3.164)$$

where the first term is the same definition we had in Section 3.3, and in the second term is

$$\mathcal{E}_A := \frac{\partial e}{\partial \chi^A} - \frac{1}{\sqrt{g_x}} \left(\sqrt{g_x} \frac{\partial e}{\partial (\chi^A, a)} \right)_{,a}. \quad (3.165)$$

We can derive the second term as follows. We find

$$\delta(\sqrt{g_x}e) = e\delta(\sqrt{g_x}) + \sqrt{g_x}\delta(e) \quad (3.166)$$

$$= \frac{1}{2}\sqrt{g_x}T_{ab}\delta g^{ab} + \sqrt{g_x} \left[\frac{\partial e}{\partial \chi^A} \delta \chi^A + \frac{\partial e}{\partial (\chi^A, a)} \delta(\chi^A, a) \right], \quad (3.167)$$

where the first term comes about as before. We can look more closely at the last term:

$$\frac{\partial e}{\partial (\chi^A, a)} \delta(\chi^A, a) = \frac{\partial e}{\partial (\chi^A, a)} \delta(\nabla_a \chi^A) \quad (3.168)$$

$$= \nabla_a \left[\frac{\partial e}{\partial (\chi^A, a)} \delta \chi^A \right] - \nabla_a \left[\frac{\partial e}{\partial (\chi^A, a)} \right] \delta \chi^A \quad (3.169)$$

$$= \text{b.t.} - \frac{1}{\sqrt{g_x}} \left(\sqrt{g_x} \frac{\partial e}{\partial (\chi^A, a)} \right)_{,a} \delta \chi^A, \quad (3.170)$$

where b.t. stands for a boundary term (which we neglect here). Because χ^A is a scalar on spacetime, we can say that $\chi^A, a = \nabla_a \chi^A$ on the first line. Then we swap the order of the derivatives, and use the product rule to get to the second line. In the second line, the first term is of the form $\nabla_a v^a$, where v^a is some vector; this means that this term is a boundary term. We can then rewrite the second term in terms of partial derivatives.

The variations are then generated by an infinitesimal change in coordinates:

$$x^a \rightarrow x^a + \zeta^a, \quad (3.171)$$

and so they take the form

$$\delta \chi^A = \mathcal{L}_\zeta \chi^A = \zeta^c \psi^A{}_c \quad (3.172)$$

and

$$\delta g^{ab} = \mathcal{L}_\zeta g^{ab} = 2\nabla^{(a} \zeta^{b)}. \quad (3.173)$$

Our action must be invariant under such changes, so we plug in these values and get

$$\delta S = \int \left(\frac{1}{2} T_{ab} 2\nabla^{(a} \zeta^{b)} + \mathcal{E}_A \zeta^c \psi^A{}_c \right) \sqrt{g_x} d^4x. \quad (3.174)$$

We can write

$$T_{ab}\nabla^a\zeta^b = \nabla^a(T_{ab}\zeta^b) - \zeta^b\nabla^aT_{ab}, \quad (3.175)$$

using the product rule. Since the first term is a boundary term, we can see that, in order for δS to be 0, we must have

$$\nabla^bT_{ab} = \psi^A{}_a\mathcal{E}_A. \quad (3.176)$$

This means that we have linked stress-energy conservation to our second order system, \mathcal{E}_A , which represents the elastic matter field equations; stress-energy conservation will only be achieved if $\psi^A{}_a\mathcal{E}_A = 0$ is satisfied. It is not immediately obvious how this can be true, because $\mathcal{E}_A = 0$ only has 3 independent components, while $\nabla^aT_{ab} = 0$ appears to have 4. However, it is possible to show that one component of the latter, namely $u^a\nabla^bT_{ab} = 0$, is equivalent to

$$-n\dot{\epsilon} + \frac{p}{n}\dot{n} - \pi_{ab}\nabla^b u^a = 0, \quad (3.177)$$

where, for example, $\dot{n} = u^a\nabla_a n$. However, this equation is just the first law of thermodynamics, which should automatically be satisfied by any thermodynamically consistent equation of state. With an equation of state, the number of independent components for stress-energy conservation reduce to 3, which is the same as for $\mathcal{E}_A = 0$.

The matter equations can be written in second order form as follows:

$$\mathcal{E}_A = M^{ab}{}_{AB}\chi^B{}_{,ba} - G_A = 0, \quad (3.178)$$

where G_A represents all lower-order terms, and

$$M^{ab}{}_{AB} := \frac{\partial^2 e}{\partial \psi^A{}_a \partial \psi^B{}_b}. \quad (3.179)$$

This can be seen from the definition of \mathcal{E}_A and the fact that

$$\left(\frac{\partial e}{\partial (\chi^A{}_{,a})} \right)_{,a} = \frac{\partial^2 e}{\partial \psi^A{}_a \partial \psi^B{}_b} \frac{\partial \psi^B{}_b}{\partial x^a} = \frac{\partial^2 e}{\partial \psi^A{}_a \partial \psi^B{}_b} \chi^B{}_{,ba}, \quad (3.180)$$

and all of the other terms from the definition of \mathcal{E}_A are comprised of lower-order derivatives.

We call $M^{ab}{}_{AB}$ the *principal symbol*, and note that, while it is not symmetric in ab alone, it *is* symmetric in the sense that $M^{ab}{}_{AB} = M^{ba}{}_{BA}$. Although the principal symbol is not symmetric in ab , the antisymmetric part does not contribute

to \mathcal{E}_A , because these indices are contracted with the symmetric object $\chi^A_{,ba}$.

Next, we would like to write the principal symbol explicitly in terms of quantities involved in the equation of state and information about the shear. We start by noting, again, that $e(\psi^A_a, g^{ab}) = e(g^{AB})$; we use this and the chain rule for partial derivatives to write

$$M^{ab}_{AB} = \frac{\partial^2 g^{CD}}{\partial \psi^A_a \partial \psi^B_b} \frac{\partial e}{\partial g^{CD}} + \frac{\partial g^{CD}}{\partial \psi^B_b} \frac{\partial g^{EF}}{\partial \psi^A_a} \frac{\partial^2 e}{\partial g^{EF} \partial g^{CD}}. \quad (3.181)$$

Then with the derivatives

$$\frac{\partial g^{CD}}{\partial \psi^A_a} = 2\delta^{(D}_A \psi^{C)}_c g^{ca}, \quad (3.182)$$

$$\frac{\partial^2 g^{CD}}{\partial \psi^B_b \partial \psi^B_b} = 2\delta^{(D}_A \delta^{C)}_B g^{ab}, \quad (3.183)$$

we can write

$$M^{ab}_{AB} = 4 \frac{\partial^2 e}{\partial g^{AC} \partial g^{BD}} \psi^{Ca} \psi^{Db} + 2 \frac{\partial e}{\partial g^{AB}} g^{ab}, \quad (3.184)$$

where

$$\psi^{Aa} := \psi^A_b g^{ab}. \quad (3.185)$$

We now recall that $h_{ab} = \psi^A_a \psi^B_b g_{AB}$. We can split M^{ab}_{AB} into parts parallel and normal to the 4-velocity by replacing $g^{ab} = -u^a u^b + \psi^{Aa} \psi^{Bb} g_{AB}$ to get the following:

$$M^{ab}_{AB} = -\mu_{AB} u^a u^b + U_{ACBD} \psi^{Ca} \psi^{Db}, \quad (3.186)$$

where we have the following definitions:

$$\mu_{AB} := 2 \frac{\partial e}{\partial g^{AB}}, \quad (3.187)$$

$$U_{ACBD} := 4 \frac{\partial^2 e}{\partial g^{AC} \partial g^{BD}} + 2 \frac{\partial e}{\partial g^{AB}} g_{CD}. \quad (3.188)$$

From this, we notice that there are no cross terms; in other words, $u_a h_{bc} M^{ab}_{AB} = 0$.

Next we can evaluate our expressions for μ_{AB} and U_{ACBD} further. The first becomes

$$\mu_{AB} = n\tau_{AB} + eg_{AB}, \quad (3.189)$$

via the chain rule, where τ_{AB} was defined earlier. We can also write

$$U_{ACBD} = n[g_{AC}\tau_{BD} + g_{BD}\tau_{AC} + \tau_{AB}g_{CD} + \tau_{ACBD}] + 2eg_{A[C}g_{D]B}, \quad (3.190)$$

where

$$\tau_{ABCD} := 4 \frac{\partial^2 \epsilon}{\partial g^{AB} \partial g^{CD}}. \quad (3.191)$$

The above expression for U_{ACBD} is derived using the chain rule for partial derivatives, and we recall that

$$\frac{\partial g_{BD}}{\partial g^{AC}} = -g_{B(A} g_{C)D}. \quad (3.192)$$

Next we use the chain rule to write the objects τ_{AB} and τ_{ABCD} as a sum of terms. We already have

$$\tau_{AB} = \frac{p}{n} g_{AB} + 2f_\alpha \pi_{AB}^\alpha. \quad (3.193)$$

Now we can also write

$$\begin{aligned} \tau_{ABCD} = & -2 \frac{p}{n} g_{A(C} g_{D)B} + \left(c_s^2 - \frac{p}{n} \right) g_{AB} g_{CD} + 2n (g_{AB} f_{n\alpha} \pi_{CD}^\alpha + g_{CD} f_{n\alpha} \pi_{AB}^\alpha) \\ & + 4f_{\alpha\beta} \pi_{AB}^\alpha \pi_{CD}^\beta + 4f_\alpha \pi_{ABCD}^\alpha, \end{aligned} \quad (3.194)$$

where we have introduced the objects

$$c_s^2 := \frac{\partial p}{\partial n} \quad (3.195)$$

$$f_{n\alpha} := \frac{\partial^2 \epsilon}{\partial n \partial I^\alpha} \quad (3.196)$$

$$f_{\alpha\beta} := \frac{\partial^2 \epsilon}{\partial I^\alpha \partial I^\beta} \quad (3.197)$$

$$\begin{aligned} \pi_{ABCD}^1 := & \frac{\partial^2 I^1}{\partial g^{AB} \partial g^{CD}} \\ = & \left(\frac{1}{3} g_{A(C} g_{D)B} + \frac{1}{9} g_{AB} g_{CD} \right) I^1 - \frac{1}{3} (\eta_{AB} g_{CD} + \eta_{CD} g_{AB}) \end{aligned} \quad (3.198)$$

$$\begin{aligned} \pi_{ABCD}^2 := & \frac{\partial^2 I^2}{\partial g^{AB} \partial g^{CD}} \\ = & \left(\frac{2}{3} g_{A(C} g_{D)B} + \frac{4}{9} g_{AB} g_{CD} \right) I^2 - \frac{4}{3} (\eta_{AE} \eta^E_B g_{CD} + \eta_{CE} \eta^E_D g_{AB}) \\ & + 2\eta_{A(C} \eta_{D)B}. \end{aligned} \quad (3.199)$$

3.4.2 Unsheared state

To get a better sense of the meaning of these equations, we look at the structure of these equations in the unsheared state. We will denote this state by a circle. In the unsheared state, we have

$$\dot{k}_{AB} = n^{2/3} g_{AB}. \quad (3.200)$$

From this, we get the following

$$\overset{\circ}{I}{}^\alpha = 3 \quad (3.201)$$

$$\overset{\circ}{\pi}{}^\alpha_{AB} = 0 \quad (3.202)$$

$$\overset{\circ}{\pi}{}^1_{ABCD} = g_{A(C}g_{D)B} - \frac{1}{3}g_{AB}g_{CD} \quad (3.203)$$

$$\overset{\circ}{\pi}{}^2_{ABCD} = 4\overset{\circ}{\pi}{}^1_{ABCD}. \quad (3.204)$$

From this, we can write

$$n\overset{\circ}{\tau}_{AB} = pg_{AB} \quad (3.205)$$

$$n\overset{\circ}{\tau}_{ABCD} = 2rg_{A(C}g_{D)B} + qg_{AB}g_{CD}, \quad (3.206)$$

where

$$r := -p + 2n(f_1 + 4f_2), \quad (3.207)$$

$$q := nc_s^2 - p - \frac{4}{3}n(f_1 + 4f_2). \quad (3.208)$$

Note that only the linear combination $(f_1 + 4f_2)$ appears in these expressions. This is discussed further in Section 3.4.5. Finally we can write

$$\overset{\circ}{\mu}_{AB} = (p + e)g_{AB}, \quad (3.209)$$

$$\overset{\circ}{U}_{ACBD} = (2p + q + e)g_{AC}g_{BD} + (p + r)g_{AB}g_{CD} + (r - e)g_{AD}g_{BC}. \quad (3.210)$$

3.4.3 First order system

We now reduce the second order system to the first order system. This reduction must have the form

$$\bar{\mathcal{E}}_A := \bar{M}^{ab}{}_{AB}\psi^B{}_{b,a} - G_A = 0. \quad (3.211)$$

with

$$\bar{M}^{ab}{}_{AB} := M^{ab}{}_{AB} + D^{ab}{}_{AB}, \quad (3.212)$$

where $D^{ab}{}_{AB} = D^{[ab]}{}_{AB}$, which is the constraint addition. This form means that the first order system is made up of the symmetric part, which is the same as the second order system, with the addition of an antisymmetric part, which corresponds to the necessary constraint addition.

We also need $\nabla_b T^{ab} = 0$ to give us the dynamical part of the equations; however, we need the *full* system to be hyperbolic, so we may also need constraints for these terms. To formalize this constraint addition, we define the system of

equations to be

$$\bar{E}^a := \nabla_b T^{ab} + \bar{\Lambda}_{ACBD} \psi^{Aa} \psi^{Cc} \psi^{Db} \psi^B_{[b,c]} = 0, \quad (3.213)$$

where $\bar{\Lambda}_{ACBD} = -\bar{\Lambda}_{ADBC}$. This parameterizes a family of constraint additions. We now need to write $\nabla_b T^{ab} = 0$ in terms of ψ^A_a , starting with the expression

$$T^{ab} = 2 \frac{\partial e}{\partial g^{AB}} \psi^{Aa} \psi^{Bb} - eg^{ab}. \quad (3.214)$$

Now we expand this, and keep only the principal part, consisting of terms of the form $\psi^B_{b,c}$. To do this, we plug the above expression for T^{ab} into $\nabla_b T^{ab} = 0$. We recall that

$$\nabla_a \psi^{Aa} = \psi^{Aa}_{,a} + \text{l.o.}, \quad (3.215)$$

where “l.o.” represents all lower order terms. We can then see that, using the chain rule and expanding using the product rule,

$$\frac{\partial g^{CD}}{\partial x^b} = 2 \psi^{(C}_{d,b} \psi^{D)d} + \text{l.o..} \quad (3.216)$$

After rearranging terms, we get

$$\nabla_b T^{ab} = \left(\psi^{Aa} M^{cb}{}_{AB} - 4 \frac{\partial e}{\partial g^{AB}} \psi^{A[b} g^{c]a} \right) \psi^B_{b,c} + \text{l.o..} \quad (3.217)$$

Next, we substitute in $g^{ab} = -u^a u^b + \psi^{Aa} \psi^{Bb} g_{AB}$, recalling that $\mu_{AB} := 2 \frac{\partial e}{\partial g^{AB}}$, our definition for \bar{E}^a , and the symmetries in $\bar{\Lambda}_{ACBD}$, and we get

$$\bar{E}^a = (\psi^{Aa} \bar{M}^{cb}{}_{AB} + 2u^a \mu_{AB} \psi^{A[b} u^{c]}) \psi^B_{b,c} + \text{l.o.}, \quad (3.218)$$

where our definition of $\bar{M}^{ab}{}_{AB}$ becomes

$$\bar{M}^{ab}{}_{AB} := M^{ab}{}_{AB} + (\bar{\Lambda}_{ACBD} - 2g_{A[C}\mu_{D]B}) \psi^{Ca} \psi^{Db}. \quad (3.219)$$

Then this modification of $M^{ab}{}_{AB}$ can be written as a modification of U_{ACBD} :

$$\bar{U}_{ACBD} := U_{ACBD} + \bar{\Lambda}_{ACBD} - 2g_{A[C}\mu_{D]B}, \quad (3.220)$$

so that we have

$$\bar{M}^{ab}{}_{AB} = -\mu_{AB} u^a u^b + \bar{U}_{ACBD} \psi^{Ca} \psi^{Db}. \quad (3.221)$$

We then split \bar{E}^a into the parts parallel and normal to the 4-velocity as follows:

$$u_a \bar{E}^a = -2\mu_{AB}\psi^{Ab}u^c\psi^B_{[b,c]} \quad (3.222)$$

and

$$\psi^A{}_a \bar{E}^a = g^{AD}\bar{M}^{cb}{}_{DB}\psi^B{}_{b,c} - G^A, \quad (3.223)$$

where G^A are lower order terms. However, if μ_{AB} is invertible, then $u_a \bar{E}^a = 0$ is equivalent to the kinematic evolution equations with the hyperbolicity fix (Eq. 3.19), because we can write

$$u^c\psi^B_{[b,c]} = 0, \quad (3.224)$$

which is equivalent to Eq. 3.18 in Section 3.2.1.

Showing symmetric hyperbolicity for the first-order system For symmetric hyperbolicity, the principal symbol must be symmetric, and, roughly, the time component must be positive definite. (More precisely, there must exist a time-like covector, such that the norm constructed from this is positive definite.) We start by showing the first condition, and then move on to the second.

Neither of the principal symbols used so far have the correct index structure, which should be of the form, $p_{\alpha\beta}^{c}$, where the composite index is defined by $\omega^\alpha := \psi^A{}_a$, and will be represented by Greek indices. We need to obtain a symbol of this index structure, because this is the form in which the definition of hyperbolicity is written, so to show hyperbolicity, we must write the equations in this way. To achieve our goal of symmetry, we start by defining the symbol

$$W^{ab}{}_{AB}{}^c := u^a \bar{M}^{cb}{}_{AB} - 2u^{[c} \bar{M}^{b]a}{}_{BA}. \quad (3.225)$$

We then consider a system of first order equations:

$$\mathcal{E}^a{}_A := W^{ab}{}_{AB}{}^c \psi^B{}_{b,c} - G_A u^a = 0 \quad (3.226)$$

and

$$\mathcal{A}_A := -u^c g_{AB} \chi^B{}_{,c} = 0, \quad (3.227)$$

where χ^A and $\psi^A{}_a$ are now considered to be independent variables, and u^a is derived from them via j^a , which comes from the volume forms on matter space and spacetime. Now we show that each solution, χ^A , to the second order system above generates a solution, $\psi^A{}_a = \chi^A{}_{,a}$, to the first order system. Beig and

Schmidt have also proven the converse of this: a solution, $\psi^A{}_i$, to the first order system, obeying $\psi^A{}_{[i,j]} = 0$, gives rise to a second order solution, χ^A [30].

In summary, we started with a second order system, \mathcal{E}_A . Then, from this we constructed the general form of the first order system, $\bar{\mathcal{E}}_A$. Once we had this, we made a particular choice of constraint addition which gave us the specific first order system, $\psi^A{}_a \bar{E}^a$, which is of the form, $\bar{\mathcal{E}}_A$. Now we have constructed another, modified first order system, $\mathcal{E}^a{}_A$, in order to have the necessary form of the principle symbol to show when and if the system is hyperbolic. We must now show that solutions to this new first order system ($\mathcal{E}^a{}_A$) are also solutions to our chosen first order system ($\psi^A{}_a \bar{E}^a$), and that they therefore generate solutions to the second order system, \mathcal{E}_A .

Equivalence of solutions to $(\mathcal{E}^a{}_A, \mathcal{A}_A)$ and (E^a) We see that our symbol, $W^{ab}{}_{AB}{}^c$, is symmetric in terms of the composite indices:

$$W^{ab}{}_{AB}{}^c = W^{ba}{}_{BA}{}^c, \quad (3.228)$$

if and only if $\bar{M}^{ab}{}_{AB}$ has the same sort of symmetry:

$$\bar{M}^{ab}{}_{AB} = \bar{M}^{ba}{}_{BA}. \quad (3.229)$$

To ensure that this is the case, we set

$$\bar{\Lambda}_{ACBD} = \Lambda_{ACBD} + 2g_{A[C}\mu_{D]B}, \quad (3.230)$$

where $\Lambda_{ACBD} = \Lambda_{BDAC} = -\Lambda_{ADBC}$, and which we will determine later, when we look at positivity. We can see that this object produces the right symmetries in $\bar{M}^{ab}{}_{AB}$ as long as $\Lambda_{ACBD}\psi^{Ca}\psi^{Db}$ has the right symmetries. We can see that it does as follows:

$$\Lambda_{ADBC}\psi^{Ca}\psi^{Db} = \Lambda_{BCAD}\psi^{Db}\psi^{Ca} \quad (3.231)$$

$$= \Lambda_{BDAC}\psi^{Cb}\psi^{Da}, \quad (3.232)$$

where we first use the symmetries mentioned above, and then relabel the contracted indices C and D .

We now must verify that this is equivalent to our evolution equations. We know that

$$u_a \bar{M}^{ab}{}_{AB} = u_a M^{ab}{}_{AB} = u^b \mu_{AB}. \quad (3.233)$$

This means that

$$u_a W^{ab}{}_{AB}{}^c = -\bar{M}^{cb}{}_{AB}. \quad (3.234)$$

Also, we have

$$\psi_{Ca} W^{ab}{}_{AB}{}^c = -2\psi^{D[b} u^{c]} \bar{U}_{ACBD}. \quad (3.235)$$

If \bar{U}_{ACBD} is invertible as a matrix in the compound indices AC and BD , then we get the final decomposition

$$u_a \mathcal{E}^a{}_A = 0 \Rightarrow \bar{M}^{cb}{}_{AB} \psi^B{}_{b,c} = G_A \quad (3.236)$$

$$\psi^E{}_a \mathcal{E}^a{}_A = 0 \Rightarrow u^b \psi^A{}_{[a,b]} = 0 \quad (3.237)$$

$$\mathcal{A}_A = 0 \Rightarrow u^a \chi^C{}_{,a} = 0. \quad (3.238)$$

The second line comes from the fact that we can write

$$\psi_E{}_a \mathcal{E}^a{}_A = -2\psi^{Db} u^c \bar{U}_{AEBD} \psi^B{}_{[b,c]}; \quad (3.239)$$

if \bar{U}_{AEBD} is non-singular, then we can write this simply as $u^c \psi^B{}_{[b,c]} = 0$. It can then be seen that the first two lines are equivalent to Eq. 3.222 and Eq. 3.223. It turns out that the third line is equivalent to the advection equation for the matter-space metric, $u^a k_{AB,a} = 0$, as we have the following equality:

$$u^a k_{AB,a} = k_{AB,C} u^a \chi^C{}_{,a}, \quad (3.240)$$

because of the chain rule. We have now shown that solutions to our chosen system, $\mathcal{E}^a{}_A$ and \mathcal{A}_A , will be the same as the solutions to the advection equation for k_{AB} , and the first order system with constraint addition, E^a .

Positive definiteness of the energy norm for the first-order system In order to show positive definiteness, we show the existence of a time-like covector, t_a , that makes the following quadratic form (the energy norm), positive definite:

$$E := t_c (W^{ab}{}_{AB}{}^c m^A{}_a m^B{}_b - u^c g_{AB} l^A l^B). \quad (3.241)$$

This energy norm is constructed as in Eq. 3.162. The first term comes from $\mathcal{E}^a{}_A$ (Eq. 3.226), and the second comes from \mathcal{A}_A (Eq. 3.227), where $m^A{}_a$ and l^A are the characteristic variables, which can be thought of as either perturbations of $\psi^A{}_a$ and χ^A about the background, or general versions of the variables $\psi^A{}_a$ and χ^A themselves. For the first order system, the norms of both $m^B{}_b$ and l^A must be bounded for the system ($\mathcal{E}^a{}_A$ and \mathcal{A}_A) to be well-posed, so both are included in

the constructed energy norm. We decompose using

$$m^A{}_a =: \alpha^A u_a + \alpha^{AC} \psi_{Ca}, \quad (3.242)$$

and choosing $t_a = u_a$, to get

$$E = \mu_{AB} \alpha^A \alpha^B + \bar{U}_{ACBD} \alpha^{AC} \alpha^{BD} + g_{AB} l^A l^B. \quad (3.243)$$

This means that u_a is a subcharacteristic vector (meaning it produces positive definiteness of E) as long as μ_{AB} and \bar{U}_{ACBD} are positive definite (which means that they are also invertible). For simplicity, we examine only the unsheared state. We know that μ_{AB} is positive definite in the unsheared state if $(p + e) > 0$, so we just need to examine the positive definiteness of the term containing \bar{U}_{ACBD} , or its value in the unsheared state, $\dot{\bar{U}}_{ACBD}$.

We choose

$$\Lambda_{ACBD} = 2(d - e - p)g_{A[C}g_{D]B} \quad (3.244)$$

or

$$\begin{aligned} \bar{\Lambda}_{ACBD} &= 2(d - e - p)g_{A[C}g_{D]B} + 2g_{A[C}\mu_{D]B} \\ &= 2(d - e - p)g_{A[C}g_{D]B} + 2g_{A[C}\delta_D{}^E [pg_{EB} + 2nf_\alpha\pi_{EB}^\alpha + eg_{EB}] \\ &= 4nf_\alpha g_{A[C}\pi_{D]B}^\alpha + 2dg_{A[C}g_{D]B}. \end{aligned} \quad (3.245)$$

To find the value of the total constraint addition, we will need to determine the value of d .

Next we look again at the unsheared state, and use the decomposition,

$$\alpha^{AB} = \omega^{AB} + \kappa^{AB} + \frac{\kappa}{3}g^{AB}, \quad (3.246)$$

where ω^{AB} is antisymmetric, κ^{AB} is symmetric and trace free, and g^{AB} is the spacetime metric on matter space as defined before. We first note that

$$\begin{aligned} \dot{\bar{U}}_{ACBD} &= (2p + q + e)g_{AC}g_{BD} + (p + r)g_{AB}g_{CD} \\ &\quad + (r - e)g_{AD}g_{BC} + 2(d - p - e)g_{A[C}g_{D]B}. \end{aligned} \quad (3.247)$$

From this we get

$$\dot{\bar{U}}_{ACBD}\alpha^{AC}\alpha^{BD} = \left(nc_s^2 + \frac{2d}{n} \right) \kappa^2 + d\omega^{AB}\omega_{AB} + [4n(f_1 + 4f_2) - d]\kappa^{AB}\kappa_{AB}. \quad (3.248)$$

In order for our energy norm, E , to be positive definite, this expression must be positive definite. This will be the case as long as $0 < d < 4n(f_1 + 4f_2)$. Since $d > 0$, adding constraints to $\nabla_b T^{ab} = 0$ is necessary for symmetric hyperbolicity. For example, we could choose $d = 2n(f_1 + 4f_2)$, and then the constraint addition becomes

$$\bar{E}^a = \nabla_b T^{ab} + h^{ac} \psi^{Db} [4n f_\alpha \pi_{DB}^\alpha + 4n(f_1 + 4f_2) g_{DB}] \psi^B_{[b,c]}, \quad (3.249)$$

where the first term of the constraint addition makes the principal part of the second order system symmetric, and the second term makes it positive definite. This can be seen, because the first term comes from the second term in $\bar{\Lambda}_{ACBD}$, $2g_{A[C}\mu_{D]B}$, which we need to include in order to make $\bar{M}^{ab}_{AB} = \bar{M}^{ba}_{BA}$, which is necessary for the symmetry of $W^{ab}_{AB}{}^c$ in its compound indices. The second term includes d , which we know needs to be non-zero for the term including $\mathring{\bar{U}}_{ACBD}$ to be positive definite.

3.4.4 Characteristics of the first-order system

To get a better sense of the wave structure of solutions to this system, and to evaluate whether this approaches the expected Newtonian limit, we now examine the characteristics of the elasticity formulation.

The variable, k_a , is a characteristic covector of the first order system of equations with characteristic variable, $\omega^\beta = m^B{}_b$, if

$$W^{ab}_{AB}{}^c m^B{}_b k_c = 0. \quad (3.250)$$

(Compare this to $k_c P_{\alpha\beta}{}^c \omega^\alpha = 0$ from Appendix A.6. When k_c is decomposed into a time-like part and a spatial part—i.e. Eq. A.146—it is clear that this becomes the usual equation defining an eigenvector.) We now use the decomposition of $m^B{}_b$ as before, fix the irrelevant overall factor, and parameterize $k_a = \lambda u_a - e_a$, where $e_a = \psi^A{}_a e_A$ is a unit covector on spacetime, normal to u^a and e_A is the corresponding unit covector on matter space. Then λ is the physical velocity of the mode relative to the matter. If we use a decomposition where $u_a W^{ab}_{AB}{}^c = -\bar{M}^{cb}_{AB}$ and $\psi_{Ca} W^{ab}_{AB}{}^c = -2\psi^{D[b} u^{c]} \mathring{\bar{U}}_{ACBD}$ (Eq. 3.234 and Eq. 3.235), then we can convert the characteristic variable equation into the pair,

$$\bar{U}_{ACBD} (\alpha^B e^D + \lambda \alpha^{BD}) = 0, \quad (3.251)$$

$$\lambda \mu_{ABA} \alpha^B + \bar{U}_{ACBDE} e^C \alpha^{BD} = 0. \quad (3.252)$$

Symmetric hyperbolicity implies that \bar{U}_{ACBD} is invertible, so the first of these equations becomes

$$\alpha^B e^D + \lambda \alpha^{BD} = 0. \quad (3.253)$$

Then this has two classes of solutions. The first class consists of solutions where $\lambda = 0$ and $\alpha^B = 0$, which gives

$$\bar{U}_{ACBD} e^C \alpha^{BD} = 0, \quad (3.254)$$

from Eq. 3.252. The fact that $\lambda = 0$ means that the modes are stationary with respect to the material. Eq. 3.254 consists of 3 equations for 9 unknowns (the components of α^{BD}), so we have 6 degrees of freedom, which result in 6 modes for the solutions. We can then parameterize these solutions as follows:

$$\alpha^{BD} = (\bar{U}^{-1})^{ACBD} v_A w_C, \quad (3.255)$$

where $w_C e^C = 0$, and w_C and v_C are otherwise arbitrary covectors on matter space that parameterize our six degrees of freedom. ($w_C e^C = 0$ ensures that the object, $v_A w_C = \bar{U}_{ACBD} \alpha^{BD}$, is normal to e^C , so that our original equation is satisfied.)

The second class of solutions occurs where $\lambda \neq 0$, and

$$\alpha^{BD} = -\lambda^{-1} \alpha^B e^D, \quad (3.256)$$

or equivalently,

$$m^B_b = \lambda^{-1} \alpha^B k_b. \quad (3.257)$$

Since α^A is a characteristic variable of the system, we can write it as a plane wave,

$$\alpha^A = \bar{\alpha}^A e^{ik_c u^c}, \quad (3.258)$$

where $\bar{\alpha}^A$ is the amplitude, and then the gradient of this is

$$\alpha^A_{,a} = \bar{\alpha}^A i k_a e^{ik_c u^c} = i k_a \alpha^A \quad (3.259)$$

From this, we see that we can interpret m^B_b as the gradient of the plane wave, $\lambda^{-1} \alpha^B$, with wave vector, k_b ; this means that it is a possible perturbation of ψ^B_b (since ψ^B_b is a gradient itself, it should have perturbations of this form), which means that the modes are physical, and obey the constraints. The previously considered modes are not generally gradients, and therefore, cannot be guaranteed to be allowed perturbations of ψ^A_i ; this means that they are not physical, and all the physical modes are of the form of Eq. 3.257.

Next, we substitute our equation for α^{BD} from the second class of solutions into the second equation of the decomposition (Eq. 3.252). This gives

$$(-\lambda^2 \mu_{AB} + \bar{U}_{ACBDE}{}^C e^D) \alpha^B = 0, \quad (3.260)$$

or

$$(-\lambda^2 \mu_{AB} + U_{ACBDE}{}^C e^D) \alpha^B = 0, \quad (3.261)$$

because the constraint addition drops out, because we know that $\Lambda_{ACBD} = \Lambda_{BDAC} = -\Lambda_{ADBC}$, so

$$\Lambda_{ACBDE}{}^C e^D = -\Lambda_{ADBC}{}^C e^D = -\Lambda_{ACBDE}{}^D e^C, \quad (3.262)$$

where in the last step we have relabeled indices. From this, we see that this term must be 0. We can write this as

$$\Delta_{AB} \alpha^B = 0, \quad (3.263)$$

where

$$\Delta_{AB} := M^{ab}{}_{AB} k_a k_b. \quad (3.264)$$

However, this is precisely the condition for k_a to be a characteristic covector of the second order system with the characteristic variable α^A , so the modes of the first order system correspond 1-1 to the modes of the second order system. Hence, there will be 6 modes that consist of 3 pairs with speeds, $\pm\lambda$.

3.4.5 Characteristics of the second-order system in the un-sheared state

We now look at finding the characteristics of the second order system. We start by looking at the preceding solutions in more detailed. To illuminate these solutions, we examine them in the unsheared state. We can write the unsheared-state solutions as

$$\dot{\Delta}_{AB} = (-A\lambda^2 + B)g_{AB} + C e_A e_B, \quad (3.265)$$

where

$$A = e + p, \quad (3.266)$$

$$B = p + r, \quad (3.267)$$

$$C = 2p + r + q. \quad (3.268)$$

We can now read off the characteristic covectors and variables by inspection. Disturbances to the solutions travel along $n^a + \lambda e^a$ (if the characteristic direction is parameterized as $k_a = \lambda n_a + e_a$); in other words, they travel in the direction of e^a with speed λ as measured by an observer in the n^a reference frame. (We have chosen $n^a = u^a$.) This means that transversal waves, which have perturbations in directions normal to the direction of propagation of the wave, have eigenvectors obeying $\alpha^B e_B = 0$, so $\dot{\Delta}_{AB} \alpha^B = 0$ implies

$$\lambda^2 = \frac{B}{A} = \frac{2f_1 + 8f_2}{1 + \epsilon + p/n} =: \lambda_T^2. \quad (3.269)$$

The longitudinal waves produce disturbances in the direction of propagation of the wave, and therefore have eigenvalues such that the characteristic variable, α^B is proportional to e^B , so $\dot{\Delta}_{AB} \alpha^B = 0$ gives

$$\lambda^2 = \frac{B + C}{A} = \frac{c_s^2}{1 + \epsilon + p/n} + \frac{4}{3} \lambda_T^2 =: \lambda_L^2. \quad (3.270)$$

The relationship between the shear modulus and the speed of transversal waves (or shear speed) in a material is

$$\mu = n \lambda_T^2. \quad (3.271)$$

From this, it is clear that, in the Newtonian limit, the shear modulus becomes

$$\mu = 2n(f_1 + 4f_2). \quad (3.272)$$

We can see above that we only get one set of transverse wave speeds where we would ordinarily expect to get two sets, corresponding to two left-traveling and two right-traveling transverse waves (one for each independent transverse direction). This degeneracy arises from assumed isotropy in the crystal; in the unsheared state, there is no way to distinguish between the two transverse directions, and therefore, these two waves should propagate at the same velocity. This is related to the fact that the two generalized forces, f_1 and f_2 , only appear as the linear combination $(f_1 + 4f_2)$ in the expressions describing the unsheared state. In the Newtonian limit, this linear combination is proportional to the shear modulus. In the regime of linear perturbations around the unsheared state, we can only measure the value of this linear combination (the shear modulus); it is impossible to isolate the values of f_1 and f_2 . This corresponds to the regime where linear (or Hookean) elasticity holds, and stress is proportional to strain. For larger perturbations away from the unsheared state, where we expect larger anisotropy in the stress, we see

the non-linear effects of hyperelasticity, and the two transverse waves separate into four. Observing systems in the hyperelastic regime enables us to measure f_1 and f_2 independently.

Likewise, the longitudinal wave speed should be related to the bulk modulus by

$$K = n\lambda_L^2, \quad (3.273)$$

and in the Newtonian limit, the longitudinal wave speed reduces to the sound speed:

$$K = nc_s^2. \quad (3.274)$$

These are in units where $c = 1$, and n represents the rest-mass density.

3.4.6 Characteristics of the second-order system in general

If we keep the same decomposition of our characteristic direction covector, $k_a = \lambda u_a - e_a$, then we can write Δ_{AB} in the following form:

$$\Delta(\lambda) = \lambda^2 \Delta_2 + \Delta_0, \quad (3.275)$$

where the indices have been left out for clarity. This is the form of Eq. 3.260. However, solving this by computer algebra does not give an illuminating result. For numerical algorithms in the code, it will be useful to have the characteristic speeds in the coordinate frame ($\lambda = dx^i/dt$), so we parameterize $k_a = \lambda(dt)_a - (dx^i)_a$. (Note: this is a different λ from that in the previous decomposition.) Then our characteristic equation (eliminating the indices) has the form

$$\Delta\alpha = (\lambda^2 \Delta_2 + \lambda \Delta_1 + \Delta_0)\alpha = 0, \quad (3.276)$$

where λ is no longer related to the speed relative to the matter in a simple way, because of the Lorentz factor in relativistic velocity addition. We solve this system numerically by using a linear algebra package to find the eigenvalues of

$$\begin{pmatrix} -(\Delta_2)^{-1}\Delta_1 & -(\Delta_2)^{-1}\Delta_0 \\ I & 0 \end{pmatrix}. \quad (3.277)$$

The above is implemented in the `ElasticEvolution` code, and can be used to set the maximum characteristic speeds for the HLL flux as well as the dynamic time step calculations. However, in practice, this process is slow; any advantages gained by having more accurate characteristic speeds are outweighed by the additional time taken to solve this system at every grid point and time-step. Instead, in the

`ElasticEvolution` code, we use an input parameter to set an overall maximum value for the maximum characteristic speed. In the `MultiModel` code, we estimate λ_{\max} by calculating the maximum characteristic speed in the unsheared state and then multiplying by some fudge factor (an input parameter).

Typically, this fudge factor is set to ~ 2 , but for most tests, the results are not very sensitive to the choice of this parameter. The maximum characteristic speed is used in the `MultiModel` code to both dynamically calculate the time step size and to calculate the HLL flux. The time step size calculation sets the lower limit on this fudge factor: the maximum characteristic speed should not be underestimated such that the CFL condition is violated, causing numerical instabilities. In the HLL flux, the λ_{\max} term acts like a dissipative term with a coefficient $\lambda_{\max}\Delta x$ (Δx is the grid spacing), so setting the fudge factor too large can cause unnecessary smearing of sharp features in the solution, and possible overdamping instabilities. Fine tuning of this parameter is typically achieved by trial and error.

3.4.7 Hyperbolicity without constraint addition

In the Newtonian literature, the evolution equations for ψ^A_i use the constraint addition, but the constraint addition for $\nabla_b T^{ab} = 0$ is not included. Generally, we see that this system will not be symmetric hyperbolic: without the constraint addition, in the unsheared state, $\mathring{\bar{U}}_{ACBD}$ becomes

$$\mathring{\bar{U}}_{ACBD} = (p+q)g_{AC}g_{BD} + (p+r)g_{AB}g_{CD} + (p+r)g_{AD}g_{BD}, \quad (3.278)$$

and while $p+r = 2n(f_1 + 4f_2)$ is positive definite, $p+q = nc_s^2 - \frac{4}{3}n(f_1 + 4f_2)$ is not. We also see that, in general, \bar{U}_{ACBD} will no longer be symmetric, since it becomes

$$\bar{U}_{ACBD} = U_{ACBD} - 2g_{A[C}\mu_{D]B}, \quad (3.279)$$

and the second term here is not symmetric in $AC \leftrightarrow BD$.

It is clear that the constraint addition is necessary in order for the system to be symmetric hyperbolic. However, we would like to see whether the system may still be strongly hyperbolic without the constraint addition to $\nabla_a T^{ab} = 0$. For strong hyperbolicity, we require that the system admits a complete set of characteristic variables; we now examine our previous calculation to see how it changes if we do not assume symmetry of the object, \bar{U}_{ACBD} .

We first see that, in finding the characteristic variables, we can no longer go from $\bar{U}_{ACBD}(\alpha^B e^D + \lambda\alpha^{BD}) = 0$ to $\alpha^B e^D + \lambda\alpha^{BD} = 0$ (Eq. 3.251 to Eq. 3.253), because \bar{U}_{ACBD} will no longer generally be invertible. However, all solutions to the

latter equation will be solutions to the former equation. If \bar{U}_{ACBD} is not invertible, then we will simply find *more* solutions to $\bar{U}_{ACBDE}e^C\alpha^{BD} = 0$ (Eq. 3.254). We will still find the same six unphysical modes that we found before, although we may find extra modes, and we will not generally be able to parameterize them as in Eq. 3.255.

Our three physical modes are unaffected by the constraint addition, because we find them using $(-\lambda^2\mu_{AB} + U_{ACBDE}e^C e^D)\alpha^B = 0$ (Eq. 3.261). Physically, this is what we should expect; the constraint addition should make no physical difference to the system, and therefore should not affect the physical modes of the solutions. We still find the same three physical modes that we found with the constraint addition.

In summary, we see that without the constraint addition, we still find all of the same modes that we found with the constraint addition; the difference is that we may find some extra unphysical modes. Since our characteristic variables with the constraint addition formed a complete set, they will still form a complete set without the constraint addition, meaning that the system without the constraint addition to $\nabla_a T^{ab} = 0$ will still be strongly hyperbolic.

3.5 Numerical elasticity code

3.5.1 Description of the ElasticEvolution code

All of the variables in our computer code are 3-dimensional, but they are assumed to depend only on one or two spatial coordinates. The numerical methods employed are those in [60]. Briefly, the code uses a HRSC method with a third-order Runge-Kutta time evolution. In the reconstruction, standard slope limiting techniques, applied to the primitive variables are used—all results shown used van Leer’s MC limiter ([25]). The HLL approximate Riemann solver ([26]) is used to calculate the fluxes. The code can be run using either the relativistic or Newtonian set of governing equations.

The HLL flux is

$$\mathbf{f}_{i-\frac{1}{2}} = \frac{\mathbf{f}(\mathbf{q}_{i-1}^R) + \mathbf{f}(\mathbf{q}_i^L) + \bar{\lambda}_{\text{HLL}} (\mathbf{q}_{i-1}^R - \mathbf{q}_i^L)}{2}, \quad (3.280)$$

where \mathbf{q}_{i-1}^R and \mathbf{q}_i^L are the right and left reconstructed vectors of conserved variables for the $(i-1)^{\text{th}}$ and i^{th} cells, respectively, and $\bar{\lambda}_{\text{HLL}}$ is an estimate of the absolute value of the largest coordinate characteristic speed. (Compare this to Section 2.8, and note that \mathbf{q}_{i-1}^R here refers to the right edge of the cell to the left,

which is referred to as Q_L in that section.) We set this either to $\max |\lambda|$ at one point (using the numerical calculation outlined in Section 3.4.6), to $\max |\lambda|$ over the whole grid, or to a constant. These three possibilities are in order of increasing efficiency and increasing dissipation but all can be made to work (possibly multiplying $\max |\lambda|$ by a constant fudge factor to get $\bar{\lambda}_{\text{HLL}}$). Setting $\max |\lambda|$ over the whole grid allows us, in particular, to not update the calculation of the characteristic speeds within a complete Runge-Kutta step in the method of lines. Setting $\bar{\lambda}_{\text{HLL}} = 1$ globally is sensible in highly relativistic situations (assuming the matter evolution is causal).

In two dimensions standard directional splitting techniques are used. Specifically, on our (logically) Cartesian grid we compute the appropriate one dimensional fluxes \mathcal{F}^i required by Eq. (3.129) by sweeping through the grid lines one dimension at a time. The update terms are accumulated and applied simultaneously to minimize symmetry errors caused by the splitting.

We briefly note that the performance of the code has been compared to a relativistic hydrodynamics code by reducing the elasticity code explicitly to the hydrodynamic limit. As none of our codes have been optimized for performance, any comparisons will be approximate. Nevertheless, as the elasticity code is approximately 8 times slower than the hydrodynamic code on the same problem. We believe this is partly due to the greater number of evolved variables (19 instead of 5) and partly due to the fact that the conserved-to-primitive conversion requires a 4-dimensional instead of a 1-dimensional root find, and so there is no obvious way of reducing this ratio. (Adding spacetime evolution and adaptive mesh refinement to both hydro and elasticity would obviously reduce the overall ratio).

3.5.2 n_D versus n_ψ

As discussed in Section 3.2.3, the particle number density n can either be obtained from the conserved variable $\psi^A{}_i$, or from the conserved variable D . If we evolve D as a dynamical variable and use n_D to represent the primitive variable, we have one more variable than if we use n_ψ . However, we have shown in Section 3.2.3 the evolution equations for n_D and n_ψ are equivalent even if the constraints are not obeyed, and so we expect that both formulations have identical stability properties.

In fact, when these two evolutions are compared, the RMS relative error in the resulting data is small; we expect that this is finite-differencing error, as it converges away between first and second order. However, when n_ψ and n_D are compared for a single evolution where n_D is dynamical, the difference is of the

order of round-off error rather than finite-differencing error; we suspect that this is an artifact of planar symmetry in those tests.

3.5.3 ψ versus F

A mixed framework using the inverse $F^A{}_i$ of the configuration gradient $\psi^i{}_A$ is outlined in [1]. For constraint satisfying initial data the results of the two frameworks should be the same. We have implemented both frameworks numerically and compared them. We find that the difference is on the order of the finite differencing error for constraint satisfying initial data, as expected.

Some of the tests given in [61] and [62] do not satisfy the constraints and so are unphysical—namely the second test in [61] and the fifth test in [62]. As the evolution of such data depends on the choice of constraint addition to the equations, we would expect it to depend on the framework used. Our numerical results obtained in the “mixed” framework, which is that used in [61, 62], matches their numerical results for all tests. Our results using the Eulerian framework (presented in Section 3.2 and also used by [55, 63]) match only for the physical tests, where the initial data obey all constraints.

3.5.4 Newtonian Riemann tests

To validate our Newtonian code, and the Newtonian limit of our relativistic code, we have compared our results to two previously published studies ([61] and [62]). These results use the Newtonian theory and the mixed framework outlined in [1].

Broadly the results obtained from our codes matched those shown in [61] and [62]. As an example, we show the results for the first test of [61] in Figs. 3.7–3.8, using the results of the Newtonian code. The precise initial data used is outlined in Section 3.5.12. We see the seven waves expected for this solution; three left traveling rarefactions (the second is very small), a contact, two right traveling rarefactions (again the second is very small), and a fast shock. For clarity, the wave structure of the exact solution is shown in Fig. 3.6. All waves are captured with only minor under and overshoots, and the numerical solutions converge to the exact solution [64] with resolution, as seen by comparing Fig. 3.7 with Fig. 3.8.

Similar results are seen for all comparison tests. However, not all of the tests run robustly for all numerical methods possible within our code. An example is the sonic point test problem outlined in [62] (see in particular Figs. 5–7 there). At the contact discontinuity there is an unphysical “dip” in the density and a corresponding “jump” in the internal energy. This is the classical “wall-heating” effect seen by most numerical methods when strong rarefactions separate (e.g.,

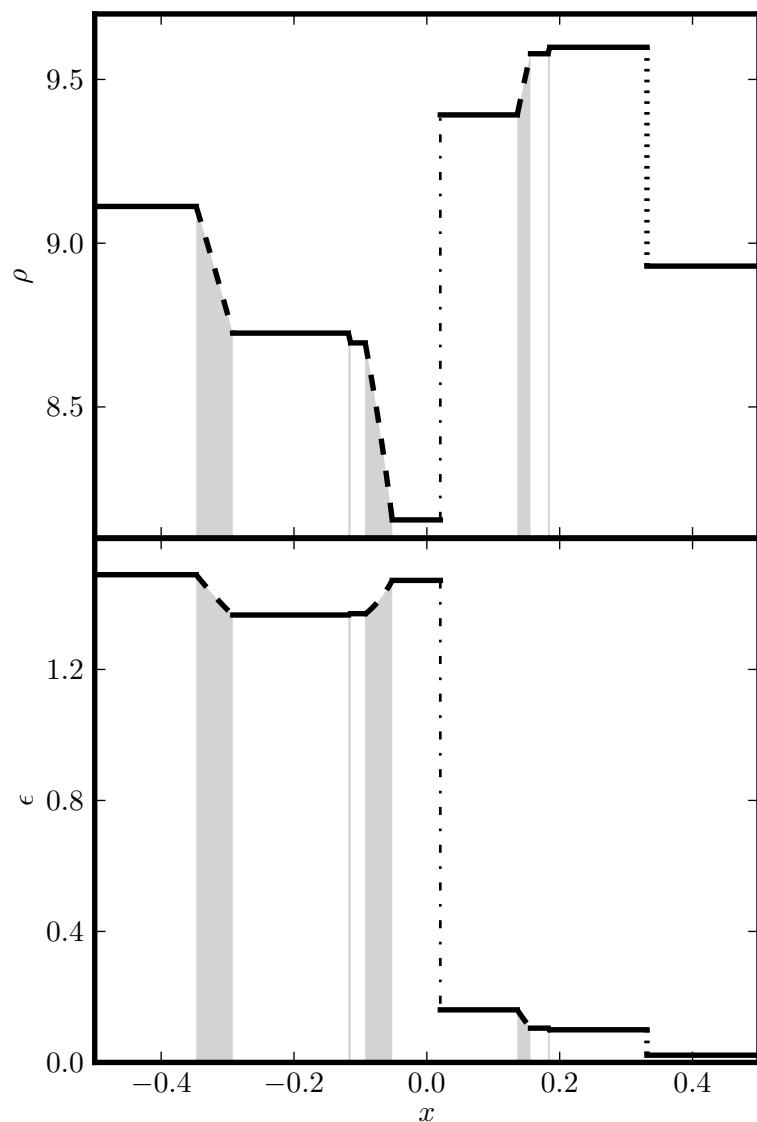


Figure 3.6: The density and specific internal energy in the first Newtonian Riemann test from [61] (from now on BDRT1), illustrating the seven waves possible in elastic matter. This is the exact solution, illustrating the wave structure in detail. The rarefactions—the 1, 2, 3, 5 and 6-waves—are given by the dashed lines and are shaded beneath to show the width of the fan. The contact—the linear 4-wave—is given by the dash-dotted line. The shock—the 7-wave—is given by the dotted line. It is clear that resolving some of the rarefaction waves will be difficult at moderate resolution.

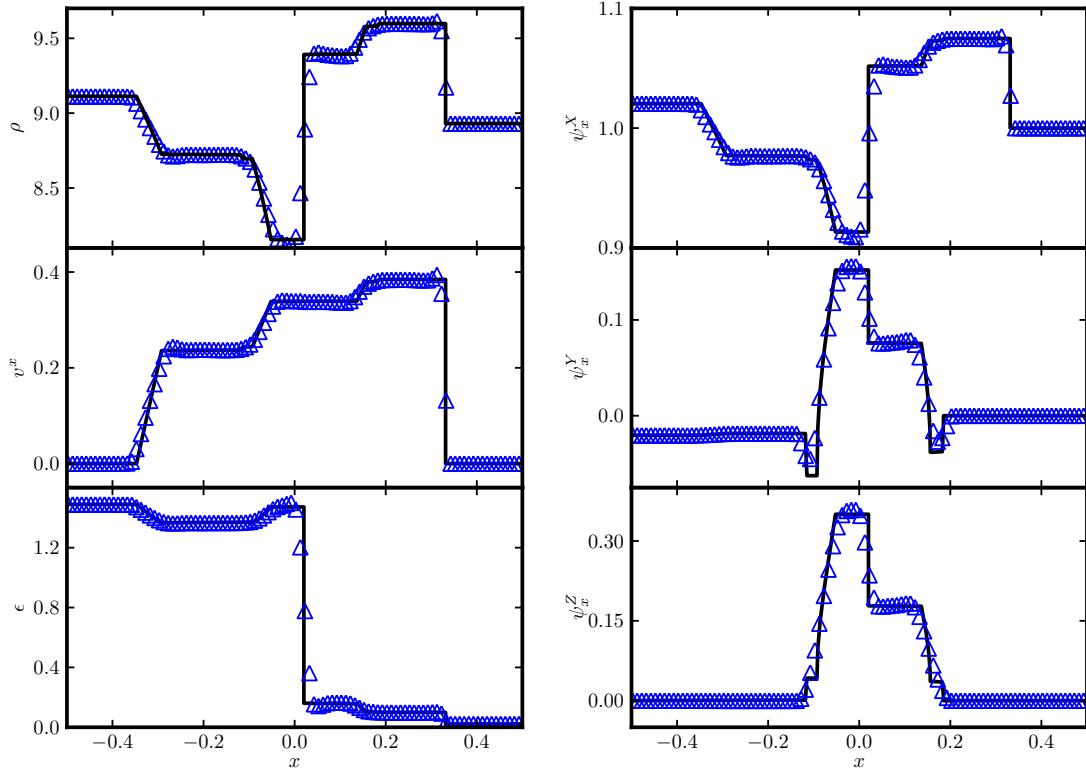


Figure 3.7: Numerical solution of the BDRT1 test at coordinate time $t = 0.06$. We show the results of our Newtonian code using 200 points (only 100 are plotted for clarity), with the exact solution given by the solid line. Density, specific internal energy and normal velocity are shown in the left panel and components of the configuration gradient in the right. Only minor under and overshoots are visible.

on reflections from walls or the origin in spherical symmetry—see [65] for the classical case and [66] for a brief discussion of the relativistic case). This error can lead to unphysical states, and particular care is required with the choice of the Courant factor and $|\lambda|_{\text{HLL}}$. The code fails if c_s^2 or n become (unphysically) negative (although the variant that sets $|\lambda|_{\text{HLL}}$ a priori can limp on for a while with negative c_s^2). (Generally, there are two constraints on the Courant parameter and the value of λ_{HLL} . The CFL (necessary) stability criterion requires $|\lambda|_{\text{max,global}} < \Delta x / \Delta t$ while, depending on the time discretization, $\lambda_{\text{HLL}} < O(1) \Delta x / \Delta t$ to avoid an overdamping instability.)

Finally, we note that a direct and comprehensive comparison to the results of [62] is complicated by two issues. Firstly the units for the entropy appear inconsistent there, as detailed in Section 3.5.12. Secondly we do not find agreement in the comparison of the pressure tensor p_{ij} (denoted σ there). As all other values and wave speeds match up well, and we have comprehensive quantitative agreement with the results of [61], we believe our results to be correct.

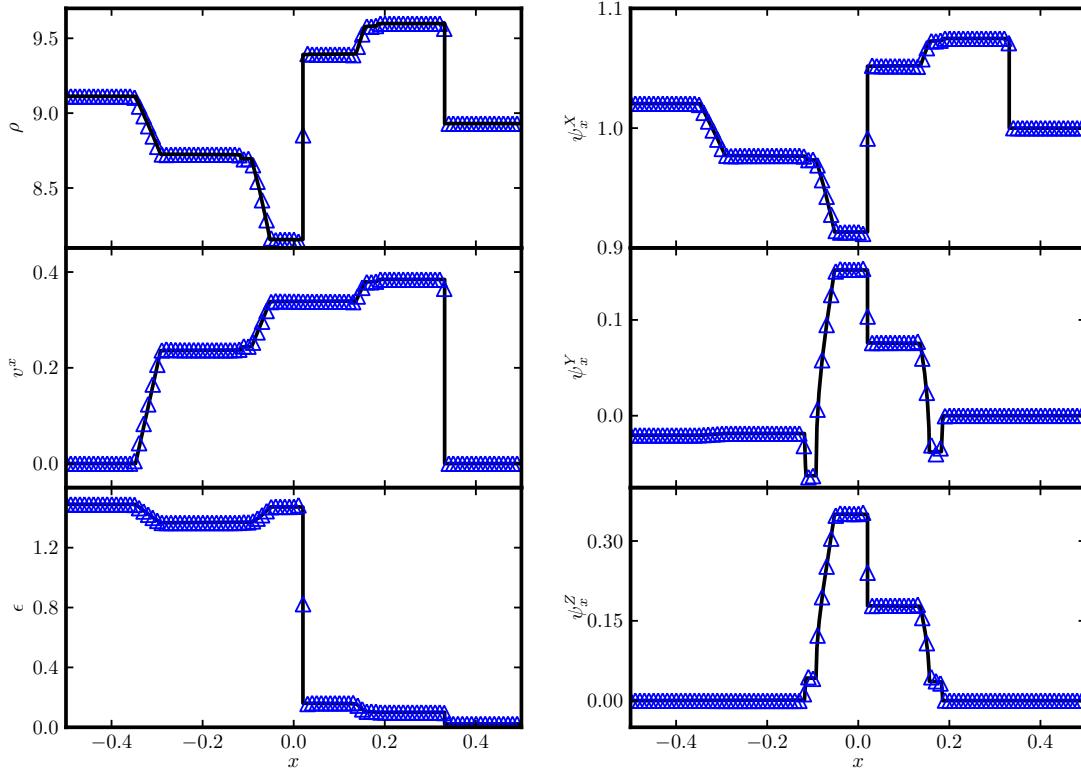


Figure 3.8: Numerical solution of the BDRT1 test at coordinate time $t = 0.06$. We show the results of our Newtonian code using 1000 points (only 100 are plotted for clarity), with the exact solution given by the solid line. Density, specific internal energy and normal velocity are shown in the left panel and components of the configuration gradient in the right. Only minor under and overshoots are visible. Comparing against the results in Fig. 3.7 we see convergence to the correct weak solution.

3.5.5 Newtonian limit vs Newtonian code

The code (both relativistic and Newtonian) uses geometric units where the speed of light is one. In particular, all velocities are of the form $v = \bar{v}/\bar{c}$ where v is a dimensionless velocity, \bar{v} its value in conventional units and \bar{c} the value of the speed of light in the same units. All parameters in the equation of state, such as ϵ and c_s^2 , are treated analogously. There is no need to rescale rest mass and length, as long as units are used consistently.

Changing \bar{c} while keeping \bar{v} etc. fixed is a trivial scale invariance of the Newtonian equations and their solution, but in the relativistic equations decreasing \bar{c} with \bar{v} etc. fixed makes the same test problem more relativistic. We can use this to obtain an insight into the effects of (special) relativity, and to verify that our relativistic code has the correct Newtonian limit as $\bar{c} \rightarrow \infty$.

In Fig. 3.9 we show the results from the relativistic code run with a small range

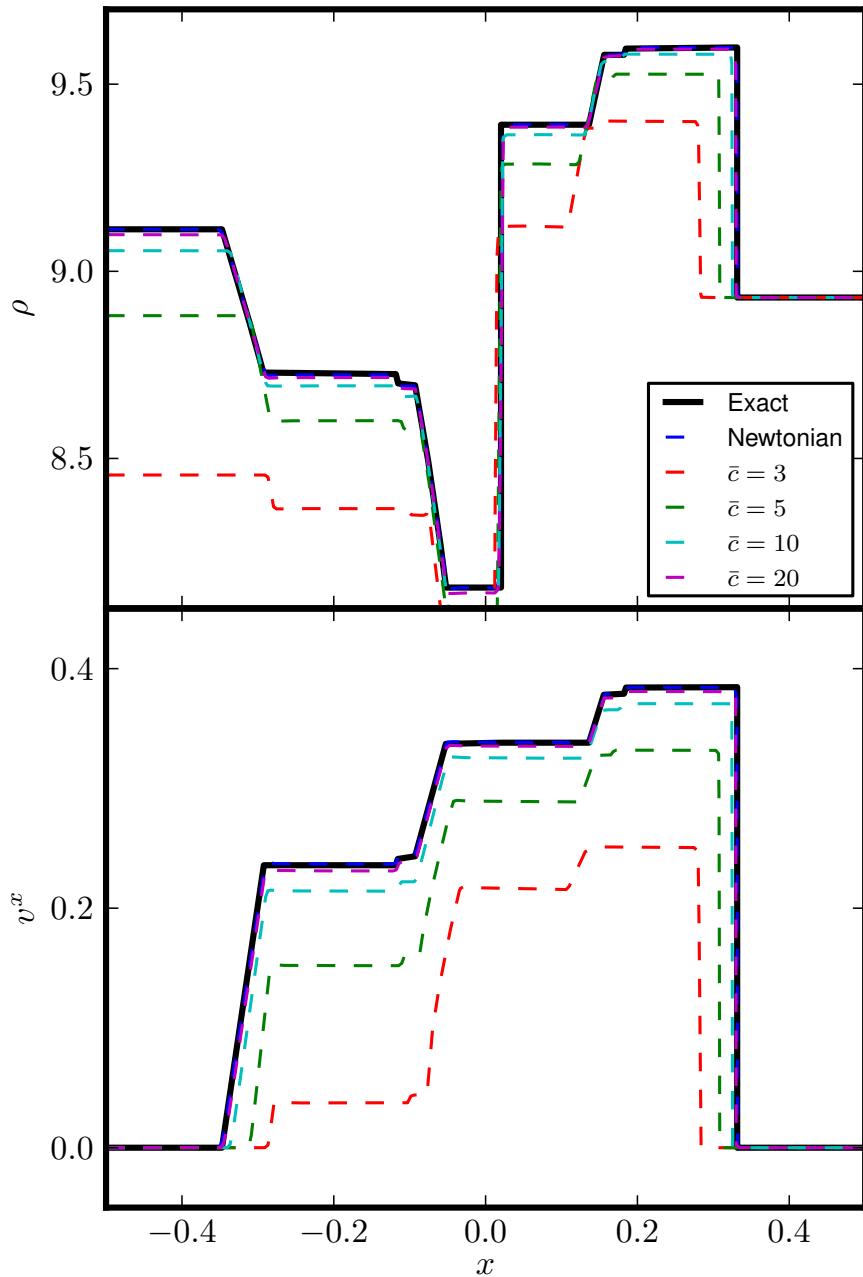


Figure 3.9: BDRT1 again, at coordinate time $t = 0.06$, but now run using the *relativistic* code with various values of \bar{c} , and compared against the Newtonian results. The results for ρ, v^x (appropriately scaled by \bar{c}) are representative of the behavior of all quantities. We see that as \bar{c} increases the Newtonian limit is approached. 10000 points were used in each case to ensure that we are at the continuum limit within the resolution of this plot.

of values for \bar{c} . Only relatively small values of \bar{c} are shown—3, 5, 10 and 20 km s^{-1} , compared to a typical velocity in the (Newtonian) Riemann problem of 1 km s^{-1} —as for sufficiently large values of \bar{c} the results are visually indistinguishable. We see that the results from the relativistic code are qualitatively similar, in terms of wave structure and accuracy, and approach the Newtonian results in the limit $\bar{c} \rightarrow \infty$.

3.5.6 Relativistic Riemann tests

In the genuinely relativistic limit we have tested our code against exact solutions constructed by solving a pre-determined wave structure. The explicit procedure is detailed in [1] and follows the method used in the Newtonian case outlined in [61], without constructing a full Riemann problem solver.

We have verified that the code behaves correctly for single shocks and rarefactions in the relativistic limit, and for some invented initial data sets that test a range of wave structures. As an example, we show in Figs. 3.11–3.12 the results for a four wave problem. For clarity, the wave structure of the exact solution is shown in Fig. 3.10. There are two left-going rarefactions (1 and 2-waves), one right-going rarefaction (a 6-wave) and a right going shock (7-wave). The central three waves—the nonlinear 3 and 5-waves and the contact—are all trivial. We note that some of the quantities change so rapidly across some rarefaction waves that they are only visually distinguishable from shocks at high magnification.

Even with the violent behavior displayed across some waves in this four wave test, we find our code matching the exact solution well, with no unphysical oscillations and only minor under and overshoots that converge away with resolution. There are the expected minor oscillations near the trivial waves, most noticeable near the contact, but again these converge with resolution.

3.5.7 A method for Riemann tests on a 2D grid

As a first test of the role of the constraints in hyperbolicity, we numerically solve Riemann problems on a 2-dimensional grid, with the initial discontinuity at an angle to the grid. Assume the grid consists of $n_x \times n_y$ cells, surrounded by the necessary number of ghost cells. After each time update, the ghost points are filled using periodic boundary conditions, identifying cell (i, j) with $(i + n_x, j)$ in the x direction, but (i, j) with $(i + \delta_x, j + n_y)$ in the y direction, where δ_x is an offset. Consistently with these boundary conditions, the initial discontinuity is then placed on a line of $x/y = \delta_x/n_y$ (assuming that the grid spacing is the same in the x and y directions). This is illustrated in Fig. 3.13.

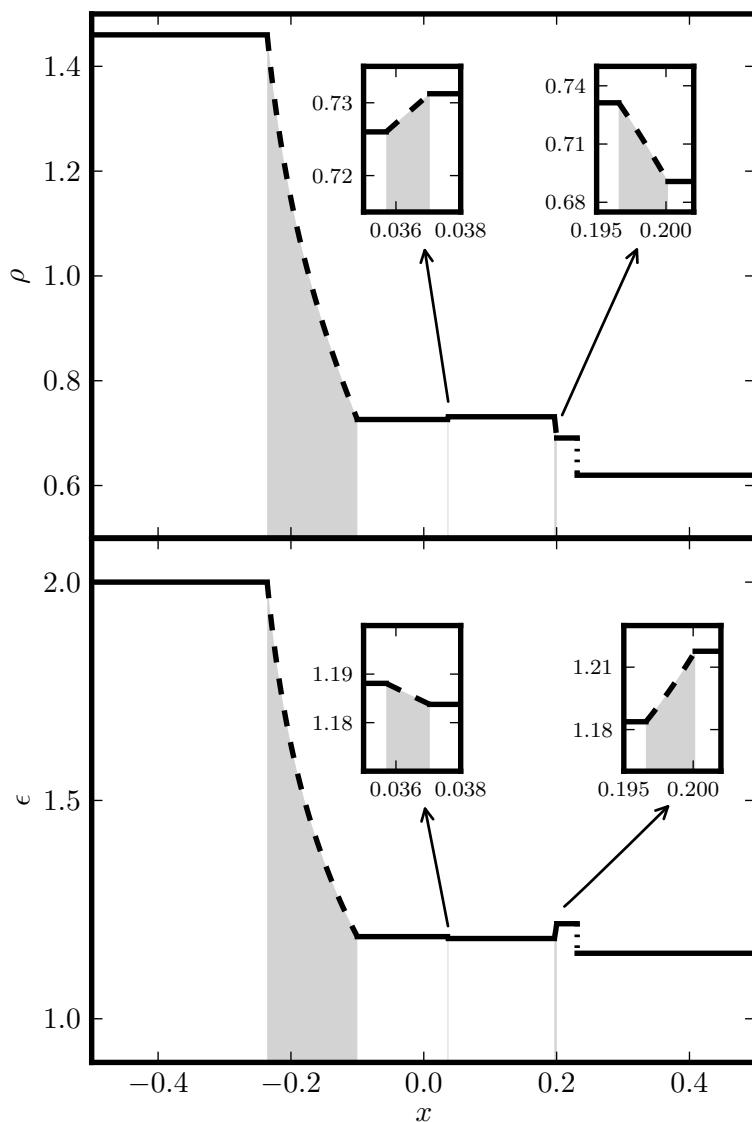


Figure 3.10: The density and specific internal energy ϵ for the relativistic 4-wave test at coordinate time $t = 0.25$. This is the exact solution, illustrating the wave structure in detail. The rarefactions—the 1, 2, and 6-waves—are given by the dashed lines and are shaded beneath to show the width of the fan. The very narrow 2 and 6-waves are shown in detail in the insets. The contact—the linear 4-wave—is trivial. The shock—the 7-wave—is given by the dotted line. It is clear that resolving some of the rarefaction waves will be difficult at moderate resolution.

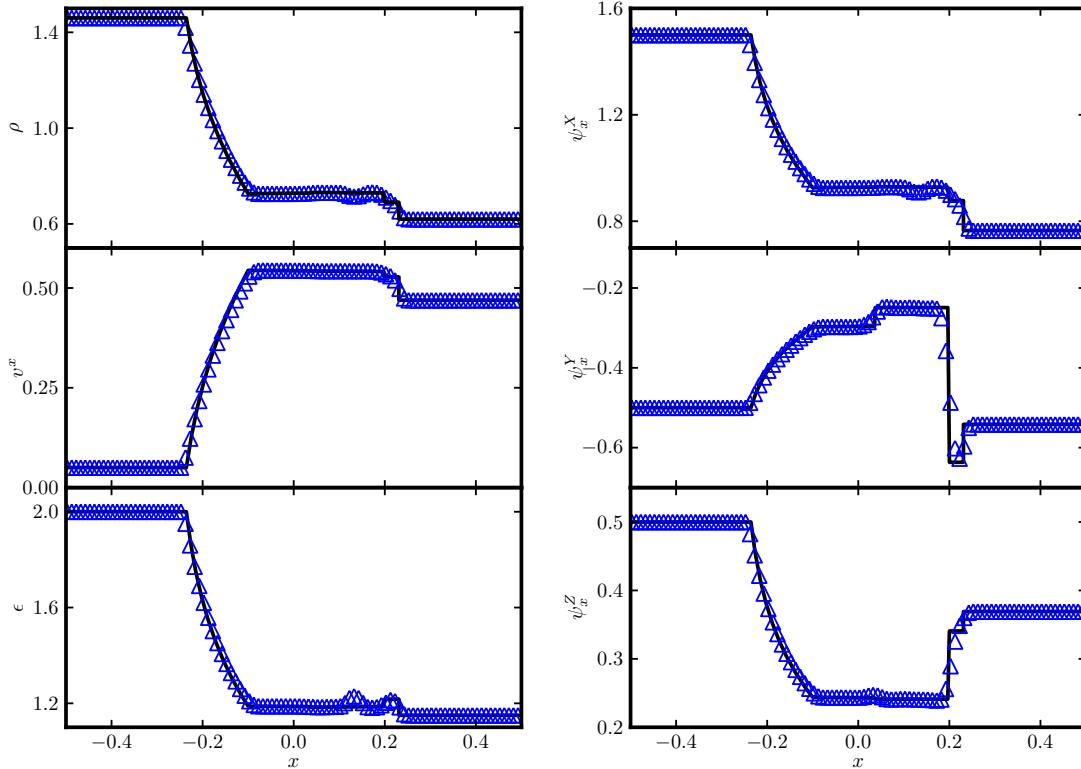


Figure 3.11: Numerical solution of the relativistic 4-wave test at coordinate time $t = 0.25$. Density, specific internal energy and normal velocity are shown in the left panel, and components of the configuration gradient in the right. The 4 wave structure (two left-going rarefactions, one right-going rarefaction and one right going shock) is most clearly seen in the plot of ψ_x^Y . The solution is computed using 200 points but only 100 are plotted for clarity. We see that all waves are captured well and with only minor under and overshoots, most visible for the second rarefaction wave in quantities such as ϵ .

As the x and y directions are interchangeable, the slope δ_x/n_y and its inverse pose the same Riemann test. (Less obviously, in our implementation those two tests also have roughly equal computational cost.) We choose $n_y \geq \delta_x$ (and typically $\delta_x = 1$) so that the initial discontinuity is always closer to the y axis (where it is in the 1D tests), and use the x axis as an approximation to a line normal to the initial discontinuity when taking a cut through the solution.

3.5.8 Two-dimensional Riemann tests

The constraints are trivial if all variables depend only on one coordinate, for example when a Riemann problem is aligned with a Cartesian grid. As a first test of the behavior in three dimensions and the role of the constraints, we have solved Riemann problems also at an arbitrary angle to a two-dimensional Cartesian

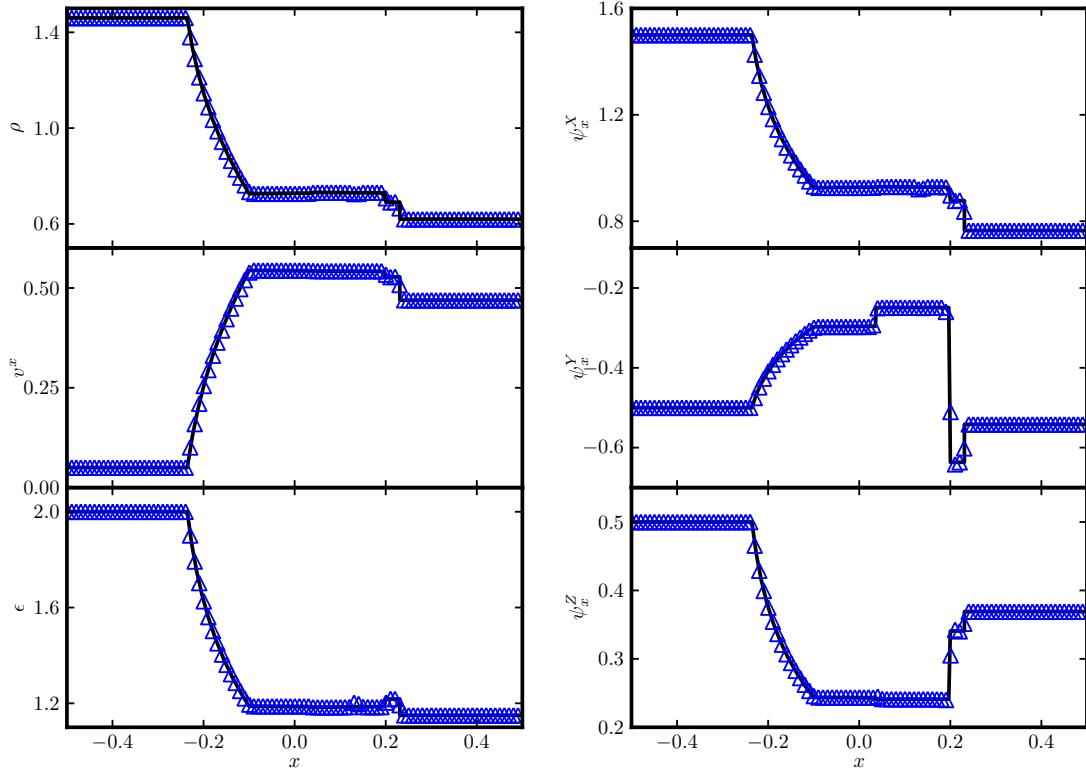


Figure 3.12: Numerical solution of the relativistic 4-wave test at coordinate time $t = 0.25$. Density, specific internal energy and normal velocity are shown in the left panel, and components of the configuration gradient in the right. The solution is computed using 1000 points but only 100 are plotted for clarity. We see that all waves are captured well and with only minor under and overshoots, and comparing to Fig. 3.11 we see the expected convergence.

grid. A method for carrying out such 2D simulations efficiently is described in Section 3.5.7.

We put the initial discontinuity along lines $x/y = 0$ (our 1D tests), 1, $1/2$ and $1/5$, and use a cut along the x axis as an approximation to a line normal to the initial discontinuity. We compare this cut, suitably foreshortened, against the exact solution, in Fig. 3.14, with results similar to our 1D tests.

We have not implemented the “hyperbolicity fix” constraint additions for either the kinematic or dynamical evolution equations. In 1D the equations are symmetric hyperbolic anyway, as there are no constraints then, but in 2D our equations are not even strongly hyperbolic. Nevertheless, there is no sign of numerical instability in 2D. We have no explanation for this unexpected stability, but expect that constraint addition will be necessary in other tests in the future.

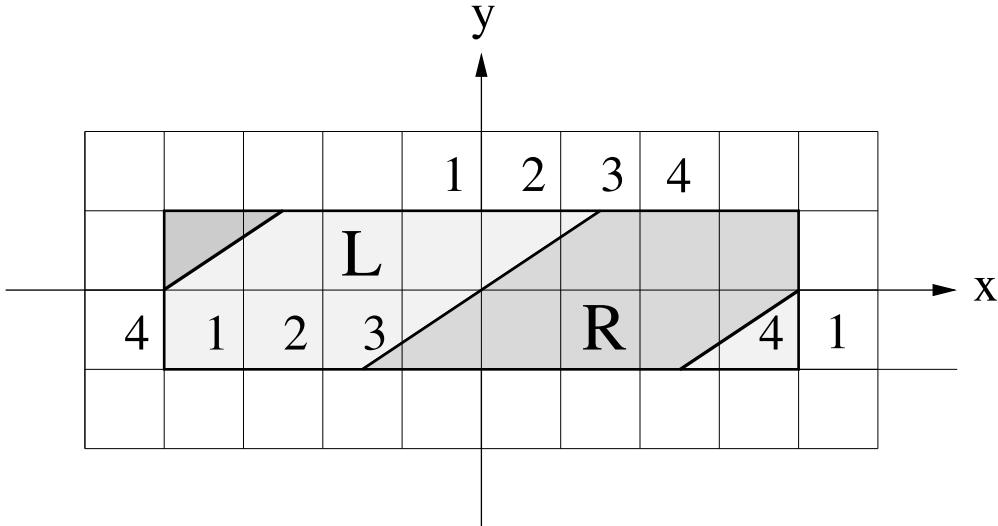


Figure 3.13: Example of a two-dimensional grid with shifted periodic boundary conditions, with $n_x = 8$, $n_y = 2$ and $\delta_x = 3$. The physical grid is surrounded by one ghost cell on each side. In reality, n_x would be much larger, while n_y ranges from 1 to a few, and δ_x from 0 to a few, with no common factor. The placement of the left and right state of a Riemann problem is shown by the letters L and R and shading. Cells are initialized with the left or right state depending on the position of the cell *center*. The numbers 1, 2, 3, 4 identify four physical cells and the ghost cells they donate values to. The initial discontinuity is at an angle α from the y axis, with $\tan \alpha = \delta_x/n_y$ ($= 3/2$ in this example), and goes through the point $x = y = 0$.

3.5.9 Two-dimensional rotor tests

To study a problem with nontrivial dependence on two Cartesian coordinates we consider a test suggested by [67]. The initial data, detailed in Section 3.5.12, represents an elastic rotor problem, where an inner rotating bearing is instantaneously welded to the non-rotating exterior, causing the rotor to slow and propagating elastic waves through the material. This is a cylindrically symmetric problem simulated in Cartesian coordinates. In all cases the rotor has coordinate radius 0.1, whilst the exterior is at rest. In all numerical experiments shown here 400^2 points were used.

Results for the Newtonian case are shown at representative coordinate times in Fig. 3.15. These should be compared to the results shown by Dumbser *et al* [67]. The results in the literature use a considerably more accurate numerical method, which is both higher order and uses finite elements better adapted to the symmetry of the problem. Despite this, we see qualitative agreement in the waves emitted during the evolution of the problem. Note that these plots looks slightly different from those presented in [1]; this due to an error in the plotting script.

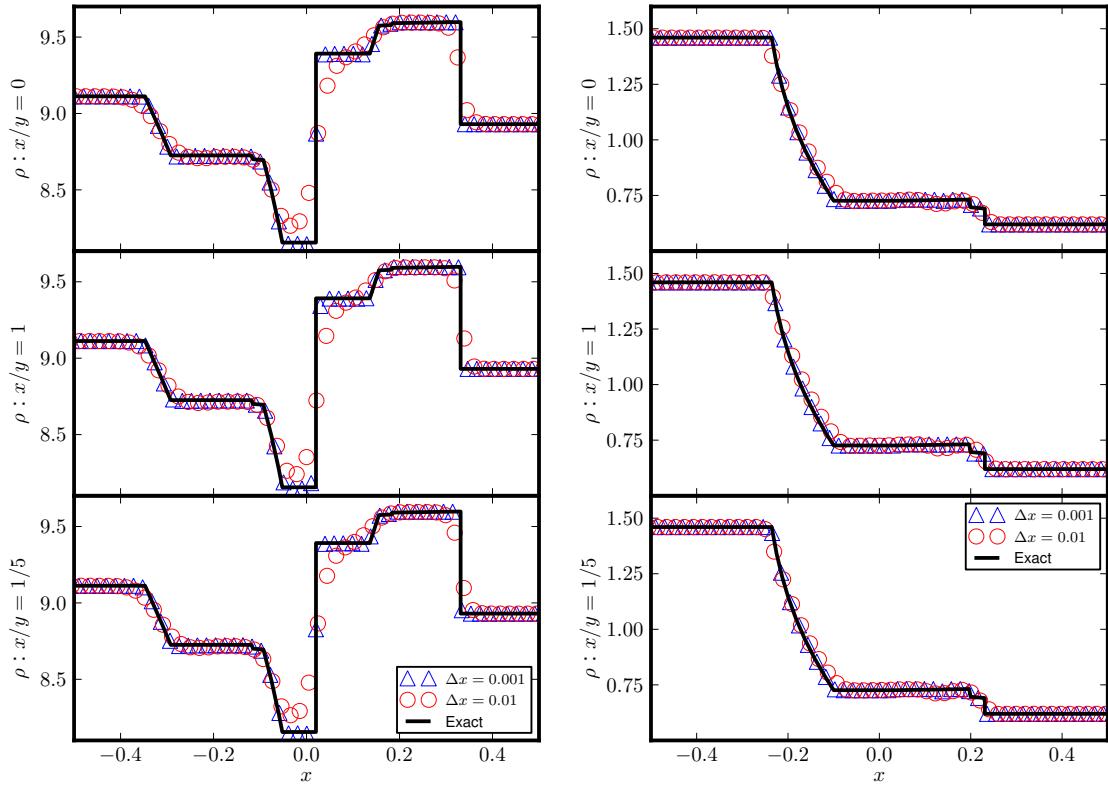


Figure 3.14: Results for the BDRT1 Riemann test at coordinate time $t = 0.06$ (left panel) and relativistic 4 wave test at coordinate time $t = 0.25$ (right panel), calculated on a two-dimensional grid for three different angles between the initial discontinuity and the grid, each at two resolutions. The initial discontinuity was placed on the line given alongside each plot. In order to compare the results to the exact solution, a slice through the two-dimensional grid is taken along the x axis (as an approximation to a line normal to the waves), and x is scaled to correspond to distance perpendicular to the initial discontinuity. The spatial resolution is independent of the angle of the initial discontinuity, and the snapshot is always taken at the same time, for all angles. The relativistic code is used in both cases (in the Newtonian limit for the BDRT test). The high-resolution results were produced using $\Delta x = \Delta y = 0.001$. (The effective resolution in the relevant direction, normal to the initial discontinuity, is lower—by a factor of $\sqrt{2}$ in the worst case, where the initial discontinuity is on the grid diagonal). The low-resolution version was produced using $\Delta x = \Delta y = 0.01$. (For clarity, only 1 in 2 or 1 in 20 points are plotted for the $x/y = 0$ and $x/y = 1/5$ cases, while 1 in 3 or 1 in 28 points are plotted for $x/y = 1$.) All evolutions look similar, with the results approaching the exact solution as the resolution is increased.

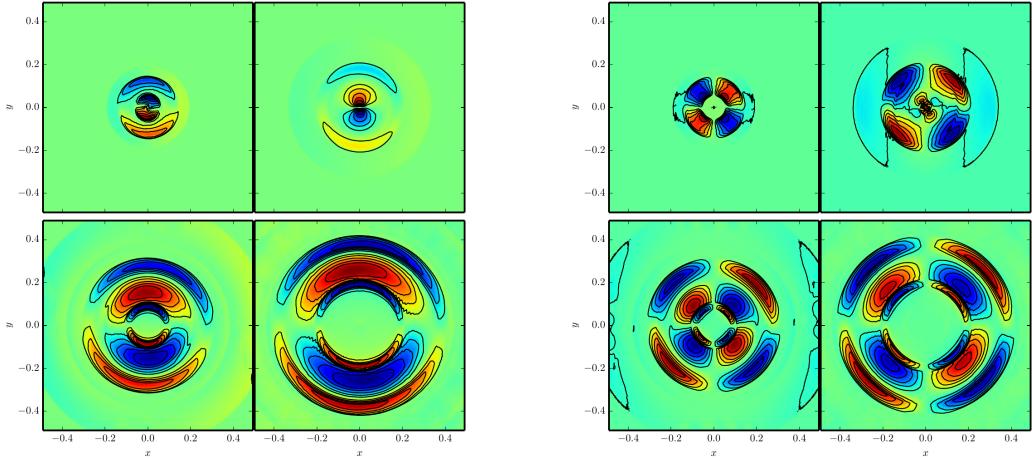


Figure 3.15: Newtonian rotor test, following [67]. An initially rotating central cylinder is slowed by the interaction with the exterior, which is initially at rest. The figures in the left panel show ρv^x whilst those in the right show ρF^y_Y , at coordinate times $t = 0.02, 0.05, 0.1$ and 0.15 . The results qualitatively match those in Fig. 24 of [67].

We can now see the expected symmetry:

$$F^y_Y(x, y) = F^y_Y(-x, -y)$$

and

$$v^x(x, y) = -v^x(-x, -y).$$

In the continuum, we also expect that

$$F^y_Y(x, y) = -F^y_Y(-x, y)$$

and

$$v^x(x, y) = v^x(-x, y);$$

however, due to the use of a square grid for these tests, we get a systematic asymmetry when we reflect in x and y . This asymmetry does not disappear with resolution.

Results for the relativistic case are shown at representative coordinate times in Fig. 3.16. Again we see qualitative agreement in the emitted wave structure, despite the differences in the models.

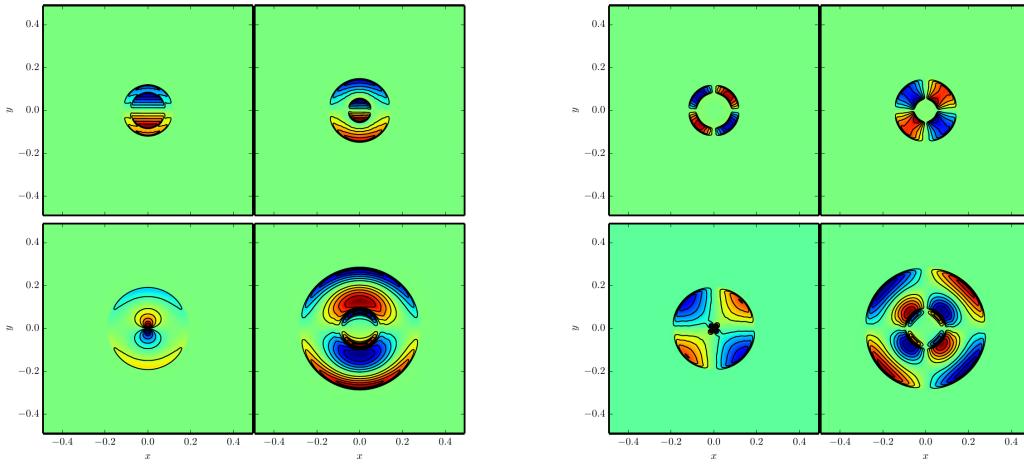


Figure 3.16: Relativistic rotor test, to be compared with the Newtonian results in Fig. 3.15. An initially rotating central cylinder is slowed by the interaction with the exterior, which is initially at rest. These figures in the left panel show ρv^x whilst those in the right show ρF^y_Y , at coordinate times $t = 0.02, 0.05, 0.1$ and 0.2 . The emitted waves are qualitatively similar to the Newtonian results.

3.5.10 Considerations for a two-dimensional cylindrical grid

Source terms One important difference when moving from Cartesian to cylindrical coordinates is that the source terms for the equations of motion derived from the stress-energy tensor become non-zero. When the cylindrical Minkowski metric, $\text{diag}(-1, r^2, 1, 1)$, is used, we find that the only non-zero source term is

$$\mathcal{S}(S_r) = r^2 T^{\theta\theta}. \quad (3.281)$$

Transforming initial data and output We start by setting the initial data using Cartesian coordinates; we then transform both the matter-space and space-time coordinates to cylindrical coordinates. We use the Jacobian matrix

$$\frac{\partial x^{a'}}{\partial x^b} = \begin{pmatrix} \frac{x}{r} & \frac{y}{r} & 0 \\ -\frac{y}{r^2} & \frac{x}{r^2} & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.282)$$

and its inverse to transform covariant and contravariant spatial indices, where $x^{a'} = (r, \theta, z)$ and $x^b = (x, y, z)$, and the usual transformation between (x, y) and (r, θ) is used:

$$x = r \cos \theta, \quad y = r \sin \theta. \quad (3.283)$$

The same transformation applies to the matter space indices, with $\xi^{A'} = (R, \Theta, Z)$ and $\xi^A = (X, Y, Z)$ in place of $x^{a'}$ and x^b . To do this for the initial data, we find the matter-space coordinates in terms of the spacetime coordinates by integrating $\psi^A{}_i$, which is, by definition, $\psi^A{}_i = \partial \xi^A / \partial x^i$. Where the initial data for $\psi^A{}_i$ is piecewise constant, as is the case for all our tests here, this integration just gives

$$\xi^A = \psi^A{}_i x^i + \text{const}, \quad (3.284)$$

with the constants chosen so that the ξ^A are continuous at the initial discontinuity. In order to compare the results in cylindrical coordinates with those in Cartesian coordinates, we need to transform v^i and $\psi^A{}_i$ between the two coordinate systems at later times. For this diagnostic purpose (only), we advect the (Cartesian) matter coordinates $\xi^A(x^i, t)$ with the fluid. $\xi^{A'}$ and ξ^A are then always linked by the standard formula (Eq. 3.283).

Limitations on the initial data In these tests, we have hidden the planar or axisymmetry by making the problem appear to depend nontrivially on two coordinates (r, θ) . However, we do not want to hide the z -translation symmetry in order to keep simulations two-dimensional. Now, if the Cartesian components, $\psi^A{}_z$, were non-zero, then the matter-space coordinates, ξ^A , would be dependent on z , making the transformation, as well as the cylindrical components of $\psi^A{}_i$, z -dependent.

To avoid these problems, we use only tests with $\psi^A{}_z = 0$ when we are using cylindrical coordinates. Because we wanted to compare to published results, we use a version of BDRT1 [61] which differs from the original by a linear transformation of the matter space coordinates (and so changes $\psi^A{}_i$), and transform the results back to the original coordinates for the purpose of comparison. The initial data for this test is described in Section 3.5.12.

Boundaries We have used simple copy boundary conditions in the r direction, and periodic boundaries in the θ direction, while ensuring that the computational grid extends over a range of 2π in θ . The copy boundaries are not correct in general; these incorrect boundary conditions are what cause the visible problems near the boundaries in the BDRT1 test on the cylindrical grid, visible in Fig. 3.17b. Since we do not anticipate using cylindrical coordinates (or spherical coordinates) in the future, we use copy boundary conditions for simplicity's sake, and note where the incorrect boundary conditions have caused problems.

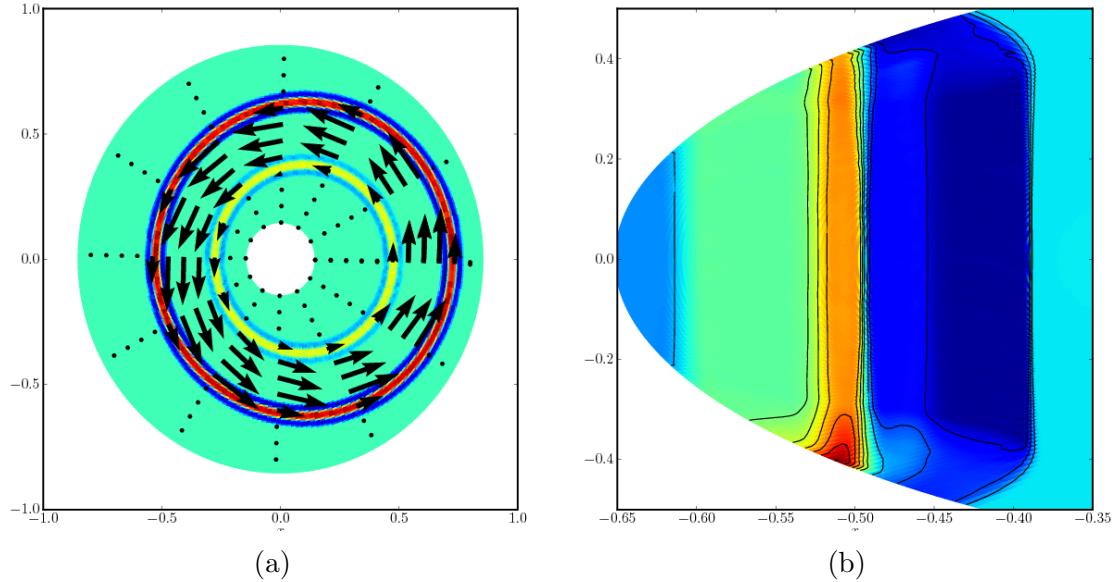


Figure 3.17: Tests on a two-dimensional cylindrical grid. Fig. 3.17a shows a pseudocolor density plot and velocity vector plot for a ring rotor at coordinate time $t = 0.0095$. The initial velocity within the rotating region is a rigid rotation, while the outer and inner regions are static. The numerical domain is an annulus, with the origin of cylindrical coordinates cut out. In the density plot, waves are visible propagating away from the initial discontinuity. In an equivalent shear test in planar symmetry with a static inner and outer region and a central region with non-zero velocity parallel to the initial discontinuities, one would expect the Riemann problem solutions around the two discontinuities to be symmetrical; the asymmetry visible in the ring rotor version is due to the curvature of the discontinuity and centrifugal forces. Fig. 3.17b shows a pseudocolor density plot of the BDRT1 Riemann test at coordinate time $t = 0.02$. Plane waves can be seen propagating away from the initial discontinuity at $x = -0.5$. Only a part of the annular numerical domain is shown. In the plot, the effect of the incorrect boundary conditions can also be seen as the solution near the boundaries is incorrect—this is seen also in Fig. 3.18.

3.5.11 Tests in two-dimensional curvilinear coordinates

To test the functioning of the code with a metric other than the Minkowski metric in Cartesian coordinates, we have implemented several tests in Minkowski cylindrical coordinates. We describe the details of implementing this cylindrical grid in Section 3.5.10. We use two different Newtonian tests to demonstrate that the code can handle curvilinear coordinates: a Riemann problem (BDRT1, [61]) and an off-axis ring rotor, with initial data described in further detail in Section 3.5.12. The symmetry of each of these tests is obscured on the cylindrical grid; this allows us to compare the two-dimensional cylindrical evolutions to one-dimensional evolutions that take advantage of the symmetry of the physical systems (cylindrical

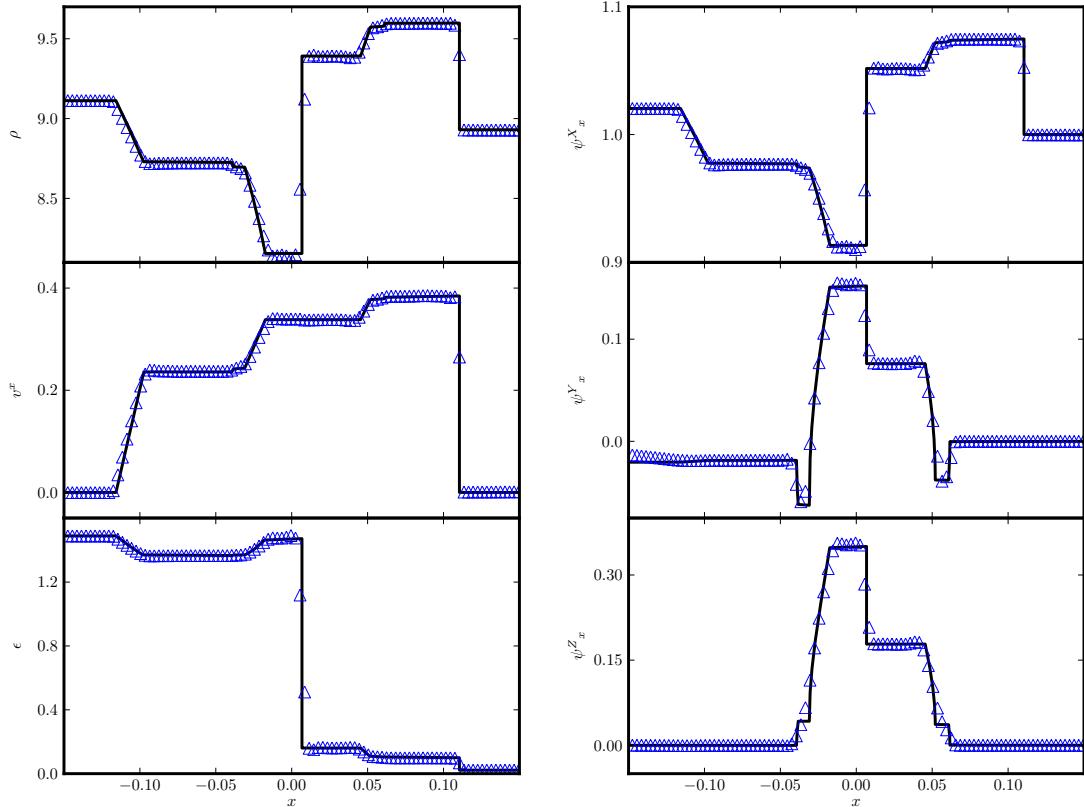


Figure 3.18: One-dimensional slice of BDRT1 test at coordinate time $t = 0.02$ evolved on a two-dimensional cylindrical grid. Here the numerical domain was an annulus with 300 points in the r direction and 601 in the angular direction; only 1 out of every 3 points is plotted here for clarity. The results have also been transformed from cylindrical coordinates, which are used for evolution, to Cartesian coordinates, as described in Section 3.5.10. In addition, while the transformed version of the BDRT1 initial data was used (see Section 3.5.12), the matter-space indices have been transformed back to the coordinate system of the original BDRT1 test for comparison with the exact solution presented in [61]. The results using a two-dimensional cylindrical grid visually match well with the exact solution; the only notable feature is that the effect of the incorrect boundary conditions can be seen near the left edge of the plot for ψ^{Y_x} .

symmetry for the ring rotor and planar symmetry for the Riemann test), giving us confidence that the code is self-consistent.

The results of the off-axis ring rotor evolved on the two-dimensional cylindrical grid show good agreement with both the results of a one-dimensional evolution of an axisymmetric ring rotor, suitably shifted, as well as the results of a two-dimensional Cartesian evolution of the same ring rotor. As the resolution is increased, the results of the 2D cylindrical evolution approach those of the 1D evolution. Fig. 3.17a shows a pseudocolor plot of the density with a vector plot of the velocity at coordinate time $t = 0.0095$. The initial velocity within the rotating ring is a rigid rotation, while the outer regions are static. In the density plot, waves can be seen propagating away from both edges of the ring.

The Riemann problem results on the 2D cylindrical grid also approach the published exact solution [61] with increased resolution. Fig. 3.17b shows a pseudocolor plot of the density for the Riemann problem on a 2D cylindrical grid at coordinate time $t = 0.02$. Waves can be seen propagating away from the initial discontinuity. The boundary conditions are not consistent with the symmetries of the initial data, giving rise to error at the boundaries, as was mentioned in Section 3.5.10.

3.5.12 Initial data for numerical tests

We used several sets of initial data that were defined in published papers; this was done to ensure that our code agreed with Newtonian results produced previously [61] [62]. Because both papers chose entropy, s , as a primitive variable, instead of the pressure, p , we list the initial entropy value here, and calculate the pressure from the entropy when the system is initialized.

For the following sets of initial data, the spacetime metric is the Minkowski metric, and the matter-space metric is the Euclidean metric in Euclidean coordinates normalized with the initial density of the elastic medium, n_0 ; we note that while we must convert units of velocity to geometrized units, we do not need to convert units of density or of length, as long as we are consistent throughout the code. For this paper the value $n_0 = 8.93 \text{ g/cm}^3$ was used for the BDRT tests (from [61]) and $n_0 = 8.9 \text{ g/cm}^3$ was used for the TRT tests (from [62]). In addition to this, for each of these situations, the Cranfield equation of state, described in Appendix C, was used. For comparison purposes, the velocities in this section are taken to be in km s^{-1} , while the entropy is in $\text{kJ g}^{-1}\text{K}^{-1}$.

BDRT1 This is the same as *Testcase 1* in [61]. It allows us to examine the entire seven-wave structure of the solution. Using the Cranfield EOS mentioned above, the solution consists of three left-traveling rarefaction waves, a right-traveling contact, two right-traveling rarefactions, and a right-traveling shock wave. The initial data is presented for the state vector $\mathbf{w} = (v^i, F^i{}_A, s)$ in the mixed framework given in [1], and all other quantities are derived from them:

$$\begin{aligned}\mathbf{w}_L &= \left\{ \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.98 & 0 & 0 \\ 0.02 & 1 & 0.1 \\ 0 & 0 & 1 \end{pmatrix}, 0.001 \right\}, \\ \mathbf{w}_R &= \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.1 \\ 0 & 0 & 1 \end{pmatrix}, 0 \right\}.\end{aligned}\quad (3.285)$$

BDRT1 transformed As explained in Section 3.5.10, we cannot directly use the original BDRT1 test with the cylindrical grid. To remedy this, we obtain a transformed version of the BDRT1 test presented in [61] by transforming the lower matter-space indices as $F^i{}_{A'} = F^i{}_B \Lambda^B{}_{A'}$ where

$$\Lambda^B{}_{A'} := \frac{\partial \xi^B}{\partial \xi^{A'}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -0.1 \\ 0 & 0 & 1 \end{pmatrix}, \quad (3.286)$$

and the upper matter-space indices using its inverse. The transformed version is as follows:

$$\begin{aligned}\mathbf{w}_L &= \left\{ \begin{pmatrix} 0 \\ 0.5 \\ 1 \end{pmatrix}, \begin{pmatrix} 0.98 & 0 & 0 \\ 0.02 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, 0.001 \right\}, \\ \mathbf{w}_R &= \left\{ \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, 0 \right\},\end{aligned}\quad (3.287)$$

where the matter-space metric is also transformed so that, over the whole computational domain, it is

$$k_{AB} = \begin{pmatrix} n_0^{2/3} & 0 & 0 \\ 0 & n_0^{2/3} & -0.1n_0^{2/3} \\ 0 & -0.1n_0^{2/3} & n_0^{2/3} \end{pmatrix}. \quad (3.288)$$

4-wave relativistic solution We constructed a range of relativistic solutions, mostly consisting of a single shock or rarefaction, using the technique outlined in [1]. The toy relativistic equation of state given in Appendix C, with parameters $\Gamma = 5/3$, $\lambda = 4/3$, and $\kappa = 1/2$. In particular, we present a solution with four nonlinear waves. The two left-going waves (1- and 2-waves) are rarefactions. The contact is trivial, as are the central (3- and 5-waves) nonlinear waves. The slower right-going wave (a 6-wave) is a rarefaction, and the fast right-going 7-wave is a shock. The initial data is presented for the state vector $\mathbf{w} = (v^i, \psi^A_i, p)$, truncated to 6 significant figures, and all other quantities are derived from them:

$$\mathbf{w}_L = \left\{ \begin{pmatrix} 0.05 \\ 0.1 \\ 0.2 \end{pmatrix}, \begin{pmatrix} 1.5 & 0 & 0 \\ -0.5 & 1 & 0 \\ 0.5 & 0 & 1 \end{pmatrix}, 1.86054 \right\}, \quad (3.289)$$

$$\mathbf{w}_R = \left\{ \begin{pmatrix} 0.469381 \\ -0.0332532 \\ 0.349709 \end{pmatrix}, \begin{pmatrix} 0.764910 & 0 & 0 \\ -0.541672 & 1 & 0 \\ 0.369075 & 0 & 1 \end{pmatrix}, 0.450123 \right\}. \quad (3.290)$$

Rotor tests In addition to Riemann-problem style tests we consider a genuinely two-dimensional rotor test. The Newtonian rotor test was suggested by [67], where the evolution was shown using a high-order finite element technique. The domain is cylindrical, of total radius 0.5. The material is initially at rest except in the rotor, represented by a cylinder of radius 0.1, within which it rotates with angular velocity $\omega = 10$. The material is not deformed (i.e., F^i_A is the unit matrix) nor hot (i.e., $s = 0$). All other matter properties follow the Riemann tests above. That is, the initial density is given by $n_0 = 8.93 \text{ g/cm}^3$ and the Cranfield equation of state, described in Appendix C, was used. Here, as we have used a Cartesian grid, we have simulated the full domain $x, y \in [-0.5, 0.5]$.

We have developed a new ring rotor version of the Dumbser Rotor from [67] so that, while using two-dimensional cylindrical coordinates, we do not have to include the axis of the coordinate system. The inner and outer areas of the grid have the same initial data, while the central region of the grid differs from these states only in that it has a non-zero velocity. Aside from the velocity, the initial data for this test are exactly the same as for the Dumbser Rotor; the velocity in the inner and outer regions is 0, while the central region has a uniform angular velocity of $\omega = 1.0$.

We suggest a relativistic rotor test as a direct comparison with the Newtonian version. The domain remains the same as the Newtonian case. The angular velocity is reduced to $\omega = 0.5$. The material is initially set so that ψ^A_i is the

unit matrix and $p = 1$. As the shear also depends on the velocity through $\psi^A{}_t$, the material is sheared within the rotor initially, in contrast to the Newtonian case, but this is small. As in the Riemann tests above we use the toy relativistic equation of state given in Appendix C, with parameters $\Gamma = 5/3$, $\lambda = 4/3$, and $\kappa = 1/2$.

Chapter 4

Interfaces

An estimate of the scale of the crust-core transition is given by the size of the viscous boundary layer between the crust and the core, called the Ekman layer. For typical neutron star values, this region is very small (~ 10 cm) [68, 60] compared to the size of the neutron star (~ 10 km); this means that it is best modeled by a sharp transition. However, sharp transitions in the equation of state can lead to large unphysical oscillations in the pressure [69, 70, 71, 72]. These oscillations are present in first-order computations and are not removed by moving to higher order; they are due to unphysical (thermodynamically inconsistent) mixing of the two materials introduced by the smearing of sharp discontinuities that is inherent in HRSC schemes [73, 74]. To solve this numerical problem, we must think carefully about how we treat the crust-core interface.

To represent interfaces, we consider the spacetime to be composed of a union of separate regions, where each region is described by a different physical model [60]. These regions are joined at dynamical boundaries. We track these boundaries using a *level-set function*, and then determine and apply appropriate boundary conditions for a variety of interface types using an extension of the *ghost fluid method* [75].

4.1 Level-set methods

In order to implement a method that produces the correct behavior at the interface, we first need to locate the interface. To keep track of the location of the interface during evolution, we use a *level-set function*. A level-set function is a scalar grid function which is positive where one material is located and negative where the other material is located. The interface between the two materials occurs where the level-set function is equal to 0.

If more than two materials are present in an evolution, more than one level-set function is needed. In general, one needs $n - 1$ level-set functions for an n -material evolution—one for each material—and then the area where the n th material is located is where all level-set functions are negative. However, in practice it is difficult to ensure that multiple level sets will evolve together, meaning that n level-set functions must be used for n materials, and corrections must be made in order to eliminate areas of the grid where two level-set functions are positive at once [76].

To evolve the level-set function, it is only necessary to ensure that the interface moves at the correct velocity. In our codes, this is done in two ways. In the `ElasticEvolution` code, the level set is simply advected along with the materials; the level-set advection velocity is taken to be the velocity of the real material in any given cell. In the `MultiModel` code, we find the interface velocity by finding the component of velocity normal to each interface in the real cells adjacent to that interface. Once the advection velocity is determined next to the interfaces, it is extrapolated to the remainder of the grid.

The methods described above, which are used to estimate the interface velocity for level-set advection, are correct in the continuum limit as long as the velocity is not discontinuous at the interface. In both cases, advection is in the direction normal to the interface.¹ Since this is the case, the velocity should only stay discontinuous for sustained times (i.e. aside from shocks passing through the interface or initial interface Riemann problems) in the vacuum-interface case (where the physical velocity in the vacuum is not defined). In this case, we advect the level-set using the velocity in the cell immediately to the material side of the interface. In essence, we are defining the vacuum velocity in such a way that it is continuous across the interface. In principle, the correct thing to do would be to solve the interface Riemann problem and use the interface velocity to advect the level-set function, regardless of what type of interface we use; however, in most cases, our approach will produce the correct result.

In the `ElasticEvolution` code, we use the upwind method for advection; if the velocity is positive, we use the left-hand difference, and if it is negative, we use the right-hand difference. In the `MultiModel` code, we treat the advection equation as a Hamilton-Jacobi equation, and use the Lax-Friedrichs scheme with a WENO reconstruction of the derivatives [77]. We use the fast-marching method [78] both to extrapolate the interface velocity to the remainder of the grid, and to extrapolate the value of the level-set outward for reinitialization. This helps to

¹In 1D, the advection velocity is normal to the interface by definition. In the `MultiModel` code, the advection velocity is the component of the material velocity normal to the interface.

prevent the loss of sharp corners or pockets of material, or excessive steepening of the level-set function.

4.2 Ghost fluid method

Once we have located the interface, using the level-set function, we need to determine what should happen at the interface between the two materials. We have chosen to used the ghost-fluid method (GFM), first developed by Fedkiw *et al* 1999 [75] for interfaces between two fluids with differing equations of state. In this method, each material is evolved independently, filling a ghost fluid into the cells occupied by the other material on the original grid. The ghost fluid is of the same material as the material to be evolved, but is chosen to attempt to reproduce the physically correct behavior (although for the original GFM, the behavior is only physically correct in the quiescent state) at the interface between the two materials. The way that we choose the values of the variables in the ghost fluid cells determines the physics that occurs at the interface.

In the original GFM of Fedkiw *et al* [75], the population of these ghost cells is relatively intuitive. Basically, we look at a stationary contact discontinuity between the two materials (fluids in the original GFM), and determine which variables are continuous, and which are generally allowed to be discontinuous. We then assign variables as follows:

- *Taking*: If the variable should be continuous across the interface, then we assign that variable in the ghost cell to be the value of that variable in the real-material cell located at the same physical location. For clarity, we will call this *taking*.
- *Extrapolating*: If the variable can generally be discontinuous across the contact, then we extrapolate (we will call this *extrapolating*) from the nearest real-material cell of the same material as the cell to be populated. Although generally the value could be extrapolated at any order, we use zeroth order extrapolation; in other words, we simply copy the value from the nearest real-material cell.
- *Calculating*: Generally, we might have multiple variables that are allowed to be discontinuous, but that must be assigned consistently. For example, in the fluid-fluid case, the entropy and density can both generally be discontinuous, but must be assigned such that their values are thermodynamically consistent with the pressure via the equation of state. In this particular case,

Type	Cont.	Disc.	Set	Calc.
Fluid-Fluid (with ρ)	$p, v^i k_i$	$s, v^i \perp^j_i$	NA	ρ
Fluid-Fluid (with ψ^A_i)	$p, v^i k_i$	$s, v^i \perp^j_i$	$\psi^A_i k^i = k(\psi^A_i k^i)_{\text{ext}}$	k
Fluid-Solid (set fluid, with ρ)	$p_{ij} k^i k^j, v^i k_i$	$s, v^i \perp^j_i$	NA	ρ
Fluid-Solid (set fluid, with ψ^A_i)	$p_{ij} k^i k^j, v^i k_i$	$s, v^i \perp^j_i$	$\psi^A_i k^i = k(\psi^A_i k^i)_{\text{ext}}$	k
Fluid-Solid (set solid)	$p_{ij} k^i k^j, v^i k_i$	$s, v^i \perp^j_i$	$p_{ij} k^i \perp^j_k = 0$	$\psi^A_i k^i$
Solid-Solid (slip)	$p_{ij} k^i k^j, v^i k_i$	$s, v^i \perp^j_i$	$p_{ij} k^i \perp^j_k = 0$	$\psi^A_i k^i$
Solid-Solid (stick)	$p_{ij} k^i, v^i$	s	NA	$\psi^A_i k^i$
Fluid-Vacuum (set fluid)	$p = 0$	s, v^i	NA	ρ
Solid-Vacuum (set solid)	$p_{ij} k^i = 0$	s, v^i	$p_{ij} k^i = 0$	$\psi^A_i k^i$
Solid-Atmosphere (set solid)	$p_{ij} k^i k^j = p_{\text{atm}}$	s, v^i	$p_{ij} k^i \perp^j_k = 0$	$\psi^A_i k^i$
Fluid-Atmosphere (set fluid)	$p = p_{\text{atm}}$	s, v^i	NA	ρ

Table 4.1: Summary of how each interface type is handled using the simple ghost-fluid method. Note that we always extrapolate the components $\psi^A_i \perp^i_j$ and $p_{ij} \perp^i_k \perp^j_l$, because these are generally allowed to be discontinuous.

we choose to extrapolate the entropy, and then we calculate the density using the equation of state.

- *Setting:* For some interface times and certain boundary conditions, we may also need to *set* variables to specific values. This could be to enforce certain physical conditions (i.e. there should be no traction at the interface when we use slip boundary conditions between two solids) or it could be to address certain numerical concerns (i.e. as in the case of using the elasticity formulation in the fluid limit, where we should only have one free parameter instead of three).

This will become clearer with an example. The following subsections describe how this is done for various types of Newtonian interfaces, and are summarized in Table 4.1. In the following, we shall call the covector normal to the interface, k_i . In the Newtonian case, it is raised by $k^i = \delta^{ij} k_j$, and normalized $k_i k^i = 1$.

Because the original GFM assumes that the discontinuity at the interface is a stationary material contact, it is *not* correct when pressure and normal velocity are discontinuous at the interface, as when a shock hits; this is because it assumes that these values are continuous, rather than solving the full Riemann problem. Methods that seek to address this are mentioned in Section 4.3. In practice, the GFM performs well where strong shocks are not present; in our shattering simulations, the shocks should be small, and therefore, the GFM should be sufficient for our purposes.

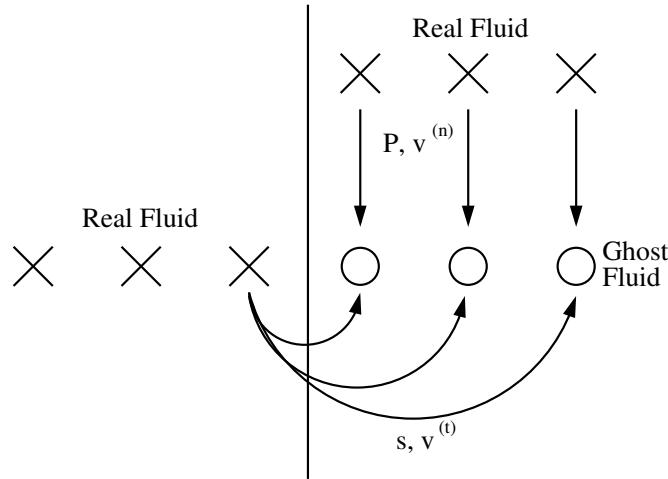


Figure 4.1: To populate the ghost fluid for a fluid-fluid interface, the variables that must be continuous across the interface contact ($p, v^i k_i =: v^{(n)}$) are *taken* from the real-fluid cell located at the same position as the ghost-fluid cell being filled. The variables that are generally discontinuous at the contact ($s, v^i \perp^j_i =: v^{(t)j}$) are *extrapolated* from the nearest real-fluid cell of the same material as the ghost fluid. Extra discontinuous variables that must be consistent with these (ρ) are then calculated from the variables that are already assigned.

4.2.1 Fluid-fluid interface

For fluids in the usual Eulerian formulation, we have the primitive variables: pressure p , density ρ , velocity v^i , and entropy s . Pressure and the component of velocity normal to the interface should be continuous across the contact, so we take these values from the real-material cells located at the same position as the cells we are populating (so we take these values from the other material). Tangential velocity and entropy should generally be discontinuous across the interface, so we fill in these values using extrapolation from the nearest real-material cell of the same material. Once we have pressure and entropy, we can calculate the density from these, using the equation of state. In principle, we could instead populate the density using extrapolation and calculate the entropy from this, but we follow [75], and extrapolate the entropy. This process is illustrated for the fluid-fluid interface in Fig. 4.1.

4.2.2 Fluid-fluid interface (using the ψ^A_i formulation)

In the `ElasticEvolution` code, we have a choice between using the evolved rest-mass density, ρ , and the rest-mass density derived from ψ^A_i , and the matter-space and spacetime metrics, k_{AB} and g^{ab} . This means that we can choose to either produce a fluid evolution by evolving D , the relativistic rest-mass density

introduced in Section 3.3.2, or by taking the fluid limit of the ψ^A_i formulation. In general, the latter will not be correct, because of the difference in shear behavior mentioned in Section 3.2.2, where a discontinuity in the velocity tangential to a shock causes a non-zero shock speed in elasticity, but not in a fluid. However, the fluid tests presented here contain no shear, so we use the fluid limit of the ψ^A_i formalism without incident.

This version of the GFM is almost exactly the same as the GFM using D , except that, instead of calculating ρ from p and s , we calculate $\psi^A_i k^i$. This construction has three independent components. However, the pressure is only one independent component, so we should only solve for one value. To do this, we set $\psi^A_i k^i = k(\psi^A_i k^i)_{\text{ext}}$, where $(\psi^A_i k^i)_{\text{ext}}$ is extrapolated from the nearest real-fluid cell of the material we are populating, and then use a root finder to calculate the constant factor k . We extrapolate all of the tangential components, $\psi^A_i \perp^i{}_j$.

4.2.3 Solid-fluid interface (setting the fluid)

Setting the ghost fluid in a solid-fluid interface is mostly the same as setting the ghost-fluid cells in the fluid-fluid interface situation while using the ψ^A_i formulation. The only difference here is that, instead of the pressure being continuous, the normal-normal component of the pressure tensor should be continuous (i.e. $p_{ij} k^i k^j$). In the Newtonian case, for a fluid, $p = p_{ij} k^i k^j$; however, this is not the case for elastic materials, so it is important that we take $p_{ij} k^i k^j$, instead of taking p . If we are evolving D instead of ψ^A_i , we take our Newtonian fluid $p_{\text{fluid}} = (p_{ij} k^i k^j)_{\text{solid}}$, and calculate ρ from p and s , as in the fluid-fluid interface case.

One might expect, that since we have set the three components $p_{ij} k^i$, and we have to solve for three components $\psi^A_i k^i$, we could just use a three-dimensional root finder. However, for a fluid equation of state, the components $p_{ij} k^i \perp^j{}_k$ are guaranteed to be 0, so we really only have one independent component. Instead of solving for all three components of $\psi^A_i k^i$ independently, we again set these values to the extrapolated value multiplied by some factor, and then solve for that factor. If we are evolving D , we simply calculate ρ from s and p .

4.2.4 Solid-fluid interface (setting the solid)

In the solid-fluid interface configuration, the solid is set in almost the same way as the fluid. Again, we have the case where $p_{ij} k^i k^j$ should be continuous, and $p_{ij} k^i \perp^j{}_k$ should be 0; however, away from the interface, the normal-tangential components can generally be anything. This means that the three components

$p_{ij}k^i$ are all independent, and we use a three-dimensional root finder to find all of the components $\psi^A_i k^i$.

4.2.5 Solid-solid interface (slip conditions)

There are several options for boundary conditions when two solids meet at an interface. One is for there to be no friction between the two solids; in other words, for them to slip completely against each other with no resistance. In this case, the conditions are exactly the same as for the solid-fluid interface (except for the obvious difference that the second solid is populated like a solid, and not like a fluid).

4.2.6 Solid-solid interface (stick conditions)

Another possibility for interface boundary conditions between the two materials is for them to stick completely with no slipping at all. In this situation, the two materials have essentially become one; if both materials are chosen to have identical equations of state, the results using these boundary conditions should produce exactly the same thing as if there were no interface at all. The normal components of the pressure tensor ($p_{ij}k^i$) and all of the components of the velocity should be continuous, so these variables are taken. The entropy, tangential components of the pressure tensor, and tangential components of ψ^A_i are all extrapolated. Then the components $\psi^A_i k^i$ are calculated from $p_{ij}k^i$ as in the solid-fluid and solid-solid-slip cases.

In addition, one could imagine a situation somewhere between our “stick” and “slip” conditions, where the two pieces of material are sliding against each other with some resistance. Although one could imagine modeling this by assigning the normal-tangential components of the stress tensor to some intermediate values, resistance (i.e. friction) implies that some energy will be dissipated, while the system of equations that we use is inviscid. This means that they include no dissipative mechanism. So, although we could dream up some intermediate boundary conditions to govern the motion of the material, we could not accurately model the full physics of such a system.

4.2.7 Solid-vacuum interface

In the vacuum, we have no matter grid variables. In the solid, we expect the normal components of the pressure tensor, $p_{ij}k^i$, to go to 0 at the interface; there should be no forces acting at the interface, either normal to it or parallel to it.

Because of this, we set $p_{ij}k^i = 0$. This can be thought of as either *setting* or *taking*: we *are* setting the variables to a particular value, but this is also the value that we would expect to be physically correct in the vacuum. Conceptually, this process is equivalent to *taking*, but in the code, what we do looks more like *setting*. The velocity, entropy, and tangential components of $\psi^A{}_i$ are all extrapolated; then the components, $\psi^A{}_i k^i$, are calculated from s and $p_{ij}k^i$.

4.2.8 Fluid-vacuum interface

In the fluid-vacuum case, we expect the pressure itself, p , to go to 0 at the interface. This means that, with most equations of state that we might like to use to model neutron stars, the rest-mass density, ρ , will also go to 0 at the interface, which is likely to cause numerical problems (because small errors could cause the density to become negative, and negative density is not allowed as an input to many of the equations used). Additionally, to calculate the speed of sound, we need to find the ratio, p/ρ , which will fluctuate wildly as the density approaches zero. Indeed, a naive implementation of the fluid-vacuum interface fails; however, following the approach in [79], we can implement a positivity preserving flux limiter, which keeps both density and pressure positive. The results for preliminary tests are promising, but not fully tested.

Neutron-star surface The crust of a neutron star will typically be covered by a fluid envelope; however, to simplify the neutron star toy model, we will transition directly from the solid crust to the vacuum exterior. This means that there will be a solid-vacuum interface at the surface of the star. However, the properties of the solid will not be like those tested in the sections below. For the Cranfield equation of state, when the pressure goes to zero the density does not. This means that, at the vacuum interface, there is no reason for numerical errors to cause the density to become negative. Additionally, the internal energy does not approach 0, and this prevents this from becoming negative due to fluctuations as well.

However, in the stellar environment, the equation of state has the property that the shear modulus scales with density; since pressure is a function of powers of the density, the density will necessarily approach 0 as pressure approaches 0, as will the specific internal energy. This means that, as in the fluid-vacuum case above, small fluctuations can cause the density or internal energy to become unphysically negative. A positivity preserving scheme could help to address this issue, but it is unclear how to extend the work in [79] to the system of equations used for elasticity.

4.2.9 Atmosphere interfaces

To attempt to address the problems at the stellar surface mentioned above, we introduce an *atmosphere*. This atmosphere is treated as a model present in the `MultiModel` code. However, this model consists only of an atmospheric pressure value: while the fluid model consists of grid functions describing density, velocity, internal energy, and pressure at every point in the grid, the atmosphere model simply consists of a single grid function which stores the pressure everywhere. This pressure is then simply initially set to a constant value (set using an input parameter) all over the grid, and then is never evolved.

Material-atmosphere interfaces are treated in much the same way as material-vacuum interfaces. In the atmosphere, nothing is set: we only have the atmospheric pressure, which is never updated. In a solid, we set the normal components of the pressure tensor as we would in the solid-fluid interface case (i.e. $p_{ij}k^i k^j = p_{\text{atm}}$ and $p_{ij}k^i \perp^j k = 0$). Velocity, entropy, and the tangential components of ψ^A_i are then all extrapolated, and the normal components, $\psi^A_i k^i$, are calculated from $p_{ij}k^i$. In a fluid, we set $p = p_{\text{atm}}$, extrapolate velocity and entropy, and calculate the density from these values using the equation of state.

4.2.10 Isobaric fix

One problem with the original GFM is that it exhibits overheating effects at the interfaces between materials. This causes an unphysical increase in entropy, and a corresponding decrease in the density. This effect was already observed in the original GFM paper [75], and a simple fix was suggested to ameliorate the observed overheating issues.

The *isobaric fix* attempts to prevent overheating errors by overwriting the real-material cell immediately adjacent to the interface, copying the entropy from the next real-material cell. (For example, if the interface is located between i and $i + 1$, with the real material in cells $j < i$, then we overwrite the entropy in cell i by copying the value from cell $i - 1$.) This fix has been implemented in the `ElasticEvolution` code, but not in the `MultiModel` code.

4.2.11 GFM in general relativity

In general relativity, our interface is a $2 + 1$ hypersurface, \mathcal{S} , in spacetime with normal covector, s_a . We can separate this into space and time-like parts as follows:

$$s_a =: \lambda n_a + k_a, \quad (4.1)$$

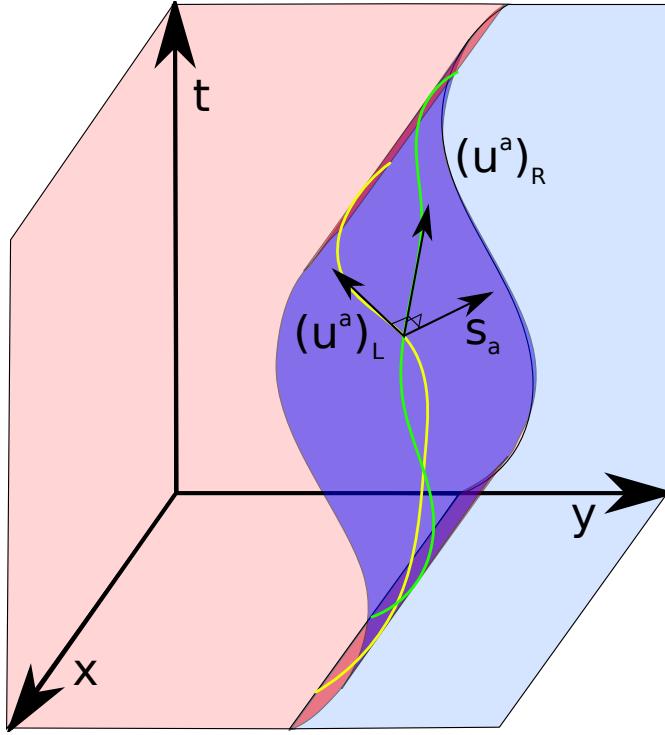


Figure 4.2: A sketch of the interface, shown as a 3D hypersurface in spacetime. World lines of individual particles may slide along the interface (or even move away from the interface, if there is circulation in the flow), but may not pierce the interface. This means that, while the four-velocity can jump across the interface, it must still be tangential to the interface, so that $s_a u^a = 0$, where s_a is the covector normal to the interface.

where n_a is a covector normal to the constant- t hypersurfaces, which we use for the $3 + 1$ split, and $n^a k_a = 0$, where n^a is raised by the metric, g^{ab} . Then k_a is purely spatial, and we can think of it as being a 3D vector, k_i , which is normal to the instantaneous interface at any given time.

Since particle world lines cannot pierce the interface (as illustrated in Fig. 4.2), we must have

$$u^a s_a = W(n^a + v^a)(\lambda n_a + k_a) = W(-\lambda + v^a k_a) = 0, \quad (4.2)$$

where u^a is split according to the spacetime split as

$$u^a = W(n^a + v^a), \quad (4.3)$$

so that v^a is purely spatial ($v^a = (0, v^i)$ in the $3 + 1$ split). Then we see that λ must be

$$\lambda = v^a k_a = v^i k_i =: v_\perp, \quad (4.4)$$

which is the component of the 3-velocity normal to the instantaneous interface, and we can henceforth write

$$s_a = v_\perp n_a + k_a, \quad (4.5)$$

and

$$s^2 = s_a s^a = 1 - v_\perp^2. \quad (4.6)$$

Because the two materials are in contact at the interface, s_a and k_a must be continuous there. It does not make sense to have different time-slicing on the different sides of the interface, so we assume that n_a is also continuous. This means that we must have the normal component of the 3 velocity, v_\perp , continuous across the interface.

Now, what we really want to be continuous across the interface is the following quantity:

$$[[T^{ab} s_a s_b]] = 0, \quad (4.7)$$

or for solid-solid stick boundary conditions:

$$[[T^{ab} s_a]] = 0, \quad (4.8)$$

where the double square brackets indicate the difference between the quantity in the left and right states. For the former, we can rewrite this as

$$[[e u^a u^b s_a s_b + p^{ab} s_a s_b]] = [[p^{ab} s_a s_b]] = 0. \quad (4.9)$$

We then split p^{ab} into the isotropic and anisotropic stress parts, so that

$$p^{ab} = p h^{ab} + \pi^{ab}, \quad (4.10)$$

note that $h^{ab} = g^{ab} + u^a u^b$ is the projection normal to the 4-velocity, and that

$$\pi_{ab} = \psi^A{}_a \psi^B{}_b \pi_{AB}. \quad (4.11)$$

It can be shown that

$$\psi^A{}_a s^a = \psi^A{}_a (k^a - v_\perp v^a), \quad (4.12)$$

so we get

$$p^{ab} s_a s_b = p (1 - v_\perp^2) + \pi_{ij} (k^i - v_\perp v^i) (k^j - v_\perp v^j). \quad (4.13)$$

In the Newtonian limit, this goes to $p + \pi_{ij} k^i k^j$, which is the Newtonian quantity that we expect to be continuous. In a relativistic fluid-fluid interface, this is

automatically continuous as long as both p and v_\perp are continuous.

For the no slip boundary conditions, we have

$$[[p^{ab}s_a]] = 0. \quad (4.14)$$

We consider the construction

$$p_{ab}s^a\gamma^{bc} = (ph_{ab} + \pi_{ab})s^a\gamma^{bc} = pk^c + \pi_{ab}(k^a - v_\perp v^a)\gamma^{bc}, \quad (4.15)$$

where γ^{ab} is the inverse of $\gamma_{ab} = g_{ab} + n_a n_b$. Again, this has the expected Newtonian limit.

4.3 Other interface methods

The level-set function paired with the GFM are not the only ways to produce the correct behavior at the interface between two materials; they are just the methods that we found to be simplest and most intuitive. Other methods seek to improve the performance of these methods through various additions.

Instead of using a level-set function to track the interface, Miller and Colella 2002 [63] choose to evolve volume fractions to keep track of how much of each material is present in each cell; from this, the position of the interface can be calculated. Mass redistribution is then used to produce a fully conservative method.

Another option was proposed by Liu, Khoo and Yeo in 2003 [80] and developed further by Sambasavian in 2009 [81] and then applied by Barton and Drikakis in 2010 [82]. This method is largely the same as the GFM—it differs only in the method for choosing ghost-cell values—and is therefore called the *modified* or *Riemann GFM*. Instead of using the simple criteria based on the continuity or discontinuity of various variables across the interface contact, an interface Riemann problem is used to determine the correct behavior at the interface. The ghost cells are then populated such that the correct behavior is produced on the real-fluid side of the interface. This method produces correct results when the pressure and normal velocity are discontinuous at the interface, where the simple GFM can sometimes have trouble.

4.4 Results

To test our code, we have used various published Riemann problems that show different types of interfaces. In all the presented cases, our results can be compared

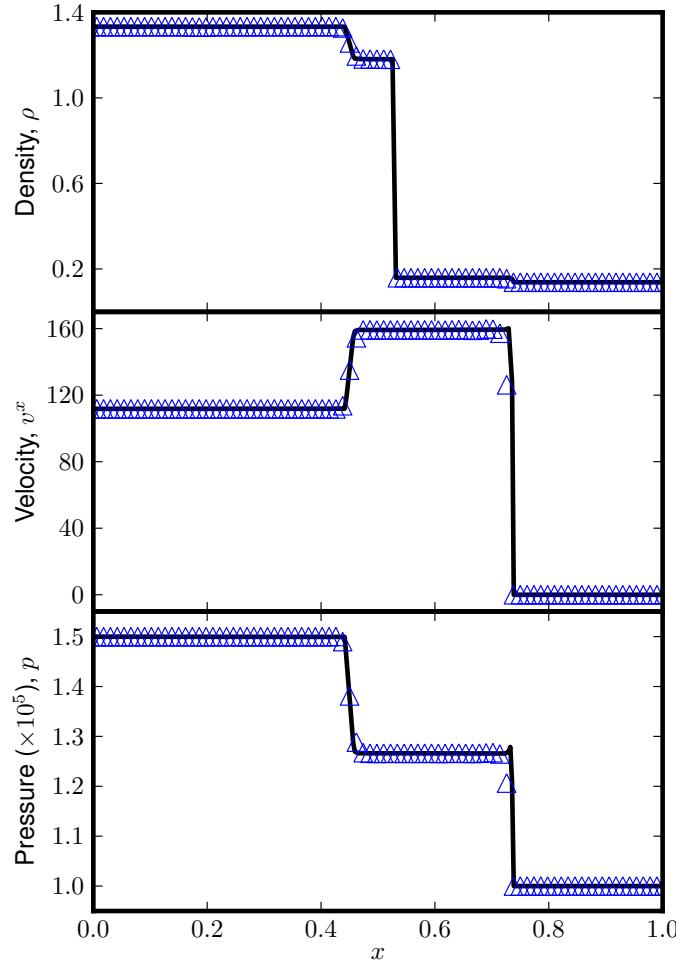


Figure 4.3: The results of Test B from [75]. The blue triangles show the results of an evolution using 250 cells, with only every third cell shown for clarity. The black line shows the results of higher resolution run using 1000 grid cells. These results were produced using an evolution that evolved ψ^A_i instead of D , but the results of an evolution using D would be visually indistinguishable from this.

to the results published in other work; this allows us to ensure that our code is functioning correctly.

4.4.1 Newtonian fluid-fluid test: Test B

This test of a Newtonian fluid-fluid interface is *Test B*, published in [75]. In this test, a shock comes in from the left side of the grid, and hits an interface at the center of the grid. The interface is between two fluids, each described by a gamma law equation of state with a different value for the ratio of specific heats, γ . The

initial data are given as follows:

$$\begin{aligned}\gamma_L &= 1.4, & \rho_L &= 1.3333, & p_L &= 1.5 \times 10^5, & u_L &= 0.3535\sqrt{10^5} \\ \gamma_M &= 1.4, & \rho_M &= 1, & p_M &= 1 \times 10^5, & u_M &= 0 \\ \gamma_R &= 1.67, & \rho_R &= 0.1379, & p_R &= 1 \times 10^5, & u_R &= 0\end{aligned}\quad (4.16)$$

where

$$x_L < 0.05, \quad 0.05 \leq x_M < 0.5, \quad \text{and} \quad x_R \geq 0.5. \quad (4.17)$$

Here, u represents the x component of the velocity, and all other components are 0. In the case where ψ^A_i is evolved and used to calculate the rest mass density instead of D , the initial values of ψ^A_i are simply set as follows:

$$\psi^A_i = \begin{pmatrix} \rho & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.18)$$

Since, in this case, we set the matter space metric, k_{AB} , to the identity matrix, this will produce the correct calculated rest mass density. Results are shown at coordinate time $t = 0.0012$.

Our results for Test B are shown in Fig. 4.3. The blue markers show the results of an evolution using 250 grid cells; only 1 in every 3 cell values is shown for clarity. The black line shows a higher resolution evolution using 1000 grid cells. From these plots, it can be seen that the numerical solutions converge, and if the values of the constant states and wave speeds are compared to the results published in [75], then good agreement is observed. These results were produced using the `ElasticEvolution` code, but have been reproduced using the `MultiModel` code.

4.4.2 Special relativistic fluid-fluid test: separation

We demonstrate that the simple GFM works for interfaces in our code in special relativistic situations using a fluid-fluid separation test, published in [74]. Two gamma-law fluids are initially traveling away from each other; aside from the equation of parameters, the only initially discontinuous quantity is the velocity. The initial data are as follows:

$$\begin{aligned}\gamma_L &= \frac{5}{3}, & \rho_L &= 1, & p_L &= 1, & u_L &= -0.3 \\ \gamma_R &= \frac{4}{3}, & \rho_R &= 1, & p_R &= 1, & u_M &= 0.3,\end{aligned}\quad (4.19)$$

where the interface is located at $x = 0.5$. Again, all other velocity components are set to zero. To assign ψ^A_i in a relativistic test in order to produce the correct

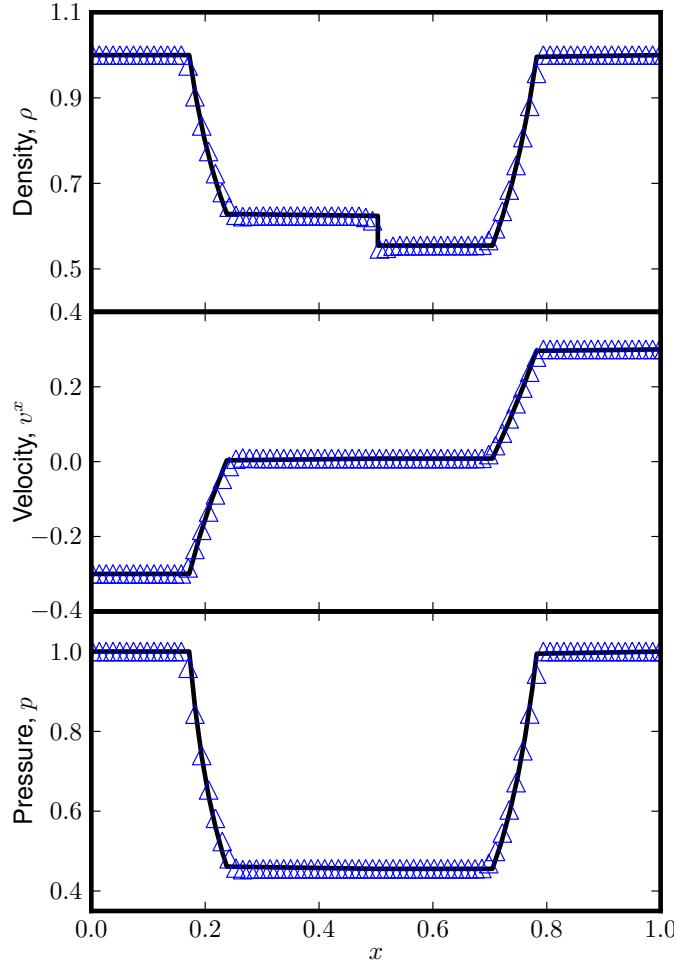


Figure 4.4: The density, x -component of velocity, and pressure after evolution for the special relativistic separation test from [74]. The numerical results (blue triangles) are shown with the exact solution (black line). Here it is easy to see that, across the interface contact discontinuity, the pressure and x -component of velocity must be continuous, but the density is allowed to be discontinuous. The density exhibits undershoots on either side of the interface; this is due to the fact that the x -component of the velocity in the initial data is discontinuous at the interface, but in our assignment of ghost-fluid cells, we have assumed that this value is continuous (this is what is called the *startup error* in [74]).

density, we set it to the following:

$$\psi^A_i = \begin{pmatrix} \rho/\sqrt{1-v_x^2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (4.20)$$

The results are shown at coordinate time $t = 0.4$.

The results produced by our code for the separation test are shown in Fig. 4.4. The blue markers show an evolution using 250 grid cells; there is good agreement between these results and those presented in [74] as well as with the exact solution, shown in black. There is an undershoot in the density on either side of the interface. As discussed in Section 4.2, our method assumes that those quantities that should generally be continuous across an interface contact discontinuity are continuous across the interface; it does not take into account the situation where there is a shock or Riemann problem at the interface. The fact that the normal component of the velocity is initially discontinuous at the interface means that our method causes errors near the interface in the first few time steps, which do not disappear over the evolution, and which do not converge away with increased resolution. The results were produced using the `ElasticEvolution` code.

4.4.3 Solid-fluid test: BOD1

Our first solid-fluid test comes from [83], and we therefore refer to it as *BOD1*. In this test, a high-pressure gas is in contact with a solid surface. For the solid, we use the Cranfield equation of state with parameters for copper: $\rho_0 = 8.93 \text{ g/cm}^3$, $c_0 = 4.651 \text{ km/s}$, $b_0 = 2.141 \text{ km/s}$, $C_V = 3.9 \times 10^{-4} \text{ kJ g}^{-1} \text{ K}^{-1}$, $T_0 = 300 \text{ K}$, $\alpha = 1.0$, $\beta = 3.0$, $\gamma = 2.0$. The gas is completely reacted PBX-9404, which uses the JWL equation of state with the following parameters: $\rho_0 Q = 10.2 \text{ GPa}$, $\rho_0 = 1.84 \text{ g/cm}^3$, $A = 854.45 \text{ GPa}$, $B = 20.493 \text{ GPa}$, $\omega = 0.25$, $R_1 = 4.6$, $R_2 = 1.35$, and $\rho_0 \epsilon^* = 49.8051 \text{ GPa}$ (the JWL equation of state is given in Appendix C—we use parameters from [84]). The fluid is on the left ($x < 0.5$), while the solid is on the

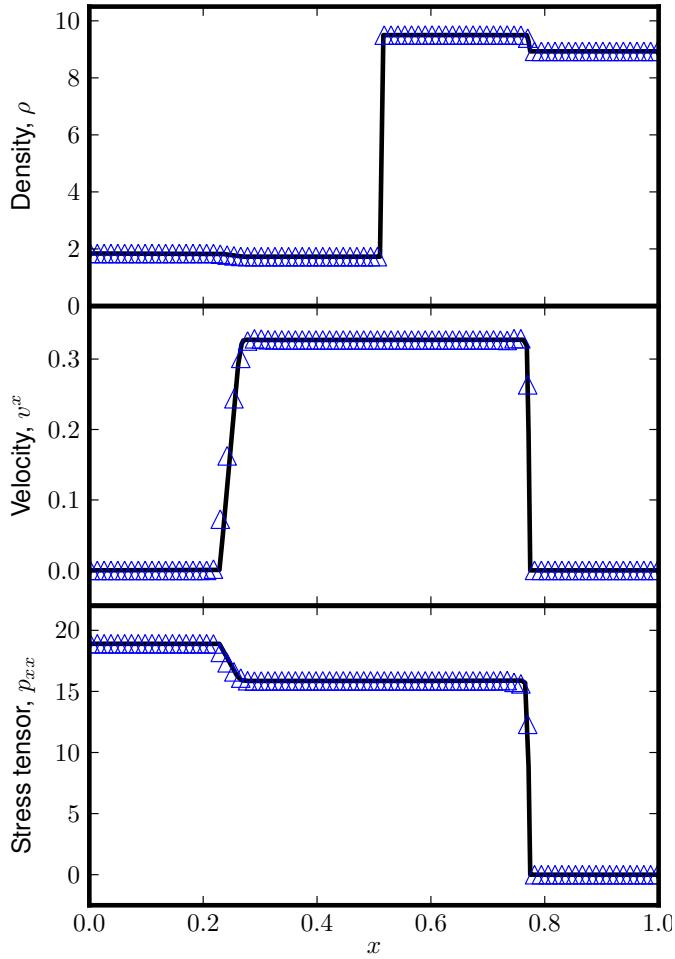


Figure 4.5: The solid-fluid test published in [83]. The blue triangles show the results of an evolution with 250 grid cells (1 in every 3 points are shown for clarity), while the black line shows a higher resolution evolution with 1000 grid cells in the computational grid. These results can be compared to the first test in [83]. Although our results for the pressure, p , do not match theirs, our results for the component of the stress tensor, p_{xx} , *do* match their results for p . We suspect that there is some discrepancy between their definition of the pressure in a solid and ours; it is unclear what definition has been used.

right ($x \geq 0.5$). The initial data are as follows:

$$\begin{aligned} (\psi^A_i)_L &= \begin{pmatrix} \left(\frac{1.84}{\rho_0}\right)^{1/3} & 0 & 0 \\ 0 & \left(\frac{1.84}{\rho_0}\right)^{1/3} & 0 \\ 0 & 0 & \left(\frac{1.84}{\rho_0}\right)^{1/3} \end{pmatrix} & p_L &= 18.9, \quad v_L = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \\ (\psi^A_i)_R &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & s_R &= 0, \quad v_R = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \quad (4.21)$$

We use the matter space metric, $k_{AB} = \text{diag}(\rho_0^{2/3})$. In principle, we have some freedom with how we choose k_{AB} and ψ^A_i , as long as the correct density and shear scalars are produced as a result. In the fluid limit of the elasticity formulation, we have even *more* freedom, since only the density comes into the equation of state and has an effect on physical behavior. Our choice of k_{AB} is completely arbitrary, but is the reason for the slightly strange form of ψ^A_i in the fluid; we need to ensure that the resulting initial rest mass density is $\rho = 1.84 \text{ g/cm}^3$ in order to match the initial state of the published test. In the solid, we are given the initial value of the entropy, s , instead of the initial value of the pressure, p ; to handle this, we simply calculate the pressure from the entropy before the initial time step. The results are shown for a coordinate time $t = 0.05$ (which is $t = 0.5 \mu\text{s}$ given the units of the input parameters and initial data).

The results for this test are shown in Fig. 4.5. The blue markers show a 250-cell evolution, while the black line shows a 1000-cell evolution; for the lower resolution plot, only 1 in 3 points are shown for clarity. We see good agreement when our results are compared to those published in [83], aside from in the variable p . However, their P matches the results of the component of our stress tensor, p_{xx} , so we suspect that this may be due to a difference between our definitions for p in a solid; it is not clear what definition has been chosen. The results shown here were produced using the `ElasticEvolution` code; the `MultiModel` code is currently limited to the gamma-law equation of state for fluids.

4.4.4 Solid-fluid test: BOD2

We refer to the second solid-fluid test from [83] as *BOD2*. It consists of a moving, stressed solid in contact with a gas; again, the solid is copper, using the Cranfield equation of state. The gas is now completely unreacted PBX-9404; this again uses the JWL equation of state, and all of the material parameters are the same as for the previous test (aside from the parameter that tells the proportion of reacted to

unreacted gas). In this test, the solid is on the left ($x < 0.5$), and the fluid is on the right ($x \geq 0.5$). The initial data for this test are as follows:

$$\begin{aligned} (F_A^i)_L &= \begin{pmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{pmatrix} & s_L &= 0, & v_L &= \begin{pmatrix} 2 \\ 0 \\ 0.1 \end{pmatrix} \\ (\psi^A_i)_R &= \begin{pmatrix} \left(\frac{1.84}{\rho_0}\right)^{1/3} & 0 & 0 \\ 0 & \left(\frac{1.84}{\rho_0}\right)^{1/3} & 0 \\ 0 & 0 & \left(\frac{1.84}{\rho_0}\right)^{1/3} \end{pmatrix} & p_R &= 1 \times 10^{-4}, & v_R &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \end{aligned} \quad (4.22)$$

Again we use $k_{AB} = \text{diag}(\rho_0^{2/3})$ for the matter space metric; the resulting initial density in the fluid will again be $\rho = 1.84$. In the solid, p must again be calculated from s before the initial time step. In addition, in [83], the authors use the matrix inverse of ψ^A_i , F_A^i , as the dynamical variable, so their initial data are given in terms of this instead of ψ^A_i . To handle this, we simply invert the initial value for F_A^i before the initial time step.

Unfortunately, we have not been able to get the results for this test to match the published results. Although we have not yet been able to account for the discrepancy, we think it may be problem with the way we are relating the parameter, Y , in [83], which gives the fraction of the PBX-9404 that is reacted, to the parameters used in the JWL equation of state. Perhaps this correspondence is less straightforward than we had originally assumed.

4.4.5 Solid-solid test: slip

Our solid-solid slip test comes from [82]. It consists of two solids with different equation of state parameters that are allowed to slip freely against each other. Both solids are described by the Cranfield equation of state; the solid on the left is aluminum, while the solid on the right is copper. The copper equation of state parameters take the same values as in previous tests. The aluminum equation of state parameters are as follows: $\rho_0 = 2.71$, $c_0 = 6.22$, $b_0 = 3.16$, $C_V = 9.0 \times 10^{-4}$, $T_0 = 300$, $\alpha = 1.0$, $\beta = 3.577$, $\gamma = 2.088$. The initial data are given by the

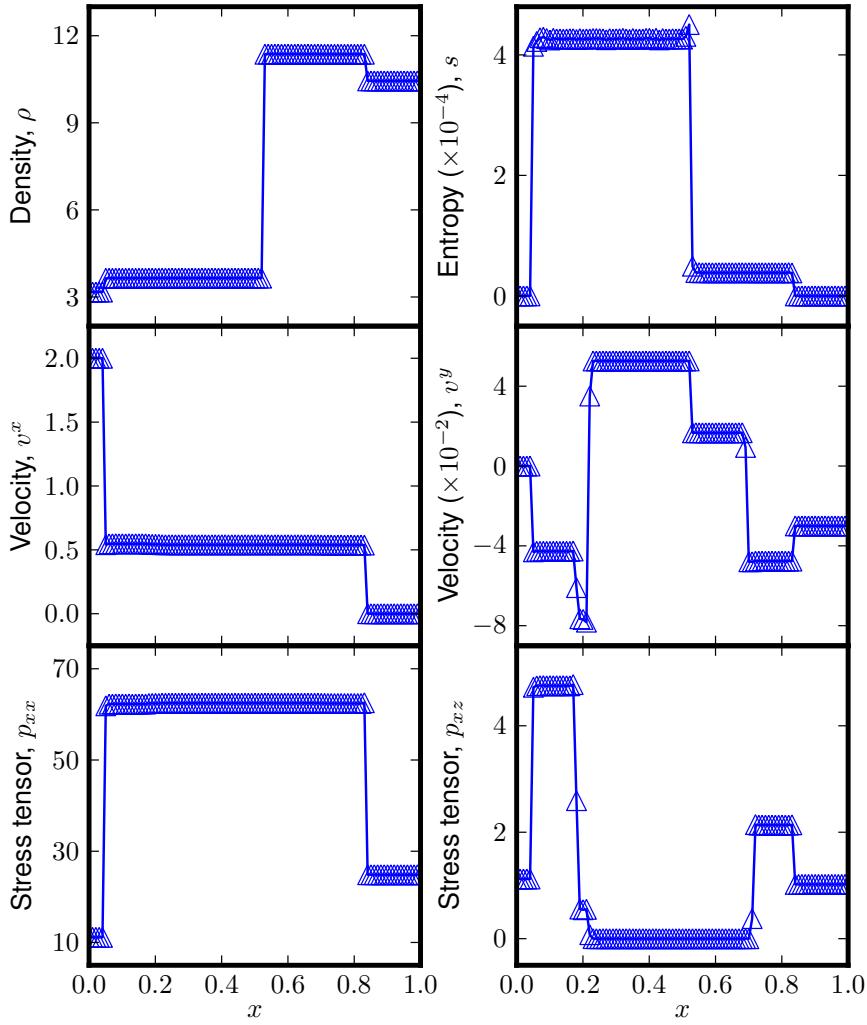


Figure 4.6: The solid-solid interface test using slip conditions published in [82]. These results were produced by an evolution using 1000 grid cells; the values in only 1 in every 10 cells is shown for clarity. The lines shown simply connect the cell values, again for clarity of plotting here. These results match well with the published results in [82].

following:

$$\begin{aligned}
 (F_A^i)_L &= \begin{pmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{pmatrix} & s_L &= 0, \quad v_L = \begin{pmatrix} 2 \\ 0 \\ 0.1 \end{pmatrix} \\
 (F_A^i)_R &= \begin{pmatrix} 1 & 0 & 0 \\ 0.015 & 0.95 & 0 \\ -0.01 & 0 & 0.9 \end{pmatrix} & s_R &= 0, \quad v_R = \begin{pmatrix} 0 \\ -0.03 \\ -0.01 \end{pmatrix}.
 \end{aligned} \tag{4.23}$$

We again use $k_{AB} = \text{diag}(\rho_0^{2/3})$ for the matter space metric. We also need to calculate p from s and $\psi^A{}_i$ from $F_A{}^i$ before the initial time step. The results are given at the coordinate time $t = 0.05$.

Our results are shown in Fig. 4.6. The blue markers show the values in 100 cells out of a 1000 grid cell evolution. The lines simply connect the points on the plot so that the various discontinuities can be seen more easily. Our results match well with those published in [82]. It should also be noted that our stress tensor, p_{ij} , corresponds to their stress tensor, σ_{ij} , as follows: $p_{ij} = -\sigma_{ij}$. These results were produced by the `ElasticEvolution` code, and have been reproduced in the `MultiModel` code.

4.4.6 Solid-solid test: stick

The solid-solid stick test also comes from [82]. The test consists of two solids that stick completely at the interface between them; since both materials are considered to be copper in this test, this is physically just a single material with discontinuous initial data. The initial data for this test are exactly the same as for the solid-solid slip test; the only differences are that both material are copper, and we use stick instead of slip interface conditions. The results are shown at coordinate time $t = 0.06$.

The results for this test are shown in Fig. 4.7. The blue markers show 100 of the 1000 cells in a high resolution evolution. The lines shown simply connect the markers so that discontinuities are more apparent. Our results show good agreement with those published in [82]. These results were produced by the `ElasticEvolution` code, and have been reproduced in the `MultiModel` code.

4.4.7 Solid-vacuum test

We have implemented the solid-vacuum test published in [82]. This test consists of stressed solid with a free surface, and the remainder of the computational grid is occupied by vacuum. The left hand region of the grid ($x < 0.5$) is occupied by the solid, which is taken to be aluminum, using the same equation of state parameters as in the solid-solid slip test. The initial conditions for the solid are as follows:

$$(F_A{}^i) = \begin{pmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{pmatrix} \quad s = 0, \quad v = \begin{pmatrix} 2 \\ 0 \\ 0.1 \end{pmatrix} \quad (4.24)$$

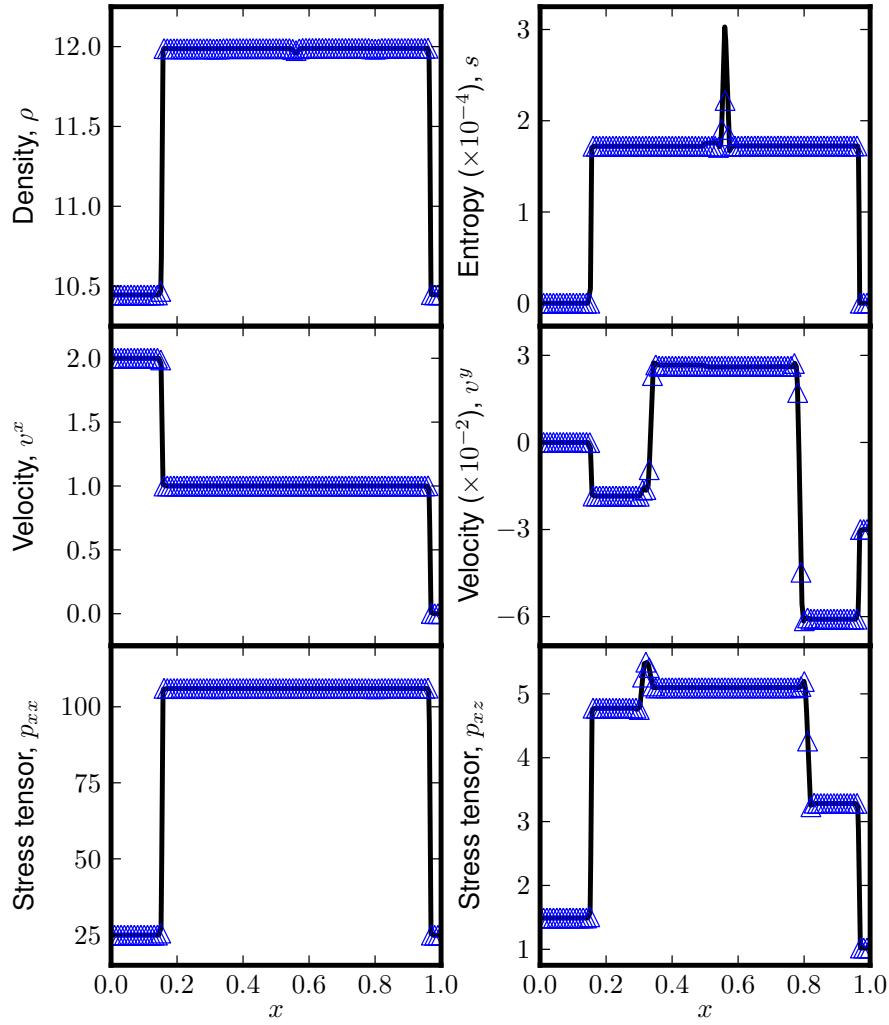


Figure 4.7: The solid-solid interface test using stick conditions published in [82]. These results were produced by an evolution using 1000 grid cells; only 1 in every 10 cells is shown for clarity, and are shown in blue. Since the interface here is trivial—both materials use the same equation of state with the same parameter values—we can compare these results to a single material evolution, shown in black. Both of these results also match the published results in [82].

Again we use $k_{AB} = \text{diag}(\rho_0^{2/3})$, and we calculate p and $\psi^A{}_i$ from s and $F_A{}^i$ before the initial time step. We show the results at the coordinate time $t = 0.06$.

The results for the solid-vacuum test are shown in Fig. 4.8. The blue markers show 100 out of 1000 grid cells used for evolution, while the lines connect these markers to make any large discontinuities readily distinguishable. All vacuum cells have been removed, so the interface between the solid and the vacuum will be at the last grid cell (to the right) shown on the plot. The values of the constant states, as well as the wave speeds and the location of the interface match well

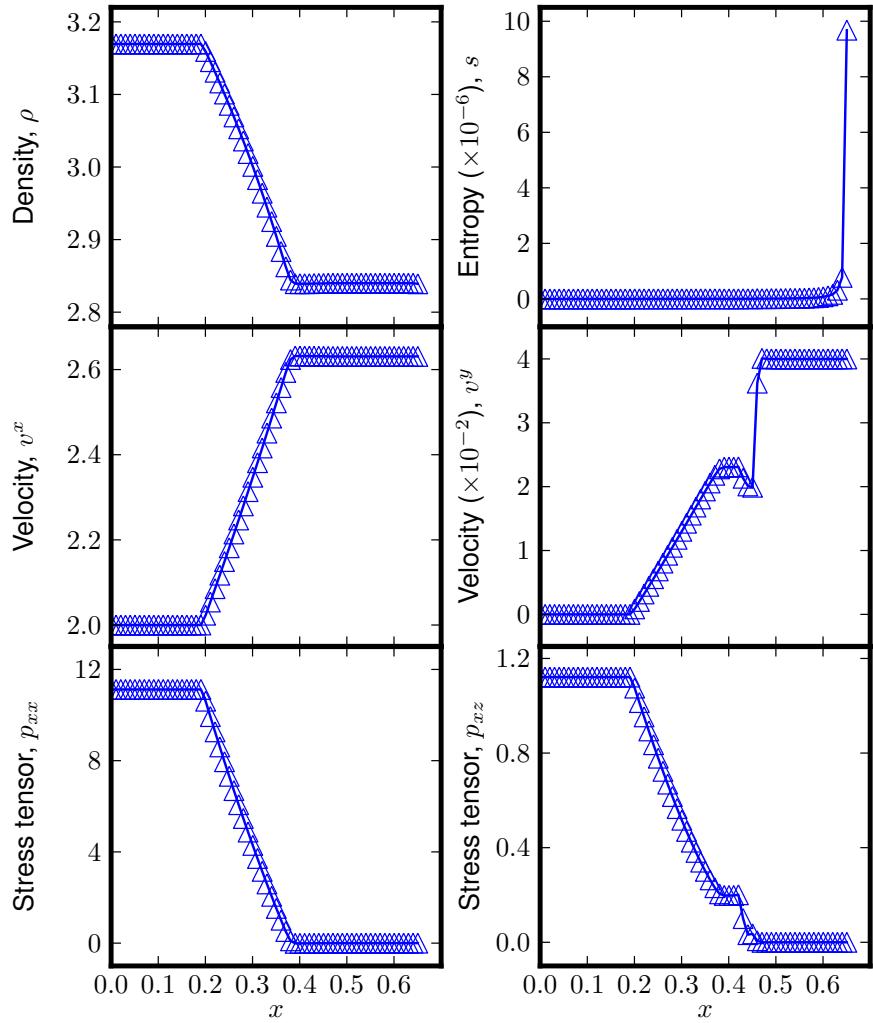


Figure 4.8: The solid-vacuum test published in [82]. This evolution used 1000 grid cells, but only 100 are shown for clarity. Here, instead of simply advecting the level-set function, we must evolve it instead using the velocity in the cell immediately to the solid side of the interface; when this is done, the results match well with the published results for this test. We note that the seemingly large error in entropy is not visible in the other variables (i.e. the density), because the thermal term in the equation of state is still much smaller than the other terms for this test. A similar increase in entropy at the vacuum surface is seen in the results published in [82].

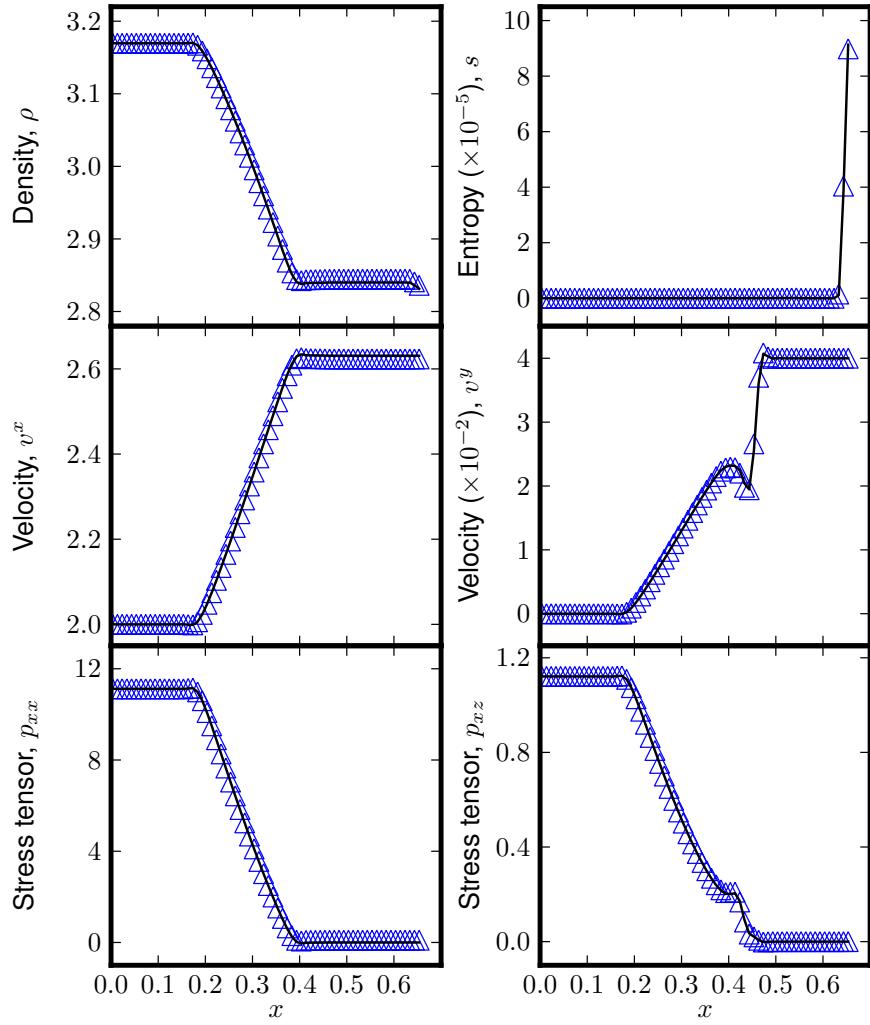


Figure 4.9: A solid-atmosphere evolution of the solid-vacuum test published in [82], compared to the solid-vacuum results. These evolutions used 1000 grid cells, but only 100 are shown for clarity. The blue triangles show the results of a solid-atmosphere evolution with an atmospheric pressure of 10^{-1} , compared to an initial pressure in the solid of ~ 18 ; the results are reasonably close to the results of the solid-vacuum evolution, shown in black. It is also possible to see the larger entropy and corresponding dip in density due to overheating in the `MultiModel` code; these are not present in the `ElasticEvolution` code results because of the presence of the isobaric fix, as described in Section 4.2.10.

between our results and those published in [82]. These results were produced by the `ElasticEvolution` code, and have been reproduced in the `MultiModel` code.

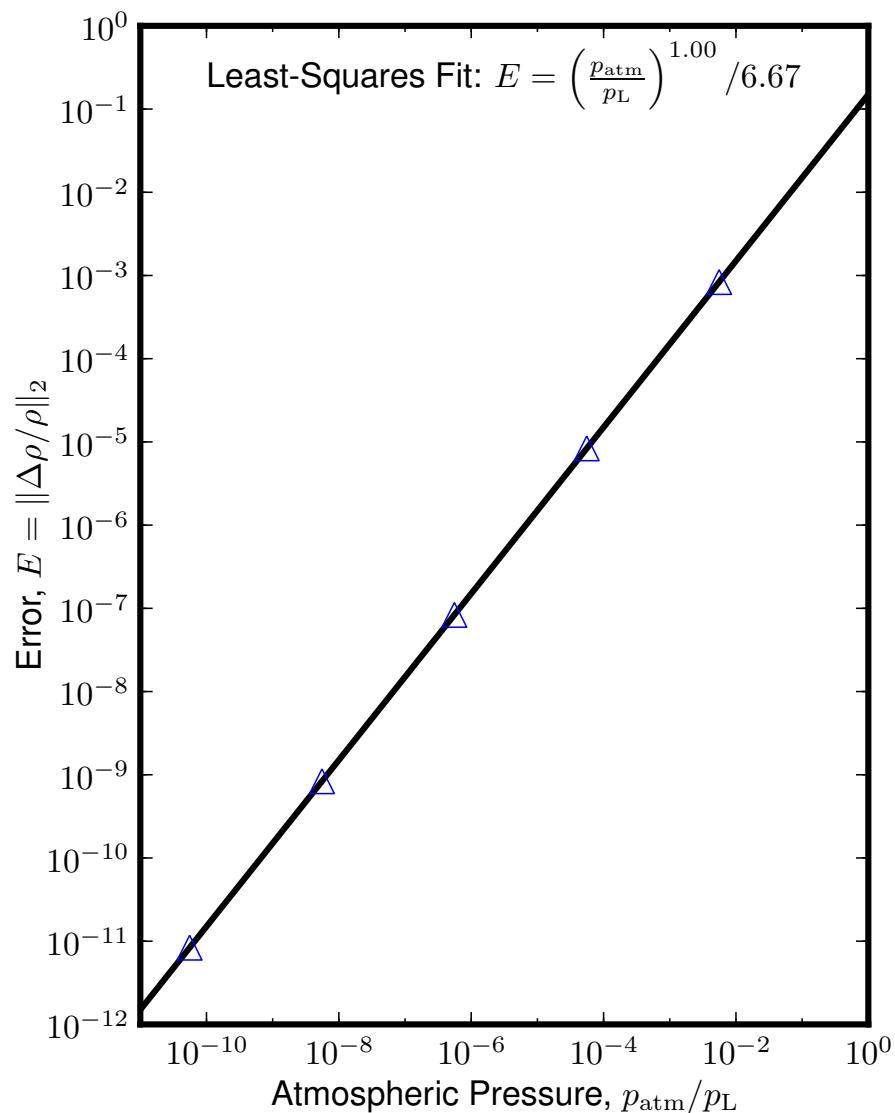


Figure 4.10: Error between the solid-atmosphere test and the solid-vacuum test as a function of atmospheric pressure. All of the tests were performed with 1000 grid cells; the error is computed by finding the L^2 -norm of the relative error in the density, as described in Appendix D. The error is shown as the blue triangles, and a least-squares fit is shown in black. The overall error from the vacuum solution is linear in the atmospheric pressure.

4.4.8 Solid-atmosphere test

To test the accuracy of atmosphere interfaces, we have adapted the solid-vacuum test presented above to the solid-atmosphere scenario. The initial data are exactly the same as above aside from the fact that there is an atmosphere in the right-hand region of the grid. We test various atmospheric pressures to assess the impact of this parameter on the results.

The results for the solid-atmosphere test are shown in Fig. 4.9. An atmospheric pressure of 10^{-1} (compared to an initial pressure of ~ 18 in the solid) is shown along with the vacuum result. This plot was produced using the `MultiModel` code as opposed to the `ElasticEvolution` code, which was used to produce the results shown in Fig. 4.8. The `MultiModel` results show the increased entropy and dip in density characteristic of overheating; these features are not present in the `ElasticEvolution` results because of the isobaric fix implemented there (a description of overheating and the isobaric fix is in [85]. Fig. 4.10 shows the error between the vacuum and atmosphere results as a function of atmospheric pressure. It is clear that as atmospheric pressure is decreased, the results approach the solid-vacuum result. A least-squares fit of the L^2 norm of the error shows that it is linear in the atmospheric pressure.

This test shows that the atmosphere is able to approximate the vacuum well for situations where the vacuum results are well-behaved, but we have not yet been able to quantify how well the atmosphere treatment works in situations where the vacuum interface fails. We can force stellar-surface type tests to *run* using an atmosphere with a suitably high atmospheric pressure, but we do not have an estimate as to whether or not the results are actually physically *correct*. However, as of yet, we have no better method for approximating a stellar-surface type solid-vacuum interface, so we will use this method for our shattering simulations.

4.4.9 2D Riemann test

To test the 2D capabilities of the `MultiModel` code, we extended one of the 1D Riemann problems from [82] to 2D using the method described in Section 3.5.7. Basically, the initial discontinuity is set at an angle to the grid, and the vectors and tensors are rotated accordingly.

We first extended the solid-solid slip test to 2D, using a 20 by 160 grid and a 20 by 800 grid. The results for this test, compared to the 1D results (produced by the `ElasticEvolution` code) are shown in Fig. 4.11. The 1D and 2D results look qualitatively similar, but some waves are smeared out in the 2D results, similar to the effects seen in Section 3.5.8 (i.e. see Fig. 3.14). As in that section, the 2D

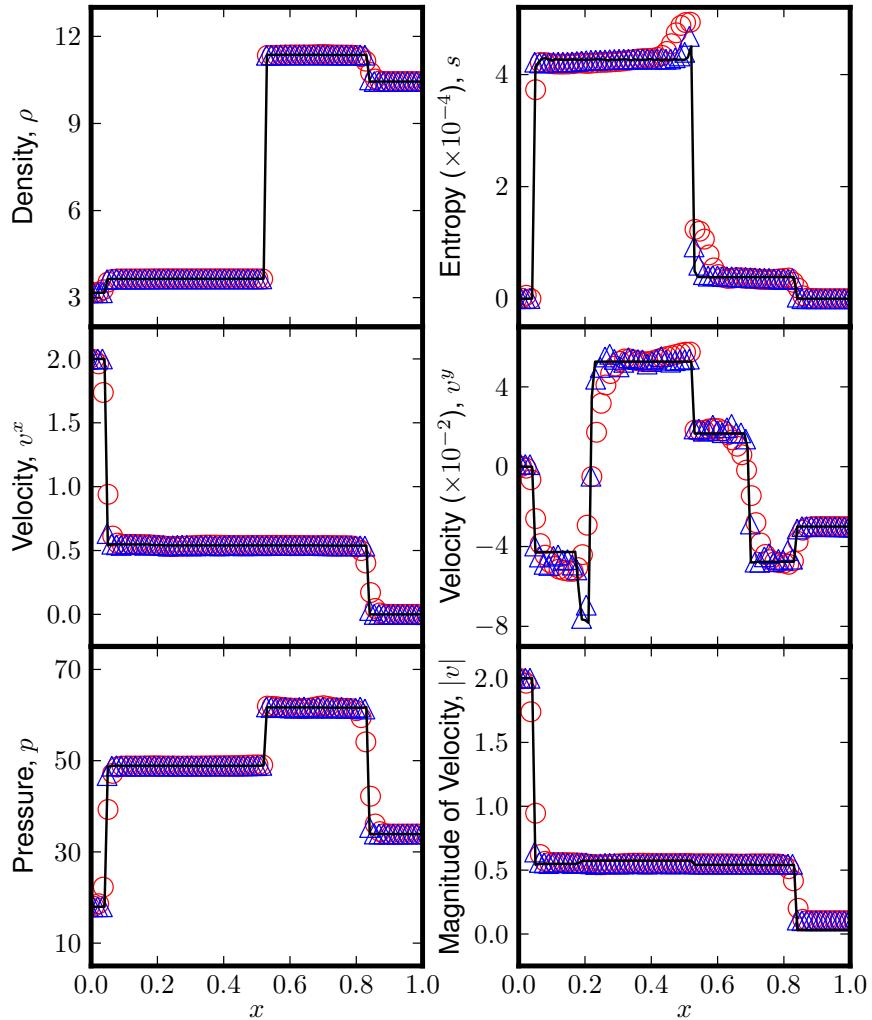


Figure 4.11: A 2D evolution of the solid-solid slip test published in [82], compared to the 1D results. The 1D evolution used 1000 grid cells, and is shown in black; the 1D Riemann problem is extended to 2D using the method described in Section 3.5.7. Results of a 20 by 160 grid are shown in red, and results of a 20 by 800 grid are shown in blue. Qualitatively, the 2D results are similar to the 1D results; however, there is some smearing of some features. Still, as resolution increases, the 2D results appear to converge to the 1D results. Again, the 2D results show a jump in the entropy near the interface. This is likely due to the overheating, which is evident in the MultiModel code.

results approach the 1D results as the resolution increases. We also see increased entropy at the interface between the two solids; this is likely due to overheating, which is present in the MultiModel code.

4.4.10 Conservation errors

The ghost fluid method is not conservative. This is due to two issues. First of all, the flux to the left and right of the interface between two different materials will not be the same. To understand this, consider the mass continuity equations. The flux in the x direction is given by ρv^x ; if the interface is normal to the x direction, then v^x will be the same in the real fluid and ghost fluid at the same location, but ρ can differ, so the flux will generally differ. Essentially this means that the flux flowing out of one cell is not the same as the flux flowing into the adjacent cell, leading to a loss of conservation.

Another contribution to the loss of conservation is advection of the level-set function. First of all, the location of the interface can only be as accurate as the method used to advect the level set. We also know that, while the actual location of the interface will be somewhere between two cell centers, the GFM assumes that the interface is located exactly at the boundary between the two cells. This contributes periodic errors in conservation by over- or underestimating the volume occupied by each material, depending where the interface is located relative to the boundary between the cells.

To determine how these conservation errors affect our results, we have evaluated the error in conservation by comparing the conserved quantities integrated over the computation grid to their initial integrated values evolved forward in time using the fluxes at the edges of the grid. As an example, we look at how we calculate the conservation error in the situation where we use a simple forward-in-time time integrator. The error in conservation is computed as follows:

$$E = Q - \sum_{\text{grid}} q. \quad (4.25)$$

Q , the initially integrated and then evolved conserved quantity, could, for example, be evolved using simple forward-in-time differencing as follows:

$$Q^{n+1} = Q^n + \frac{\Delta t}{\Delta x} (F_{np+\frac{1}{2}} - F_{\frac{1}{2}}), \quad (4.26)$$

where the indices $i = \frac{1}{2}$ and $i = n + \frac{1}{2}$ are the edges of the grid, $F_{i-\frac{1}{2}}$ is the numerical flux through the left edge of the i th cell, and Q is initialized as

$$Q^0 = \sum_{\text{grid}} q^0. \quad (4.27)$$

If simple forward-in-time differencing is used to evolve Q , then it should also be used to evolve the conserved quantity, q , as follows:

$$q_i^{n+1} = q_i^n + \frac{\Delta t}{\Delta x} (F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}). \quad (4.28)$$

In general, we will not use forward-in-time differencing for the time integration. Instead, we use a Runge-Kutta method. When an RK method is used, the flux terms above are simply used as an input for the RK method. We use the same RK method for the time evolution of Q and q .

When we calculate the cumulative error in conservation for the single-model Riemann tests discussed in Section 3.5, we see that this error typically does not go above the level of round-off error for Newtonian tests.² However, when we look at the error in conservation for our interface tests using a level-set function with the GFM, we see that the error is much larger, and is periodic in time. These results are shown for the BOD1 interface test (solid-fluid interface) in Fig. 4.12.

As mentioned above, there is no aspect of the algorithm that identifies or uses the exact location of the interface, so from a computational perspective, the interface can only be located at the boundary between two cells; however, physically, the location of the interface varies continuously in space. This error in the position of the interface causes an error in the overall conservation of the system; the error is periodic as the interface moves from one cell boundary to the next. This is supported by the fact that the error converges with resolution, and that the period of the error also decreases linearly as the grid spacing decreases, as can be seen in Fig. 4.13.

In Fedkiw *et al* 1999 [75], the convergence rate for the error in conservation is estimated for a helium bubble advected in air. Because of the nature of the advection problem, the GFM achieves the exact state in each fluid, and the only contribution to the error in conservation is the advection of the level-set function. With this test, they observe second-order convergence with resolution. Because

²We expect that the error should not go above the level of round-off error for *any* single material tests, but we have a bug that prevents this for relativistic tests. Although this is unresolved, it is not relevant to our discussion of the conservation error for Newtonian tests, and is not integral to the overall functioning of the code. A description of this bug is given in Appendix E.

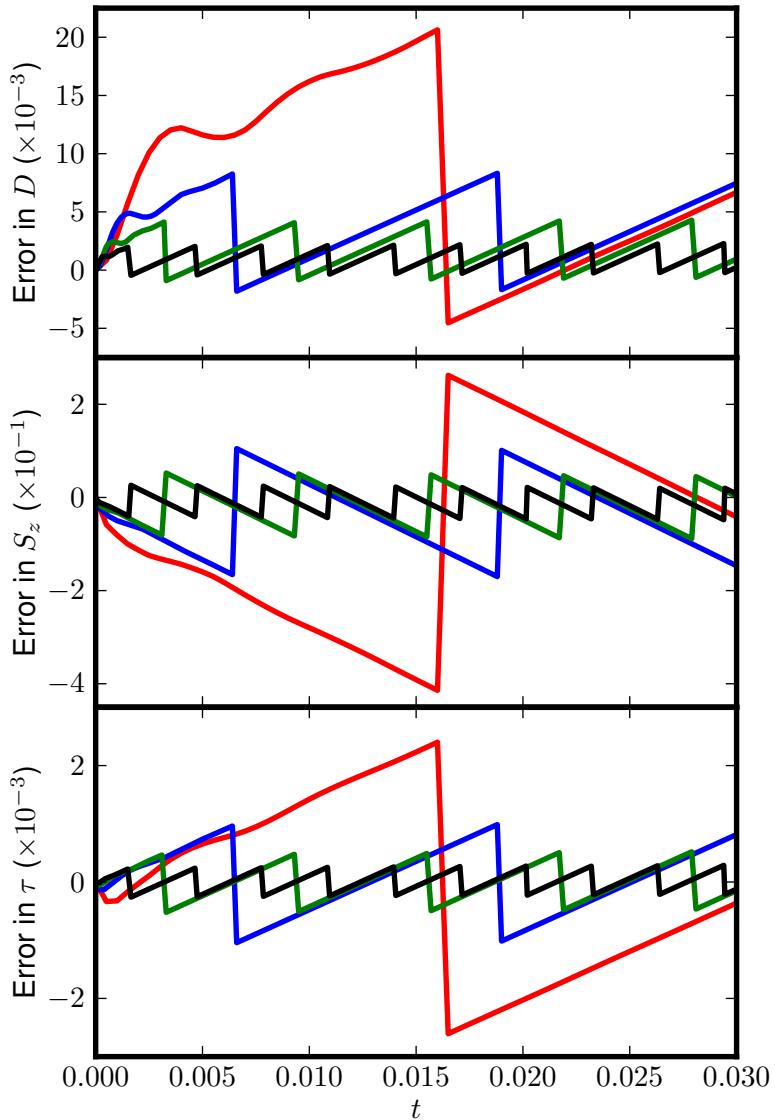


Figure 4.12: The cumulative error in conservation over time for the BOD1 interface test. The error in conservation is defined by Eq. 4.25 and the following equations. The red line shows the error as a function of time for a 100 grid cell evolution; the blue, green, and black lines are from evolutions using 250, 500, and 1000 grid cells, respectively. By eye, one can see that the error converges by somewhere between first and second order in the resolution, and the period of the error also decreases linearly with a decrease in grid spacing. The error is periodic, because of the contribution from the discrepancy between the physical location of the interface and the location assigned by the level-set/GFM algorithm, which can only be at cell edges.

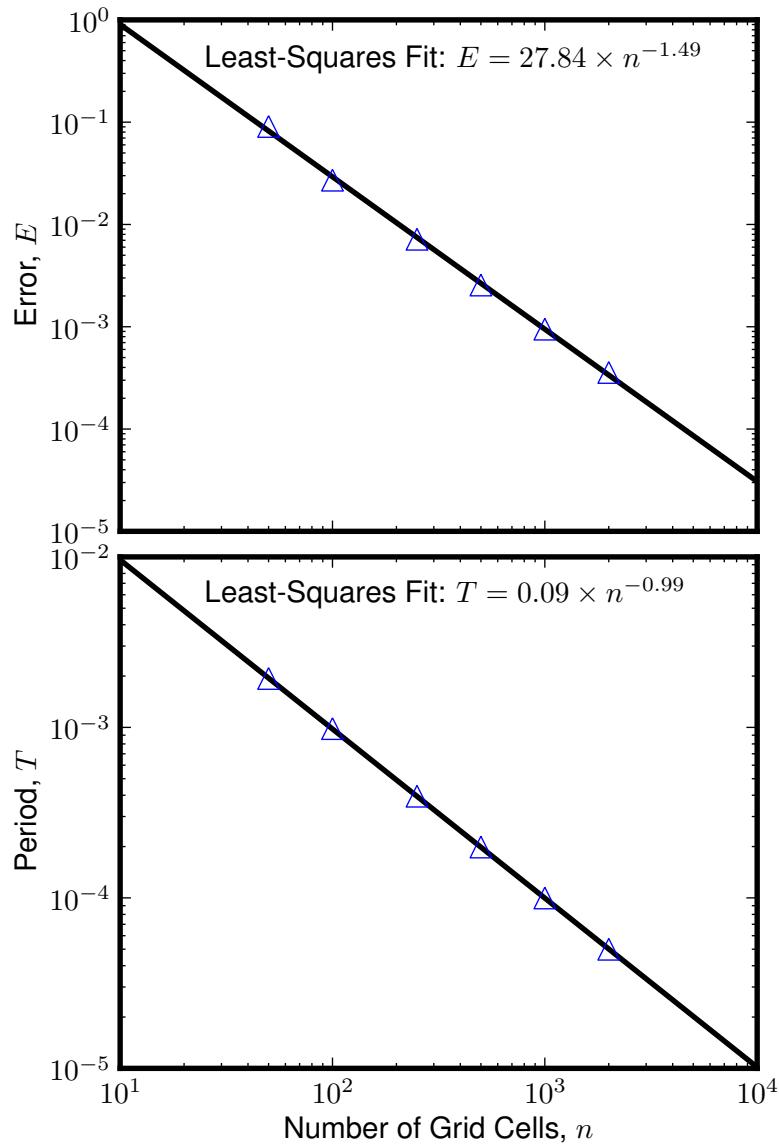


Figure 4.13: The top plot shows the convergence of the cumulative error in conservation of S_z as the number of points on the computational grid increases (as the grid spacing, Δx , decreases). Each blue markers shows the average error over an evolution (the root mean square value over time), and the black line shows a least-squares fit to the data. Using the least-squares fit, we see that the error converges at an order of about 1.5 in the resolution. The bottom plot shows the decrease in the period of the error in conservation of S_z with resolution: the blue markers are the period for each evolution and the black line is a fit. The fit shows that the period decreases by $1/n$, and therefore increases linearly with increasing Δx , as expected. The period was found by finding the length of the first half-period of the evolution, and multiplying by 2.

our test *will* be affected by both sources of error, it is to be expected that the order of convergence should be somewhat lower than this value—as shown in Fig. 4.13, we measure a convergence order of ~ 1.5 .

Chapter 5

Shattering

As we know, a neutron star has a solid crust and a fluid core. There are various different mechanisms that could break the crust, including the decay of magnetic fields in the star, spin down of a rotating star, and tidal forces due to a companion star in a binary merger system. In fact, it has been suggested that starquakes play a role in the mechanisms behind observed effects such as pulsar glitches, quasi-periodic oscillations after magnetar flares, precursors to short-hard gamma-ray bursts, and flares and outbursts from soft gamma-ray repeaters and anomalous x-ray pulsars.

In the following chapter, we discuss how breaking of the crust is likely to occur, how it is achieved in our model, and the results of numerical tests.

5.1 Earthquakes

It is possible that the neutron star crust material will fail by cracking and slipping along surfaces. This is the type of breaking that you would intuitively think of if you were making the analogy between an earthquake and a starquake: the crust breaks along some fault line, slips, and causes waves to propagate away from this initial slippage area.

We could imagine that it might be easy to simulate cracking in our model using the interface methods described in Chapter 4. At a certain point in the simulation, we could simply split the numerical grid into two separate solids, and allow these two solids to slip against one another at that interface. We could even include a small vacuum region between the two solids, which would allow voids to form at the crack.

However, this simple treatment of cracks brings up some obvious questions. When should the crack form? Would the crack be instantly infinite (or surrounding some limited region) or would we need to transition between slipping behavior at

the center of the crack and sticking at the edges? If the crack *does* have edges, how would these edges propagate through the crust? There is extensive literature about crack formation and propagation in terrestrial solids in the Newtonian regime, but more study would be needed to determine how this should be applied to relativistic systems.

In any case, evidence suggests that this type of breaking will *not* occur in the neutron star crust. Molecular dynamics simulations show that, because of the high density of the material, localized features are not present, and therefore, cracks do not tend to form [11]. These results agree with an earlier prediction by Jones 2003 [12], arguing that any cracks that form would immediately be rehealed by the high pressure in the system.

Levin and Lyutikov 2012 [15] point out that the length scale of these simulations is much smaller than the size of a neutron star, and suggest that catastrophic failure on a microscopic scale could look like crack formation on a macroscopic scale; however, their own work shows that even if cracks *do* form, the magnetic fields permeating the crust essentially hold the two sides of the crack in place, allowing very little slippage to occur, and not much elastic energy is released. For these reasons, we believe that it is more likely that the crust will fail in a catastrophic, global way.

5.2 Shattering

Another scenario for the failure of the neutron star crust is *shattering* or catastrophic failure in some volume of the crust. This type of failure is suggested by the molecular dynamics simulations of Horowitz and Kadau [11], which show that this type of catastrophic failure is favored over cracking. This is roughly because the screened Coulomb interaction in their simulations has no explicit length scale (i.e. the system at twice the density behaves like the system at the original density at a lower temperature), and particles can interact with other particles that are a large distance away from them. This arises from the high density in the neutron star, which means that the electrons are not associated with particular nuclei, but instead form an electron gas that permeates the crystal evenly. This means that the material fails abruptly in a collective fashion, whereas, for example, terrestrial metals can have localized features which allow localized defects to appear [11].

Horowitz and Kadau also see that molecular dynamics simulations starting with a cylindrical hole initialized in an otherwise perfect crystal quickly heal due to the high pressure in the system [11]. This agrees with the predictions in [12]. All of this suggests that, once a region has shattered, it is expected that it will refreeze

almost instantaneously due to the high pressure in the system. The shattering releases any shear stresses present so that when the material has refrozen, the stress is completely isotropic.

Although the shear stresses are released as a result of both, shattering is different from *melting*. Penner *et al* [86] show that the built up local strain required to melt the crust upon release of the elastic energy was

$$u \gtrsim 0.5 \left(\frac{\alpha}{5} \right)^{1/2}, \quad (5.1)$$

which is greater than the breaking stress estimates in [11]. This means that, while we expect catastrophic *shattering* of the crust, we *do not* anticipate melting of the crust.

In our model, what we mean by *shattering* is the instantaneous relaxation of the material (i.e. the removal of all shear stresses). Recall, from Section 3.2, that the unsheared state occurs when the matter-space metric is proportional to the spacetime metric pushed forward onto matter space:

$$k_{AB} = n^{2/3} g_{AB}. \quad (5.2)$$

All we need to do to achieve instantaneous relaxation is to reset quantities in order to make the above expression true. To do this, we can either reset the matter-space metric, k_{AB} , or the configuration gradient, $\psi^A{}_i$; since it is much simpler to reset k_{AB} , this is what we choose to do in practice.

It is also worth noting that, when we reset the matter-space metric, the density remains constant in the shattered cells; generally, we can only keep one thermodynamic quantity constant (i.e. entropy, internal energy, or pressure), since the relationship between them depends on the shear scalar, which will change at shattering. Physically, the specific internal energy, ϵ , is the variable that should be kept constant at shattering due to conservation of energy. The pressure and entropy are then allowed to change. For physically reasonable equations of state, the reduction in the shear scalar associated with shattering will need to be matched by an increase in entropy in order to keep the internal energy constant; this represents some of the elastic energy being converted into heat, which is what we expect physically. A missing element in this model is some mechanism for the dissipation of that localized heating.

5.3 Results

We have implemented a few shattering examples to demonstrate the method's effectiveness in neutron star simulations. First we shatter a circular region in a homogeneously stressed block of solid. Next we set up a star-like toy model with a crust, core, atmosphere, and background potential, and then shatter a small region of the crust. Details about the initial data for these systems, as well as the results, are in the following sections.

5.3.1 Homogeneous initial data

Our first test starts with a stressed block of copper, with homogeneous but non-zero anisotropic stress, and then a circular region at the center of the 2D computational grid is shattered, and refrozen in the relaxed state. This is a largely artificial situation, with limited physical interpretations, but you could think of a block of copper that is stressed in some way (say in some vice grips, but with the shear stresses non-zero, so maybe one edge is being pushed one way and the other in the other direction), and then, artificially, the shear stresses are suddenly relieved in some circular region in the center. The Cranfield equation of state (Appendix C.2) is used, with the material parameters for copper as discussed in Section 4.4. The initial data are as follows:

$$F_A^i = \begin{pmatrix} 0.98 & 0 & 0 \\ 0.02 & 1 & 0.1 \\ 0 & 0 & 1 \end{pmatrix} \quad s = 0.001, \quad v = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (5.3)$$

The matter-space metric is set (before shattering) to $k_{AB} = \text{diag}(\rho_0^{2/3})$. The central circular region ($x^2 + y^2 < 0.25$) is then shattered as discussed in Section 5.2, and the system is allowed to evolve.

Shattering results in a discontinuity in the components of the matter-space metric, k_{AB} . If we choose to evolve this shattering simulation using a single-material evolution (i.e. no level-set function or ghost-fluid method), then advection of the matter-space metric causes this discontinuity to smear unphysically, which causes large errors in the density and the entropy. To avoid these errors, we can simply use two different materials: one for the shattered region and one for the unshattered region. The ghost-fluid method for elasticity has us extrapolate the matter-space metric components into the ghost fluid; this means that the method will perfectly preserve the initial values of the matter-space metric, and prevent

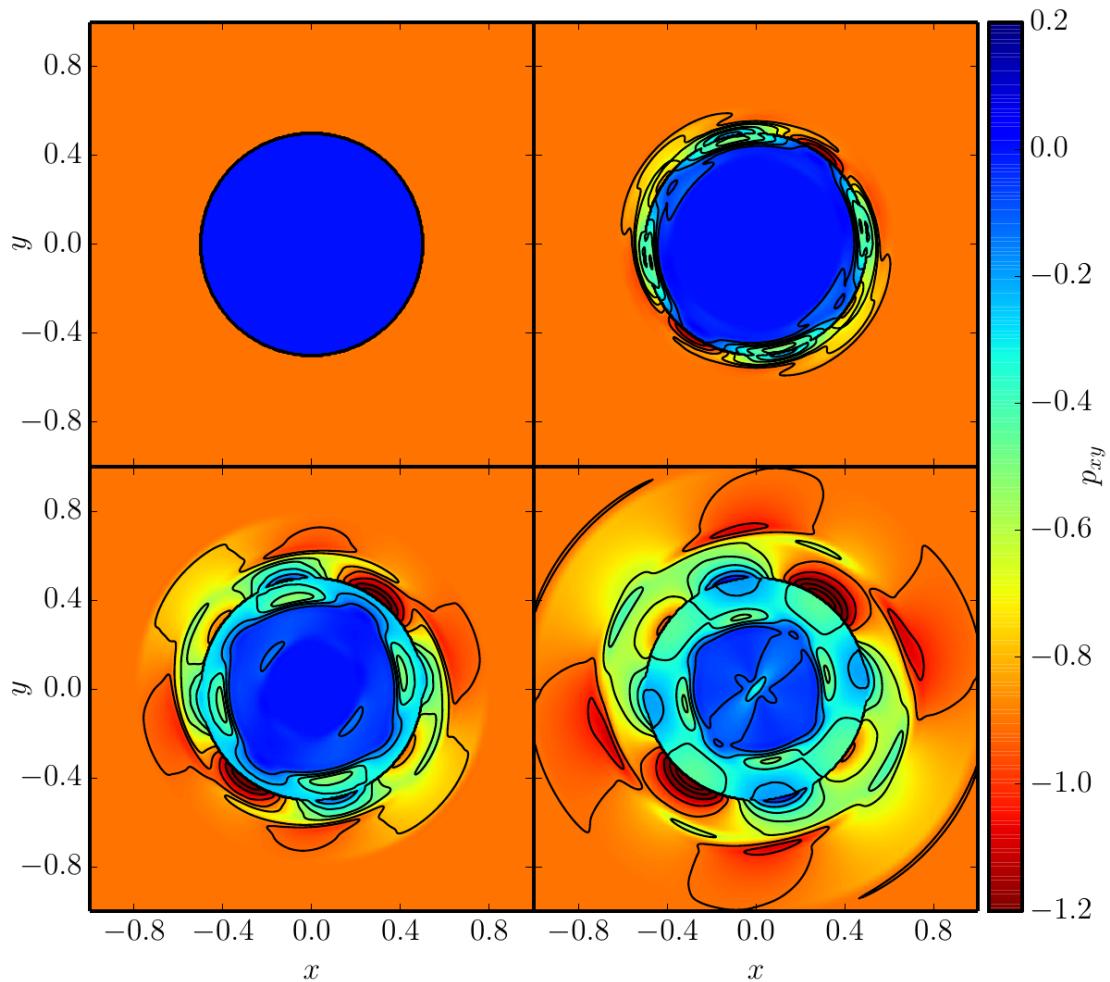


Figure 5.1: The shear stress component, p_{xy} , of the stress tensor at four times during a shattering evolution. The initial time step shows the component immediately after shattering; in the outer regions, the shear stress is homogeneous, but non-zero, but in the circular region where shattering has occurred, the shear stress has been reset to zero. In the subsequent time steps, waves are seen propagating away from the shattered region; no numerical problems were encountered as this occurred. The simulation was produced by the `MultiModel` code using two materials and a grid resolution of 400×400 ; the shown coordinate times are $t = 0.00$, $t = 0.02$, $t = 0.05$, and $t = 0.09$.

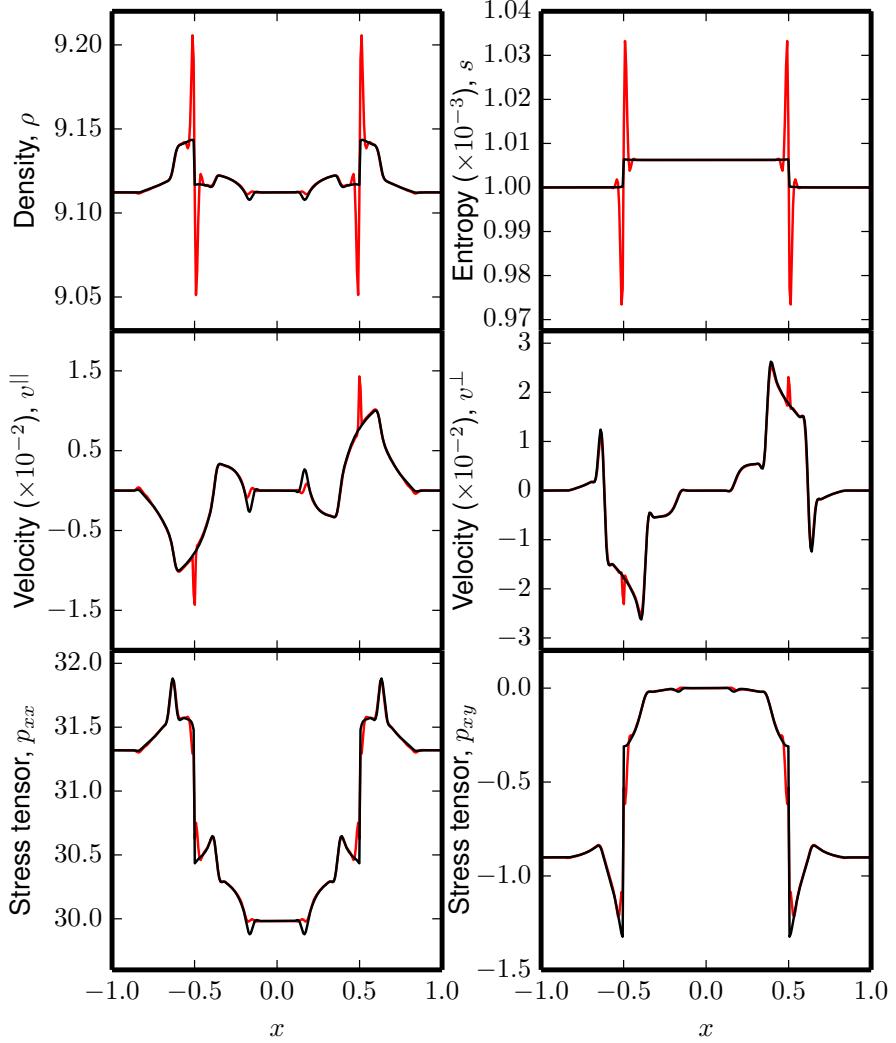


Figure 5.2: The results for a 1D slice along the diagonal, $y = x$, are shown for the initially homogeneously stressed shattering simulation. The results are shown at coordinate time $t = 0.06$, and were produced using the MultiModel code and a grid resolution of 400×400 . Examining the 1D results allows us to distinguish the different waves types. The velocity is split into components parallel to and perpendicular to the slice so that shear and pressure waves can be distinguished more readily. For example, a fast pressure wave can be seen around $x = \pm 0.85$ and $x = \pm 0.15$ in ρ , v^{\parallel} , and p_{xx} , while a slightly slower shear wave can be seen at $x = \pm 0.63$ and $x = \pm 0.37$ in v^{\perp} and p_{xy} . In this plot, we also compare a single model evolution (red) to an evolution where we treat the shattered and unshattered regions as separate materials (black). Using separate materials prevents large errors in the density and entropy, caused by smearing of the matter-space metric, k_{AB} , during advection.

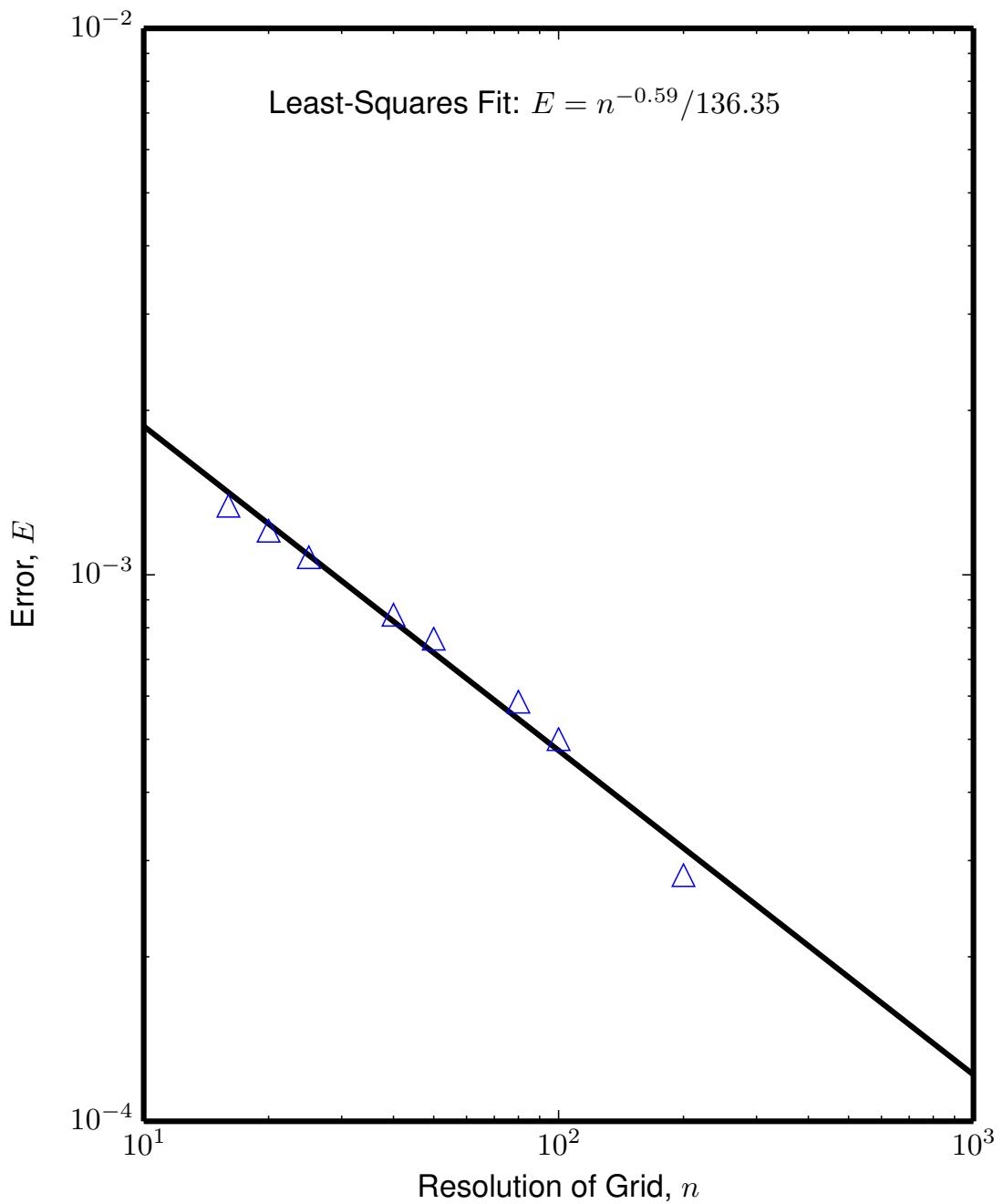


Figure 5.3: The convergence of the homogeneous shattering simulation. The same initial data were used for simulations run on square grids of various resolutions ranging from 16×16 to 200×200 . These were then compared to a high resolution (400×400) run; the measure of error is as described in Appendix D. The error in the density was evaluated at coordinate time $t = 0.06$. This is plotted for various grid resolutions. A least-squares fit shows that the convergence is of order -0.59 in n , where the grid is $n \times n$.

all smearing. The results shown here use a two-material evolution to eliminate these errors.

The results are shown in Fig. 5.1. This figure shows a shear-stress component, p_{xy} , at coordinate times $t = 0.00$, $t = 0.02$, $t = 0.05$, and $t = 0.09$. Initially, the shear stress component is homogeneous and non-zero, aside from the central region, where it has been reset to zero by the shattering process. In the subsequent time steps, waves can be seen propagating away from the initial discontinuity between the shattered and unshattered region. At the final time shown here, the fastest waves have interacted at the center of the grid, and have reached the boundaries.

It is easier to examine the wave structure of the numerical results by looking at a 1D slice of different variables. Such a slice was taken along the diagonal, $y = x$, and the values of several variables are shown at $t = 0.06$; these are shown in Fig. 5.2. A single-material evolution is shown (red), and compared to the two-material evolution (black). The large errors in the density and entropy in the single-material evolution are due to smearing of the matter-space metric components; these are eliminated by using two materials instead of one. From these plots it is possible to see fast-traveling longitudinal (pressure) waves, which travel at the sound speed and are characterized by jumps in ρ , v^{\parallel} , and p_{xx} at around $x = \pm 0.85$ and $x = \pm 0.15$. It is also possible to see slower-traveling transverse (shear) waves (traveling at the shear velocity) in v^{\perp} and p_{xy} at around $x = \pm 0.63$ and $x = \pm 0.37$. Although we do not calculate the wave speeds exactly in the `MultiModel` code, we can nevertheless see that the wave speeds traveling inward are approximately the same as the wave speeds traveling outward, as the waves are equal distances from the initial discontinuity (at $x = \pm 0.5$).

It is also possible to see the increased heat in the central region due to the shattering; as mentioned before, a release of elastic energy means that there should be an increase in the thermal energy in that region.

Fig. 5.3 shows a convergence plot of the relative error in the density between a high resolution run and lower resolution runs. The error is calculated at coordinate time $t = 0.06$. The plot clearly shows that, as the grid resolution increases, the results are converging; a least-squares fit shows us that the order of convergence is -0.59 .

Nominally, the methods used for this simulation should converge at second order in the resolution. Since the observed order of convergence is much lower than this, we examine a simpler test to try to illuminate what could be causing this low convergence order.

To do this, we construct a 1D shattering simulation with initial data equivalent to a slice along the diagonal, $y = x$, of the 2D homogeneously stressed shattering

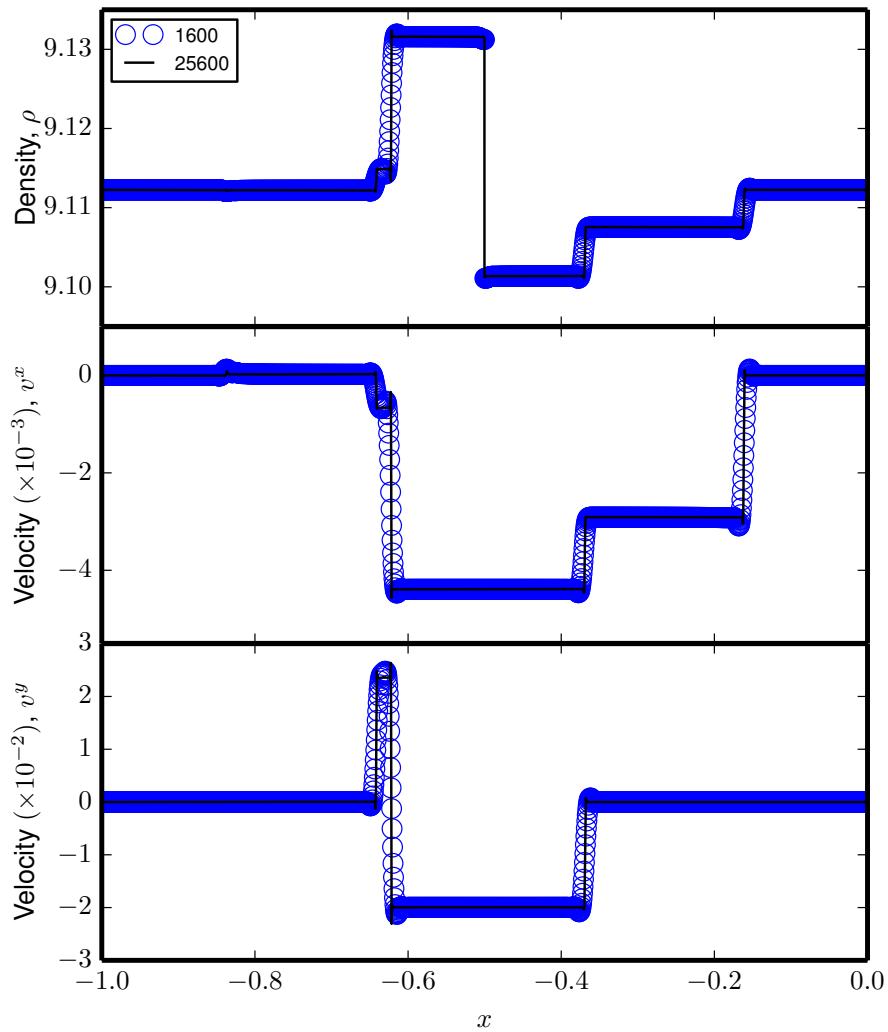


Figure 5.4: A 1D homogeneously stressed shattering simulation with initial data equivalent to the left half of a diagonal slice along $y = x$ of the 2D shattering simulation. It is evolved in 1D with planar symmetry. This is a good approximation of the diagonal slice at early times in the evolution, when the curvature of the discontinuity has not yet begun to affect the shape of the waves. We study the convergence of this 1D test as a comparison to the 2D results. Here a grid-resolution of 1600 cells (blue circles) is compared to a high-resolution simulation using 25600 cells. The wave structure of ρ , v^x , and v^y , here can be compared to the wave structure of ρ , v^{\parallel} , and v^{\perp} in Fig. 5.2. With the higher resolution here, we can see not only the pressure waves at around $x = -0.85$ and $x = -0.15$, but we can also see that, while we have one shear wave traveling into the shattered (unsheared) region at around $x = -0.37$, we actually have *two* shear waves traveling into the unshattered (sheared) region centered around $x = -0.63$. This is precisely due to the degeneracy in the shear waves discussed in Section 3.4.5.

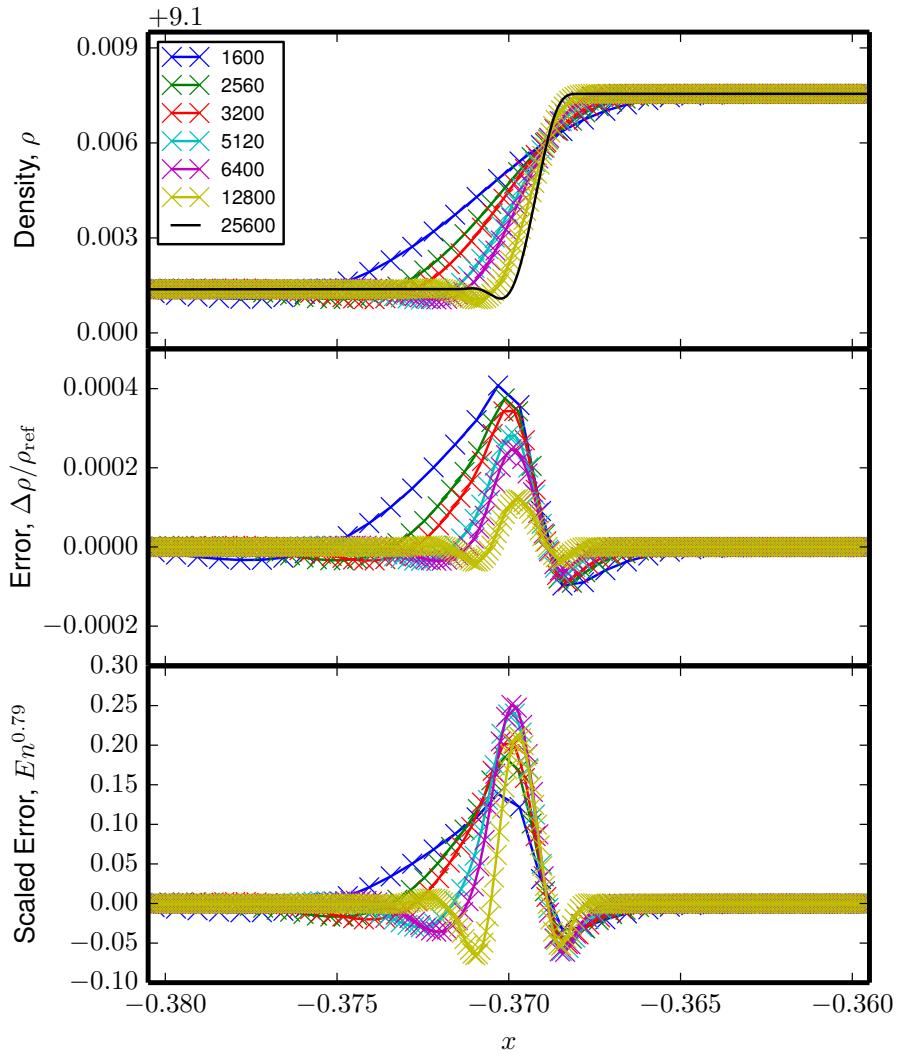


Figure 5.5: A close up of the density over the right-traveling shear rarefaction wave for the 1D homogeneously stressed shattering simulation. The rarefaction wave is shown for various resolutions; it is clear that the solution is converging in this area. We can also see this when we look at the relative error (measured against a high-resolution run using 25600 cells). We also see Richardson scaling of this error when we scale the error using a power of the resolution; however, the power necessary for this scaling is somewhat lower than the nominal second order convergence rate for the methods used. Here, the scaling power appears to be around 0.79, which matches the fit of the error for the higher resolution simulations shown later in Fig. 5.6.

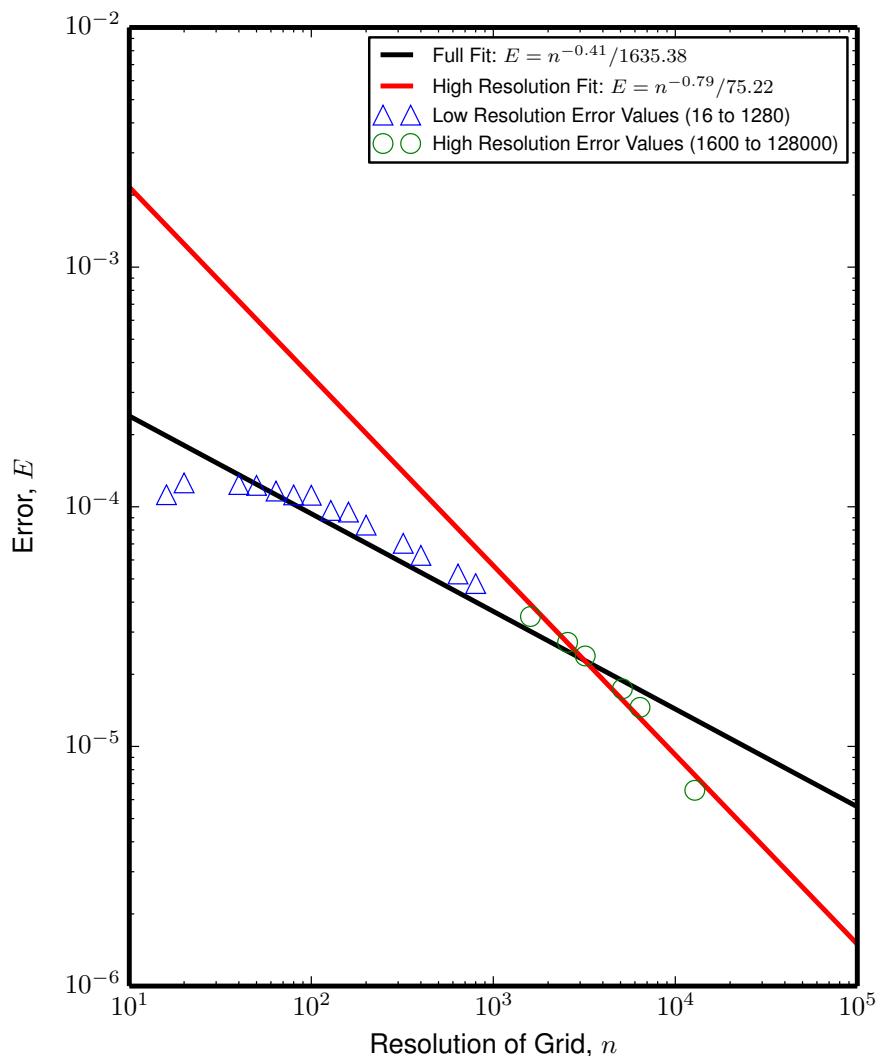


Figure 5.6: The convergence of the 1D homogeneously stressed shattering simulation. For high-resolution shock capturing methods, we only expect higher-than-first order convergence for smooth regions, so here we have calculated the norm of the error in the density in only the smooth region $x > -0.4$. If we include all of the simulations, then the fit of the error shows convergence at only an order of -0.41 in n . However, we can see in Fig. 5.4 that, although the waves in this region are smooth rarefaction waves, they are quite sharp and look almost like shocks, meaning that they are not well resolved at low resolution. Even if we restrict the fit to the highest resolution simulations (1600 cells to 12800 cells), we still only observe a convergence order of -0.79 . We believe this to be due to the fact that, even at these high resolutions, the rarefaction waves are still not well-resolved. This is supported by the fact that the wave in Fig. 5.5 is still getting steeper with resolution, and exhibits an undershoot to the left of the wave: the numerical method is still reacting as if these regions were discontinuous. This means that we should expect no better than first order convergence for this simulation.

simulation discussed above. The simulation is evolved in 1D with planar symmetry, so its results are not equivalent to the results of a 1D slice of the above test, but should be comparable for early times, when the curvature of the initial discontinuity has negligible effect. The results of this 1D test at coordinate time $t = 0.06$ are shown in Fig. 5.4. These results can be compared to the diagonal slice of the 2D simulation, shown in Fig. 5.2.

Looking at a limited region of the density around the shear, rarefaction wave at $x = -0.37$ (Fig. 5.5), we see that the solution *is* converging with resolution. This is even clearer when we plot the relative error with respect to a high-resolution simulation using 25600 cells. Fig. 5.6 shows a log-log plot of the error for a large range of different resolutions. Since we only expect high-resolution shock capturing methods to converge at higher-than-first order in smooth regions, we restrict the error calculation to take the norm of the error over only the smooth region where $x > -0.4$. A fit over *all* of the resolutions gives an order of convergence of just -0.41 in n . Since the rarefaction waves in this region are quite steep, these features may not be well-resolved at low resolutions; indeed, if we restrict our fit to higher resolutions, we see that we get convergence at -0.79 order in n . Using this, we check for Richardson scaling¹ of the error in Fig. 5.5, and see that the error *does* seem to scale at approximately this order.

This order of convergence is still significantly lower than the nominal second order convergence expected for these methods. However, we would argue that the rarefaction waves are still not well-resolved, even at the high resolutions shown above. Even for the highest resolutions shown, the rarefaction wave in Fig. 5.5 continues to steepen with increased resolution. It also exhibits an undershoot to the left of the wave at all the resolutions shown. Essentially, the numerical methods are still reacting to this wave as if it were a discontinuity, meaning that we should expect, at best, first order convergence for these tests.

5.3.2 Starquake toy model

In a more realistic neutron-star simulation, the solid crust should be situated in some background potential, and be in equilibrium before shattering. This will require gradients in density, pressure, and internal energy. The crust should also be coupled to a fluid core, and we should have a free surface. To test a more realistic scenario, we have set up a toy model.

¹See Appendix D.

In this 2D toy model, we have a solid crust, coupled to a fluid core and external atmosphere, all in a Newtonian background potential. Initially, we have an atmosphere where $y > 0$, solid where $-0.75 < y < 0$, and fluid where $y < -0.75$. The fluid is described by a Gamma-law equation of state for an ideal gas; the equation of state for the solid is a toy equation of state (Appendix C.1), which generalizes the Gamma-law equation of state by adding an additional term proportional to the shear scalar (this is the Toy equation of state described in [1]). For the fluid, we set the equation of state parameter $\Gamma = 2$, and for the solid, we set $\Gamma = \lambda = 2$, and $\kappa = 0.1$, where λ is the power for the dependence of the shear term on the density, and κ is a factor that determines the size of this term relative to the thermal (fluid) term.

To set up the problem, we start by finding the hydrostatic equilibrium for a fluid described by the Gamma-law equation of state. For this hydrostatic equilibrium, the density, pressure, and internal energy are as follows:

$$\rho = \left[\frac{\Gamma - 1}{K(s)\Gamma} (-g(y - R)) \right]^{\frac{1}{\Gamma-1}}, \quad (5.4)$$

$$p = K(s) \left[\frac{\Gamma - 1}{K(s)\Gamma} (-g(y - R)) \right]^{\frac{\Gamma}{\Gamma-1}}, \quad (5.5)$$

$$\epsilon = \frac{1}{\Gamma} (-g(y - R)), \quad (5.6)$$

where R is the location of the surface of the star, and g is the acceleration in a constant, plane-symmetric gravitational field. We choose the isentropic solution for a fluid star, where $K(s) = 1$, and as mentioned above, $\Gamma = 2$. As a result, ρ and ϵ are linear in y , and p is quadratic. We will use the above equations to calculate the initial data in the fluid part of the star.

Next, we must find the corresponding equilibrium solution in the solid part of the star. Because there are multiple configuration gradients, $\psi^A{}_i$, and matter-space metrics, k_{AB} , that will produce the same density profile, we have some freedom with how we choose these variables. However, we also have the additional requirement that the solid should be anisotropically stressed: if it were not, shattering would have no effect.

To do this, we start with $\psi^A{}_i = \text{diag}(1, 1, 1)$ and $k_{AB} = \text{diag}(\rho^{\frac{2}{3}}, \rho^{\frac{2}{3}}, \rho^{\frac{2}{3}})$, which is a solid in its relaxed state. We then divide one term of the matter-space metric by a constant factor, a , and multiply another term by the same factor; this maintains the value of the determinant of $k^A{}_B$,² and therefore the density remains the same

²Here we use the Newtonian metric, $g^{ab} = \delta^{ab}$, and $\psi^A{}_i$ is the identity matrix, so the transformation between k_{AB} and $k^A{}_B$ is trivial

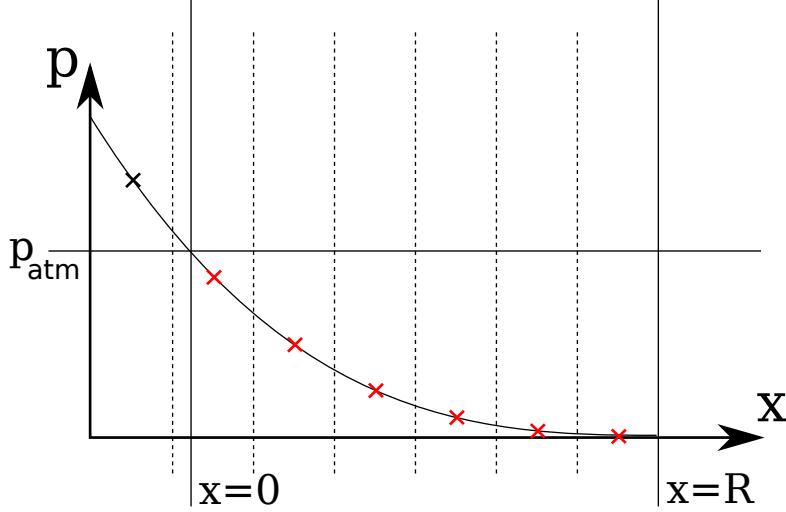


Figure 5.7: A sketch showing the pressure at the surface of the neutron star toy model. In the continuum, the pressure goes to zero at the surface of the star; however, since we are using an atmosphere, the pressure will instead go to some non-zero atmospheric pressure at the surface. To achieve this in the initial data, we set R so that $p = p_{\text{atm}}$ at $x = 0$, which is the location of the solid-atmosphere material interface. Effectively, this clips off the surface of the star; for the simulation shown below, which has a grid resolution of 50×200 and an atmospheric pressure of $p_{\text{atm}} = 0.01$, 6 cells are clipped off, shown here in red.

as in the relaxed state (it is still equal to ρ). Physically, this is effectively stretching the solid in one direction and squeezing it in the other (with no shearing), so that the density remains the same.

To be sure the system is in equilibrium, the pressure in the solid crust should be the same as it would be if it were in a fluid in the hydrostatic equilibrium shown above. Since we chose $\Gamma = \lambda$, this is automatically achieved as long as the density and internal energy are both the same as they were in fluid hydrostatic equilibrium (see Eq. C.7). If the internal energy is the same between the solid and fluid cases, then we can derive the relationship between $K(s)$ in the solid and the fluid:

$$K(s)_{\text{solid}} = K(s)_{\text{fluid}} - (\Gamma - 1) \kappa \rho^{\lambda-\Gamma} \mathcal{S} = 1 - \kappa \mathcal{S}, \quad (5.7)$$

where we have plugged known quantities into the last equality.

Since we have an atmosphere, the pressure should go to p_{atm} at the surface, rather than to zero. From this, we can calculate the constant, R , which is the location where p , ρ , and ϵ go to zero. We also use $g = 4$ for the toy model simulation shown here.

In summary, the fluid is set as follows:

$$\rho = -2y + p_{\text{atm}}^{\frac{1}{2}}, \quad \epsilon = -2y + p_{\text{atm}}^{\frac{1}{2}}, \quad v = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \quad (5.8)$$

The solid is then set like this:

$$\psi^A_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad k_{AB} = \begin{pmatrix} \frac{\rho^{\frac{2}{3}}}{a} & 0 & 0 \\ 0 & \rho^{\frac{2}{3}} & 0 \\ 0 & 0 & a\rho^{\frac{2}{3}} \end{pmatrix} \quad v = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad K(s) = 1 - \kappa s, \quad (5.9)$$

where ρ is determined as in the fluid case, and a is a scalar factor, which determines the degree to which the crust is stressed. In this simulation, we set $a = 2.5$. An atmospheric pressure of $p_{\text{atm}} = 0.01$ was used for this simulation. For reference, the maximum pressure in the initial data is $p_{\text{max}} \approx 10.0$. In the simulation shown here (meaning for a 50×200 grid), this atmospheric pressure effectively cuts off 6 grid points at the surface; this is illustrated in Fig. 5.7. In a hydrostatic equilibrium where the pressure goes to zero at the surface, the atmospheric pressure is reached after moving 6 grid points away from the surface (where the pressure is zero).

Once we have this set up, we shatter a circular region centered at $(0, -0.475)$, with a radius of $r = 0.1375$. We note that, because we already have 3 materials in the evolution, and because smearing of the matter-space metric does not seem to cause errors as drastic as those seen in the homogeneously stressed shattering simulation above, we do not use separate materials for the shattered and unshattered regions for this particular test. Some errors result from this smearing, but running with an additional material would become extremely time-consuming. The p_{xx} component of the stress tensor, along with velocity vectors, and the difference between p_{xx} and its equilibrium value, are shown in Figs. 5.8-5.11 at coordinate times, $t = 0.0$, $t = 0.3$, $t = 0.6$, and $t = 1.1$. For reference, the free-fall time across the grid is $t \approx 1$, and the sound speed in the equilibrium configuration is $0.5 \lesssim c_s \lesssim 2.5$, so the crossing time for sound waves is $0.7 \lesssim t \lesssim 3.6$, where the sound speed is lowest at the surface, so sound waves will travel more slowly there. The shear speed in the crust ranges from $0.15 \lesssim \lambda_T \lesssim 0.55$; so the crossing time for shear waves is longer than that for sound waves.

Fig. 5.8 shows the initial data for the toy model shattering simulation. The stress-tensor component, p_{xx} , in the solid is the sum of the pressure and the anisotropic stress; the gradient in this variable is due to the gradient in the pressure, which is necessary to keep the system in hydrostatic equilibrium. In the

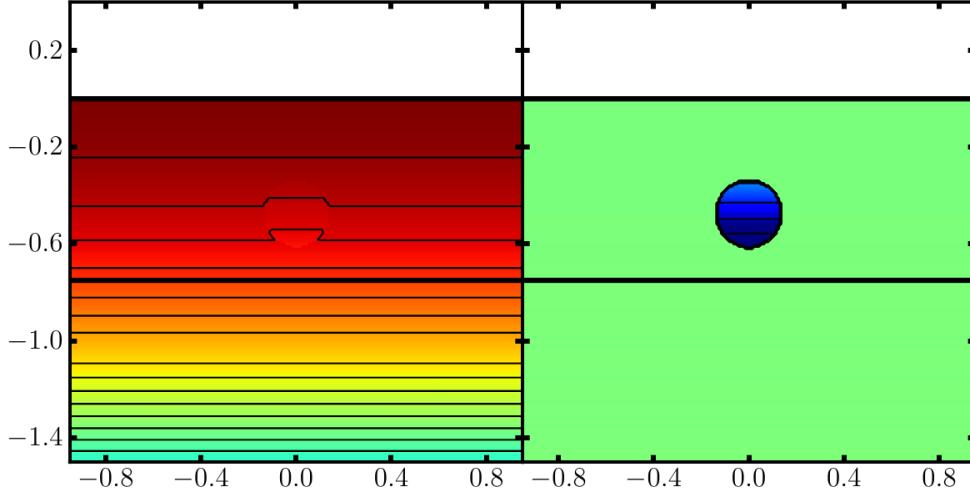


Figure 5.8: The toy model shattering simulation at coordinate time, $t = 0.0$. On the left, the stress-tensor component, p_{xx} , is shown, while on the right, the difference between this component and its equilibrium value is shown. The interface boundaries at the top and bottom of the crust are also shown as a thick black line. Initially, the toy model consists of a solid crust on top of a fluid core with an atmosphere outside, all in equilibrium in a background potential. The crust is initially anisotropically stressed; the result of this can be seen in the right-hand plot. In the outer regions, p_{xx} is equivalent to its equilibrium value, which is partially due to pressure and partially due to anisotropic stress, but in the shattered region, the value of p_{xx} has been altered. The results are produced using a 50×200 grid.

shattered region, all anisotropic stress has been removed; this can be seen in the left-hand picture, but it is especially apparent in the right-hand picture, when we look at the difference from the equilibrium configuration.

Fig. 5.9 shows the results at coordinate time $t = 0.3$. Here, velocity vectors are also included. Shattering causes an initial discontinuity in the pressure; this is the source of the fast-traveling pressure wave that can clearly be seen in the right-hand image, and that is also seen in the velocity. Because the sound-speed is higher in the interior of the star, sound waves move more quickly in the downward direction. This is easily observed in the right-hand plot where the wave has already propagated well into the fluid in the downward direction, but is only just beginning to reach the surface in the upward direction. The non-zero velocity along the surface is *not* a result of shattering, as the first wave has not yet reached most of the surface. This is due to errors in the surface treatment. A similar wave can be seen propagating into the fluid; this is, again, due to small errors in the interface treatment between the crust and core. We also see these errors in equilibrium solution evolutions without any shattering.

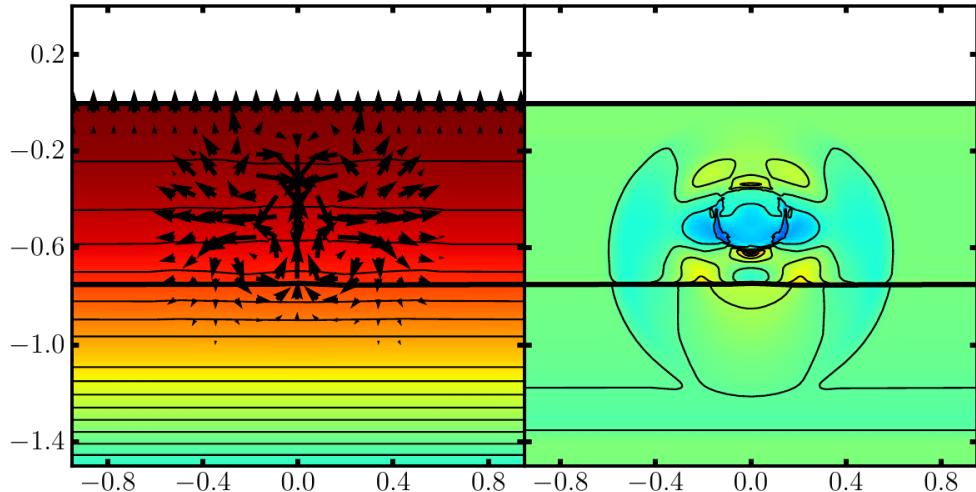


Figure 5.9: The toy model shattering simulation at coordinate time, $t = 0.3$. In addition to the variables shown in Fig 5.8, here we also see velocity vectors in the left-hand image. Here we can see a fast pressure wave moving outward from the initial shattered region, as well as vorticity being generated there. The results of errors in the surface and interface treatments are also visible as plane-symmetric waves in the results.

In the left-hand panel of Fig. 5.9, it is also possible to see that some vorticity is generated at the boundary between the shattered and unshattered regions. When the anisotropic stress is released, the stress in the x -direction between the shattered and unshattered regions is unbalanced; this causes outward motion in the x -direction everywhere along the curved discontinuity. The outward motion will be largest at around $y = -0.45$, where the discontinuity is normal to the direction of the stress gradient, and will go to zero at the top and bottom of the shattered region. The difference in horizontal motion, combined with the curved shape of the discontinuity causes the vorticity that we see here.

Fig 5.10 shows the results at coordinate time $t = 0.6$. Here, we see that the vorticity generated by the initial shattering discontinuity has interacted with both interfaces, causing them to noticeably deform. The initial pressure wave has reached the right and left edges of the grid, and, because we use periodic boundary conditions in x , interacts with itself; this can be seen as the blue arcs at the left and right edges of the grid. Errors similar to those seen in the homogeneous shattering simulation (Fig. 5.2) can be seen persisting along the edges of the region that was initially shattered; as mentioned earlier, these are due to smearing of the discontinuous matter-space metric. Interestingly, the shape of the error region is deformed (in Fig. 5.10, the region is wider than it is high) by the vorticity of the flow in the same way that the interfaces are. Essentially, the fast outward flow due to the stress gradient in the horizontal direction causes the density and pressure in

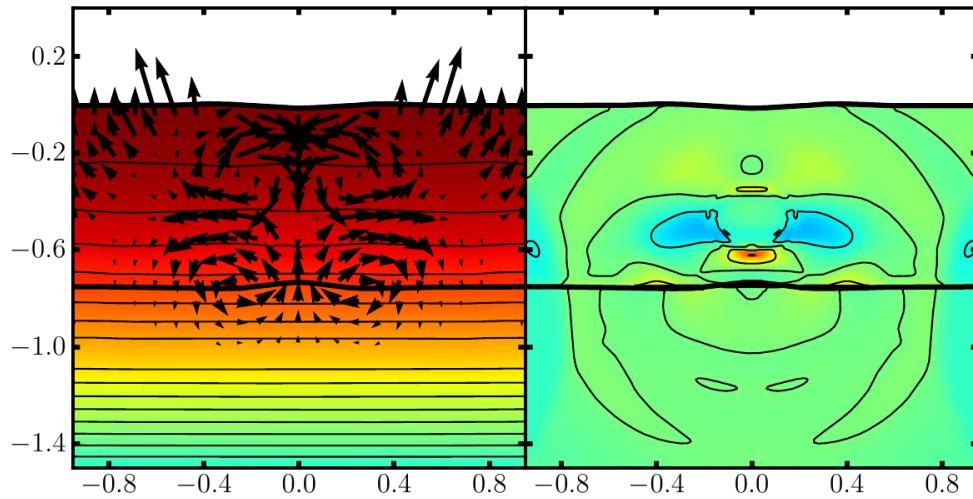


Figure 5.10: The toy model shattering simulation at coordinate time, $t = 0.6$. Here, the fast-moving pressure wave has interacted with the periodic boundaries at the left and right edges of the grid. The interfaces at the top and bottom of the crust are both deformed by the vorticity of the flow. We also see error persisting at the edge of the shattered region due to smearing of the matter-space metric; the shape of this error region is also deformed by the vorticity.

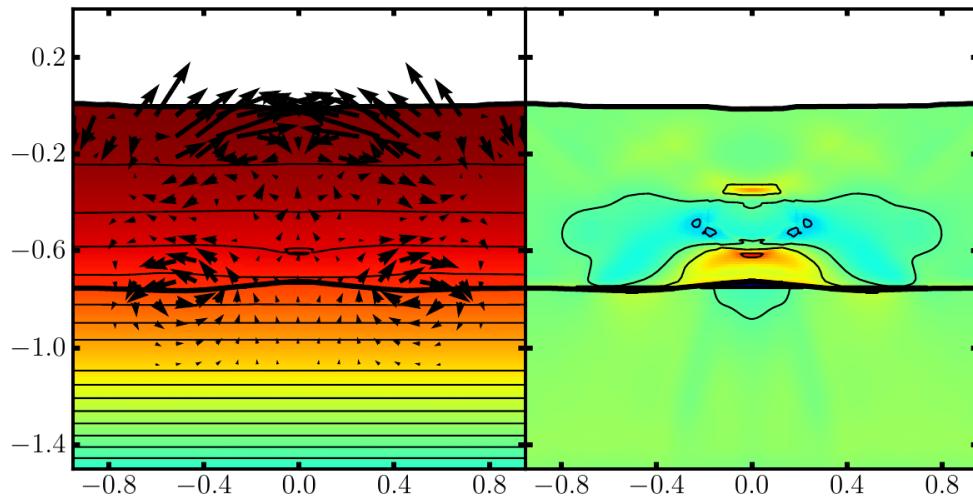


Figure 5.11: The toy model shattering simulation at coordinate time, $t = 1.1$. After interacting with itself at the boundary, the pressure wave has reflected back, and interacted again at the center of the grid. The vorticity travels rather slowly along the interface between the crust and the core; the main driving force behind this is shear in the solid. The matter-space-metric-smearing error continues to persist. Low and high stress regions also form around the initial shattered region.

the center of the crust to drop; the material then responds with an inward velocity in the vertical direction, causing the pinching effect.

Fig. 5.11 shows the results at coordinate time $t = 1.1$. Here the fast pressure waves have interacted at the boundary, and their reflections have interacted again at the center of the grid. The vorticity continues to deform the interface between the crust and core, while traveling outward along this interface. The vorticity moves along the interface due to shear waves in the crust; this is supported by the fact that the center of vorticity moves a distance of around $\Delta x \approx 0.45$ by this coordinate time, giving a speed of ~ 0.4 , which is well within the range of shear speeds in the star.

That said, the deformation of the top surface of the star varies much more quickly than the deformation of the crust-core interface. The dominant force behind this effect must be pressure, since the waves have traveled outward, interacted at the periodic boundary, reflected back, and are traveling inward again.

We also see that the density errors due to matter-space metric smearing still persist, and that the shape of the error region has deformed even more. Low stress areas have also developed at the top and bottom of this region, while high stress areas have developed to either side.

Fig. 5.12 shows the convergence of the toy model shattering simulation. It is clear that, as the resolution increases, the results converge. The error is computed at coordinate time $t = 0.3$. A least-squares fit shows that the results converge at order -1.25 in n , where the grid resolution is $n \times n$. The numerical methods are nominally second-order convergent, so the convergence order between first and second order is as expected. This test differs from the homogeneously stressed shattering simulation in that it does not include extremely steep rarefaction waves.

All of this demonstrates that we can combine all of the methods developed into one simulation: namely, the solid crust, crust-core and atmosphere-crust interfaces, and shattering.

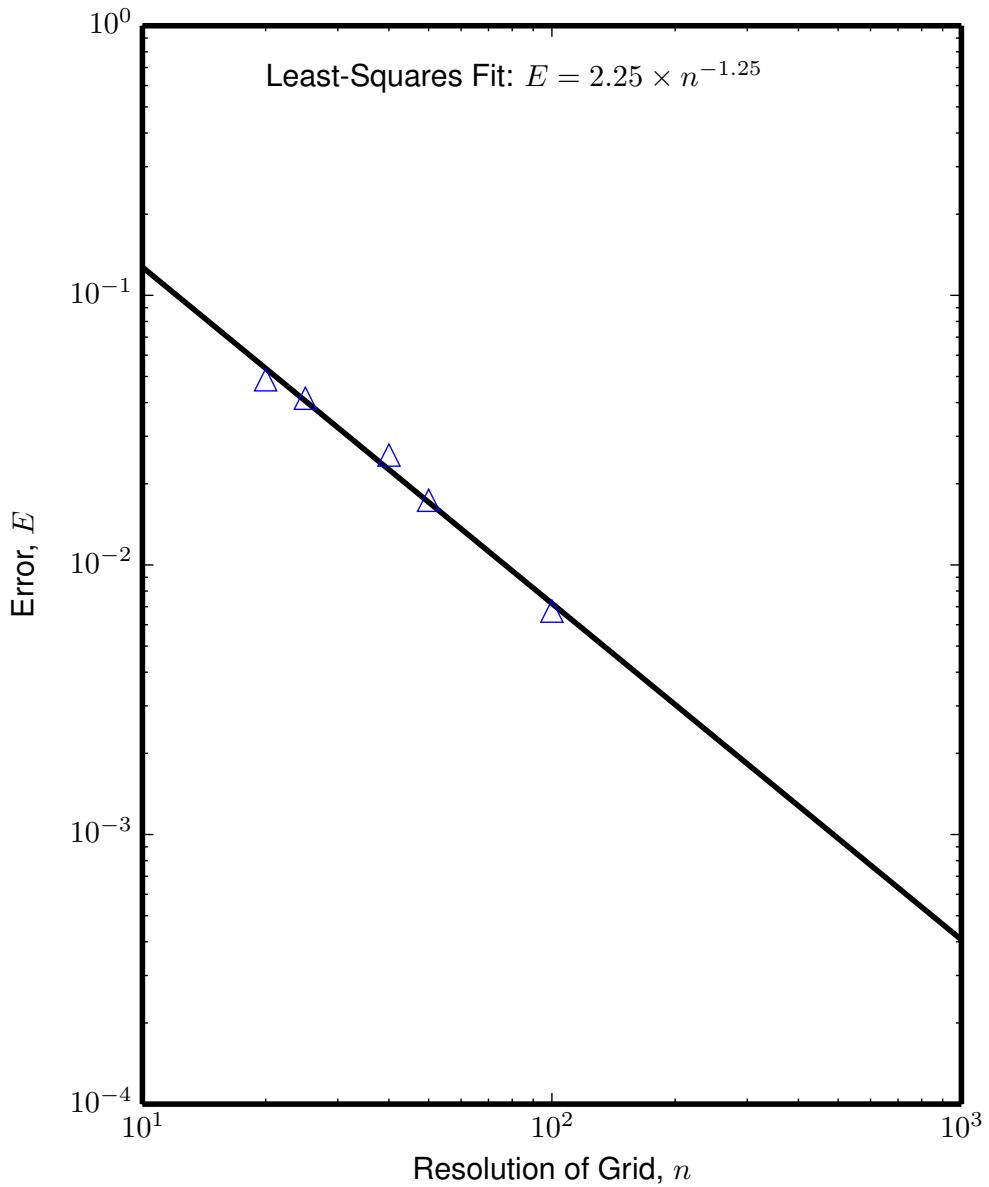


Figure 5.12: A convergence plot for the toy model neutron star shattering simulation. The error was calculated for various grid resolutions ranging from 5×20 to 25×100 ; these were then compared to a high resolution simulation (50×200). The error is estimated as described in Appendix D. The error is calculated at coordinate time $t = 0.3$. As the resolution increases, the results converge. A least-squares fit shows that the error converges with a power -1.25 in n . This is as expected: the methods used should nominally converge at second order, so for a simulation including discontinuities, we expect somewhere between first and second order convergence.

Chapter 6

Conclusions

6.1 Summary

Although the mass of the crust of a neutron star makes up only a small portion of the total mass of the star, starquakes could have significant effects on the dynamics of neutron star systems. The shear modulus of the crust is much smaller than the compression (or bulk) modulus: shear modes will be at lower frequencies than fundamental fluid modes of the star. This means that these modes can be excited in situations where the fluid modes would not be readily excited. In particular, starquakes have been suggested as possible mechanisms behind pulsar glitches, quasi-periodic oscillations after giant flares in soft gamma-ray repeaters, and the precursor signals to short-hard gamma-ray bursts. In short, breaking of the neutron star crust can have observable effects, and we would like to examine the dynamics of these starquakes in neutron star binary systems as well as isolated neutron stars.

With this long-term aim in mind, the goal of my PhD has been to develop a toy model that demonstrates several of the different methods that we need in order to perform more realistic numerical simulations of neutron star systems.

Chapter 2 discusses some of the numerical methods that are commonly used to simulate systems of conservation laws; these are methods that are used both for the codes implemented to develop the toy model, as well as for the matter-evolution parts of 3D fully-relativistic codes, which could be used to do more realistic neutron star simulations in the future.

With the basics covered, Chapter 3 goes on to describe the solution to the first of the technical challenges of simulating neutron starquakes; namely, the problem of evolving a solid crust in general relativity. Here, a conservation-law formulation for non-linear elasticity is described. The conservation-law formulation means

that the equations should be relatively easily integrated into existing codes for neutron star evolutions; it also admits *weak solutions*, or shocks, which we would expect to arise naturally from shattering or other types of breaking of the crust. In Section 3.5, a numerical implementation of the formulation is described, as are the results of several numerical tests.

With the problem of evolving the solid crust solved, we move on to coupling that crust to a fluid core. In particular, the transition between the crust and the core is much smaller than the overall size of the neutron star, so the boundary should be modeled as a sharp transition, or an *interface*. In Chapter 4, we discuss methods for evolving material interfaces, including using a level-set function to track their location, and an extension of the ghost-fluid method other types of interfaces and to general relativity. The results of several numerical tests are discussed.

In addition to the solid-fluid interface between the crust and the core, to model the system, we will also need some way of treating the surface of the star, where the density, pressure, and internal energy all go to zero. For the toy model presented here, we use an atmosphere; Section 4.2.9 discusses exactly how this atmosphere is implemented in the toy model, and an evaluation of how well this atmosphere approximates a vacuum is presented in Section 4.4.8.

The last piece necessary to put together the toy model is some mechanism for breaking the crust. Molecular dynamics simulations suggest that the neutron star crust fails catastrophically rather than forming local dislocations, or cracks; because of this, we implement a mechanism for *shattering*, which amounts to instantaneous relaxation of the material in some volume of the crust. This is described in Chapter 5. Some results for a simple shattering simulation are shown.

The final 2D toy model is presented in Section 5.3.2. It consists of a solid crust, a fluid core, and an external atmosphere all in some background Newtonian gravitational potential. A region of the crust is then shattered, and the resulting evolution is examined.

Although the results here may not be physically significant, they represent the first time that all of these methods have been combined into a single simulation. As such, this work represents a significant step forward, showing that realistic simulations of starquakes are technically feasible. These can then be used to study binary neutron star mergers or isolated neutron stars, and to determine the mechanisms behind various observed phenomena.

6.2 Future work

The first thing that we would like to investigate in the near future is the accuracy of the atmosphere method that we use at the surface of the star. In Section 4.4.8, we have shown that the results of solid-atmosphere Riemann problems approach the solid-vacuum results as the atmospheric pressure is reduced, *as long as we use a terrestrial equation of state*. We also know that evolutions of the stellar surface of our toy model are stable for sufficiently high atmospheric pressures. However, we *do not* know whether this stellar surface evolution is correct; in principle, the atmosphere treatment is an approximation of the physical system and is not expected to be correct, but we would like to quantify exactly *how* wrong it is, and how this affects the results of the overall simulation.

In addition to this, we would like to perform a toy model simulation similar to that presented here, but in general relativity with a fixed background metric. This will give us an increased level of confidence that the methods implemented here will work in fully general relativistic simulations.

A future goal for this project is to take the methods used here and to import them into a 3D fully relativistic code such as Cactus (i.e. as an extension to the GRHydro thorn) so that more realistic and physically relevant neutron star simulations can be performed. A first step would probably be to import elasticity into the matter part of one of these codes by making appropriate additions to the part of the code that handles the hydrodynamic evolution. Specifically, we would need to include additional evolution equations for the configuration gradient, ψ^A_i , and for the matter space metric, k_{AB} ; we would also need to adapt the matter evolution equations as well as the equation of state to include anisotropic-stress terms. In relativistic elasticity, the conversion of conserved variables to primitive variables also requires a 4-dimensional root finder rather than the 1-dimensional root finder used for relativistic hydrodynamics. We would also want to think carefully about how to implement an elastic atmosphere.

Adding elasticity to this type of code, by itself, could already potentially lead to some interesting physics results, providing upper limits on the effect that elasticity could have on the dynamics of binary neutron star mergers, for example. Of course, if one is only interested in these upper limits; there may be simpler ways to achieve this goal. For example, one could potentially determine the maximum influence of elasticity on the tidal deformations of neutron stars in a binary merger system, and therefore the phase evolution of the gravitational waves emitted from this system, by importing elasticity into a LORENE-like code to find a series of equilibrium

orbits. However, to simulate starquakes, we *do* need to import elasticity into a fully relativistic hydrodynamics code.

The next step would be to import the methods for interfaces into the code. This would allow us to do simulations of even more realistic neutron stars, including both a crust and a core. We could then gain an even better insight into the dynamics of starquake systems, and how these compare to the observed phenomena that may be linked to this behavior.

Of course, in order for the observed behavior to actually become observable, a magnetic field will need to be present. Some of the proposed mechanisms that could break the crust are due to magnetic field evolution—clearly, realistic simulations of these phenomena will require the presence of a magnetic field—but even where the magnetic field is not directly involved in the starquake mechanism itself, it *is* involved in converting that into an observable effect. Any complete description of these observed events will involve a magnetic field, so the long-term goal is to produce a simulation of a neutron star with a solid crust, fluid core, and magnetic fields permeating both, as well as the magnetosphere outside the star. Perhaps it would even make more sense to import elasticity directly into an MHD code, rather than a purely hydrodynamic code—in this case, we would combine elasticity and magnetic fields first, and worry about the interface with the fluid core later.

Appendix A

Introduction to hyperbolicity

To better understand what hyperbolicity is, and why it is important, we include a summary of the material presented in the lecture notes of Kreiss and Busenhart from 2001 [29]. The section here follows this work closely.

A.1 An example

To illustrate the fundamental concepts behind hyperbolicity and the idea of well-posedness, we start with a simple example. Consider the system

$$\begin{aligned} u_{,t}(x, t) + u_{,x}(x, t) &= 0, \quad x \in \mathbb{R}, \quad t \geq 0, \\ u(x, 0) &= f(x), \end{aligned} \tag{A.1}$$

where $f(x) = f(x + 2\pi)$ is a smooth, 2π -periodic function. Now assume that

$$f(x) = \hat{f}(\omega)e^{i\omega x} \tag{A.2}$$

consists of only one wave, where $\hat{f}(\omega)$ is the Fourier transform of $f(x)$. Then we construct an ansatz solution to the system from Eq. A.1:

$$u(x, t) = \hat{u}(\omega, t)e^{i\omega x}. \tag{A.3}$$

We substitute this into Eq. A.1, and get the following:

$$\begin{aligned} \hat{u}(\omega, t)_{,t} + i\omega\hat{u}(\omega, t) &= 0, \\ \hat{u}(\omega, 0) &= \hat{f}(\omega), \end{aligned} \tag{A.4}$$

and the solution to this is simple to find:

$$\hat{u}(\omega, t) = \hat{u}(\omega, 0)e^{-i\omega t}. \quad (\text{A.5})$$

We substitute this back into the ansatz, Eq. A.3, and get

$$u(x, t) = \hat{u}(\omega, 0)e^{i\omega(x-t)} = \hat{f}(\omega)e^{i\omega(x-t)} = f(x - t). \quad (\text{A.6})$$

This is a solution to the system, Eq. A.1. More generally, our function, $f(x)$, could contain more than one wave. A general 2π -periodic function can be written as

$$f(x) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} \hat{f}(\omega)e^{i\omega x}. \quad (\text{A.7})$$

Then, by superposition, the general solution becomes

$$u(x, t) = \frac{1}{\sqrt{2\pi}} \sum_{\omega=-\infty}^{\infty} \hat{f}(\omega)e^{i\omega(x-t)} = f(x - t). \quad (\text{A.8})$$

We denote the complex conjugate value of f by \bar{f} . With this, we can define the scalar product

$$(f, g) := \int_0^{2\pi} \bar{f}g dx. \quad (\text{A.9})$$

From this, the L_2 norm is defined as follows:

$$\|f\| := (f, f)^{1/2}. \quad (\text{A.10})$$

With Parseval's relation, we can write

$$\sum_{\omega=-\infty}^{\infty} |\hat{f}(\omega)|^2 = \|f(\cdot)\|^2. \quad (\text{A.11})$$

Then for any fixed time, t , we have

$$\|u(\cdot, t)\|^2 = \sum_{\omega=-\infty}^{\infty} |\hat{u}(\omega, t)|^2 \quad (\text{A.12})$$

$$= \sum_{\omega=-\infty}^{\infty} |\hat{f}(\omega)e^{-i\omega t}|^2 \quad (\text{A.13})$$

$$= \sum_{\omega=-\infty}^{\infty} |\hat{f}(\omega)|^2 \quad (\text{A.14})$$

$$= \|f(\cdot)\|^2. \quad (\text{A.15})$$

This means that the amplitude of the solution to Eq. A.1 does not increase with time, and it is said to be *norm conserving*. This means that small errors introduced into the initial data (or at each step) do not grow as the system is evolved in time. In order to solve a problem numerically, we need our system to be *well-posed*; in other words, the solution should be bounded.

A.2 Definition of well-posedness

We now define well-posedness more specifically. The Cauchy problem is

$$\begin{aligned} u_{,t}(x, t) &= P(\partial/\partial x)u(x, t), \quad x \in \mathbb{R}^d, \quad t \geq 0, \\ u(x, 0) &= f(x). \end{aligned} \quad (\text{A.16})$$

The symbol, $P(\partial/\partial x)$, is the differential operator of order m :

$$P(\partial/\partial x) = \sum_{|\nu| \leq m} A_\nu D^\nu, \quad (\text{A.17})$$

where A_ν are constant, complex, matrices of size $n \times n$, where n is the size of the vector, u . Then ν is a multi-index, i.e. a vector of non-negative integers, and has order

$$|\nu| := \nu_1 + \nu_2 + \dots + \nu_d. \quad (\text{A.18})$$

Each unique ν has a corresponding A_ν , regardless of whether the order, $|\nu|$, is the same. The derivatives are denoted by D^ν such that

$$D^\nu = \frac{\partial^{|\nu|}}{\partial x_1^{\nu_1} \dots \partial x_d^{\nu_d}}. \quad (\text{A.19})$$

The result of all this notation is that $P(\partial/\partial x)$ is a linear combination of derivatives up to order m . It can also be thought of as a $n \times n$ -matrix. It could, for example, look something like this:

$$P(\partial/\partial x) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} \frac{\partial^2}{\partial x \partial y} + \begin{pmatrix} 0 & 5 \\ 3 & 1 \end{pmatrix} \frac{\partial^2}{\partial z^2}. \quad (\text{A.20})$$

In this example, x is a 3-dimensional vector, and u is a 2-dimensional vector. This in turn means that ν is a vector of 3 non-negative integers, and the coefficients, A_ν , are a 2×2 matrices.

Now, we again assume that $f(x)$ is smooth and 2π -periodic in all spatial dimensions. As in the example, we can write the function as a Fourier series:

$$f(x) = \sum_{\omega=-\infty}^{\infty} \hat{f}(\omega) e^{i\langle \omega, x \rangle}, \quad (\text{A.21})$$

where

$$\langle \omega, x \rangle := \sum_{i=1}^d \omega_i x_i \quad (\text{A.22})$$

is the scalar product, and ω is a vector of integer components. Then, application of the differential operator is equivalent to multiplication by the matrix

$$P(i\omega) := \sum_{|\nu| \leq m} A_\nu (i\omega_1)^{\nu_1} \dots (i\omega_d)^{\nu_d}, \quad (\text{A.23})$$

and the matrix $P(i\omega)$ is called the *symbol* of the differential operator $P(\partial/\partial x)$. It is simple to show that our previous example, Eq. A.20, becomes

$$P(i\omega) = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix} (i\omega_x)(i\omega_y) + \begin{pmatrix} 0 & 5 \\ 3 & 1 \end{pmatrix} (i\omega_z)^2 \quad (\text{A.24})$$

$$= \begin{pmatrix} (i\omega_x)(i\omega_y) & 5(i\omega_z)^2 \\ 3(i\omega_z)^2 & 2(i\omega_x)(i\omega_y) + (i\omega_z)^2 \end{pmatrix}. \quad (\text{A.25})$$

Given this definition of the Cauchy problem, we say that it is *well-posed* if it can be bounded in the following way:

$$|e^{P(i\omega)t}| \leq K e^{\alpha t}, \quad (\text{A.26})$$

where here we have the matrix norm,

$$|A| := \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}, \quad (\text{A.27})$$

and where K and α are constants, and the above expression is true for all $t \geq 0$ and all ω . Now, in the following, we can see that if we assume well-posedness, then a solution to the Cauchy problem exists. To show this, we start with the ansatz

$$u(x, t) = \sum_{\omega=-\infty}^{\infty} \hat{u}(\omega, t) e^{i\langle \omega, x \rangle}. \quad (\text{A.28})$$

From this, the Cauchy problem, Eq. A.16, becomes

$$\hat{u}_{,t}(\omega, t) = P(i\omega)\hat{u}(\omega, t), \quad (\text{A.29})$$

$$\hat{u}(\omega, 0) = \hat{f}(\omega), \quad (\text{A.30})$$

which has the solution

$$\hat{u}(\omega, t) = \hat{f}(\omega) e^{P(i\omega)t}. \quad (\text{A.31})$$

When we transform this back to the spatial domain, we get

$$u(x, t) = \sum_{\omega=-\infty}^{\infty} \hat{f}(\omega) e^{i\langle \omega, x \rangle} e^{P(i\omega)t}. \quad (\text{A.32})$$

Since the system is well-posed, the norm, $|e^{P(i\omega)t}|$, is bounded, so the series, Eq. A.32, will converge. (Basically, the $e^{P(i\omega)t}$ term converges because the norm is bounded, and the other terms converge because they form the Fourier transform of $f(x)$.) In summary, well-posedness of the Cauchy problem implies the existence of a solution.

Now, we would like to show that the Cauchy problem is well-posed if and only if the norm of the solution is bounded relative to the initial data in the following way:

$$\|u(\cdot, t)\|^2 \leq K^2 e^{2\alpha t} \|u(\cdot, 0)\|^2. \quad (\text{A.33})$$

This is simply shown as follows:

$$\|u(\cdot, t)\|^2 = \sum_{\omega=-\infty}^{\infty} |\hat{u}(\omega, t)|^2 \quad (\text{A.34})$$

$$= \sum_{\omega=-\infty}^{\infty} |\hat{f}(\omega) e^{P(i\omega)t}|^2 \quad (\text{A.35})$$

$$\leq \sup_{\omega} |e^{P(i\omega)t}|^2 \sum_{\omega=-\infty}^{\infty} |\hat{f}(\omega)|^2 \quad (\text{A.36})$$

$$\leq K e^{2\alpha t} \|u(\cdot, 0)\|^2, \quad (\text{A.37})$$

where the first and second equalities come simply from Parseval's relation, and then substituting in the solution for the Cauchy problem in the frequency domain. The third line comes about by first recalling the inequality, $|AB| \leq |A||B|$, and then replacing the factor of $|e^{P(i\omega)t}|^2$ in each term by its supreme value over ω . Then we know that $|e^{P(i\omega)t}| \leq K e^{\alpha t}$ for all ω , and we transform the second term back to the spatial domain. This shows that well-posedness implies that the growth of the norm of the solution over time will be bounded by the exponential function, and the other direction is trivial.

A.3 The first-order, constant-coefficient, 1D system

Now we look at the constant-coefficient, first-order Cauchy problem:

$$\begin{aligned} u_{,t}(x, t) &= Au_{,x}(x, t), \\ u(x, 0) &= f(x), \end{aligned} \quad (\text{A.38})$$

where $u = (u_1, u_2, \dots, u_n)^T$, and A is a complex $n \times n$ matrix. We now show that the constant-coefficient system, Eq. A.38, is well-posed if and only if the eigenvalues of A are real and A had a complete set of eigenvectors, i.e. if the eigenvectors of A form a basis that spans \mathbb{R}^n .

We start by showing that real eigenvalues and a complete set of eigenvectors imply that the system is well-posed; we will later return to showing the other direction. If we have a complete set of eigenvalues, then there exists a transformation, T , such that

$$TAT^{-1} = \Lambda, \quad (\text{A.39})$$

where Λ is a diagonal matrix with the eigenvalues along the diagonal. In other words, the matrix is diagonalizable. Therefore, we see the following:

$$|e^{i\omega At}| = |T^{-1}Te^{i\omega At}T^{-1}T| \quad (\text{A.40})$$

$$= |T^{-1}e^{i\omega(TAT^{-1})t}T| \quad (\text{A.41})$$

$$\leq |T^{-1}||e^{i\omega\Lambda t}||T| \quad (\text{A.42})$$

$$= |T^{-1}||T|. \quad (\text{A.43})$$

The first inequality comes from simply multiplying twice by $T^{-1}T = I$, the identity matrix. Then, it can be seen that the transformation can be moved inside the exponential by the following:

$$Te^{i\omega At}T^{-1} = T \left(\sum_{n=0}^{\infty} \frac{(i\omega At)^n}{n!} \right) T^{-1} \quad (\text{A.44})$$

$$= I + i\omega tTAT^{-1} + \frac{(i\omega t)^2}{2}TAT^{-1}TAT^{-1} + \dots \quad (\text{A.45})$$

$$= I + i\omega t\Lambda + \frac{(i\omega t)^2}{2}\Lambda^2 + \dots \quad (\text{A.46})$$

$$= \sum_{n=0}^{\infty} \frac{(i\omega\Lambda t)^n}{n!} \quad (\text{A.47})$$

$$= e^{i\omega\Lambda t}. \quad (\text{A.48})$$

Then the third inequality comes from $|AB| \leq |A||B|$, and finally, we can eliminate $|e^{i\omega\Lambda t}| = I$ if and only if all of the eigenvalues (which are the diagonal values of Λ) are real. This then shows that the system is well-posed, with $K = |T^{-1}||T|$ and $\alpha = 0$.

Showing that well-posedness requires a complete set of eigenvectors and all real eigenvalues is a little more involved. Generally, we can always transform the matrix, A , to a Jordan's normalform using a transformation, T , such that

$$TAT^{-1} = \begin{pmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_r \end{pmatrix}, \quad (\text{A.49})$$

where $J_j = \lambda_j I + D_j$, and

$$D_j = \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}. \quad (\text{A.50})$$

If we had a complete set of eigenvalues, then the Jordan's normalform would be the diagonal matrix, Λ ; we only get ones on the first diagonal if we have eigenvectors which are not linearly independent. From this, we will get a repeated eigenvalue. Next we work out that

$$|e^{i\omega At}| = |T^{-1} e^{i\omega t(TAT^{-1})} T| \quad (\text{A.51})$$

$$= \frac{|T||T^{-1} e^{i\omega t(TAT^{-1})} T||T^{-1}|}{|T||T^{-1}|} \quad (\text{A.52})$$

$$\geq \frac{1}{|T||T^{-1}|} |e^{i\omega t(TAT^{-1})}|, \quad (\text{A.53})$$

where the first equality is similar to the steps we took in Eq. A.40, and then we multiply by $|T||T^{-1}|/|T||T^{-1}| = 1$, and use the $|AB| \leq |A||B|$ inequality in reverse.

Now suppose that we have some matrix B of block diagonal form:

$$B = \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix}. \quad (\text{A.54})$$

Then the matrix norms of the two blocks, $|B_1|$ and $|B_2|$, are given by

$$|B_j| := \sup_{u_j \in \mathbb{R}^{n_j}} \frac{\|B_j u_j\|}{\|u_j\|}, \quad (\text{A.55})$$

where n_j is the size of the block B_j (so B_j is $n_j \times n_j$), and the double vertical lines indicate some vector norm in \mathbb{R}^{n_j} . Now, since

$$\left\| \begin{pmatrix} u_1 \\ 0 \end{pmatrix} \right\|_{p, \mathbb{R}^n} = \|u_1\|_{p, \mathbb{R}^{n_1}}, \quad (\text{A.56})$$

where $n > n_1$ and the 0 represents the zero vector of dimension $1 \times (n - n_1)$, we can say that

$$|B_1| = \sup_{u_1 \in \mathbb{R}^{n_1}} \frac{\|B_1 u_1\|_{p, \mathbb{R}^{n_1}}}{\|u_1\|_{p, \mathbb{R}^{n_1}}} \quad (\text{A.57})$$

$$= \sup_{u_1 \in \mathbb{R}^{n_1}} \frac{\left\| \begin{pmatrix} B_1 u_1 \\ 0 \end{pmatrix} \right\|_{p, \mathbb{R}^n}}{\left\| \begin{pmatrix} u_1 \\ 0 \end{pmatrix} \right\|_{p, \mathbb{R}^n}} \quad (\text{A.58})$$

$$= \sup_{u_1 \in \mathbb{R}^{n_1}} \frac{\left\| B \begin{pmatrix} u_1 \\ 0 \end{pmatrix} \right\|_{p, \mathbb{R}^n}}{\left\| \begin{pmatrix} u_1 \\ 0 \end{pmatrix} \right\|_{p, \mathbb{R}^n}}. \quad (\text{A.59})$$

We see from this that we can always produce the matrix norm of any particular block, i.e. $|B_1|$ or $|B_2|$ here, from the operation $\|Bu\|/\|u\|$, as long as we choose the appropriate value for u , i.e. $u = (u_1, 0)^T$ for $|B_1|$, etc. Now suppose that, to find the matrix norm of B , we tried a value of u such that the value of $\|Bu\|/\|u\|$ was less than $\max(|B_1|, |B_2|)$. Then this value of u would not be maximizing $\|Bu\|/\|u\|$, since if we chose $u = (u_1, 0)^T$ (for $|B_1| > |B_2|$) or $u = (0, u_2)^T$ (for $|B_2| > |B_1|$), we could increase the value of $\|Bu\|/\|u\|$. This means that $|B| \geq \max(|B_1|, |B_2|)$.

In our case, this means that

$$|TAT^{-1}| \geq \max_j |J_j|, \quad (\text{A.60})$$

and since, for any particular Jordan block, J_j , we can write

$$\left| \frac{(i\omega t)^n (TAT^{-1})^n}{n!} \right| \geq \left| \frac{(i\omega t)^n (J_j)^n}{n!} \right|, \quad (\text{A.61})$$

we therefore get

$$|e^{i\omega t(TAT^{-1})}| \geq \max_j |e^{i\omega J_j t}|. \quad (\text{A.62})$$

Our inequality then becomes

$$|e^{i\omega At}| \geq \frac{1}{|T||T^{-1}|} \max_j |e^{i\omega J_j t}|. \quad (\text{A.63})$$

Now, we know that if A and B commute, then we have $e^{A+B} = e^A e^B$. From this,

and the fact that the identity matrix, I , commutes with any matrix, we can see that

$$|e^{i\omega At}| \geq \frac{1}{|T||T^{-1}|} \max_j |e^{i\omega J_j t}| \quad (\text{A.64})$$

$$= \frac{1}{|T||T^{-1}|} \max_j |e^{i\omega \lambda_j It} e^{i\omega D_j t}| \quad (\text{A.65})$$

$$= \frac{1}{|T||T^{-1}|} \max_j |e^{i\omega \lambda_j t} e^{i\omega D_j t}| \quad (\text{A.66})$$

$$= \frac{1}{|T||T^{-1}|} \max_j |e^{i\omega \lambda_j t}| |e^{i\omega D_j t}|, \quad (\text{A.67})$$

where we can take the $|e^{i\omega \lambda_j t}|$ out of the norm, because it is just the absolute value of a number. Then, if we denote the J_k that maximizes $|e^{i\omega J_k t}|$ simply by $J = \lambda I + D$ and call the corresponding eigenvalue $\lambda = a + bi$, then the above inequality becomes

$$|e^{i\omega At}| \geq \frac{1}{|T||T^{-1}|} |e^{i\omega at}| |e^{-\omega bt}| |e^{i\omega Dt}|. \quad (\text{A.68})$$

Now suppose that $b \neq 0$. If this is the case, then for any α , we can choose ω such that $|e^{-\omega bt}|$ grows faster than $e^{\alpha t}$. This means that, for well-posedness, we must have $b = 0$, which means that our eigenvalues must all be *real*.

If D is a $p \times p$ block, then we can write

$$e^{i\omega Dt} = \sum_{j=0}^{p-1} \frac{(i\omega Dt)^j}{j!}, \quad (\text{A.69})$$

because D is nilpotent, with $D^p = 0$, and therefore, all of the terms after the $(p - 1)$ st term will be 0. The largest term of $|e^{i\omega Dt}|$ grows like $|\omega t|^{p-1}$, and so $|e^{i\omega Dt}|$ can only be bounded if $p = 1$, which means that all eigenvectors are linearly independent, i.e. that they form a complete set. Now we see that, in order to have a well-posed system, we must have a complete set of eigenvectors, and we must have all real eigenvalues.

A.4 The symmetrizer in a constant coefficient system

Now consider the constant coefficient system as we had before, so

$$\hat{u}_{,t}(\omega, t) = i\omega A\hat{u}(\omega, t), \quad (\text{A.70})$$

and then assume that the matrix, A , is Hermitian, so $A = A^* := \bar{A}^T$, where the star represents the Hermitian operator, which is equivalent to taking the complex-conjugate transpose of the matrix. Equivalently, a Hermitian operator satisfies

$$\langle Au, v \rangle = \langle u, Av \rangle. \quad (\text{A.71})$$

Then we look at the time derivative of the energy:

$$\frac{\partial}{\partial t} \langle \hat{u}, \hat{u} \rangle = \langle \hat{u}, \hat{u}_{,t} \rangle + \langle \hat{u}_{,t}, \hat{u} \rangle \quad (\text{A.72})$$

$$= \langle \hat{u}, i\omega A \hat{u} \rangle + \langle i\omega A \hat{u}, \hat{u} \rangle \quad (\text{A.73})$$

$$= \langle -i\omega A \hat{u}, \hat{u} \rangle + \langle i\omega A \hat{u}, \hat{u} \rangle \quad (\text{A.74})$$

$$= 0. \quad (\text{A.75})$$

The second to last equality comes about because, if A is Hermitian, then the matrix, $i\omega A$, will be anti-Hermitian, where

$$\langle Au, v \rangle = -\langle u, Av \rangle. \quad (\text{A.76})$$

We see that the time-derivative of the energy is zero, and therefore, the norm of $\hat{u}(\omega, t)$ does not change over time:

$$|\hat{u}(\omega, t)|^2 = |\hat{u}(\omega, 0)|^2 \quad (\text{A.77})$$

for all $t \geq 0$. This means that well-posedness is satisfied with $K = 1$ and $\alpha = 0$. In fact, we will now go on to show that for any well-posed constant coefficient problem, we can always construct a norm $|\hat{u}|_H^2 = \langle \hat{u}, H \hat{u} \rangle$, such that this new norm is conserved.

We start by showing the following lemma. If A is an $n \times n$ matrix, then we can find a positive definite, Hermitian matrix, H , such that $HA + A^*H = 0$ if and only if all the eigenvalues of A are purely imaginary, and A has a complete set of eigenvectors. We then call H the symmetrizer of A .

First we show that purely imaginary eigenvalues and a complete set of eigenvectors imply that this matrix can be constructed. We know that these conditions mean that we can produce a transformation, T , such that

$$TAT^{-1} = \Lambda, \quad (\text{A.78})$$

where Λ is the diagonal matrix of the eigenvalues of A . Then we see that the symmetrizer is $H = T^*T$, because

$$HA + A^*H = T^*TA + A^*T^*T \quad (\text{A.79})$$

$$= T^*(TAT^{-1} + (T^*)^{-1}A^*T^*)T \quad (\text{A.80})$$

$$= T^*(\Lambda + \Lambda^*)T \quad (\text{A.81})$$

$$= 0. \quad (\text{A.82})$$

The second equality comes from multiplying by $I = T^{-1}T$ at the end of the first term, and $I = T^*(T^*)^{-1}$ at the beginning of the second term. In the third line, we note that $(TAT^{-1})^* = (T^*)^{-1}A^*T^*$, because the order of multiplication reverses when we take the transpose. Then $\Lambda = -\Lambda^*$, because we have purely imaginary eigenvalues. We see that a diagonalizable matrix with purely imaginary eigenvalues will have a symmetrizer, H , that behaves as described above.

We now must show the other direction: that the existence of the symmetrizer, H , implies that the eigenvalues of A are purely imaginary, and that A has a complete set of eigenvectors. We start by assuming a positive definite Hermitian matrix, H , which can be written as $H = S^*S$, because

$$H^* = (S^*S)^* = \left(\overline{\bar{S}^T S}\right)^T \quad (\text{A.83})$$

$$= (S^T \bar{S})^T \quad (\text{A.84})$$

$$= \bar{S}^T S \quad (\text{A.85})$$

$$= S^*S \quad (\text{A.86})$$

$$= H. \quad (\text{A.87})$$

We know that

$$S^*SA + A^*S^*S = 0, \quad (\text{A.88})$$

but then, from this we can say

$$(S^*)^{-1} [S^*SA + A^*S^*S] S^{-1} = SAS^{-1} + (S^*)^{-1}A^*S^* = 0, \quad (\text{A.89})$$

and so the matrix, SAS^{-1} , is anti-Hermitian. Recall that an anti-Hermitian matrix, M , is one that satisfies

$$\langle Mx, y \rangle = -\langle x, My \rangle. \quad (\text{A.90})$$

This means that all of the eigenvalues of M must be purely imaginary, because, for any eigenvalue, λ_j , with associated eigenvector, e_j , we get

$$\lambda_j \langle e_j, e_j \rangle = \langle \lambda_j e_j, e_j \rangle \quad (\text{A.91})$$

$$= \langle M e_j, e_j \rangle \quad (\text{A.92})$$

$$= -\langle e_j, M e_j \rangle \quad (\text{A.93})$$

$$= -\langle e_j, \lambda_j e_j \rangle \quad (\text{A.94})$$

$$= -\bar{\lambda}_j \langle e_j, e_j \rangle, \quad (\text{A.95})$$

so $\lambda_j = -\bar{\lambda}_j$.

Now consider the vector space $K_1 = \text{span}(e_1)^\perp$. If $\text{span}(e_1)$ is all the vectors made from a linear combination of e_1 , then this can be thought of as a line (for example, in 3D space if M is 3×3). Then K_1 is all the vectors normal to that line (i.e. in 3D space, this is a plane). We can then show that K_1 is M -invariant; in other words, if a vector, v is in K_1 , then Mv will also be in K_1 (i.e. $\forall v \in K_1 : Av \in K_1$). We can see this by choosing $v \in K_1$, then

$$\langle Mv, e_1 \rangle = -\langle v, Me_1 \rangle \quad (\text{A.96})$$

$$= -\langle v, \lambda_1 e_1 \rangle \quad (\text{A.97})$$

$$= -\bar{\lambda}_1 \langle v, e_1 \rangle \quad (\text{A.98})$$

$$= \lambda_1 \langle v, e_1 \rangle, \quad (\text{A.99})$$

but v was from K_1 , which is all the vectors normal to e_1 , meaning that this will be 0, and $\langle Mv, e_1 \rangle = 0$. So Mv is also normal to e_1 , and is therefore in K_1 .

Now we show that we can find the next eigenvector, e_2 , in K_1 ; there is no need to look in the general vector space. Suppose we have some general vector $c_1 e_1 + c_2 v$, where $v \in K_1$ and e_1 is our first eigenvector. Then, in order for this to be an eigenvector, we must have

$$A(c_1 e_1 + c_2 v) - \lambda'(c_1 e_1 + c_2 v) = 0, \quad (\text{A.100})$$

where λ' is the corresponding eigenvalue. From this we get

$$(\lambda_1 - \lambda')c_1 e_1 + (A - \lambda' I)c_2 v = 0. \quad (\text{A.101})$$

Then we can contract this with e_1 to get

$$\langle (\lambda_1 - \lambda') c_1 e_1, e_1 \rangle + \langle (A - \lambda' I) c_2 v, e_1 \rangle = 0 \quad (\text{A.102})$$

$$(\lambda_1 - \lambda') c_1 \langle e_1, e_1 \rangle = 0. \quad (\text{A.103})$$

Now if $\lambda' \neq \lambda_1$, then c_1 must be 0, and our new eigenvector must be entirely in K_1 . If $\lambda' = \lambda_1$, then we get

$$(A - \lambda' I) c_2 v = 0 \quad (\text{A.104})$$

or

$$Av - \lambda' v = 0. \quad (\text{A.105})$$

This means that our general vector can be an eigenvector only if we have a repeated eigenvalue ($\lambda' = \lambda_1$), and if this is the case, then the part that is entirely in K_1 (which is proportional to v), will also be an eigenvector. This means that if we restrict ourselves to K_1 , then we will always be able to find e_2 . In turn, this means that we can always find e_2 such that it is orthogonal to e_1 . We can then repeat the process (i.e. $K_2 = \text{span}(e_1, e_2)^\perp$, then K_2 is also M -invariant, and we can find e_3 in K_2 , etc.) until we run out of dimensions, showing that all of our eigenvectors will be orthogonal to one another.

Now we can choose the eigenvectors of our anti-Hermitian matrix, M , to be an orthonormal basis. In this basis, M will be diagonal, so M must be diagonalizable. From this, we go back and see that our anti-Hermitian matrix, SAS^{-1} , will be diagonalizable; meaning that it has a complete set of eigenvectors. We also know that SAS^{-1} will have purely imaginary eigenvalues, because it is anti-Hermitian. Since SAS^{-1} is simply a similarity transform of A , we know that A will also be diagonalizable, and that it will have purely imaginary eigenvalues. So we have shown that the existence of a symmetrizer, H , with the property $HA + A^*H = 0$, implies that A has purely imaginary eigenvalues and a complete set of eigenvectors.

Now we return to the system

$$\hat{u}_{,t}(\omega, t) = i\omega A \hat{u}(\omega, t). \quad (\text{A.106})$$

If the problem is well-posed, then the eigenvalues of the matrix A are real, and the eigenvalues of the matrix $\tilde{A} = i\omega A$, are purely imaginary. \tilde{A} will also have a complete set of eigenvectors, since A has this property. From this, we know we can construct a symmetrizer, $H\tilde{A} + \tilde{A}^*H = 0$, and therefore we can construct a

new norm such that

$$\frac{\partial}{\partial t} \langle \hat{u}, H\hat{u} \rangle = \langle \hat{u}_t, H\hat{u} \rangle + \langle \hat{u}, H\hat{u}_{,t} \rangle \quad (\text{A.107})$$

$$= \langle i\omega A\hat{u}, H\hat{u} \rangle + \langle \hat{u}, Hi\omega A\hat{u} \rangle \quad (\text{A.108})$$

$$= \left\langle \hat{u}, (\tilde{A}^*H + H\tilde{A})\hat{u} \right\rangle \quad (\text{A.109})$$

$$= 0. \quad (\text{A.110})$$

This shows that, if our constant coefficient system is well-posed, then we can always construct a norm from our symmetrizer, H , such that we have

$$|\hat{u}(\omega, t)|_H^2 = |\hat{u}(\omega, 0)|_H^2. \quad (\text{A.111})$$

In other words, this specific norm does not increase as time evolves. It is often called the *energy norm*, because, in physical systems, energy is typically conserved, and the value of this norm is often associated with the energy.

A.5 The constant-coefficient, first-order, multi-dimensional system

Next we examine the Cauchy problem in multiple space dimensions:

$$\begin{aligned} \frac{\partial u}{\partial t}(x, t) &= \sum_{\nu=1}^d A_\nu \frac{\partial u}{\partial x_\nu}(x, t), \quad x \in \mathbb{R}^d, \quad t \geq 0 \\ u(x, 0) &= f(x), \end{aligned} \quad (\text{A.112})$$

where $f(x)$ is 2π -periodic in all spatial dimensions, and the solutions should have the same property. We now normalize the symbol, $P(i\omega)$, as follows:

$$P(i\omega) = i \sum_{\nu=1}^d A_\nu \omega_\nu =: |\omega| P(i\omega'), \quad (\text{A.113})$$

where $|\omega|^2 = \sum |\omega_j|^2$ and $\omega' = \omega/|\omega|$. Then we have the following conditions for well-posedness:

1. For all $\omega' \in \mathbb{R}^d : |\omega'| = 1$, all of the eigenvalues of $P(i\omega')$ are purely imaginary.

2. $P(i\omega')$ has a complete set of eigenvectors. This means that they are linearly independent. It also means that there is some constant, K , and transformation, $T(\omega')$, for every ω' such that $|T| + |T^{-1}| \leq K$, and $T(\omega')P(i\omega')T^{-1}(\omega') = \Lambda(i\omega')$, where $\Lambda(i\omega')$ is the diagonal matrix of eigenvalues.

We then say that our first-order, multi-dimensional system, Eq. A.112, is *weakly hyperbolic* if it satisfies condition 1, and *strongly hyperbolic* if it satisfies both conditions 1 and 2. For the system to be *strictly hyperbolic*, $P(i\omega)$ must have distinct, purely-imaginary eigenvalues for all $\omega \in \mathbb{R}^d : \omega \neq 0$. This is more restrictive than strong hyperbolicity, since the eigenvalues do not necessarily have to be distinct in order for the matrix to be diagonalizable.

Symmetric hyperbolicity occurs when all of the coefficient matrices, A_ν , are Hermitian ($A_\nu = A_\nu^*$). Since Hermitian matrices are diagonalizable and have real eigenvalues, the symbol, $P(i\omega')$, will also be diagonalizable, and will have purely imaginary eigenvalues; therefore, symmetric hyperbolic systems automatically satisfy conditions 1 and 2, and are therefore strongly hyperbolic.

Next we show that, if we have a strongly hyperbolic, multi-dimensional, constant-coefficient, first-order system, then it will be well-posed. In Fourier space, our system, Eq. A.112, becomes

$$\hat{u}(\omega, t) = |\omega| P(i\omega') \hat{u}(\omega, t). \quad (\text{A.114})$$

We can then construct a Hermitian matrix, $\hat{H}(\omega') = T^*(\omega') T(\omega')$, such that

$$\hat{H}(\omega') P(i\omega') + P^*(i\omega') \hat{H}(\omega') = 0. \quad (\text{A.115})$$

Let (u, v) be the usual L_2 scalar product, and define a new scalar product, $(u, v)_H = (u, Hv)$, where

$$Hv(x, t) = \sum_{\omega} e^{i\langle \omega, x \rangle} \hat{H}(\omega') \hat{v}(\omega, t) \quad (\text{A.116})$$

is well-defined. Then Parseval's relation tells us that

$$(u, Hu) = \sum_{\omega} \left\langle \hat{u}(\omega, t), \hat{H}(\omega') \hat{u}(\omega, t) \right\rangle. \quad (\text{A.117})$$

Now recall that $|T| + |T^{-1}| \leq K$ and $\hat{H} = T^*T$. From this we can write

$$K^{-2}|\hat{u}|^2 \leq \frac{1}{|T^{-1}|^2}|\hat{u}|^2 \quad (\text{A.118})$$

$$\leq \langle \hat{u}, \hat{H}\hat{u} \rangle = |T\hat{u}|^2 \quad (\text{A.119})$$

$$\leq |T|^2|\hat{u}|^2 \quad (\text{A.120})$$

$$\leq K^2|\hat{u}|^2. \quad (\text{A.121})$$

The first inequality comes from $|T| + |T^{-1}| \leq K$, because then

$$K^{-2} \leq \frac{1}{(|T| + |T^{-1}|)^2} \leq \frac{1}{|T^{-1}|^2}, \quad (\text{A.122})$$

since $|T|, |T^{-1}| \geq 0$, so $|T| + |T^{-1}| \geq |T^{-1}|$. We can show the next step by seeing that

$$\frac{1}{|T^{-1}|^2}|\hat{u}|^2 = \frac{1}{|T^{-1}|^2}|T^{-1}T\hat{u}|^2 \quad (\text{A.123})$$

$$\leq \frac{|T^{-1}|^2}{|T^{-1}|^2}|T\hat{u}|^2 \quad (\text{A.124})$$

$$= |T\hat{u}|^2. \quad (\text{A.125})$$

The equality in the second line comes from

$$\langle \hat{u}, \hat{H}\hat{u} \rangle = \langle \hat{u}, T^*T\hat{u} \rangle \quad (\text{A.126})$$

$$= \langle T\hat{u}, T\hat{u} \rangle \quad (\text{A.127})$$

$$= |T\hat{u}|^2. \quad (\text{A.128})$$

We then can get to the last line by recalling again that $|T|^2 \leq (|T| + |T^{-1}|)^2 \leq K^2$.

Now we can see that our norm is bounded by

$$K^{-2}|\hat{u}|^2 \leq \langle \hat{u}, \hat{H}\hat{u} \rangle \leq K^2|\hat{u}|^2. \quad (\text{A.129})$$

Now we convert all of these to their spatial domain values using Parseval's relation:

$$K^{-2}\|u\|^2 = K^{-2} \sum_{\omega} |\hat{u}|^2 \quad (\text{A.130})$$

$$\leq (u, Hu) = \sum_{\omega} \langle \hat{u}, \hat{H}\hat{u} \rangle \quad (\text{A.131})$$

$$\leq K^2 \sum_{\omega} |\hat{u}|^2 = K^2\|u\|^2, \quad (\text{A.132})$$

and so (u, Hv) defines a scalar product. Note that, generally, in the spatial domain, H is not a matrix, but an operator. In addition to this, from $\hat{H}P + P^*\hat{H} = 0$, we see that

$$(u, HPu) + (Pu, Hu) = \sum_{\omega} \langle \hat{u}, \hat{H}P\hat{u} \rangle + \sum_{\omega} \langle P\hat{u}, \hat{H}\hat{u} \rangle \quad (\text{A.133})$$

$$= \sum_{\omega} \langle \hat{u}, \hat{H}P\hat{u} \rangle + \sum_{\omega} \langle \hat{u}, P^*\hat{H}\hat{u} \rangle \quad (\text{A.134})$$

$$= \sum_{\omega} \langle \hat{u}, (\hat{H}P + P^*\hat{H})\hat{u} \rangle \quad (\text{A.135})$$

$$= 0. \quad (\text{A.136})$$

This means that our norm stays invariant:

$$\frac{\partial}{\partial t}(u, Hu) = 0. \quad (\text{A.137})$$

This shows that strong hyperbolicity implies well-posedness.

In this section, we have not proven that hyperbolicity implies well-posedness for the general system; instead we have simply aimed to introduce the concepts of strong, weak, strict, and symmetric hyperbolicity, and their link to the well-posedness of a system. By showing that strong hyperbolicity implies well-posedness for the first-order, multi-dimensional Cauchy problem with periodic initial data, we can get a sense of how this might work in more general systems, and an intuitive understanding of why hyperbolicity is so important in numerics. We now move on to defining hyperbolicity in differential geometry notation, so that we can apply it to the elastic-matter system.

A.6 Hyperbolicity in differential geometry notation

We now discuss the definition of hyperbolicity in differential geometry notation. The definitions in this section originally come from Beig and Schmidt [30], and Anile [31], and are also written in much the same form as this in an appendix in [1]. They are included here for completeness.

We express our system of equations in the following form:

$$P_{\alpha\beta}{}^c \omega^{\beta}_{,c} + \text{l.o.} = 0, \quad (\text{A.138})$$

where l.o. refers to lower order terms, the indices α label the equations, and the indices β label the variables. In order for this system to have a unique solution, we must have the number of values of α to be the same as the number of values of β . If we relate this to the previous section where we looked at hyperbolicity in separate time and space coordinates, ω^β takes the place of our vector of variables, u , and $P_{\alpha\beta}^c$ is similar to A or A_ν in the constant-coefficient, first-order systems. The difference is that the time derivative coefficient is also included in this object.

If α and β are of the same type (which is not true in the original form of our evolution equations—consider the linearized form, $\delta^a_{[a}\delta^d_{b]}\psi^A{}_{d,c} = \text{l.o.}$, where β would correspond to the index d , and α would correspond to $[a, b]$), and if we have the property $P_{\alpha\beta}^c = P_{\beta\alpha}^c$, then we can construct a conserved current,

$$J^c,_c = \text{l.o.}, \quad (\text{A.139})$$

where

$$J^c := P_{\alpha\beta}^c \omega^\alpha \omega^\beta. \quad (\text{A.140})$$

If there exists a covector, t_c , such that

$$E(\omega, \omega) := t_c J^c = t_c P_{\alpha\beta}^c \omega^\alpha \omega^\beta, \quad (\text{A.141})$$

is positive definite, then the system is symmetric hyperbolic, where t_c here is called the *subcharacteristic covector*. In relativity, we expect t_c to be time-like. This E is the same energy norm that we discussed previously; it should be conserved over time if the problem is well-posed. We know from before that E allows us to estimate an L_2 norm of the solution in terms of the initial and boundary data.

The characteristic direction is a covector, k_c , such that $\det(k_c P_{\alpha\beta}^c) = 0$. This can be compared a standard eigenvalue equation in linear algebra, where $\det(A - \lambda I) = 0$ for the eigenvalues, λ of A . Then, the characteristic variable corresponding to k_c is the non-zero vector, ω^β , such that

$$k_c P_{\alpha\beta}^c \omega^\beta = 0. \quad (\text{A.142})$$

This equation defines ω^β as an eigenvector of the spatial part of $k_c P_{\alpha\beta}^c$. Then we can say that one solution to the principle part of the equation will be a plane wave, ω^β , with wave number, k_c . For a causal system, information should not propagate faster than the speed of light, so k_c should be space-like or null. We can work this out another way by thinking of the the standard plane-wave in 1D form, which

gives us

$$e^{ik_c u^c} = e^{i(k_0 t + k_1 x)} = e^{ik_1 \left(x - \left(\frac{-k_0}{k_1} \right) t \right)} = e^{ik_1(x - \lambda t)}, \quad (\text{A.143})$$

where we define $\lambda = -k_0/k_1$. Then we can read off λ to be the standard wave speed, which, for causality, should have the property, $|\lambda| \leq 1$. This will require that $|k_0| \leq |k_1|$, which implies that k_c is space-like.

Now we move on to describing the second-order system. The second order system had the form

$$P_{\alpha\beta}{}^{cd} \omega^\beta{}_{,cd} + \text{l.o.} = 0. \quad (\text{A.144})$$

Then our definition of the characteristic variable, ω^β , is as follows:

$$k_c k_d P_{\alpha\beta}{}^{cd} \omega^\beta = 0, \quad (\text{A.145})$$

which has the same plane-wave-solution interpretation as for the first-order system.

We also note that it is often useful to decompose the characteristic equation with respect to a preferred direction by decomposing the characteristic direction as follows:

$$k_a = \lambda n_a - e_a, \quad (\text{A.146})$$

where n_a is a unit time-like covector, and e_a is a unit space-like covector, which is normal to n_a . We can then say that λ is the characteristic velocity of the characteristic variable, ω^β , relative to a n_a observer, in the direction of e_a . Now k_c is normal to the plane spanned by the set of vectors,

$$v^a = n^a - \lambda e^a + s^a, \quad (\text{A.147})$$

where s^a is any vector normal to n_a and e_a , so that $k_a v^a = 0$. The plane spanned by the set of v^a is called the characteristic plane. The relative speed between n^a and v^a can be read off from the Lorentz factor:

$$\frac{1}{\sqrt{1 - u^2}} = \frac{n^a v_a}{|n||v|}, \quad (\text{A.148})$$

where u is the relative speed. Working this out shows that the relative speed is

$$\sqrt{\lambda^2 + s^a s_a} \geq \lambda. \quad (\text{A.149})$$

A disturbance to the solution should move along with speed, λ , relative to a n^a observer, so we see that $s^a = 0$, and the disturbance will move along the vector $n^a - \lambda e^a$.

There are a few logical choices for n^a in the decomposition of the characteristic direction. One is to use the unit normal surfaces of constant time; this is how we get the coordinate characteristic speeds calculated and used in our numerical scheme. Another option is to set $n^a = u^a$, which is generally easier to compute, and gives the characteristic speeds relative to the matter.

To more formally make a connection to the previous section where we discussed the non-relativistic understanding of hyperbolicity, we write the first-order characteristic equation instead as

$$\mathcal{P}_e \omega = \lambda \omega, \quad (\text{A.150})$$

where

$$\mathcal{P}_e := (n_a P^a)^{-1} (e_b P^b), \quad (\text{A.151})$$

and the indices have been omitted for clarity. Note that the above equation is equivalent to Eq. A.142. We also recall that if n_a is subcharacteristic, then $n_a P^a$ will have an inverse, since it will be positive definite. From the above equation, we see that ω is an eigenvector of \mathcal{P}_e in the traditional sense, and the associated eigenvalue is λ .

We can then write the definitions for hyperbolicity for a relativistic system:

- The system is *weakly hyperbolic* if \mathcal{P}_e has real eigenvalues for all e_a normal to n_a .
- The system is *strongly hyperbolic* if \mathcal{P}_e has a basis of real eigenvectors that depend continuously on e_a .
- The system is *symmetric hyperbolic* if \mathcal{P}_e is symmetric. As before, a real symmetric matrix is always diagonalizable with real eigenvalues, so symmetric hyperbolicity implies strong hyperbolicity.

These definitions are equivalent to those mentioned in the previous section.

More generally, if there exists a positive definite Hermitian matrix, \mathcal{H} , such that $\mathcal{P}_e \mathcal{H}$ is symmetric independent of e^a , then we call this system *symmetrizable*, where \mathcal{H} is called the *symmetrizer*. Then, instead of choosing the positive definite energy norm as above, we choose

$$E := \mathcal{H}_{\alpha\beta} \omega^\alpha \omega^\beta, \quad (\text{A.152})$$

so in the Newtonian case, this becomes

$$E = u^T \mathcal{H} u = (u, \mathcal{H} u), \quad (\text{A.153})$$

which is the general energy norm we saw in the previous section.

Appendix B

Additional information about the elasticity formulation

B.1 3+1 split of spacetime

For reference, we assemble some standard formulas. In 3+1 numerical relativity, the spacetime metric g_{ab} is split into a spatial metric γ_{ij} with inverse γ^{ij} , a lapse α and shift β^i , as

$$g_{00} = -\alpha^2 + \beta_i \beta^i, \quad g_{0i} = \beta_i, \quad g_{ij} = \gamma_{ij}, \quad (\text{B.1})$$

where we define indices on β^i to be moved implicitly with γ_{ij} . The (absolute value of the) determinant of the 4-metric is given by

$$g_x = \alpha^2 \gamma_x \quad (\text{B.2})$$

and hence the volume forms on M^3 and M^4 are related by

$$\epsilon_{0ijk} = \alpha \epsilon_{ijk}. \quad (\text{B.3})$$

The inverse 4-metric is

$$g^{00} = -\alpha^{-2}, \quad g^{0i} = \alpha^{-2} \beta^i, \quad g^{ij} = \gamma^{ij} - \alpha^{-2} \beta^i \beta^j. \quad (\text{B.4})$$

The covector normal to the surfaces of constant t has components

$$n_0 = -\alpha, \quad n_i = 0, \quad (\text{B.5})$$

and hence

$$n^0 = \alpha^{-1}, \quad n^i = -\alpha^{-1}\beta^i. \quad (\text{B.6})$$

Hence the projector into the surfaces of constant t

$$\gamma_{ab} := g_{ab} + n_a n_b \quad (\text{B.7})$$

has components

$$\gamma_{00} = \beta_i \beta^i, \quad \gamma_{0i} = \beta_i, \quad \gamma_{ij} = \gamma_{ij}, \quad (\text{B.8})$$

and

$$\gamma^{00} = 0, \quad \gamma^{0i} = 0, \quad \gamma^{ij} = \gamma^{ij}. \quad (\text{B.9})$$

We define the convective derivative to be the derivative along the 4-velocity,

$$u^a \frac{\partial}{\partial x^a} \propto \frac{\partial}{\partial t} + \hat{v}^i \frac{\partial}{\partial x^i}. \quad (\text{B.10})$$

The factor of proportionality is given by the normalization condition

$$u^a u^b g_{ab} = -1. \quad (\text{B.11})$$

We find

$$u^a = (u^t, u^i) = \alpha^{-1} W(1, \hat{v}^i), \quad (\text{B.12})$$

$$u_a = (u_t, u_i) = W(-\alpha + v_j \beta^j, v_i), \quad (\text{B.13})$$

where

$$\hat{v}^i := \alpha v^i - \beta^i, \quad (\text{B.14})$$

$$W := (1 - v_i v^i)^{-1/2}, \quad (\text{B.15})$$

and where we define the indices on v^i to be moved implicitly with γ_{ij} . The scalar

$$-u^a n_a = W \quad (\text{B.16})$$

gives the Lorentz factor of the relative velocity between the matter and the time slices.

B.2 The Newtonian limit

We obtain the limit of Newtonian motion in the absence of gravity in two steps. In the first step, we let the spacetime go to Minkowski spacetime in adapted coordinates,

$$ds^2 = -dt^2 + \gamma_{ij} dx^i dx^j, \quad (\text{B.17})$$

where γ_{ij} is flat and independent of t , but x^i could still be curvilinear coordinates. Hence

$$\hat{v}^i = v^i, \quad (\text{B.18})$$

and the advection equation (Eq. 3.55) becomes

$$(\partial_t + v^i \partial_i) k_{AB} = 0. \quad (\text{B.19})$$

In the second step, we use dimensional analysis of the special relativistic equations of motion to insert a parameter c representing the speed of light, as follows:

$$n, \quad (\text{B.20})$$

$$c^{-1} v^i, \quad (\text{B.21})$$

$$c^{-2} \epsilon, \quad c^{-2} p, \quad c^{-2} \pi_{ij}, \quad (\text{B.22})$$

$$c^{-3} \pi_{0i}, \quad c^{-4} \pi_{00}, \quad (\text{B.23})$$

for the primitive variables, and

$$D, \quad c^{-1} S_i, \quad c^{-2} \tau, \quad (\text{B.24})$$

$$c^{-1} \mathcal{F}(D)^i, \quad c^{-2} \mathcal{F}(S_j)^i, \quad c^{-3} \mathcal{F}(\tau)^i, \quad (\text{B.25})$$

for the conserved variables. We then take the limit $c \rightarrow \infty$ of the relevant equations for Minkowski spacetime. In this limit,

$$W = 1, \quad (\text{B.26})$$

$$u^a = n^a, \quad (\text{B.27})$$

$$h_{ab} = \gamma_{ab}, \quad (\text{B.28})$$

$$\psi^A_t = 0, \quad (\text{B.29})$$

$$\pi = \gamma^{ij}\pi_{ij} = 0, \quad (\text{B.30})$$

$$D = n, \quad (\text{B.31})$$

$$S_i = nv_i, \quad (\text{B.32})$$

$$\tau = n(v^2/2 + \epsilon), \quad (\text{B.33})$$

$$\mathcal{F}(D)^i = nv^i, \quad (\text{B.34})$$

$$\mathcal{F}(S_j)^i = nv_jv^i + p\delta^i_j + \pi^i_j, \quad (\text{B.35})$$

$$\mathcal{F}(\tau)^i = n(v^2/2 + \epsilon)v^i + pv^i + \pi^i_jv^j, \quad (\text{B.36})$$

$$\mathcal{S}(D) = 0, \quad (\text{B.37})$$

$$\mathcal{S}(\tau) = 0, \quad (\text{B.38})$$

$$\mathcal{S}(S_j) = \frac{1}{2}\sqrt{\gamma_x}T^{ik}\partial_j\gamma_{ik}. \quad (\text{B.39})$$

where v^i and π_{ij} are now the Newtonian velocity and stress tensor, and their indices are moved implicitly with the metric γ_{ij} of Euclidean space. Instead of requiring p , f_1 and f_2 as functions of h (the relativistic enthalpy, which includes the rest mass energy) and n , we need them as functions of ϵ and n . The reconstruction of n , v_i and ϵ from D , S_i and τ becomes explicit for the equations of state we consider.

B.3 Shear invariants and the shear scalar

The three eigenvalues of η^A_B can be parameterized as $\{a, b, 1/(ab)\}$. We then find that in the unsheared state $a = b = 1$,

$$I^1 = I^2 = 3, \quad (\text{B.40})$$

$$I_{,a}^1 = I_{,b}^1 = I_{,a}^2 = I_{,b}^2 = 0, \quad (\text{B.41})$$

$$I_{,aa}^1 = I_{,bb}^1 = 2, \quad I_{,ab}^1 = 1, \quad (\text{B.42})$$

$$I_{,aa}^2 = I_{,bb}^2 = 8, \quad I_{,ab}^2 = 4. \quad (\text{B.43})$$

Hence $4(I^1 - 3)$ and $I^2 - 3$ are the same function of the shear up to quadratic order. This is not the result of a bad choice of I^α but a property of any shear invariant. It is related to the fact that the characteristic speeds in the unsheared state depend on f_1 and f_2 only through the one combination $f_1 + 4f_2$ that appears in the shear modulus (Eq. 3.272).

Therefore, to model linear elasticity correctly, it is sufficiently general to make the ansatz

$$\epsilon(s, n, I^\alpha) = \check{\epsilon}(n, s) + \frac{\check{\mu}(n, s)}{n} \mathcal{S}(I^1, I^2), \quad (\text{B.44})$$

where the single *shear scalar* \mathcal{S} obeys

$$\mathcal{S} = 0, \quad (\text{B.45})$$

$$2\frac{\partial \mathcal{S}}{\partial I^1} + 8\frac{\partial \mathcal{S}}{\partial I^2} = 1 \quad (\text{B.46})$$

in the unsheared state $I^1 = I^2 = 3$, but is otherwise arbitrary. For any such choice of \mathcal{S} , $\check{\mu}(n, s)$ evaluates to the usual shear modulus (Eq. 3.272) in the Newtonian limit, and the equations of motion are the same when linearized about the unsheared state.

Clearly there are many possibilities of defining a shear scalar that obeys these conditions, but we are not aware of any physical reason given in the literature for why a specific choice should be preferred, or of values given for f_1 and f_2 independently. Indeed, different choices can only be distinguished in a nonlinear deformation regime, in which case there is no particular reason to assume that ϵ does not depend on I^1 and I^2 in a more generic way.

An equation of state for copper used in [62] uses the shear scalar

$$\mathcal{S}_{\text{Cran}} := \frac{3I^2 - (I^1)^2}{12}, \quad (\text{B.47})$$

which is quadratic in the eigenvalues of η^A_B . In [54] the shear scalar

$$\mathcal{S}_{\text{KS}} := \frac{(I^1)^3 - I^1 I^2 - 18}{24}, \quad (\text{B.48})$$

which is cubic, is suggested for what seem to be aesthetic reasons. Yet another shear scalar is

$$\mathcal{S}_{\text{VM}} := s^{ab} s_{ab} = \frac{I^2 - 2I^1 + 3}{4}, \quad (\text{B.49})$$

where

$$s_{ab} := \frac{1}{2}(h_{ab} - \eta_{ab}) \quad (\text{B.50})$$

is the “constant volume shear tensor” defined in [45]. In the Newtonian limit, near the unsheared state, \mathcal{S}_{VM} is related to the Von Mises stress scalar (assuming stress and strain are related linearly). It gives the same values of f_1 and f_2 in the unsheared state as $\mathcal{S}_{\text{Cran}}$.

Appendix C

Equations of state

We now consider examples of equations of state of the form of Eq. B.44. The following general expressions will be useful:

$$h = 1 + \check{\epsilon} + \frac{\check{\mu}}{n} \mathcal{S} + \frac{p}{n}, \quad (\text{C.1})$$

$$p = n^2 \frac{\partial \check{\epsilon}}{\partial n} + \left(n \frac{\partial \check{\mu}}{\partial n} - \check{\mu} \right) \mathcal{S}, \quad (\text{C.2})$$

$$f_\alpha = \frac{\check{\mu}(n, s)}{n} \frac{\partial \mathcal{S}}{\partial I^\alpha}. \quad (\text{C.3})$$

In principle we can eliminate s from these two equations to obtain p , f_1 and f_2 , as functions of (n, h, I^1, I^2) , which we need in the recovery of the primitive from the conserved variables.

C.1 A toy relativistic EOS

As a toy model for a relativistic equation of state, we take $\check{\epsilon}$ from the commonly used “Gamma-law” hot equation of state, and make the shear modulus $\check{\mu}$ a power of the density only, namely

$$\check{\epsilon}(n, s) = \frac{K(s)}{\Gamma - 1} n^{\Gamma - 1}, \quad (\text{C.4})$$

$$\check{\mu}(n, s) = \kappa n^\lambda, \quad (\text{C.5})$$

where Γ , κ and λ are constants. This is motivated by the fact that in neutron star crusts $\mu \propto n^{4/3}$, with the factor of proportionality only weakly temperature-dependent. The bulk modulus in neutron stars is given by the nuclear interactions, while the shear modulus is provided by Coulomb interactions, which makes it

independent and much smaller. Following [54], we choose \mathcal{S} as \mathcal{S}_{KS} given by Eq. B.48.

The expressions we need for the conserved to primitive variables conversion are then

$$p(h, n, I^\alpha) = \frac{\Gamma - 1}{\Gamma} n(h - 1) + \frac{\lambda - \Gamma}{\Gamma} \kappa n^\lambda \mathcal{S}, \quad (\text{C.6})$$

$$p(\epsilon, n, I^\alpha) = (\Gamma - 1)n\epsilon + (\lambda - \Gamma)\kappa n^\lambda \mathcal{S}, \quad (\text{C.7})$$

$$h(p, n, I^\alpha) = 1 + \frac{\Gamma}{\Gamma - 1} \frac{p}{n} + \frac{\Gamma - \lambda}{\Gamma - 1} \kappa n^{\lambda-1} \mathcal{S}, \quad (\text{C.8})$$

$$f_1 = \kappa n^{\lambda-1} \frac{3(I^1)^2 - I^2}{24}, \quad (\text{C.9})$$

$$f_2 = -\kappa n^{\lambda-1} \frac{I^1}{24}. \quad (\text{C.10})$$

The characteristic speeds in the unsheared state are

$$\lambda_T^2 = \frac{\kappa n^{\lambda-1}}{1 + \Gamma\epsilon}, \quad (\text{C.11})$$

$$\lambda_L^2 = \frac{\Gamma(\Gamma - 1)\epsilon + \frac{4}{3}\kappa n^{\lambda-1}}{1 + \Gamma\epsilon}. \quad (\text{C.12})$$

C.2 Cranfield EOS

The equation of state for copper used in [62] for Newtonian shock tube problems, translated into our notation, is

$$\epsilon(s, n, I^\alpha) = A(n) + B(n)K(s) + C(n)\mathcal{S}, \quad (\text{C.13})$$

$$A := \frac{K_0}{2\alpha^2} \left[\left(\frac{n}{n_0} \right)^\alpha - 1 \right]^2, \quad (\text{C.14})$$

$$B := c_V T_0 \left(\frac{n}{n_0} \right)^\gamma, \quad (\text{C.15})$$

$$K := e^{\frac{s}{c_V}} - 1, \quad (\text{C.16})$$

$$C := B_0 \left(\frac{n}{n_0} \right)^{\beta+4/3}, \quad (\text{C.17})$$

where the parameters $K_0 = c_0^2 - (4/3)b_0^2$ and $B_0 = b_0^2$ are the squared bulk sound and shear speeds, respectively, and where \mathcal{S} is $\mathcal{S}_{\text{Cran}}$ given by Eq. B.47. We need

the following forms of the equation of state:

$$p(s, n, I^\alpha) = n[nA' + \gamma BK + (\beta + 4/3)C\mathcal{S}], \quad (\text{C.18})$$

$$\begin{aligned} p(h, n, I^\alpha) &= \frac{n}{\gamma+1} \left[\gamma(h-1) - \gamma A + nA' \right. \\ &\quad \left. + (\beta + 4/3 - \gamma)C\mathcal{S} \right], \end{aligned} \quad (\text{C.19})$$

$$\begin{aligned} p(\epsilon, n, I^\alpha) &= n \left[\gamma\epsilon - \gamma A + nA' \right. \\ &\quad \left. + (\beta + 4/3 - \gamma)C\mathcal{S} \right], \end{aligned} \quad (\text{C.20})$$

$$\begin{aligned} h(p, n, I^\alpha) &= 1 + \frac{\gamma+1}{\gamma} \frac{p}{n} + A - \frac{1}{\gamma} nA' \\ &\quad - \frac{1}{\gamma} (\beta + 4/3 - \gamma)C\mathcal{S}, \end{aligned} \quad (\text{C.21})$$

$$f_1 = -\frac{CI^1}{6}, \quad (\text{C.22})$$

$$f_2 = \frac{C}{4} \quad (\text{C.23})$$

C.3 JWL

The Jones, Wilkins and Lee (JWL) equation of state is not a solid equation of state; nevertheless, we summarize it here because of its use in the BOD1 test in Section 4.4. This is a reactive equation of state for explosives; in the test here, we simply assume that the material simulated is completely reacted, and ignore the reaction process. We use the version presented in [84], with the parameters for PBX 9404. The pressure is as follows:

$$p = A \left(1 - \frac{\omega}{R_1 V} \right) e^{-R_1 V} + B \left(1 - \frac{\omega}{R_2 V} \right) e^{-R_2 V} + \frac{\omega E}{V}, \quad (\text{C.24})$$

where

$$V := \frac{n_0}{n}, \quad (\text{C.25})$$

and

$$E := n_0\epsilon + (\lambda - 1)n_0\epsilon^* + \lambda n_0 Q. \quad (\text{C.26})$$

Here, the parameter λ represents the degree to which the material is reacted, so we simply set this to $\lambda = 1$ (the reaction is already complete).

Appendix D

Estimating error

The relative error in a variable, $f(x)$, is given by:

$$e(x) = \frac{f(x) - f_{\text{ref}}(x)}{f_{\text{ref}}(x)}, \quad (\text{D.1})$$

where $f_{\text{ref}}(x)$ is some reference value for $f(x)$ (ideally the exact solution). Then, the L^2 norm of the error, e , over the grid $[a, b]$ is given by

$$\|e\|_2 := \left(\int_a^b |e(x)|^2 dx \right)^{\frac{1}{2}}. \quad (\text{D.2})$$

On a grid discretized into N cells, we only have a numerical approximation to the value of our variable (f_i) at cell centers (i). The cell centers are located at $x = x_i = a + (i - \frac{1}{2})h$, where $h = (b - a)/N$. Then our calculation of the error in each cell becomes

$$e_i = \frac{f_i - f_{\text{ref}}(x_i)}{f_{\text{ref}}(x_i)}. \quad (\text{D.3})$$

We can then approximate the L^2 norm of e to be

$$\|e\|_2 \approx \left(h \sum_{i=1}^N |e_i|^2 \right)^{\frac{1}{2}} \quad (\text{D.4})$$

In two spatial dimensions, we integrate over both x and y , meaning that

$$\|e\|_2 := \left(\int_a^b \int_c^d |e(x, y)|^2 dx dy \right)^{\frac{1}{2}}. \quad (\text{D.5})$$

In the numerical approximation, this becomes

$$\|e\|_2 \approx \left(h^2 \sum_{i=1}^N \sum_{j=1}^M |e_{i,j}|^2 \right)^{\frac{1}{2}}, \quad (\text{D.6})$$

where we have assumed that the grid satisfies $h = (b - a)/N = (d - c)/M$.

In some cases, it is not possible to obtain the exact solution, in these cases, we use a high-resolution simulation as a reference value for the variable. We find the reference value at the relevant low-resolution cell center by averaging the high-resolution solution over the corresponding cells. If the high-resolution solution is on an $n \times n$ grid, and the low-resolution solution is on an $N \times N$ grid, where $n/N =: r$, then we find f_{ref} as follows:

$$f_{\text{ref},i,j} := \frac{1}{r^2} \sum_{l=r(i-1)+1}^{r(i-1)+r} \sum_{k=r(j-1)+1}^{r(j-1)+r} f_{\text{hr},l,k}, \quad (\text{D.7})$$

where i and j run from 1 to N over the grid, and l and k run from 1 to n . We calculate the reference value for each low-resolution cell, and then calculate the error in each cell from this.

All of the errors mentioned in this work are calculated as shown above unless otherwise stated.

D.1 Richardson scaling

For a p -th order convergent method, we expect that the exact solution differs from the numerical solution by

$$f_{\text{exact}} = f_1 + C (h_1)^p, \quad (\text{D.8})$$

where f_1 is the numerical solution using a given grid spacing, h_1 , and C is a constant factor. For another resolution, f_2 , the constant factor, C , should remain the same if we have convergence in the order predicted:

$$f_{\text{exact}} = f_2 + C (h_2)^p. \quad (\text{D.9})$$

One way to evaluate convergence is to estimate the value of C ,

$$C = \frac{f_{\text{exact}} - f_n}{h_n^p}, \quad (\text{D.10})$$

or

$$C (x_{\max} - x_{\min})^p = (f_{\text{exact}} - f_n) n^p, \quad (\text{D.11})$$

where n is the number of grid points, and x_{\min} and x_{\max} are the x values at the left and right edges of the grid (so the left-hand side of Eq. D.11 should be constant for different resolutions). We can then plot this value for different resolutions and compare. This can give us more information than a simple norm over the entire grid, which does not allow us to distinguish between discontinuous and smooth regions. For example, we may be able to observe Richardson scaling at the expected order of convergence in smooth regions, but not in regions near sharp discontinuities; however, the norm of the error over the grid would only tell us that the simulation was not converging as expected. Here, we use a high resolution reference solution in the place of the exact solution.

Appendix E

Relativistic conservation-error calculation bug

For a single-material evolution of the special relativistic 4-wave Riemann test, the error in conservation should not go above the level of round-off error (without use of the GFM, the method should be conservative). However, while we see that this is true for D and S_i , the error in conservation for τ is several orders of magnitude larger than round-off error ($\sim 10^{-9}$), and is characterized by time periods of very low variation, separated by sudden jumps (of the order 10^{-9}).

The period between jumps is not fixed, but appears to typically be ~ 0.1 in coordinate time, where the time step is $\Delta t = 0.005$. These jumps originate as jumps in the value of τ that is integrated over the grid at each time step (i.e. $\sum_{\text{grid}} q$ in Eq. 4.25), while the initially integrated and then subsequently evolved value of τ (i.e. Q in Eq. 4.25 and the following discussion) displays no such jumps.

Although we were never able to find the source of the bug (and subsequently switched from using the `ElasticEvolution` code to the `MultiModel` code), we suspect that it originates somewhere in the error calculation (likely the integration calculation) itself. Because the conservation error calculation is not integral to the overall functioning of the code, and was only ever intended for use as a means of illustrating the lack of conservation inherent in the GFM, this particular issue was left unresolved.

Since the Newtonian tests are not affected by this bug, we limit our discussion to conservation errors in Newtonian tests.

Appendix F

Tensor properties of g^{AB} on matter space

The object g^{AB} is defined to be the push forward of the spacetime metric onto matter space:

$$g^{AB} := \psi^A{}_a \psi^B{}_b g^{ab}, \quad (\text{F.1})$$

where we know that $\psi^A{}_a$, the configuration gradient, is defined to be the partial derivative of the map with respect to the spacetime coordinates,

$$\psi^A{}_a := \frac{\partial \xi^A}{\partial x^a} = \frac{\partial \chi^A(x^a)}{\partial x^a}, \quad (\text{F.2})$$

where the map χ gives the matter space coordinates ξ^A corresponding to the spacetime coordinates x^a . If we construct a new coordinate system on matter space, then we can write a coordinate transformation from the old to the new as

$$\bar{\xi}^{A'} = \bar{\xi}^{A'}(\xi^A). \quad (\text{F.3})$$

In the new coordinates, we write $\bar{g}^{A'B'}$ as follows:

$$\bar{g}^{A'B'} = \bar{\psi}^{A'}{}_a \bar{\psi}^{B'}{}_b g^{ab} = \frac{\partial \bar{\xi}^{A'}}{\partial x^a} \frac{\partial \bar{\xi}^{B'}}{\partial x^b} g^{ab}, \quad (\text{F.4})$$

We can use the chain rule to rewrite these partial derivatives as follows:

$$\bar{g}^{A'B'} = \frac{\partial \bar{\xi}^{A'}}{\partial \xi^A} \frac{\partial \xi^A}{\partial x^a} \frac{\partial \bar{\xi}^{B'}}{\partial \xi^B} \frac{\partial \xi^B}{\partial x^b} g^{ab}, \quad (\text{F.5})$$

but we can simply rewrite this as the following, in terms of the matter-space object g^{AB} :

$$\bar{g}^{A'B'} = \frac{\partial \bar{\xi}^{A'}}{\partial \xi^A} \frac{\partial \bar{\xi}^{B'}}{\partial \xi^B} g^{AB}. \quad (\text{F.6})$$

This is the usual coordinate transformation for tensors on matter space.

References

- [1] C. Gundlach, I. Hawke, and S. J. Erickson. A conservation law formulation of nonlinear elasticity in general relativity. *Classical and Quantum Gravity*, 29(1):015005, January 2012.
- [2] W. Baade and F. Zwicky. Cosmic Rays from Super-novae. *Proceedings of the National Academy of Science*, 20:259–263, May 1934.
- [3] R. C. Tolman. Static Solutions of Einstein’s Field Equations for Spheres of Fluid. *Physical Review*, 55:364–373, February 1939.
- [4] J. R. Oppenheimer and G. M. Volkoff. On Massive Neutron Cores. *Physical Review*, 55:374–381, February 1939.
- [5] A. Hewish, S. J. Bell, J. D. H. Pilkington, P. F. Scott, and R. A. Collins. Observation of a Rapidly Pulsating Radio Source. *Nature*, 217:709–713, February 1968.
- [6] T. Gold. Rotating Neutron Stars as the Origin of the Pulsating Radio Sources. *Nature*, 218:731–732, May 1968.
- [7] S. Chandrasekhar. The Maximum Mass of Ideal White Dwarfs. *Astrophysical Journal*, 74:81, July 1931.
- [8] D. Page and S. Reddy. Dense Matter in Compact Stars: Theoretical Developments and Observational Constraints. *Annual Review of Nuclear and Particle Science*, 56:327–374, November 2006.
- [9] S. L. Shapiro and S. A. Teukolsky. *Black holes, white dwarfs, and neutron stars: The physics of compact objects*. 1983.
- [10] N. Chamel and P. Haensel. Physics of Neutron Star Crusts. *Living Reviews in Relativity*, 11:10–+, December 2008.
- [11] C. J. Horowitz and K. Kadau. Breaking Strain of Neutron Star Crust and Gravitational Waves. *Physical Review Letters*, 102(19):191102–+, May 2009.

- [12] P. B. Jones. Nature of fault planes in solid neutron star matter. *The Astrophysical Journal*, 595(1):342, 2003.
- [13] A. I. Chugunov and C. J. Horowitz. Breaking stress of neutron star crust. *Monthly Notices of the Royal Astronomical Society*, 407:L54–L58, September 2010.
- [14] R. Perna and J. A. Pons. A Unified Model of the Magnetar and Radio Pulsar Bursting Phenomenology. *Astrophysical Journal*, 727:L51, February 2011.
- [15] Y. Levin and M. Lyutikov. On the dynamics of mechanical failures in magnetized neutron star crusts. *Monthly Notices of the Royal Astronomical Society*, 427:1574–1579, December 2012.
- [16] A. M. Beloborodov and Y. Levin. Thermoplastic waves in magnetars. *The Astrophysical Journal Letters*, 794(2):L24, 2014.
- [17] R. C. Duncan. Global Seismic Oscillations in Soft Gamma Repeaters. *Astrophysical Journal*, 498:L45+, May 1998.
- [18] A. W. Steiner and A. L. Watts. Constraints on Neutron Star Crusts from Oscillations in Giant Flares. *Physical Review Letters*, 103(18):181101, October 2009.
- [19] G. Baym, C. Pethick, D. Pines, and M. Ruderman. Spin Up in Neutron Stars: The Future of the Vela Pulsar. *Nature*, 224:872–874, November 1969.
- [20] K. Glampedakis and N. Andersson. Hydrodynamical Trigger Mechanism for Pulsar Glitches. *Physical Review Letters*, 102(14):141101, April 2009.
- [21] P. W. Anderson and N. Itoh. Pulsar glitches and restlessness as a hard superfluidity phenomenon. *Nature*, 256:25–27, July 1975.
- [22] B. Link. Instability of superfluid flow in the neutron star inner crust. *Monthly Notices of the Royal Astronomical Society*, 422:1640–1647, May 2012.
- [23] D. Tsang, J. S. Read, T. Hinderer, A. L. Piro, and R. Bondarescu. Resonant Shattering of Neutron Star Crusts. *Physical Review Letters*, 108(1):011102, January 2012.
- [24] D. Tsang. Shattering Flares during Close Encounters of Neutron Stars. *Astrophysical Journal*, 777:103, November 2013.

- [25] B. van Leer. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In *Lecture Notes in Physics, Berlin Springer Verlag*, volume 18 of *Lecture Notes in Physics, Berlin Springer Verlag*, pages 163–168, 1973.
- [26] B. Einfeldt. On godunov-type methods for gas dynamics. 25(2):294–318, 1988.
- [27] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.
- [28] R. J. LeVeque. *Finite-volume methods for hyperbolic problems*. Cambridge University Press, New York, NY, USA, 2002.
- [29] H. O. Kreiss and H. U. Busenhart. *Time-Dependent Partial Differential Equations and Their Numerical Solution*. Lectures in Mathematics ETH Zürich. Birkhäuser, 2001.
- [30] R. Beig and B. G. Schmidt. Relativistic elasticity. *Classical and Quantum Gravity*, 20:889–904, March 2003.
- [31] A. M. Anile. *Relativistic fluids and magneto-fluids: With applications in astrophysics and plasma physics*. 1989.
- [32] P. L. Roe. Some contributions to the modelling of discontinuous flows. In B. E. Engquist, S. Osher, & R. C. J. Somerville, editor, *Large-Scale Computations in Fluid Mechanics*, pages 163–193, 1985.
- [33] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, {III}. *Journal of Computational Physics*, 71(2):231 – 303, 1987.
- [34] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. Technical Report ICASE Report No. 97-65, NASA/CR-97-206253, ICASE-Langley, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton VA 23681-2199, USA, November 1997.
- [35] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.
- [36] L. D. Landau and E. M. Lifshitz. *Theory of Elasticity*. Butterworth-Heinemann, Oxford, 3rd edition, 1986.

- [37] J. Marsden and T. J. R. Hughes. *Mathematical foundations of elasticity*. Dover Publications, Inc., New York, 1983.
- [38] J. Weber. Detection and generation of gravitational waves. *Phys. Rev.*, 117:306–313, Jan 1960.
- [39] F. J. Dyson. Seismic Response of the Earth to a Gravitational Wave in the 1-Hz Band. *Astrophysical Journal*, 156:529, May 1969.
- [40] J. L. Synge. A theory of elasticity in general relativity. *Mathematische Zeitschrift*, 72(1):82–87, 1959.
- [41] J.-F. Bennoun. Étude des milieux continus élastiques et thermodynamiques en relativité générale. *Annales de l'institut Henri Poincaré (A) Physique théorique*, 3(1):41–110, 1965.
- [42] W. C. Hernandez. Elasticity Theory in General Relativity. *Physical Review D*, 1:1013–1018, February 1970.
- [43] C. B. Rayner. Elasticity in General Relativity. *Royal Society of London Proceedings Series A*, 272:44–53, February 1963.
- [44] J. G. Oldroyd. Equations of State of Continuous Matter in General Relativity. *Royal Society of London Proceedings Series A*, 316:1–28, March 1970.
- [45] B. Carter and H. Quintana. Foundations of General Relativistic High-Pressure Elasticity Theory. *Royal Society of London Proceedings Series A*, 331:57–83, November 1972.
- [46] B. Carter. Rheometric structure theory, convective differentiation and continuum electrodynamics. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 372(1749):pp. 169–200, 1980.
- [47] J.-M. Souriau. Géométrie et relativité. Enseign. des sciences. Paris: Hermann & Cie. 511 pp. (1964)., 1964.
- [48] G. A. Maugin. Magnetized deformable media in general relativity. *Annales de l'institut Henri Poincaré (A) Physique théorique*, 15(4):275–302, 1971.
- [49] G. A. Maugin. On the covariant equations of the relativistic electrodynamics of continua. iii. elastic solids. *Journal of Mathematical Physics*, 19(5), 1978.
- [50] D. Christodoulou. On the geometry and dynamics of crystalline continua. *Annales de l'institut Henri Poincaré (A) Physique théorique*, 69(3):335–358, 1998.

- [51] J. Kijowski and G. Magli. Relativistic elastomechanics as a lagrangian field theory. *Journal of Geometry and Physics*, 9(3):207 – 223, 1992.
- [52] J. Kijowski and G. Magli. Unconstrained hamiltonian formulation of general relativity with thermo-elastic sources. *Classical and Quantum Gravity*, 15(12):3891, 1998.
- [53] J. Kijowski and G. Magli. Relativistic Elastomechanics is a Gauge-Type Theory. *ArXiv High Energy Physics - Theory e-prints*, November 1994.
- [54] M. Karlovini and L. Samuelsson. Elastic stars in general relativity: I. Foundations and equilibrium models. *Classical and Quantum Gravity*, 20:3613–3648, August 2003.
- [55] G. H. Miller and P. Colella. A High-Order Eulerian Godunov Method for Elastic-Plastic Flow in Solids. *Journal of Computational Physics*, 167:131–176, February 2001.
- [56] K. G. Powell. An approximate riemann solver for magnetohydrodynamics (that works in more than one dimension). Technical Report ICASE Report No. 94-25, ICASE-Langley, ICASE Mail Stop 132C, NASA Langley Research Center, Hampton VA 23681-2199, USA, 1994.
- [57] G. Tóth and D. Odstrčil. Comparison of some flux corrected transport and total variation diminishing numerical schemes for hydrodynamic and magnetohydrodynamic problems. *Journal of Computational Physics*, 128(1):82 – 100, 1996.
- [58] N. Andersson and G. L. Comer. Relativistic fluid dynamics: Physics for many different scales. *Living Rev.Rel.*, 10:1, 2005.
- [59] F. Banyuls, J. A. Font, J. M. Ibáñez, J. M. Martí, and J. A. Miralles. Numerical $\{3 + 1\}$ General Relativistic Hydrodynamics: A Local Characteristic Approach. *Astrophysical Journal*, 476:221–231, February 1997.
- [60] S. T. Millmore and I. Hawke. Numerical simulations of interfaces in relativistic hydrodynamics. *Classical and Quantum Gravity*, 27(1):015007–+, January 2010.
- [61] P. T. Barton, D. Drikakis, E. Romenski, and V. A. Titarev. Exact and approximate solutions of Riemann problems in non-linear elasticity. *Journal of Computational Physics*, 228(18):7046 – 7068, 2009.

- [62] V A Titarev, E Romenski, and E F Toro. Musta-type upwind fluxes for non-linear elasticity. *International Journal for Numerical Methods in Engineering*, 73(7):897–926, 2008.
- [63] G. H. Miller and P. Colella. A Conservative Three-Dimensional Eulerian Method for Coupled Solid-Fluid Shock Capturing. *Journal of Computational Physics*, 183:26–82, November 2002.
- [64] P. Barton. Exact solution of bdrt tests. personal communication, 2011.
- [65] W. F. Noh. Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. *Journal of Computational Physics*, 72:78–120, September 1987.
- [66] J. M. Martí and E. Müller. Numerical Hydrodynamics in Special Relativity. *Living Reviews in Relativity*, 6:7–+, December 2003.
- [67] M. Dumbser, D. S. Balsara, E. F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209–8253, September 2008.
- [68] N. Andersson, G. L. Comer, and K. Glampedakis. How viscous is a superfluid neutron star core? *Nuclear Physics A*, 763:212–229, December 2005.
- [69] S. Karni. Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics*, 112(1):31 – 43, 1994.
- [70] R. Abgrall. Generalisation of the roe scheme for the computation of mixture of perfect gases. *Rech. Aérosp*, 6:31–43, 1988.
- [71] B. Larroutuou. How to preserve the mass fractions positivity when computing compressible multi-component flows. *Journal of Computational Physics*, 95(1):59 – 84, 1991.
- [72] B. Larroutuou and L. Fezoui. On the equations of multi-component perfect of real gas inviscid flow. In Claude Carasso, Pierre Charrier, Bernard Hanouzet, and Jean-Luc Joly, editors, *Nonlinear Hyperbolic Problems*, volume 1402 of *Lecture Notes in Mathematics*, pages 69–98. Springer Berlin Heidelberg, 1989.
- [73] R. Abgrall and S. Karni. Computations of compressible multifluids. *J. Comput. Phys.*, 169(2):594–623, May 2001.

- [74] S. T. Millmore. *Interfaces in Numerical Relativistic Hydrodynamics*. PhD thesis, University of Southampton, Southampton, UK, April 2010.
- [75] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152:457–492, July 1999.
- [76] F. Losasso, T. Shinar, A. Selle, and R. Fedkiw. Multiple interacting liquids. *ACM Trans. Graph.*, 25(3):812–819, July 2006.
- [77] M. G. Crandall and P. L. Lions. Two approximations of solutions of hamilton-jacobi equations. *Mathematics of Computation*, 43(167):pp. 1–19, 1984.
- [78] D. Adalsteinsson and J. A. Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2 – 22, 1999.
- [79] D. Radice, L. Rezzolla, and F. Galeazzi. High-order fully general-relativistic hydrodynamics: new approaches and tests. *Classical and Quantum Gravity*, 31(7):075012, April 2014.
- [80] T. G. Liu, B. C. Khoo, and K. S. Yeo. Ghost fluid method for strong shock impacting on material interface. *Journal of Computational Physics*, 190(2):651 – 681, 2003.
- [81] S. K. Sambasivan and H. S. UdayKumar. Ghost Fluid Method for Strong Shock Interactions Part 2: Immersed Solid Boundaries. *AIAA Journal*, 47:2923–2937, December 2009.
- [82] P. T. Barton and D. Drikakis. An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces. *Journal of Computational Physics*, 229:5518–5540, August 2010.
- [83] P. T. Barton, B. Obadia, and D. Drikakis. A conservative level-set based method for compressible solid/fluid problems on fixed grids. *Journal of Computational Physics*, 230:7867–7890, September 2011.
- [84] W. Fickett and L. M. Scherr. Numerical calculation of the cylinder test. Technical Report NTIS Issue Number 7608, Contract Number W-7405-ENG-36, Accession Number LA-5906, US Energy Research and Development Administration, Los Alamos Scientific Laboratory, P.O. Box 1663, Los Alamos, NM 87545, USA, July 1975.

- [85] R. P. Fedkiw, A. Marquina, and B. Merriman. An isobaric fix for the overheating problem in multimaterial compressible flows. *J. Computational Physics*, 148:545–578, 1999.
- [86] A. J. Penner, N. Andersson, D. I. Jones, L. Samuelsson, and I. Hawke. Crustal failure during binary inspiral. 2011.