

*Communications in
Applied
Mathematics and
Computational
Science*

vol. 8 no. 1 2013

Communications in Applied Mathematics and Computational Science

msp.org/camcos

EDITORS

MANAGING EDITOR

John B. Bell
Lawrence Berkeley National Laboratory, USA
jbbell@lbl.gov

BOARD OF EDITORS

Marsha Berger	New York University berger@cs.nyu.edu	Ahmed Ghoniem	Massachusetts Inst. of Technology, USA ghoniem@mit.edu
Alexandre Chorin	University of California, Berkeley, USA chorin@math.berkeley.edu	Raz Kupferman	The Hebrew University, Israel raz@math.huji.ac.il
Phil Colella	Lawrence Berkeley Nat. Lab., USA pcollella@lbl.gov	Randall J. LeVeque	University of Washington, USA rjl@amath.washington.edu
Peter Constantin	University of Chicago, USA const@cs.uchicago.edu	Mitchell Luskin	University of Minnesota, USA luskin@umn.edu
Maksymilian Dryja	Warsaw University, Poland maksymilian.dryja@acn.waw.pl	Yvon Maday	Université Pierre et Marie Curie, France maday@ann.jussieu.fr
M. Gregory Forest	University of North Carolina, USA forest@amath.unc.edu	James Sethian	University of California, Berkeley, USA sethian@math.berkeley.edu
Leslie Greengard	New York University, USA greengard@cims.nyu.edu	Juan Luis Vázquez	Universidad Autónoma de Madrid, Spain juanluis.vazquez@uam.es
Rupert Klein	Freie Universität Berlin, Germany rupert.klein@pik-potsdam.de	Alfio Quarteroni	Ecole Polytech. Féd. Lausanne, Switzerland alfio.quarteroni@epfl.ch
Nigel Goldenfeld	University of Illinois, USA nigel@uiuc.edu	Eitan Tadmor	University of Maryland, USA etadmor@cscamm.umd.edu
		Denis Talay	INRIA, France denis.talay@inria.fr

PRODUCTION

production@msp.org

Silvio Levy, Scientific Editor

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2013 is US \$75/year for the electronic version, and \$105/year (+\$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from Mathematical Sciences Publishers.

PUBLISHED BY

 **mathematical sciences publishers**
nonprofit scientific publishing

<http://msp.org/>

© 2013 Mathematical Sciences Publishers

ON THE ORIGIN OF DIVERGENCE ERRORS IN MHD SIMULATIONS AND CONSEQUENCES FOR NUMERICAL SCHEMES

FRIEDEMANN KEMM

This paper investigates the origin of divergence errors in MHD simulations. For that purpose, we introduce the concept of discrete involutions for discretized conservation laws. This is done in analogue to the concept of involutions for hyperbolic conservation laws, introduced by Dafermos. By exploring the connection between discrete involutions and resonance, especially for constrained transport like MHD, we identify the lack of positive central viscosity and the assumption of one-dimensional physics in the calculation of intercell fluxes as the main sources of divergence errors. As an example of the consequences for numerical schemes, we give a hint how to modify Roe-type schemes in order to decrease the divergence errors considerably and, thus, stabilize the scheme.

1. Introduction

Hyperbolic conservation laws are usually equipped with additional conditions. Most important is the existence of a convex entropy, which singles out the physical relevant solution out of the large set of possible weak solutions. Sometimes, especially when there is no convex entropy, or the system degenerates into a weakly or resonant hyperbolic system, other laws have to be included to find physical solutions. In the first case (convex entropy), the additional law is for an additional variable, namely the entropy, which depends on the state variables, but is no state variable itself. In the latter case, we have additional partial differential equations for the state variables themselves. In the first case the additional law is a partial differential equation or inequality of evolution type, usually a conservation law itself, in the second it is a first-order nonevolutionary constraint. These additional constraints are, as Dafermos points out [10; 9], involutions for the underlying system of conservation laws. So the resulting system, which includes both, the evolution system and the condition, has more equations than unknowns. If the involution is satisfied by the

MSC2010: primary 76W05, 39A12, 35L45, 35L65, 35L80; secondary 35N10, 65M06, 39A70, 65Z05.

Keywords: involutions, constraint, magnetohydrodynamics, plasma physics, Maxwell equations, divergence, curl, operator scheme, finite differences, finite volume method, resonance, hyperbolic PDE, compressible flow.

initial state for the evolution equations, it is satisfied by the solution of the evolution system for all times. Thus, in the continuous setting, the constraint is merely a condition on the initial state.

These constraints play an important role in many branches of physics, the most famous of which is the area of electromagnetic modeling and plasma physics. Here we face constraints for the electric field as well as for the magnetic field. In numerical simulations this may cause severe problems, because in general it is impossible to reproduce these conditions exactly. This results in unphysical forces and thus completely useless solutions [6; 36], especially in magnetohydrodynamics (MHD). In MHD many codes fail completely. But this is not the case with all numerical schemes.

First, there are approaches that are designed to model the constraint numerically. Many of them are done on staggered grids [3; 4; 14]. Some newer approaches also work on collocated grids [41; 43; 40; 31; 32; 33; 16; 15; 45] or in the context of discontinuous Galerkin schemes [5]. Usually, this class of schemes is referred to as *constrained transport* schemes.

A second family of schemes are based on a modification of the system of partial differential equations which makes the constraint part of the evolution system itself. In the context of plasma physics, a popular approach is to transport the involution term, in this case the divergence of the magnetic field, with the flow velocity. This was first suggested for numerical simulations by Brackbill and Barnes [6] and put forward by employing Godunov's full symmetrizable form¹ of the MHD equations [20] by Powell et al. [38; 39]. In [16; 15; 45], this approach is even combined with constrained transport. Another possibility is to apply a kind of a generalized Lagrange multiplier approach [36], a method which can show up in several variants: resulting in a Hodge-projection scheme, resulting in a parabolic treatment of the involution term as was suggested by Marder [30], resulting in a hyperbolic system — the involution term is radiated with an artificial speed out of the computational domain [35; 34] — or it results in a treatment of the involution in the manner of a telegraph equation [12; 27]. (Crockett et al. [8] even combine the Marder approach with a Hodge-projection method.) In the context of electromagnetic models and plasma physics these approaches are usually referred to as *divergence cleaning*.

A third class is that of schemes that are stable without any modification or special discretization technique. This is the case in many physical contexts. For magnetohydrodynamics it is only reported very scarcely. But still there are some examples: The scheme of Zachary, Malagoli and Colella [46], an upwind method, published already in 1994, has this property. Another example is the scheme

¹It is interesting to note that this form was first discovered by Godunov [20] as symmetrizable form of MHD and then rediscovered by Powell et al. [38] as Galilean invariant form of MHD. Since it has an entropy [20; 10], one would not need any involution for the system.

presented by Balbás and Tadmor[2], which on the contrary is a central scheme. They still need some intermediate cleaning steps to obtain physical relevant solutions, but only at a few time steps in a long interval [46; 1]. Both schemes have in common that they discretize the full equations, while most schemes, for the computation of intercell fluxes, employ one-dimensional physics in the direction of the normal of the cell face. This shows that there is something special in discretizing the multidimensional equations directly.

Therefore, in this study, we push forward the investigations started in [25] and take a closer look at involutions and their relation to constrained transport and resonance. We also take a closer look at the discretization of conservation laws in terms of finite differences for the partial derivatives in the equations. We define discrete analogues of the most important types of involutions and look for sufficient conditions for the existence of discrete involutions to a given discretized conservation law. We single out a class of linear schemes for which the discrete involutions are exact. We consider the interplay between discrete involutions and resonance, and we study the role of central numerical viscosity and the assumption of one-dimensional physics in the computation of intercell fluxes. As a result, applying full physics in the computation of intercell fluxes and a suitable amount of central viscosity on the resonant wave lead to a stable scheme also in the MHD context. As an example, we show how to apply the Harten entropy fix in a smart way to adjust the viscosity on the resonant wave. There are still some divergence errors, but the work needed in divergence cleaning can be considerably reduced.

The plan of the paper is as follows. We start with an overview of the concept of involutions and its connections to resonance and constrained transport. [Section 3](#) presents a theory for discrete involutions. Also some standard schemes are investigated whether they yield exact or only approximate involutions. The key is a shift in the interpretation of numerical schemes. Some terms, traditionally considered to be part of the spatial discretization, are identified to be essentially part of the time discretization. This helps us in [Section 4](#) to investigate the interplay between discrete involutions, resonance, central numerical viscosity, and the assumption of one-dimensional physics. Also we show numerical evidence of the theoretical results ([Section 4.4](#)). In this course, we present, as an example, a modification of the Roe-scheme which minimizes the production of divergence errors.

2. Hyperbolic conservation laws with involutions

2.1. Definition and a sufficient condition. Our starting point is the general conservation law

$$\mathbf{q}_t + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0}, \quad (1)$$

where \mathbf{q} denotes the vector of conserved quantities and $\mathbf{F} = (F_1, F_2, \dots)$ denotes

the flux. The F_i are the directional fluxes in the (space) directions given by the standard unit vectors e_i . The corresponding flux Jacobians will be denoted by A_i .

The system (1) is called *hyperbolic* if for all directions

$$\mathbf{n} = \sum_i n_i \mathbf{e}_i,$$

where $\|\mathbf{n}\| = 1$, the corresponding flux Jacobian

$$A_n = \sum_i n_i A_i$$

is diagonalizable with real eigenvalues. It is called *strictly hyperbolic* if, in addition, all eigenvalues are distinct. If it is not diagonalizable but still all eigenvalues are real it is called *weakly hyperbolic* or *resonant hyperbolic*. In this survey we restrict the analysis to systems which are, at least, weakly hyperbolic.

If the system (1) can be rewritten as

$$(\tilde{G}(\tilde{\mathbf{q}}))_t + \nabla \cdot \tilde{\mathbf{F}}(\tilde{\mathbf{q}}) = \mathbf{0}, \quad (2)$$

with symmetric Jacobians of G and the F_i , then the conservation law is called symmetrizable and the quasilinear form of (2) is called its symmetric form. As a consequence, such a system is always fully and never resonant or weakly hyperbolic.

Dafermos [10, p. 9] notes that a system of conservation laws is endowed with nontrivial balance laws, such as an entropy law, if and only if it is symmetrizable. The MHD equations, among others, are not symmetrizable. For some states \mathbf{q} , the system is resonant hyperbolic. Godunov [18; 19; 20] discovered an extended system that is symmetrizable and has an entropy law, but at the price of giving up conservation. As Tóth [44] points out, in numerical schemes, this may lead to wrong jump conditions. As another way to deal with the lack of an entropy law, Dafermos [10, pp. 69 ff.] offers the concept of involutions.

The system (1) has an *involution* if there exist constant matrices M_i such that the condition

$$\sum_i M_i \mathbf{q}_{x_i} = \mathbf{0}, \quad (3)$$

also called the involution of system (1), holds true for all times if it is satisfied by the initial data.

In his work on hyperbolic systems with involutions, Dafermos [9; 10] concentrates on a subclass that includes most of the physically relevant cases:

Theorem 1. *Let the system (1) and matrices M_i be given. If the directional fluxes fulfill the antisymmetric condition*

$$M_i F_j + M_j F_i = \mathbf{0}, \quad i, j = 1, 2, \dots, \quad (4)$$

then

$$\sum_i \mathbf{M}_i \mathbf{q}_{x_i}$$

is an involution of system (1) and satisfies the additional condition

$$\frac{\partial}{\partial t} \left(\sum_i \mathbf{M}_i \mathbf{q}_{x_i} \right) = \mathbf{0}. \quad (5)$$

As a consequence, $\sum_i \mathbf{M}_i \mathbf{q}_{x_i}$ not only constitutes an involution of (1), but in addition is constant in time, independently of the initial state. This shows that (4) is a rather strong condition. But, as Section 2.2 shows, many important systems satisfy condition (4). In the following sections this is used to find discrete analogues for the concept of involutions. For the understanding of the following sections, it is necessary to understand the proof of Theorem 1. In summary, the proof consists in four steps:

1. Apply $\sum_i \mathbf{M}_i \partial/\partial x_i$ to the conservation law (1).
2. Constant matrices commute with partial derivatives.
3. Partial derivatives commute with each other.
4. Due to condition (4) all terms including fluxes vanish.

In more detail, we find after application of step 1.

$$\sum_i \mathbf{M}_i \frac{\partial}{\partial x_i} \frac{\partial}{\partial t} \mathbf{q} + \sum_i \mathbf{M}_i \frac{\partial}{\partial x_i} \sum_j \frac{\partial}{\partial x_j} F_j(\mathbf{q}) = \mathbf{0}.$$

Now we make use of the facts 2. and 3. to obtain

$$\frac{\partial}{\partial t} \left(\sum_i \mathbf{M}_i \frac{\partial}{\partial x_i} \mathbf{q} \right) + \sum_{i,j} \frac{\partial^2}{\partial x_i \partial x_j} \mathbf{M}_i F_j(\mathbf{q}) = \mathbf{0}.$$

Since the operator $\partial^2/\partial x_i \partial x_j$ is symmetric and, according to (4), $\mathbf{M}_i F_j(\mathbf{q})$ is antisymmetric, the last sum vanishes, which completes the proof. \square

2.2. Examples of hyperbolic systems with involutions. Several examples of systems with involutions satisfying (4) can be found in [9; 10] and in the studies by Torrilhon and Fey [43; 41; 42]. Here we present only few of them:

As an introductory example, Dafermos [10] presents the equations for isentropic processes of thermoelastic nonconductors of heat:

$$\begin{aligned} \mathbf{F}_t - \nabla \mathbf{v} &= \mathbf{0}, \\ \mathbf{v}_t - \nabla \cdot \mathbf{T}(\mathbf{F}) &= \mathbf{0}, \end{aligned} \quad (6)$$

with the deformation gradient \mathbf{F} , velocity \mathbf{v} , and the stress tensor \mathbf{T} . Since the time evolution of the deformation gradient \mathbf{F} is a gradient, it is curl-free. Therefore $\nabla \times \mathbf{F}$ is an involution for system (6). The matrices \mathbf{M}_i ($i = 1, \dots, 3$) are in $\mathbb{R}^{3 \times 6}$, and, while the right half is zero, the left half reads as

$$\mathbf{M}_1^{\text{left}} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{M}_2^{\text{left}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{M}_3^{\text{left}} = \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \quad (7)$$

With these matrices, condition (4) is satisfied.

An important hyperbolic system with involution is given by the vacuum Maxwell equations

$$\mathbf{E}_t - c^2(\nabla \times \mathbf{B}) = -\frac{\mathbf{j}}{\varepsilon_0}, \quad (8)$$

$$\mathbf{B}_t + (\nabla \times \mathbf{E}) = \mathbf{0}, \quad (9)$$

$$\nabla \cdot \mathbf{E} = \frac{q}{\varepsilon_0}, \quad (10)$$

$$\nabla \cdot \mathbf{B} = 0, \quad (11)$$

with the electric field \mathbf{E} , magnetic induction \mathbf{B} , electric current \mathbf{j} , charge density q , speed of light c , and the constants ε_0 and μ_0 . In the absence of electric charge and current this is a homogeneous hyperbolic conservation law, where the divergence of both fields, \mathbf{E} and \mathbf{B} is preserved. The matrices involved in condition (3) and (4) are

$$\mathbf{M}_i = \begin{pmatrix} \mathbf{e}_i^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{e}_i^T \end{pmatrix}. \quad (12)$$

Since the models of plasma-physics are obtained by using the Maxwell equations, they also inherit the involutions. In the MHD equations no evolution equation for the electric field is included. Thus, only the divergence of the magnetic field is inherited as an involution. The full equations of ideal compressible magnetohydrodynamics are

$$\rho_t + \nabla \cdot [\rho \mathbf{v}] = 0, \quad (13)$$

$$(\rho \mathbf{v})_t + \nabla \cdot [\rho \mathbf{v} \circ \mathbf{v} + (p + \frac{1}{2} \mathbf{B}^2) \mathbf{I} - \mathbf{B} \circ \mathbf{B}] = \mathbf{0}, \quad (14)$$

$$\mathbf{B}_t + \nabla \cdot [\mathbf{B} \circ \mathbf{v} - \mathbf{v} \circ \mathbf{B}] = \mathbf{0}, \quad (15)$$

$$e_t + \nabla \cdot [(e + p + \frac{1}{2} \mathbf{B}^2) \mathbf{v} - \mathbf{B}(\mathbf{v} \cdot \mathbf{B})] = 0, \quad (16)$$

$$\nabla \cdot \mathbf{B} = 0. \quad (17)$$

The last equation, (17), is the involution for the evolution system (13)–(16). The asymmetric condition (4) is satisfied with $\mathbf{M}_i = (0, 0, 0, 0, \mathbf{e}_i^T, 0)$. Thus, the MHD equations nicely fit into the framework given by Dafermos [9; 10].

In [17], Gilman argues that the classical “shallow water” equations of geophysical fluid dynamics should be useful for studying the global dynamics of the solar tachocline and demonstrates the existence of an MHD analog that would allow taking into account the strong toroidal magnetic field likely to be present there. So he presents a derivation analogous to that for the classical shallow water equations and comes up with the following system of shallow water magnetohydrodynamics (SMHD)

$$\begin{aligned} h_t + \nabla \cdot [h\mathbf{v}] &= 0, \\ (h\mathbf{v})_t + \nabla \cdot [h\mathbf{v} \circ \mathbf{v} - h\mathbf{B} \circ \mathbf{B} + \frac{1}{2}gh^2\mathbf{I}] &= \mathbf{0}, \\ (h\mathbf{B})_t - \nabla \times [\mathbf{v} \times (h\mathbf{B})] &= \mathbf{0}, \end{aligned} \quad (18)$$

with the involution

$$\nabla \cdot (h\mathbf{B}) = 0. \quad (19)$$

This inherits most of its behavior from the original MHD system (13)–(16). The main difference is that, due to the averaging over the third space dimension, the magnetic field \mathbf{B} is now replaced by $h\mathbf{B}$, where h denotes the height of the fluid layer under consideration and g is the gravitational constant.

Since the structure of the critical part of the evolution for MHD, and SMHD is similar, we also consider the linear model problem of Fey and Torrilhon [41], which resembles the common behavior of those systems in the simplest possible setting. For a given velocity field \mathbf{v} , constant in space and time, we consider

$$\mathbf{B}_t - \nabla \times (\mathbf{v} \times \mathbf{B}) = \mathbf{0}, \quad (20)$$

or in divergence form,

$$\mathbf{B}_t + \nabla \cdot (\mathbf{B} \circ \mathbf{v} - \mathbf{v} \circ \mathbf{B}) = \mathbf{0}. \quad (21)$$

Obviously the asymmetric condition (4) is satisfied with $\mathbf{M}_i = \mathbf{e}_i^T$ and the divergence of \mathbf{B} makes up an involution for the system. This is a model for divergence-preserving transport.

For the sake of completeness, we also present the model for curl-preserving transport given by Fey and Torrilhon [41]

$$\mathbf{P}_t + \nabla \cdot (\mathbf{v} \cdot \mathbf{P}) = \mathbf{0}, \quad (22)$$

or in divergence form,

$$\mathbf{P}_t + \nabla \cdot ([\mathbf{v} \cdot \mathbf{P}]\mathbf{I}) = \mathbf{0}. \quad (23)$$

Here again, it can be seen from the flux form that (4) is satisfied. The matrices \mathbf{M}_i are the same as those presented in (7), and from (22) it is seen that the resulting involution is $\nabla \times \mathbf{P}$.

The last four systems, MHD, SMHD, and the model systems for constraint preserving transport, have one point in common: dependent on the velocity field, they might lose full hyperbolicity. In general they are only weakly, or resonant, hyperbolic.

2.3. Resonant hyperbolic problems and involutions. In this section we consider the relation between involutions and resonant hyperbolic systems. It is mentioned already by Crockett et al. [8] that there is a relation between the divergence condition for MHD and resonance. Here, we want to study this relation in more detail.

2.3.1. Resonance. In physics, systems which allow for solutions growing unboundedly in time, usually are called resonant. The most famous example is the harmonic oscillator with a periodic exciting force. If the frequency of the excitation meets the eigenfrequency of the system, the amplitude grows unboundedly. A similar behavior can be found for weakly hyperbolic systems. The model equations for divergence-preserving transport (20) provide a nice example of resonance. Following Crockett et al. [8], let in the two-dimensional case $\mathbf{v} = (u, v)^T = (0, v)^T$. Then the system is only weakly hyperbolic and reads as

$$B_{1t} + vB_{1y} = 0, \quad (24)$$

$$B_{2t} - vB_{1x} = 0. \quad (25)$$

This means that B_1 is transported in y -direction and acts as a source for B_2 . If B_1 varies in x -direction, there is a nonvanishing constant source and, thus, B_2 grows unboundedly with a constant rate. We will go back to this example later.

Of the systems with evolutions provided in the previous section, some are fully hyperbolic, some are only resonant hyperbolic. Dafermos [10] points out that system (6) is hyperbolic if the inner energy, which defines the stress tensor, is rank-one convex. Thus, hyperbolicity depends on the state.

Although in any space direction all wave speeds are $\pm c$, the Maxwell equations are fully hyperbolic. They allow for no resonant effects except from those introduced by outer source terms.

In contrast, the MHD and SMHD equations allow for resonant states. By using the magnetohydrodynamic approximation for the electric field, $\mathbf{E} \approx -\mathbf{v} \times \mathbf{B}$, the induction equation attains the structure of divergence preserving transport. If we set $\mathbf{B} = (B_1 = 0, B_2, B_3)^T$, $\mathbf{v} = (u = 0, v, w)^T$, i.e., velocity and magnetic field are perpendicular to the first space direction, then the flux Jacobian in that direction has zero as a sixfold eigenvalue with five-dimensional eigenspace. The system is only resonant hyperbolic. Whenever the velocity and the magnetic field are in one plane, the flux Jacobian in the direction perpendicular to that plane is deficient, the system is only resonant hyperbolic. A similar situation occurs when the velocity component parallel to the magnetic field equals $\pm a$, where a is the speed of sound, and the

magnetic field is $a\sqrt{\rho}$. Then zero is a fourfold eigenvalue with three-dimensional eigenspace. Again the system is only resonant hyperbolic.

Due to the reduction of the physical problem to two space dimensions, for the shallow water MHD the same resonance as for MHD occurs when velocity and magnetic field are parallel. Another resonant case can be found if we have $\mathbf{v} = (u = \pm c_g, v)^T$, where $c_g = \sqrt{B_1^2 + gh}$ is the magnetogravitational speed. In this case, zero is a double eigenvalue with one-dimensional eigenspace.

The model system for divergence preserving transport (20) shares the resonant behavior, as can be seen at the beginning of this section. The flux Jacobian in the direction perpendicular to the velocity is deficient, the system is only resonant hyperbolic. For the model system for curl-preserving transport, the situation is similar.

We will go into more detail about this in the following sections.

2.4. Relation of involutions to zero eigenvalues and resonance. In this section we investigate the connection between involutions, zero eigenvalues and resonance in more detail. First we want to recall some considerations of Dafermos [10]. The antisymmetric condition (4) is equivalent to

$$\mathbf{M}_i \mathbf{A}_j + \mathbf{M}_j \mathbf{A}_i = \mathbf{0} \quad \text{for all } i, j. \quad (26)$$

If we take the unit vector $\mathbf{n} = (n_1, n_2, n_3)^T$ we find for the flux Jacobian \mathbf{A}_n in direction of \mathbf{n}

$$\mathbf{M}_n \mathbf{A}_n = \left(\sum_i n_i \mathbf{M}_i \right) \left(\sum_j n_j \mathbf{A}_j \right) = \sum_{i,j} n_i n_j \mathbf{M}_i \mathbf{A}_j = \mathbf{0}. \quad (27)$$

As a consequence, the range of \mathbf{A}_n is a subset of the kernel of \mathbf{M}_n , and therefore the dimension of the kernel of \mathbf{A}_n is greater than or equal to the rank of \mathbf{M}_n . In particular, we know that it is at least one. We always have a zero eigenvalue for systems which satisfy the antisymmetric condition (4), and equality would mean that the rows of \mathbf{M}_n are just the left eigenvectors of \mathbf{A}_n for the zero eigenvalue. As a consequence, in the case of equality, the zero eigenvalue has a full set of eigenvectors and, thus, can not destroy the hyperbolicity of the system. An example of this are the vacuum Maxwell equations with zero as a double eigenvalue and we have

$$\mathbf{M}_i = \begin{pmatrix} \mathbf{e}_i^T & \mathbf{0}^T \\ \mathbf{0}^T & \mathbf{e}_i^T \end{pmatrix}, \quad (28)$$

which makes up a linearly independent set of two left eigenvectors. In the case that the range of \mathbf{A}_n is a proper subset of the kernel of \mathbf{M}_n things might be worse, as can be seen with the above example systems.

If we apply these considerations to the divergence-free transport (20), we find that the rank of \mathbf{M}_n is one for all directions \mathbf{n} . But if $\mathbf{n} \perp \mathbf{v}$, there is no transport in direction of \mathbf{n} , and hence, the multiplicity of the zero eigenvalue of \mathbf{A}_n is three. If \mathbf{v} , and thus also \mathbf{A}_n , does not vanish completely the system matrix can not be diagonalized, because the eigenspace has dimension two. A similar consideration holds for the curl-free transport (22). In this case, except for $\mathbf{v} = \mathbf{0}$, the rows of \mathbf{M}_n always form a basis of the space of left eigenvectors of \mathbf{A}_n , proving again that both prototypes for constrained transport are merely resonant hyperbolic.

2.4.1. A quantitative view on resonance for divergence- and curl-preserving transport. For a quantitative view on resonance for divergence-preserving transport, we revisit the example (20), (21) from the end of Section 2.2 and look at it in more detail: let in the two-dimensional case $\mathbf{v} = (u, v)^T = (0, v)^T$. It follows from the considerations at the beginning of Section 2.4 that \mathbf{A}_1 is not diagonalizable, so we can expect resonance phenomena in the first space direction. Since $u = 0$ and

$$\mathbf{A}_n = (\mathbf{n}^T \cdot \mathbf{v})\mathbf{I} - \mathbf{v} \circ \mathbf{n}, \quad (29)$$

we can rewrite the system as

$$B_{1t} + vB_{1y} = 0, \quad (30)$$

$$B_{2t} + vB_{2y} = v(B_{1x} + B_{2y}) = v(\nabla \cdot \mathbf{B}). \quad (31)$$

The source is in the evolution equation of the second component of \mathbf{B} and is proportional to the involution $\nabla \cdot \mathbf{B}$. If \mathbf{B} is divergence-free there is no resonance at all. In general the two-dimensional system can be rewritten as

$$\mathbf{B}_t + \sum_i v_i \mathbf{B}_{x_i} = - \left(\sum_j A_j \mathbf{M}_j^T \right) \sum_i \mathbf{M}_i \mathbf{B}_{x_i}. \quad (32)$$

Investigating the right side of this equation, we find just the negative of the Powell correction term [38; 20]. Therefore, if we had added the Powell correction term to the right side of system (20), we would have obtained a nonresonant, fully hyperbolic system, in this simple linear case pure advection. In the full MHD equations, the Powell system, although not pure advection, due to its symmetrizability, is also fully hyperbolic without any resonance.

In the three-dimensional case there is just one additional factor to include. The system can be rewritten as

$$\mathbf{B}_t + \sum_i v_i \mathbf{B}_{x_i} = -\frac{1}{2} \left(\sum_j A_j \mathbf{M}_j^T \right) \sum_i \mathbf{M}_i \mathbf{B}_{x_i}. \quad (33)$$

Adding $\frac{1}{2} \left(\sum_j A_j \mathbf{M}_j^T \right) \sum_i \mathbf{M}_i \mathbf{B}_{x_i}$ to the right side of system (20) would lead to pure advection and, thus, to a fully hyperbolic system.

In full MHD in three space dimensions with the usual ordering of the equations, the addition of $\frac{1}{2}(\sum_j A_j M_j^T) \sum_i M_i B_{x_i}$ with $M_i = e_{i+4}$, makes the resulting system fully hyperbolic. Nevertheless, for the use in numerical schemes, the original Powell correction is more convenient due to its simpler form of left and right eigenvectors. In addition, it is Galilean invariant [38] and there is an entropy condition [20]. An issue which would affect both approaches is mentioned by Tóth [44]: In the case of nonvanishing divergence of the magnetic field, the jump conditions in the Riemann problem are wrong. This is not surprising, since the system deviates from the real physics by allowing magnetic monopoles².

In the same way, for curl-preserving transport (22), we get

$$P_t + \sum_i v_i P_{x_i} = - \left(\sum_j A_j M_j^T \right) \left(\sum_i M_i P_{x_i} \right), \quad (34)$$

where the last sum is just the involution. If the constraint is satisfied for the initial data, curl-preserving transport reduces to pure transport. Otherwise it is a transport with a source which is a linear function of the involution term. Since the involution term is constant in time, the source term is also constant in time. All in all the situation is quite similar to that in divergence preserving transport. Because of that, and because curl-preserving transport plays a minor part in practical applications, we won't go into further detail for that.

3. A discrete analog to the concept of involutions

This section is dedicated to the construction of discrete analogues of the concept of involutions for discretized conservation laws as well as a discrete analogue of [Theorem 1](#) and its proof.

For this purpose, we first give some remarks on the notion and notation of finite difference schemes for hyperbolic conservation laws. This is necessary since the usual notation doesn't allow to transfer the results of [Section 2.1](#) to the discrete case.

After that, we show how this transfer could be accomplished. We give discrete versions of [Theorem 1](#) for semidiscrete, fully discrete, and a special case of linear schemes. In this context, we have to introduce exact and approximate discrete involutions.

Finally, we investigate some standard schemes. Which discrete version of [Theorem 1](#) will apply to them? Will we find exact or only approximate discrete

²This is in general true for all divergence correction methods. But with the Powell system, the divergence errors destroy conservation and are transported with the flow instead of being radiated away like with hyperbolic or mixed type GLM [12]. In fact, as was reported to me by Powell, applying hyperbolic or mixed type GLM to the symmetrizable system yields the best results.

involutions? Although rarely used in practice, these schemes are role models for most of the usual schemes, showing which behavior we have to expect from these methods.

3.1. On the notion and notation of finite difference schemes for hyperbolic conservation laws. In this paper we employ a rather strict, but still general, notion and notation of finite difference schemes for hyperbolic conservation laws. If \mathbf{I} is the set of all index vectors \mathbf{i} involved in the computation, including both, time- and space-indexes, a difference operator for some time-derivative is given by

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j = \frac{1}{\Delta t} \sum_{\mathbf{i} \in \mathbf{I}} \alpha_{\mathbf{i},j} \mathbf{q}_i, \quad (35)$$

where the coefficients $\alpha_{\mathbf{i},j}$ are allowed to be matrix valued and to depend on anything, they only have to be bounded in time and space, and Δt is some characteristic time step size. We use the hat to distinguish difference operators from the corresponding derivatives. The inclusion of Δt into the formula makes the further considerations more convenient. For space derivatives we write in the same way

$$\frac{\hat{\partial}}{\hat{\partial}x_k} \mathbf{q}_j = \frac{1}{\Delta x} \sum_{\mathbf{i} \in \mathbf{I}} \beta_{\mathbf{i},j}^k \mathbf{q}_i, \quad (36)$$

where the index k denotes the space direction, and Δx is some characteristic space step size, for example the minimal inradius of the grid cells. All other differential operators, like divergence, curl, gradient, mixed or higher derivatives, are discretized by means of the operators given in (35) and (36), where the difference operator for each space direction and for the time are fixed. Thus, for example, the second derivative of some quantity \mathbf{q} with respect to direction x_k has to be discretized by

$$\frac{\hat{\partial}}{\hat{\partial}x_k} \left(\frac{\hat{\partial}}{\hat{\partial}x_k} \mathbf{q} \right).$$

We introduce this strict notation to be able to transfer the proof of [Theorem 1](#) to the discrete case. As a consequence of the notation, in the following, the term $\hat{\partial}/\hat{\partial}t$ is merely an abbreviation for any discrete time difference of order q . This can be done because the difference between any two difference operators of order q is also $\mathcal{O}(\Delta t^q)$. For the other partial derivatives a similar consideration holds. This is a fact which we extensively use in our arguments. With these operators a discretized hyperbolic conservation law reads as

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j + \sum_r \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{F}_r(\mathbf{q}_j) = \mathbf{0}. \quad (37)$$

Note that this is not the way the scheme is constructed. But any finite volume or finite difference scheme can be artificially rewritten in that way. This is also not the usual notation of discretized hyperbolic conservation laws in the literature. Normally, a simpler difference operator is chosen and applied to a system, where the physical flux function \mathbf{F} is replaced by a numeric flux function \mathbf{G} , which depends on the state in several grid cells. But this is not suitable for the investigation of discrete involutions, since we have to rely on the antisymmetric condition (4), which depends on \mathbf{F} and is usually not satisfied if \mathbf{F} is replaced by \mathbf{G} ³. As we will see in Section 3.2.2, sometimes parts of \mathbf{G} have to be considered as a contribution to the discrete time derivative instead of the space derivative.

3.2. Proofs for discrete involutions. To prove the existence of discrete involutions, we first have to define them:

Definition 1. *If for a discretized hyperbolic conservation law of the form (37) we have for the discretized involution*

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) \rightarrow \mathbf{0} \quad (38)$$

as the time and space step sizes go to zero; this is called an approximate discrete involution for (37). If we have equality, i.e., if

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) = \mathbf{0}, \quad (39)$$

we call it an exact discrete involution for (37).

We prove three discrete versions of Theorem 1: for the general fully discrete case, for the semidiscrete case, and finally for a linear special case.

3.2.1. The general fully discrete case. We start with the general fully discrete case. To prove that the antisymmetric condition (4) is sufficient for the existence of discrete involutions, we first have to investigate the commutators of the difference operators given in the previous section.

If we have for some quantity \mathbf{h}

$$\frac{\hat{\partial}}{\hat{\partial}x} \mathbf{h}_j = (\mathbf{h}_x)_j + \mathcal{O}(\Delta x^p) \quad (40)$$

³ By applying the concept of numerical flux functions on a one-dimensional equidistant grid, it would be even possible to write all schemes, including implicit schemes, as $(\mathbf{q}_i^{n+1} - \mathbf{q}_i^n)/\Delta t - (G_{i+1/2}^n - G_{i-1/2}^n)/\Delta x$. All details are hidden in the definition of the numerical flux function G . In the same way, for every computational grid, a difference formulation can be found which only depends on the grid itself.

and

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h}_j = \frac{1}{\Delta t} \sum_{i \in I} \alpha_{i,j} \mathbf{h}_i = (\mathbf{h}_t)_j + \mathcal{O}(\Delta t^q), \quad (41)$$

we can verify the following:

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\frac{\hat{\partial}}{\hat{\partial}x} \mathbf{h} \right)_j = \frac{1}{\Delta t} \sum_{i \in I} \alpha_{i,j} [(\mathbf{h}_x)_i + \mathcal{O}(\Delta x^p)] \quad (42)$$

$$= \frac{\hat{\partial}}{\hat{\partial}t} (\mathbf{h}_x)_j + \mathcal{O}\left(\frac{\Delta x^p}{\Delta t}\right) = (\mathbf{h}_{xt})_j + \mathcal{O}(\Delta t^q) + \mathcal{O}\left(\frac{\Delta x^p}{\Delta t}\right). \quad (43)$$

Through similar considerations for $\frac{\hat{\partial}}{\hat{\partial}x} \left(\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h} \right)_j$, we find for the commutator of both discrete partial derivatives

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\frac{\hat{\partial}}{\hat{\partial}x} \mathbf{h} \right)_j - \frac{\hat{\partial}}{\hat{\partial}x} \left(\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h} \right)_j = \mathcal{O}\left(\frac{\Delta x^p}{\Delta t}\right) + \mathcal{O}\left(\frac{\Delta t^q}{\Delta x}\right). \quad (44)$$

If for a simulation the time step and space step stay of the same order for all time, i.e., $\Delta t = \mathcal{O}_s(\Delta x)$ (the subscript s means that the order relation between Δt and Δx is symmetric), the commutator (44) simplifies to

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\frac{\hat{\partial}}{\hat{\partial}x} \mathbf{h} \right)_j - \frac{\hat{\partial}}{\hat{\partial}x} \left(\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h} \right)_j = \mathcal{O}(\Delta x^{\min\{p,q\}-1}). \quad (45)$$

But this is not always true, especially when resonance comes into play. If we consider the commutator of two different discrete space derivatives, say in the x - and y -directions, and both are of the same order of accuracy, p , we obtain

$$\frac{\hat{\partial}}{\hat{\partial}y} \left(\frac{\hat{\partial}}{\hat{\partial}x} \mathbf{h} \right)_j - \frac{\hat{\partial}}{\hat{\partial}x} \left(\frac{\hat{\partial}}{\hat{\partial}y} \mathbf{h} \right)_j = \mathcal{O}(\Delta x^{p-1}). \quad (46)$$

The commutator of a discrete derivative and a matrix \mathbf{M} can be obtained in the same way. It is

$$\mathbf{M} \frac{\hat{\partial}}{\hat{\partial}t} (\mathbf{h})_j - \frac{\hat{\partial}}{\hat{\partial}t} (\mathbf{M}\mathbf{h})_j = \mathcal{O}(\Delta t^q). \quad (47)$$

Thus, no loss of accuracy is introduced.

With these preparations, the following theorem can be proved:

Theorem 2. *Let the (weakly) hyperbolic conservation law*

$$\mathbf{q}_t + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0} \quad (48)$$

be given, together with constant matrices \mathbf{M}_l satisfying

$$\mathbf{M}_l \mathbf{F}_r + \mathbf{M}_r \mathbf{F}_l = \mathbf{0} \quad \text{for all } l, r = 1, 2, \dots \quad (49)$$

Let $\hat{\partial}/\hat{\partial}t$ be a time discretization of order q and $\hat{\partial}/\hat{\partial}x_r$ be space differences of order p .

If we discretize the conservation system (48) with these discrete operators, then we obtain the following analogue of (5):

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) = \mathcal{O} \left(\frac{\Delta x^p}{\Delta t} \right) + \mathcal{O} \left(\frac{\Delta t^q}{\Delta x} \right) + \mathcal{O}(\Delta x^{p-1}) \quad \text{for all } \mathbf{j}. \quad (50)$$

As a direct consequence, we can state:

Corollary 1. *If, in addition to the conditions of Theorem 2, the time and space step are of the same order, i.e., $\Delta t = \mathcal{O}(\Delta x)$ and $\Delta x = \mathcal{O}(\Delta t)$, then (50) can be simplified to*

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) = \mathcal{O}(\Delta x^{\min\{p,q\}-1}) \quad \text{for all } \mathbf{j}. \quad (51)$$

This applies to linear systems and, in general, to nonlinear nonresonant systems. For general nonlinear resonant systems things might be worse. We will consider the general case in more detail in Section 4.

Proof of Theorem 2. For a fixed index \mathbf{j} , the discretized conservation law reads

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j + \sum_r \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{F}_r(\mathbf{q}_j) = \mathbf{0}. \quad (52)$$

Now we apply $\sum_l \mathbf{M}_l \hat{\partial}/\hat{\partial}x_l$ to that system:

$$\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j + \sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \sum_r \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{F}_r(\mathbf{q}_j) = \mathbf{0}. \quad (53)$$

By applying the identities (46) and (47), we find for the double summation term

$$\begin{aligned} \sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \sum_r \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{F}_r(\mathbf{q}_j) &= \sum_{l,r} \frac{\hat{\partial}}{\hat{\partial}x_l} \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{M}_l \mathbf{F}_r(\mathbf{q}_j) + \mathcal{O}(\Delta x^p) \\ &= \sum_{l,r} \frac{\hat{\partial}}{\hat{\partial}x_r} \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{M}_l \mathbf{F}_r(\mathbf{q}_j) + \mathcal{O}(\Delta x^p) + \mathcal{O}(\Delta x^{p-1}) \end{aligned} \quad (54)$$

By using the identities (54) and the antisymmetric condition (49), we get

$$\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \sum_r \frac{\hat{\partial}}{\hat{\partial}x_r} \mathbf{F}_r(\mathbf{q}_j) = \mathcal{O}(\Delta x^{p-1}). \quad (55)$$

Therefore, by using the identities for the commutators (45) and (47), we can

rewrite (53) as

$$\begin{aligned} \mathbf{0} &= \sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \frac{\hat{\partial}}{\partial t} \mathbf{q}_j + \mathcal{O}(\Delta x^{p-1}) \\ &= \frac{\hat{\partial}}{\partial t} \sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_j + \mathcal{O}\left(\frac{\Delta x^p}{\Delta t}\right) + \mathcal{O}\left(\frac{\Delta t^q}{\Delta x}\right) + \mathcal{O}(\Delta x^{p-1}), \end{aligned} \quad (56)$$

which is equivalent to (50). \square

This theory is valid in the case of sufficiently smooth solutions. A numerical scheme cannot distinguish between discontinuous solutions and smooth solutions with high gradients. So at the first glance, the theorem directly transfers to that case. But due to stability reasons, one has to take measures to prevent unphysical oscillations, which results in the need of some limiting technique, like TVD, ENO/WENO etc. As a consequence of the application of limiting, the order of the scheme near discontinuities is lowered. Thus, the estimate (50) is much weaker near discontinuities than in smooth regions.

3.2.2. The semidiscrete case. For the semidiscrete case, we have to consider the construction via numerical flux functions in more detail. In the context of finite volumes, numerical schemes usually are represented in the semidiscrete form

$$\frac{\partial}{\partial t} \mathbf{q}_j - \sum_{\mathfrak{k} \in \mathfrak{R}_j} G_{\mathfrak{k}}(\mathbf{q}) = \mathbf{0}, \quad (57)$$

where \mathfrak{R}_j denotes the set of all cell faces of cell j , and G denotes a numerical flux function, normal to the cell face. This numerical flux function is allowed to depend on several \mathbf{q}_i , i.e., on the values of \mathbf{q} in several cells of the computational grid. Therefore, (57) represents a system of ordinary differential equations in time. When we solve this, using some standard scheme for ODEs, at a first glance the discrete time derivative only depends on values in the same space point. But this is not true for many choices of the numerical flux function G .

We now take a closer look at a typical example: One of the most important nonlinear schemes is the scheme by Harten, Lax, and van Leer [23], which for our purposes, is a nice model since it clearly distinguishes between the central and the upwinding part of the viscous flux. For this, the numerical flux function reads

$$\begin{aligned} &G_{\text{HLL}}(\mathbf{q}_r, \mathbf{q}_l) \\ &= \frac{1}{2}(f(\mathbf{q}_r) + f(\mathbf{q}_l)) - \frac{1}{2} \frac{S_R + S_L}{S_R - S_L} (f(\mathbf{q}_r) - f(\mathbf{q}_l)) + \frac{S_R S_L}{S_R - S_L} (\mathbf{q}_r - \mathbf{q}_l) \end{aligned} \quad (58)$$

with some bounding signal speeds S_L and S_R for the Riemann problem defined by the states left and right of the cell face, \mathbf{q}_l and \mathbf{q}_r . If $S_R = -S_L = \Delta x / \Delta t$ for equidistant Cartesian grids, this is just the numerical flux of the Lax–Friedrichs

scheme. If we have a tighter but still symmetric choice $S_R = -S_L$ of the bounding speeds we find the Rusanov– or local Lax–Friedrichs scheme. In (58) there are three contributions: a symmetric one, that would leave us with central differences of second order in space, an upwinding term, and another symmetric term, that does not depend on the flux, but only on the state \mathbf{q} itself. The second and third terms both contribute to the numerical viscosity. If we write the resulting scheme in the fully discrete difference form

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j + \hat{\mathbf{V}} \cdot \mathbf{F}(\mathbf{q}_j) = \mathbf{0},$$

the third term becomes a part of the time difference instead of the space difference. For the semidiscrete scheme, the central viscosity terms make up an additional sum:

$$\frac{\partial}{\partial t} \mathbf{q}_j + \sum_r \frac{\hat{\partial}}{\partial x_r} F_r(\mathbf{q}_j) + \sum_{k \in K} \gamma_{k,j} \mathbf{q}_k = \mathbf{0}. \quad (59)$$

Thus, an analogue of Theorem 2 is true with (50) is replaced by

$$\begin{aligned} \frac{\partial}{\partial t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_j \right) + \sum_{k \in K} \gamma_{k,j} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_k \right) \\ = \mathcal{O} \left(\frac{\Delta x^p}{\Delta t} \right) + \mathcal{O} \left(\frac{\Delta t^q}{\Delta x} \right) + \mathcal{O}(\Delta x^{p-1}), \end{aligned} \quad (60)$$

which can be interpreted as a discrete heat equation with a source term of order

$$\mathcal{O} \left(\frac{\Delta x^p}{\Delta t} \right) + \mathcal{O} \left(\frac{\Delta t^q}{\Delta x} \right) + \mathcal{O}(\Delta x^{p-1}).$$

With a suitable choice of the central part of the numerical viscosity, we can expect the discrete involution to converge to zero in time. With a poor choice, it might increase in time, even if the right side of (60) vanishes.

3.2.3. A linear special case. In this section, we consider a linear special case, which allows for exact discrete involutions. As a consequence of the previous sections, the approximation error in discrete involutions is mainly due to the commutators of the discrete differential operators. A smaller contribution is due do the commutator of these operators with the matrices \mathbf{M}_i , which make up the involution (3). If the commutators vanish, the involution is exact. We take a closer look at discrete differential operators that can be rewritten as

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h}_j = \sum_{i \in I} \tilde{\alpha}_i \mathbf{h}_{j+i} \quad (61)$$

for the time derivative and

$$\frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{h}_j = \sum_{i \in I} \tilde{\beta}_i^l \mathbf{h}_{j+i}, \quad (62)$$

where \mathbf{i} and \mathbf{j} are index vectors and I is a set of index vectors. This is a typical situation on structured grids, staggered or collocated. Here in addition, we require the coefficients $\tilde{\alpha}_i$ and $\tilde{\beta}_i^k$ to depend only on their index \mathbf{i} . Thus, the resulting scheme for a hyperbolic conservation law is linear. If now the coefficients commute with each other, for the mixed derivatives, which are just double summations, we find

$$\frac{\hat{\partial}}{\hat{\partial}x_l} \left(\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{h}_j \right) = \frac{\hat{\partial}}{\hat{\partial}t} \left(\frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{h}_j \right). \quad (63)$$

They commute; the commutator vanishes. For the coefficients to commute with the \mathbf{M}_i we have the additional requirement that the \mathbf{M}_i are square matrices or the coefficients are scalar. So, in most cases we are restricted to scalar coefficients, especially for divergence preserving transport. With these preparations we can state the following discrete analogue of 1:

Theorem 3. *Let the (weakly) hyperbolic conservation law*

$$\mathbf{q}_t + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0} \quad (64)$$

be given, together with constant matrices \mathbf{M}_i that satisfy

$$\mathbf{M}_l \mathbf{F}_r + \mathbf{M}_r \mathbf{F}_l = \mathbf{0} \quad \text{for } l, r = 1, 2, \dots \quad (65)$$

Furthermore let the linear difference operators

$$\frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j = \sum_{i \in I^l} \tilde{\beta}_i^l \mathbf{q}_{j+i}, \quad (66)$$

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_j = \sum_{k \in K} \tilde{\alpha}_k \mathbf{q}_{j+k}, \quad (67)$$

be given, where the coefficients β_k and α_i^l commute with each other and with the \mathbf{M}_i .

If we discretize the conservation law (64) with the finite difference operators (66) and (67), then the following analogue of (5) holds true:

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) = \mathbf{0} \quad \text{for all } \mathbf{j}, \quad (68)$$

and the discrete involution is exact.

If we assume the scheme to be constructed by means of numerical flux functions and consider the semidiscrete form, (68) has to be replaced by

$$\frac{\partial}{\partial t} \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_j \right) + \sum_{k \in K} \tilde{\gamma}_k \left(\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_{j+k} \right) = \mathbf{0}. \quad (69)$$

This is true if the coefficients $\tilde{\beta}_i^l$ arising from the upwind part of the numerical flux function satisfy the above mentioned requirements: they commute with each other and with the \mathbf{M}_i . If, for example, the HLL flux (58) is applied to a constant coefficient hyperbolic system, the resulting coefficients $\tilde{\beta}_i^l$ are scalar constants.

3.3. Discrete involutions and standard schemes. In the beginning of Section 3, we raised the question, which type of discrete involutions, if at all, we will find with standard schemes. Will we find exact ones or only approximate ones? Here we restrict our study to constant coefficient schemes. Thus, the only remaining question is: do the coefficients commute. We consider the Lax–Friedrichs, Lax–Wendroff, and upwind schemes, followed by a remark on the use of Runge–Kutta schemes for the time discretization. If these schemes are applied to a constant coefficient hyperbolic system, which means $F_i(\mathbf{q}) = \mathbf{A}_i \mathbf{q}$, due to the constant signal speeds, a constant time-step can be chosen, so that not only the coefficients of the space discretization are constant, but also those of the time difference.

Although the schemes investigated in this section are rarely used in their pure form, most schemes in practical use are generalizations of these simple methods and, thus, inherit some of the properties of the underlying linear scheme. The results will be explored in Section 4 to study the interplay of discrete involutions and resonance.

3.3.1. The Lax–Friedrichs scheme. The behavior of the Lax–Friedrichs scheme is best understood if we take a careful look on its derivation. The starting point is the desire for a simple symmetric scheme. Therefore, the most obvious choice is to take central differences of second order in space and forward differences in time. In one space dimension, this leads to the simple explicit scheme

$$\frac{\mathbf{q}_k^{n+1} - \mathbf{q}_k^n}{\Delta t} + \frac{F(\mathbf{q}_{k+1}^n) - F(\mathbf{q}_{k-1}^n)}{2\Delta x} = \mathbf{0}. \quad (70)$$

Since this turns out to be unconditionally unstable, one looks for a replacement. In the Lax–Friedrichs scheme this modification is done in a symmetric way. In the time discretization, the value \mathbf{q}_k^n is replaced by the arithmetic mean of its neighbors in space:

$$\frac{\mathbf{q}_k^{n+1} - \frac{1}{2}(\mathbf{q}_{k+1}^n + \mathbf{q}_{k-1}^n)}{\Delta t} + \frac{F(\mathbf{q}_{k+1}^n) - F(\mathbf{q}_{k-1}^n)}{2\Delta x} = \mathbf{0}. \quad (71)$$

An interesting consequence of this construction is that \mathbf{q}_k^{n+1} does not depend on \mathbf{q}_k^n . An advantage of this is the possibility to use the scheme in a staggered manner, meaning that in each time step we toggle between evaluating at odd and even indexes. This gave rise to the development of the Nessyahu–Tadmor scheme [37]. As a disadvantage, in non-staggered use of the scheme, high-frequency oscillations are observed [7].

It is possible to rewrite the scheme in the usual conservation form, making the difference between formulas (70) and (71) part of the numerical flux function. In the sense of applying discrete difference operators instead of the analytic ones to the conservation law (1), this would result in an additional, viscous flux. But the difference, although made part of the numerical flux, still remains part of the time discretization because the correction term does not include any contributions of the flux function $f(\cdot)$. Thus, we have

$$\frac{\hat{\partial}}{\hat{\partial}t} \mathbf{q}_k = \frac{\mathbf{q}_k^{n+1} - \frac{1}{2}(\mathbf{q}_{k+1}^n + \mathbf{q}_{k-1}^n)}{\Delta t} = \frac{\mathbf{q}_k^{n+1} - \mathbf{q}_k^n}{\Delta t} - \frac{\Delta x^2}{2\Delta t} \frac{\mathbf{q}_{k+1}^n + 2\mathbf{q}_k^n + \mathbf{q}_{k-1}^n}{\Delta x^2}. \quad (72)$$

If we apply this discrete time derivative to a scalar quantity h , the condition

$$\frac{\hat{\partial}}{\hat{\partial}t} h_k = 0 \quad \text{for all } k \quad (73)$$

is the same as applying a simple explicit method to the heat equation

$$h_t - \frac{\Delta x^2}{2\Delta t} h_{xx} = 0. \quad (74)$$

If we solve this heat equation exactly, employing homogeneous Dirichlet conditions on the boundaries, we find that h converges to zero at any place. If, instead of the scalar h , we apply Equation (73) to a vector quantity \mathbf{h} the same holds true for every component of \mathbf{h} . In several space dimensions, we get a spatial anisotropic heat equation; for three dimensions it is

$$\mathbf{h}_t - \frac{\Delta x^2}{2\Delta t} \mathbf{h}_{xx} - \frac{\Delta y^2}{2\Delta t} \mathbf{h}_{yy} - \frac{\Delta z^2}{2\Delta t} \mathbf{h}_{zz} = \mathbf{0}.$$

As a consequence, for instance in the case of homogeneous Dirichlet boundary conditions, all components of \mathbf{h} converge to zero. Therefore, if we have a conservation law

$$\mathbf{q}_t + \nabla \cdot \mathbf{F}(\mathbf{q}) = \mathbf{0}$$

with an involution

$$\sum_i M_i q_{x_i} = \mathbf{0},$$

discretized with the Lax–Friedrichs scheme and boundary conditions, which are consistent with the involution, then

$$\sum_i M_i \frac{\hat{\partial}}{\partial \hat{x}_i} q$$

is an exact discrete involution, which even converges to zero in time.

Since the Balbás–Tadmor scheme [2] by its construction is close to the Lax–Friedrichs scheme, we can already at this point expect that it produces only small divergence errors, which are even nicely damped away.

We will use this considerations later on to identify in numerical flux functions the terms which have to be considered a contribution to the discrete time derivative instead of the space derivative. And we will employ a systematic control on these terms, namely the central viscosity, to minimize the production of divergence errors in a standard scheme.

3.3.2. The Lax–Wendroff scheme. To study the Lax–Wendroff scheme, we start with the simplest possible system of conservation laws: the scalar linear advection equation

$$q_t + aq_x = 0. \quad (75)$$

The idea for the Lax–Wendroff scheme and its relatives is to start with a Taylor expansion in time:

$$q(x, t + \Delta t) = q(x, t) + \Delta t q_t(x, t) + \frac{1}{2} \Delta t^2 q_{tt}(x, t) + \mathcal{O}(\Delta t^3). \quad (76)$$

Using the original conservation law (75) and its time derivative, the time derivatives in (76) can be replaced by space derivatives:

$$q(x, t + \Delta t) = q(x, t) - a \Delta t q_x(x, t) + \frac{1}{2} a^2 \Delta t^2 q_{xx}(x, t) + \mathcal{O}(\Delta t^3). \quad (77)$$

From this we get the Lax–Wendroff scheme by applying standard second-order central differences for first and second space derivatives. If we use standard upwind differences of second order, we find the Beam–Warming scheme. The arithmetic mean of both schemes results in the Fromm scheme.

Let us now concentrate on the Lax–Wendroff scheme. Since, according to the above choice, we have

$$\frac{\hat{\partial}}{\partial \hat{x}} q_j = \frac{q_{j+1} - q_{j-1}}{2\Delta x}, \quad (78)$$

for the discrete second space derivative, we would expect

$$\frac{\hat{\partial}}{\partial \hat{x}} \left(\frac{\hat{\partial}}{\partial \hat{x}} q_j \right) = \frac{q_{j+2} - 2q_j + q_{j-2}}{4\Delta x^2}. \quad (79)$$

But the Lax–Wendroff scheme employs

$$\frac{\tilde{\partial}}{\partial \tilde{x}^2} q_j = \frac{q_{j+1} - 2q_j + q_{j-1}}{\Delta x^2}, \quad (80)$$

which is apparently not the same. To interpret this as a part of the space discretization, we would have to write it in terms of the difference operator (78). But this is impossible. Therefore, it is impossible to interpret the viscosity term of the Lax–Wendroff scheme as a part of the space discretization, even in the simple case of the one dimensional scalar advection equation. Instead, we have to view it as a part of the time difference. Thus, the time difference would read as

$$\begin{aligned} \frac{\hat{\partial}}{\partial t} q_j^n &= \frac{q_j^{n+1} - q_j^n}{\Delta t} + \frac{1}{2} a^2 \Delta t^2 \frac{q_{j+1}^n - 2q_j^n + q_{j-1}^n}{\Delta x^2} \\ &= \frac{1}{\Delta t} q_j^{n+1} - \left(\frac{1}{\Delta t} + a^2 \frac{\Delta t^2}{\Delta x^2} \right) q_j^n + \frac{1}{2} a^2 \frac{\Delta t^2}{\Delta x^2} (q_{j+1}^n + q_{j-1}^n). \end{aligned}$$

A similar formula would be found for a one-dimensional linear system of conservation laws. But then, we would have to replace a by the system matrix \mathbf{A} . Thus, the coefficients in the discrete time derivative become matrix valued. So, [Theorem 3](#) can only be applied to a small number of systems, namely those, for which the system matrix \mathbf{A} and the matrix \mathbf{M} which makes up the involution commute.

If we had used (79) instead of (80) for the second derivative, it would have been possible to interpret the viscous term as a part of the discrete space derivative. But in the case of a system this, again, leads to matrix valued coefficients — this time in the discrete space derivative. Thus, the same restrictions apply as for the original Lax–Wendroff scheme. In addition, for systems in several space directions we would have to require the matrices \mathbf{A}_i for the different space directions to commute with each other.

For several space dimensions we only show a two-dimensional example,

$$\mathbf{q}_t + \mathbf{A} \mathbf{q}_x + \mathbf{B} \mathbf{q}_y = \mathbf{0}. \quad (81)$$

For this the analogue of (77) reads as

$$\begin{aligned} \mathbf{q}(x, y, t + \Delta t) &= \mathbf{q}(x, y, t) - \Delta t (\mathbf{A} \mathbf{q}_x + \mathbf{B} \mathbf{q}_y) \\ &\quad + \frac{1}{2} \Delta t^2 (\mathbf{A}^2 \mathbf{q}_{xx} + \mathbf{A} \mathbf{B} \mathbf{q}_{yx} + \mathbf{B} \mathbf{A} \mathbf{q}_{xy} + \mathbf{B}^2 \mathbf{q}_{yy}) + \mathcal{O}(\Delta t^3). \end{aligned} \quad (82)$$

Apparently, the same arguments hold as for one space dimension. If we take the viscous term as part of the time difference, we can apply [Theorem 3](#), as long as both of \mathbf{A} and \mathbf{B} commute with both of the matrices \mathbf{M}_x and \mathbf{M}_y making up an involution of system (81). This extends to higher dimensions in a straight forward manner.

If the matrices do not commute, we only find—provided the viscous term is taken as part of the time difference—

$$\sum_l \mathbf{M}_l \frac{\hat{\partial}}{\hat{\partial}t} \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j = \mathbf{0}. \quad (83)$$

This is a much weaker condition than (68). In fact, numerical experiments show that the approximation of the involution is in no way better than for any nonlinear scheme of the same order.

For the Beam–Warming scheme, the results are quite similar. Now, most second-order schemes, especially those based on TVD limiters, are constructed by using weighted means of the Lax–Wendroff, as a central scheme, and the Beam–Warming, as an upwind scheme. Thus, for these schemes, we can not expect the conditions of Theorem 3 to hold. The best we can hope for, is an approximate involution in the sense of Theorems 2 and 1.

3.3.3. The upwind-scheme. For a scalar conservation law, the upwind scheme assigns a one sided difference operator to each space derivative. This operator takes into account the upwind direction, i.e., for positive signal speed, backward differences are used and for negative signal speeds forward differences. In the case of a linear system, the upwind method is applied to each characteristic field.

The simple case: full upwinding. The simplest case is full upwinding: in each space direction for all characteristic fields the same upwind direction is found. In this case all discrete space derivatives $\hat{\partial}/\hat{\partial}x_r$ are one sided standard differences of first order, forward or backward, depending on the upwind direction for x_r .

The effects of this can be nicely seen, when the scheme is applied to the linearized induction equation of two-dimensional magnetohydrodynamics:

$$\mathbf{B}_t - \nabla \times (\mathbf{v} \times \mathbf{B}) = \mathbf{0}, \quad \mathbf{v} = (u, v)^T \equiv \text{constant}, \quad (84)$$

with positive velocity components u and v . As Fey and Torrilhon [41] point out, this is an interesting example, modeling most of the important properties of real MHD, at least in the context of involutions. It is a linear conservation system with $\nabla \cdot \mathbf{B}$ as an involution. With the matrices $\mathbf{M}_1 = (1 \ 0)$ and $\mathbf{M}_2 = (0 \ 1)$, we find that it satisfies the conditions for Theorem 1. Thus, with appropriate difference operators, we will obtain a discrete involution.

In space, we employ two different types of differences. First we use standard upwind. Since there is only one nonzero wave speed for each space direction, we end up just with one-sided differences for $\hat{\partial}/\hat{\partial}x$ and $\hat{\partial}/\hat{\partial}y$. So we have no matrix valued coefficients, and the conditions of Theorem 3 are satisfied. For a second test, we employ the corner transport upwind (CTU) scheme, a variant of standard upwind, which takes into account the direction of the transport. This results in the

transverse upwind differences

$$\begin{aligned}\frac{\hat{\partial}}{\hat{\partial}x} h &= (1 - c_y) \frac{h_{i,j} - h_{i-1,j}}{\Delta x} + c_y \frac{h_{i,j-1} - h_{i-1,j-1}}{\Delta x}, \\ \frac{\hat{\partial}}{\hat{\partial}y} h &= (1 - c_x) \frac{h_{i,j} - h_{i,j-1}}{\Delta y} + c_x \frac{h_{i-1,j} - h_{i-1,j-1}}{\Delta y},\end{aligned}\tag{85}$$

where c_x and c_y denote the directional Courant numbers. In time, we always employ forward differences of first order. Therefore we expect the involution to be constant in time.

First example. As initial data, we discretize the divergence-free field $\mathbf{B} = (B_1, B_2)^T$ with

$$B_1 = \cos(2\pi x + \pi y), \quad B_2 = -2 \cos(2\pi x + \pi y),$$

on a 320×320 grid for the square region $[-1, 1] \times [-1, 1]$ with periodic boundary conditions. For the discrete initial values, we employ a rather naive method: we just evaluate at the cell center. Thus, the initial divergence is not exactly zero. The results are shown in [Figure 1](#). In the left picture we see that the discrete divergence measured in upwind differences is constant in time, it sticks to its initial value, if the standard upwind scheme is used. In the right picture, the same is found for the divergence measured in transverse differences with the corresponding CTU scheme employed. Although not depicted here, in both cases not only the norm of the divergence is constant. The discrete divergence itself is constant, as was predicted by the above theory.

The divergence measured in central differences, although almost zero in the initial state, grows to approach the divergence measured in terms of the difference operator used in the scheme, which is indeed much larger.

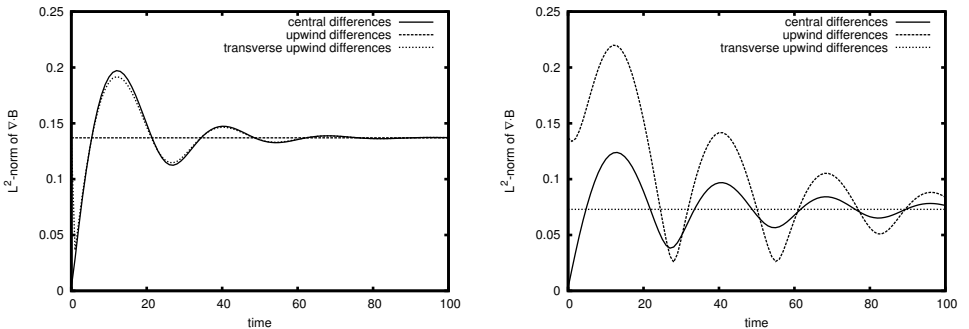


Figure 1. Smooth example: time behavior of the L^2 -norm of the discrete divergence for standard upwind (left) and corner transport upwind (right), measured with central differences, upwind differences, and transverse upwind differences.

From this, we can draw two important conclusions: First, the usual technique of projecting the magnetic field to a divergence-free field with respect to some higher-order central difference is insufficient. The projection should be done with respect to the difference operator actually used in the scheme. For general nonlinear systems with changing upwind directions, this is nearby impossible. Especially, it is impossible to provide a “divergence-free” initial state that is adequate for all cases. Second, upwind schemes, by their lack of central viscosity, are unable to damp the divergence error introduced by the initial state.

Second example. As a second example, we present an oblique Riemann-problem, a piecewise constant initial state with discontinuity normal to $(1, 1)^T$ reproduced on a Cartesian grid. The discontinuity is just the diagonal of the cells it intersects. For the left and right state and the state in the cells with the discontinuity, we take

$$\mathbf{B}_l = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \mathbf{B}_r = \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \quad \text{and thus} \quad \mathbf{B}_r^* = \mathbf{B}_l^* = \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix} \quad (86)$$

for the cells intersected by the discontinuity, i.e., we project the data onto the grid in a finite volume manner. The data are analytically divergence-free. For $u, v > 0$ they are also discrete divergence-free when we employ upwind differences. For $u > 0, v < 0$, they are not.

In [Figure 2](#) it can be seen that also in the discontinuous case the divergence measured in the differences used in the scheme is constant. [Figure 2](#) also shows that the initial state has to be divergence-free with respect to the differences used in the scheme. If not, the divergence will raise pretty soon. The worst results are obtained, when we do a wrong upwinding (lower row). For linear systems like our model problem, this is no issue. But for nonlinear systems like full MHD, this adds a new problem to the lack of exact involutions: Since the upwind direction depends on the state, it is in general impossible to know the difference operators in advance. So, the best we can get is an initial divergence in the order of the scheme itself.

The general case. For the investigation of the general case, we start with a one dimensional situation:

$$\mathbf{q}_t + \mathbf{A}\mathbf{q}_x = \mathbf{0}. \quad (87)$$

For a hyperbolic conservation law, \mathbf{A} can be decomposed into

$$\mathbf{A} = \mathbf{R}\mathbf{\Lambda}\mathbf{L}, \quad (88)$$

where \mathbf{R} and $\mathbf{L} = \mathbf{R}^{-1}$ are the matrices of the right and left eigenvectors of \mathbf{A} and $\mathbf{\Lambda}$ is the diagonal matrix of the eigenvalues of \mathbf{A} . By manipulating the entries of $\mathbf{\Lambda}$ one can easily construct matrices \mathbf{A}^+ , \mathbf{A}^- and $|\mathbf{A}|$ which have the same eigenvectors as \mathbf{A} but differ in their eigenvalues: For \mathbf{A}^+ all negative eigenvalues are replaced by zero, for \mathbf{A}^- the positive ones, and for $|\mathbf{A}|$ we replace all eigenvalues by their

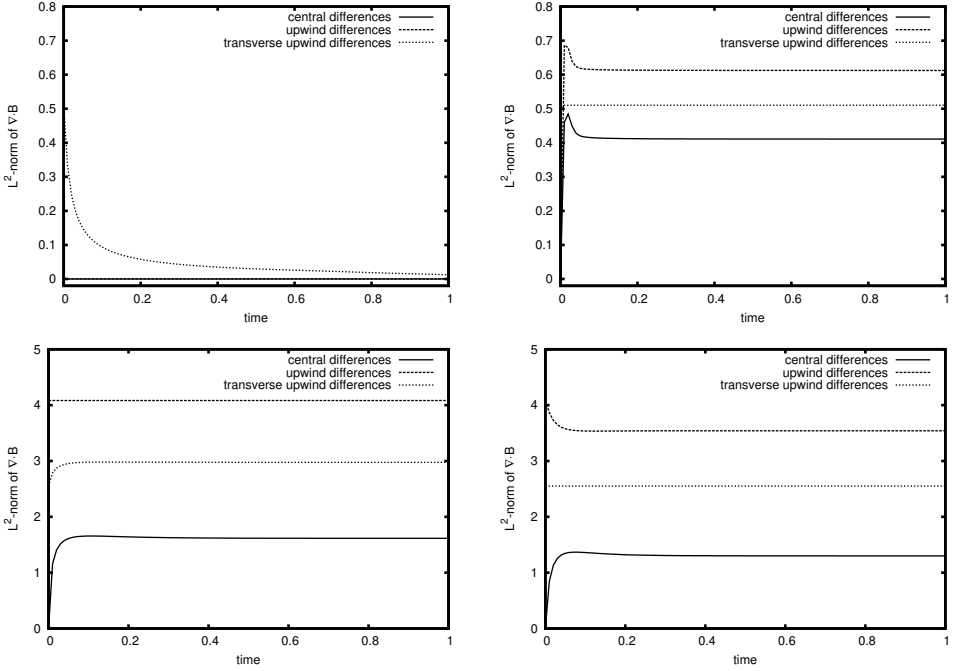


Figure 2. Discontinuous example: time behavior of the L^2 -norm of the discrete divergence for standard upwind (left) and corner transport upwind (right), measured with central differences, upwind differences, and transverse upwind differences. Upper row: original setting; lower row: sign of v changed.

absolute values. Using these matrices, we can write the resulting discrete space difference operator as

$$\begin{aligned}
 \frac{\hat{\partial}}{\hat{\partial}x}(Aq_k) &= A^+ \frac{q_k - q_{k-1}}{\Delta x} + A^- \frac{q_{k+1} - q_k}{\Delta x} \\
 &= \frac{1}{\Delta x} [A^+ q_k - A^+ q_{k-1} + A^- q_{k+1} - A^- q_k] \\
 &= \frac{1}{\Delta x} [-A^+ q_{k-1} + |A| q_k + A^- q_{k+1}] \\
 &= -\frac{1}{2\Delta x} [(|A| + A)q_{k-1} - 2|A|q_k + (|A| - A)q_{k+1}]. \quad (89)
 \end{aligned}$$

From these manipulations it can be easily seen that it is impossible to write the difference operator without matrix valued coefficients. Therefore, [Theorem 3](#) can only be applied if A and the matrix M for the involution commute. In the multidimensional case, we have to require that all A_r and M_l commute with each other. Thus, in general, we find no exact involutions for the upwind scheme, especially when the involution is a divergence.

Since most high quality numerical flux functions are based on upwinding, this implies that in real world computations, we can only expect an approximate involution in the sense of [Theorem 2](#). In addition, the lack of central viscosity prevents the scheme from damping the errors in the involution.

3.3.4. A remark on the use of Runge–Kutta schemes. Runge–Kutta schemes play an important role in numerical simulations of time-dependent problems. They are also the method of choice for the starting procedure in a multistep scheme like leapfrog and its variants. Therefore, we are interested in the effects of using them for systems with involutions.

If the space discretization is done with differences satisfying the conditions of [Theorem 3](#), then we get

$$\frac{\partial}{\partial t} \left(\sum_l M_l \frac{\hat{\partial}}{\partial x_l} \mathbf{q}_j \right) = \mathbf{0}. \quad (90)$$

If a consistent one step method is applied to that, the resulting scheme is involution preserving. When taken as a starting procedure for leapfrog, it also leads to an involution preserving scheme.

If the scheme is constructed by means of numeric flux functions, we get for the semidiscrete involution the expression given in [\(69\)](#) if the requirements given there are satisfied. This expression includes the central numerical viscosity. It corresponds to a discretized parabolic equation. When the numerical viscosity is reasonable, any stable time discretization shows the same behavior as we found in [Section 3.3.1](#) for the involution in the Lax–Friedrichs scheme.

4. Discrete involutions and resonance

In this section we identify discrete involutions and resonance as the key one needs to understand how divergence errors arise in MHD simulations and destroy them.

By means of a computational example we show how resonance makes the estimates for the discrete involution in [Theorem 2](#) worthless. We study the role of the central viscosity of the scheme and explain why the Balbás–Tadmor scheme [\[2; 1\]](#) and the Zachary–Malagoli–Colella scheme [\[46\]](#) produce only small divergence errors. In this course, we present a modification of the Roe-solver which shows the same stability. This modification is not intended to replace divergence cleaning, but to reduce the errors which have to be swept out of the computational domain.

4.1. The De Sterck test. The De Sterck test [\[11\]](#) is a special configuration for a shallow water MHD flow. It shows a strong tendency to develop resonant phenomena and, thus, to single out numerical schemes which are prone to divergence errors. The test problem imposes a supersonic horizontal grid-aligned inflow on the left

boundary of a rectangular domain. The initial state in the lower half of the domain, and also of the left boundary, contains a resonant mode. The initial data in the upper half are

$$h = 2, \quad u = 5.5, \quad v = 0, \quad B_1 = 0.5, \quad B_2 = 0, \quad (91)$$

and in the lower half

$$h = 1, \quad u = 4.5, \quad v = 0, \quad B_1 = 2, \quad B_2 = 0. \quad (92)$$

The gravitational constant is set to one. Since the discontinuity is aligned with the grid, the initial data are discrete divergence-free for any reasonable difference operator. We performed a test on a 200×200 grid for the domain $[-1, 1] \times [-1, 1]$ with the Local Lax–Friedrichs scheme (LLF). The numerical flux over the cell faces is computed with 1d-physics. This is a widespread approach. In one-dimensional physics a one-dimensional divergence constraint applies. Thus the equation for hB_1 , in the full MHD the equation for B_1 , can be eliminated. The component hB_1 , or B_1 in full MHD, is constant in space and time and, thus, only a parameter. The reduced 1d-system is fully hyperbolic. When used for multidimensional simulations, this introduces two difficulties: on each cell face the parameter for the magnetic field component normal to the face has to be chosen in some way, and we lose control over part of the viscosity of the scheme, namely the viscosity on the neglected wave. But this is exactly the wave which is responsible for resonance.

For the first six time steps the absolute value of the resulting fastest wave speed, $u - c_g$, with the magnetogravitational speed $c_g = \sqrt{B_1^2 + gh}$ is plotted in Figure 3. It turns out that, in this case, resonance, once initiated, grows very fast. It also affects the wave speeds, which depend on the magnetic field. When we consider

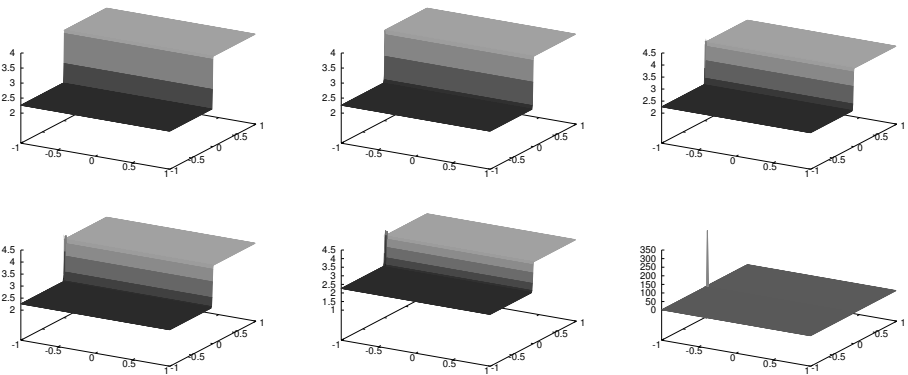


Figure 3. Absolute value of $u - c_g$ for De Sterck test with LLF based on one-dimensional physics. First six time steps (top row 1–3, bottom row 4–6). Note the different scaling in the last picture.

the estimate for the general discrete involution

$$\frac{\hat{\partial}}{\hat{\partial}t} \left(\sum_l M_l \frac{\hat{\partial}}{\hat{\partial}x_l} \mathbf{q}_j \right) = \mathcal{O} \left(\frac{\Delta x^p}{\Delta t} \right) + \mathcal{O} \left(\frac{\Delta t^q}{\Delta x} \right) + \mathcal{O}(\Delta x^{p-1}) \quad \text{for all } j, \quad (93)$$

from [Theorem 2](#), we find that the first-order term, $\mathcal{O}(\Delta x^p/\Delta t)$, is most critical. The fast growing wave speeds result in a fast decreasing time step. Thus, the estimate (93) becomes weaker each time step. Divergence errors drive resonance, and resonance weakens the bound for the growth of the divergence errors.

In computations on Cartesian grids it is common to configure the initial state in a way that all discontinuities are aligned with the grid. For a piecewise constant initial state, consistent with the constraint, this means that for any consistent difference operators the discrete initial state also satisfies the discrete constraint. The involution can only be violated by rounding errors. Since rounding errors are $\mathcal{O}(1)$, the introduced error in the involution is of order $\mathcal{O}(1/\Delta x)$. Grid refinement results in even stronger resonance phenomena. The numerical viscosity and, thus, the damping of the resonance is reduced. Hence, for a scheme which fails due to resonance, it is impossible to improve the situation by grid refinement. The situation is even worse, as can be verified by the numerical tests in [Section 4.4](#).

4.2. The role of the central numerical viscosity. Already Crockett et al. [8] realized that adding viscosity — in their case by the Marder approach [30] — reduces resonance effects in MHD. So, we go into that in more detail. To study the role of central numerical viscosity in more detail we begin with a simple example. In [Figure 4](#), we show numerical results for the situation described in the beginning of [Section 2.3.1](#). We trigger resonance by a jump of B_1 in the middle of the computational domain. Apparently the resonance effects are much weaker if we employ the Lax–Friedrichs scheme instead of the CIR scheme. The main difference between these two schemes is that the LF scheme is central while the CIR scheme employs wave wise upwinding. Thus, the LF scheme provides central viscosity, while the CIR scheme does not.

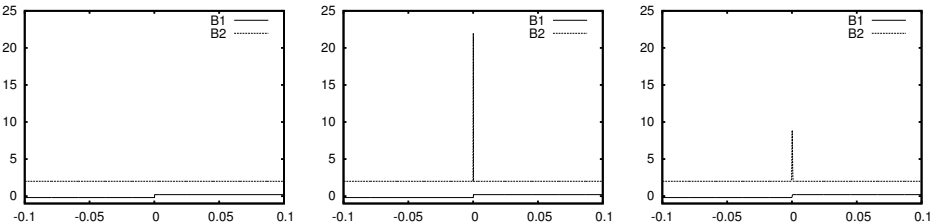


Figure 4. Effects of resonance: initial condition, result with CIR scheme, result with Lax–Friedrichs scheme.

But in the general case, the Lax–Friedrichs scheme, due to its high viscosity, is not preferable. Therefore, in practical use other schemes based on numerical flux functions are used. At this point, we reconsider the HLL flux (58):

$$G_{\text{HLL}}(\mathbf{q}_r, \mathbf{q}_l) = \frac{1}{2}(f(\mathbf{q}_r) + f(\mathbf{q}_l)) - \frac{1}{2} \frac{S_R + S_L}{S_R - S_L} (f(\mathbf{q}_r) - f(\mathbf{q}_l)) + \frac{S_R S_L}{S_R - S_L} (\mathbf{q}_r - \mathbf{q}_l). \quad (94)$$

Obviously, the viscosity terms are closely related to the signal speeds. This is a general issue [24; 28; 13; 22]. Therefore, in practice, the central viscosity can not be chosen arbitrarily high. A simple approach is HLL with $S_L = -S_R$, which refers to the local Lax–Friedrichs scheme. This choice imposes a lower bound on the viscosity for all waves, thus also for the resonant wave⁴. This is a prototype for many schemes, which do not explicitly resolve the resonant wave. Both, the Balbás–Tadmor scheme [2] and the Zachary–Malagoli–Colella scheme [46] belong to this class.

As a prototype of schemes which, by construction, explicitly resolve all waves, we consider the Harten entropy fix [21] for the Roe-solver — not to be confused with the Harten–Hyman entropy fix [22], which allows to impose a lower bound for the viscosity on each wave separately. It is constructed such that the viscosity depends smoothly on the wave speeds. Harten replaces the absolute value of an eigenvalue λ of the Roe matrix by

$$\phi(\lambda) = \begin{cases} |\lambda| & \text{if } |\lambda| \geq \delta, \\ (\lambda^2 + \delta^2)/(2\delta) & \text{if } |\lambda| < \delta, \end{cases} \quad (95)$$

where δ is a small parameter. The numerical viscosity is bounded below by $\delta/2$. Since additional numerical viscosity on a single wave is equivalent to the splitting of the wave into two weaker waves [28; 22], the optimal, i.e., the maximal admissible, choice for the parameter is twice the largest absolute value of an eigenvalue of the Roe matrix: $\delta = 2 |\lambda_{\max}|$. This puts the same amount of viscosity on the wave as in the LLF scheme. A simpler, but still reasonable choice would be $\delta = 2 |u|$. The speeds of the waves resulting from the corresponding splitting of the original resonant wave would be $\pm\lambda_{\max}$ or $\pm u$ respectively.

4.3. The assumption of one-dimensional physics in flux computations. To study the role of the assumption of one-dimensional physics in the construction of numerical flux functions, we start with an example. In Section 4.1, we demonstrated the effects of resonance by applying the LLF scheme with the numerical flux based on one-dimensional physics to the De Sterck test case. Now we repeat the same computation without the assumption of one-dimensional physics. The results are

⁴Since resonance only occurs in certain physical states, it would be more correct to call it the *wave which might become resonant*. But for the sake of readability, we stick to this simplistic formulation.

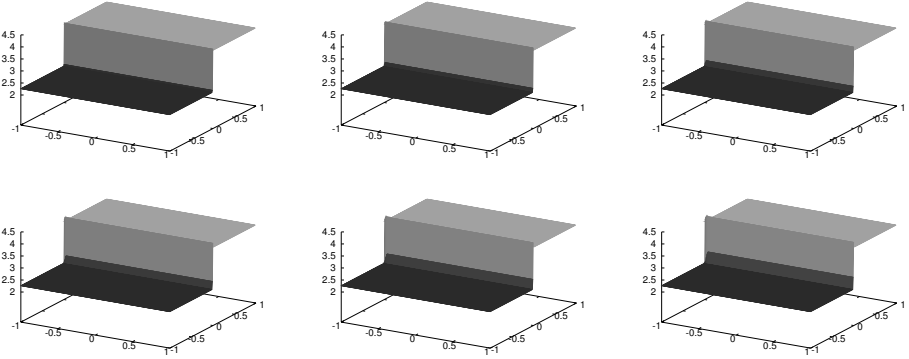


Figure 5. Absolute value of $u - c_g$ for De Sterck test with LLF for full system. First six time steps (top row 1–3, bottom row 4–6).

presented in [Figure 5](#). As one would expect by the considerations of the previous section, the resonance is nicely damped. The divergence errors are much smaller than with the one-dimensional physics in [Figure 3](#). Something got lost by the assumption of one-dimensional physics. The resulting viscosity seems to be weak or even antidiffusion on the resonant wave. Thus, in a scheme which uses projection to prevent divergence errors, the projection has to be done more often to keep the simulation stable. The work, saved by the easier flux computation, results in a much higher work for divergence cleaning.

The assumption of one-dimensional physics in the flux computation would, on a Cartesian grid, imply that all terms including B_{1x} , B_{2y} and B_{3z} are neglected. In general, this leads to a modeling error and, thus, to an error of order $\mathcal{O}(\Delta x^{-2})$ in numerical simulations. But in standard implementations of MHD it is still at least of order $\mathcal{O}(\Delta x)$. This can be verified by the following considerations:

We restrict our analysis to the x -direction in a Cartesian grid. In most codes the choice of the parameter B_1 is done in dependence on its values in the cells next the cell face at which the flux has to be evaluated. Usually it is taken to be a weighted mean of these values. Thus, for the resulting full flux function we still have, if written for some one-dimensional situation, at the i -th interface,

$$G(\mathbf{q}_{i-l}, \dots, \mathbf{q}_{i+k}) \rightarrow F_1(\mathbf{q}), \quad \text{if } \mathbf{q}_{i+r} \rightarrow \mathbf{q} \quad \text{for } r = -l, \dots, k. \quad (96)$$

Hence, the flux function and, by applying the Lax–Wendroff theorem, the scheme itself is consistent. In smooth regions this implies an order of at least one. In addition, the error introduced to the antisymmetric condition (4) (when applied to \mathbf{G} instead of \mathbf{F}) is small. The actual order of such schemes can only be tested by measuring the experimental order of convergence (EOC). There is no direct control on the differences used. As a matter of experience, these schemes are most prone

to failure due to divergence errors. The schemes by Zachary, Malagoli and Colella [46] and Balbás and Tadmor [2], mentioned in the introduction, do not employ the assumption of one-dimensional physics at any place.

For our prototype system, the linearized induction Equation (21) in two space dimensions, the flux in x -direction is $(0, -vB_1 + uB_2)^T$. Let us assume that u is positive and we employ the upwind scheme. The flux-term uB_2 is always treated with upwind differences. If we take the parameter B_1 to be the value in the cell to the left of the cell face for which the numerical flux is to be computed, we end up with full upwinding, and, according to Theorem 3, find an exact discrete involution. If we take the value of B_1 from the cell to the right of the cell face, the flux-term $-vB_1$ is discretized with downwind differences. The conditions of Theorem 3 are not longer valid. If we define $\hat{\partial}/\hat{\partial}x$ to be the upwind difference operator and $\tilde{\partial}/\tilde{\partial}x$ to be the downwind operator, the actual discretization for the second flux component at a fixed grid point x_i reads

$$\begin{aligned} \frac{\hat{\partial}}{\hat{\partial}x}(uB_2)_i - \frac{\tilde{\partial}}{\tilde{\partial}x}(vB_1)_i &= \frac{\hat{\partial}}{\hat{\partial}x}(-vB_1 + uB_2)_i - \left(\frac{\hat{\partial}}{\hat{\partial}x} - \frac{\tilde{\partial}}{\tilde{\partial}x} \right) (vB_1)_i \\ &= \frac{\hat{\partial}}{\hat{\partial}x}(-vB_1 + uB_2)_i - v \frac{B_{1i+1} - 2B_{1i} + B_{1i-1}}{\Delta x} \\ &= \frac{\hat{\partial}}{\hat{\partial}x}(-vB_1 + uB_2)_i - v \Delta x (B_{1,xxi} + \mathcal{O}(\Delta x^2)). \end{aligned} \quad (97)$$

A similar consideration can be made for the y -direction. Summed up, the divergence error introduced in one time step is of order $\mathcal{O}_s(\Delta x)$, which means that Δx is in turn of the same order as the divergence error. If instead of the value to the right of the cell face, we take a weighted mean with weight α for that value, the error is just multiplied by α but still of the same order.

This is not too bad. Thus, the main reason for the problems arising from one-dimensional physics is the loss of control on the numerical viscosity on the resonant wave.

4.4. Numerical experiments. In this section, we present some numerical experiments⁵ for the De Sterck test with a Roe-type scheme without the assumption of one-dimensional physics. Analytically, the problem results in a steady state, which has been already reached at time $t = 0.8$. To study the long-term effects, we went on to time $t = 4.8$. The left half of Figure 6 gives a comparison of the scheme with and without entropy fix. As entropy fix, we employ the above mentioned Harten fix with parameter $\delta = 2|\lambda_{\max}|$ or $\delta = 2|u|$ for the resonant wave and $\delta = 10^{-8}$ for the other waves. As Figure 6 shows, the effects of the central viscosity introduced

⁵Numerical experiments in this paper are done with clawpack [29].

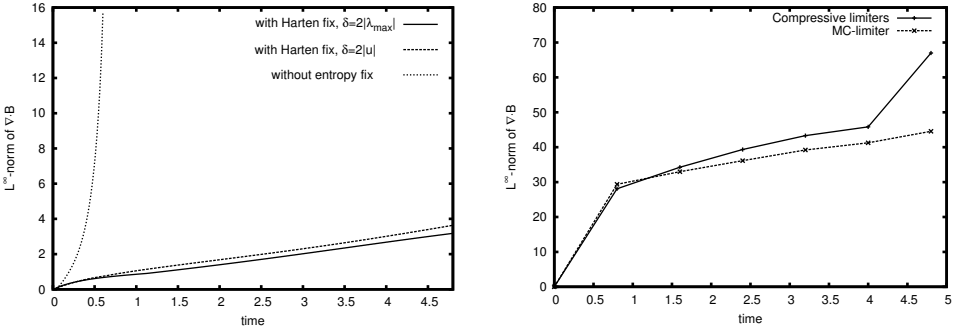


Figure 6. Left: Maximum norm of $\nabla \cdot (h\mathbf{B})$ over time for the De Sterck test problem with and without Harten fix for first-order computation on 10×10 grid. Right: Maximum norm of $\nabla \cdot (h\mathbf{B})$ for second order with Harten fix on 200×200 grid with standard MC limiter and highly compressive limiters.

by the entropy fix are strong. While the computation without the fix does not even reach the steady state, the computation with the fix survives the whole simulation without the need of an intermediate projection step. The choice $\delta = 2|u|$ is weaker, but still yields reasonable results.

The right half of Figure 6 demonstrates the influence of the limiter on the stability. Although the limiter does not change anything on the resonant wave itself, since it propagates with zero speed, the choice of limiters for the other waves show some effect. For short times, the more compressive limiters, see [26], yield better results. But the unphysical forces arising from the divergence errors are much better resolved. The better resolution of discontinuities results in steeper gradients and, thus, in higher divergence errors. In the long-term run, the error exceeds the error obtained with the classical MC limiter.

Next, we investigate the influence of the grid resolution and the order of the scheme. On the one hand, a higher grid resolution and a higher order would, by the

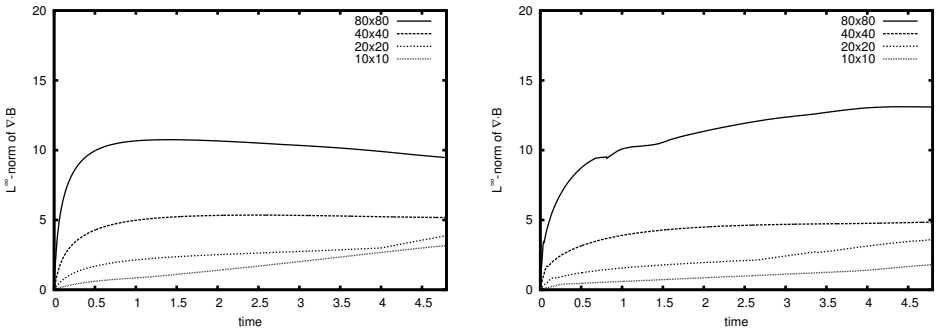


Figure 7. Influence of the grid resolution on the divergence error. Left: first order; right: second order.

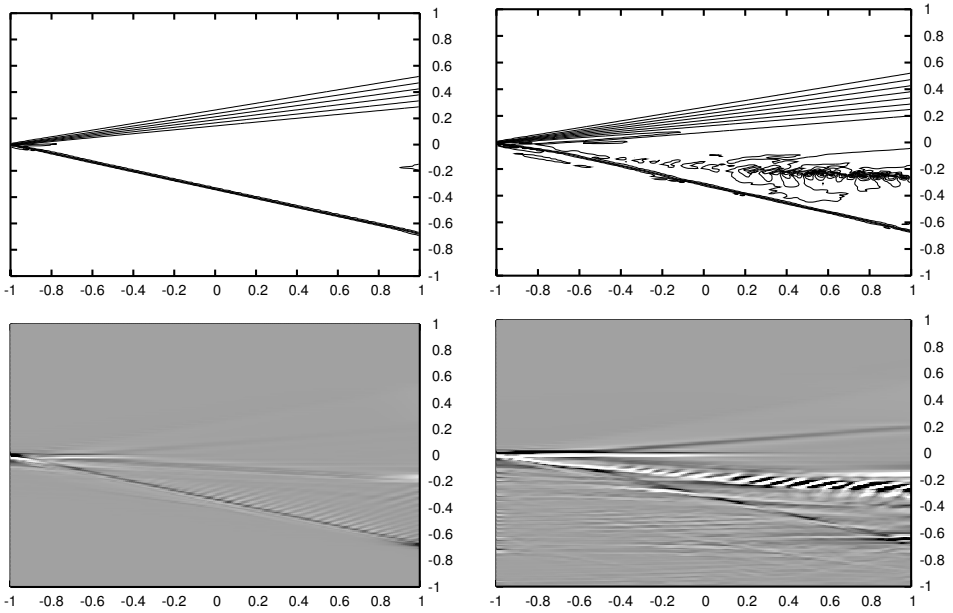


Figure 8. De Sterck test with Harten fix on 200×200 grid for second-order computation with highly compressive limiters at time $t = 0.8$ (left) and $t = 4.8$ right. Upper row: height; lower row: divergence.

estimate (50) in Theorem 2, we would expect a positive effect. But on the other hand, the higher resolution and the higher order lower the numerical viscosity and allow for steeper gradients and, thus, for higher divergence errors near shocks. As Figure 7 shows, the second argument dominates for the grid resolution. On a fixed grid, the higher-order scheme performs better.

Finally, Figure 8 presents results of highly resolved computations, 200×200 grid cells, with the second-order schemes. The basic structure of the solution is preserved even for the long-term run. But the divergence errors have infected all of the lower half of the computational domain. At the places with the highest divergence errors, disturbances of the solution can be seen in the contour plot of the height. The computations with the high resolving limiters in Figure 8 show an area with severe destruction of the solution. With the MC limiter, this effect is weaker.

The situation is the same as for the schemes by Balbás and Tadmor [2; 1] and Zachary, Malagoli, and Colella [46]. It is still reasonable to employ some sort of divergence cleaning. But one can resort to a weaker one. In the case of a projection to a divergence-free field, the time interval between two projections can be considerably increased, since the computation is still stable. In a scheme based on hyperbolic or mixed type GLM divergence cleaning [12], the divergence errors

which have to be transported out of—and thus through a significant part of—the computational domain are much smaller.

5. Conclusions and outlook

In this study, we investigated the origin of divergence errors in MHD simulations. The concept of involutions, introduced by Dafermos [10; 9], turned out to be the key of understanding of the issue. Especially when, like in MHD, the involutions are closely related to resonance, their exact reproduction in the discrete case is needed to prevent the numerical schemes from failing due to unphysical forces. If an involution satisfies Dafermos' sufficient condition (4), discrete analogues of Theorem 1 give quantitative information on the possible errors. For some linear schemes, the discrete involutions are even exact. The introduction of central viscosity in the scheme provides a tool to reduce resonant effects. It turns the discrete involution into a parabolic equation, which damps the involution and, for example in the case of MHD, the resonance. But this only works if for the computation of the intercell fluxes the full multidimensional physics is taken into account. If the intercell fluxes are computed with the assumption of one-dimensional physics, in addition to not explicitly resolving the resonant wave, we completely neglect it. The resulting central viscosity cannot be controlled and, thus, be even of the wrong sign. There is simply no possibility to control it. Employing fluxes with full physics, as in the Balbás–Tadmor scheme [1] and the Zachary–Malagoli–Colella scheme [46], considerably stabilizes the scheme. In Roe-type schemes, we can explicitly tune the amount of central viscosity introduced by the flux function. If we employ the maximal admissible amount of viscosity on the resonant wave, the scheme is stable even for very long runs. Due to the disturbances of the solution, which are caused by the growing divergence errors, it is still reasonable to employ some sort of divergence cleaning. But one can resort to a weaker one. In the case of a projection to a divergence-free field, the time interval between two projections can be considerably increased. The computation is still stable. In a GLM scheme [12], the disturbances introduced by the transport of divergence errors through the computational domain are minimized.

In summary, divergence errors in MHD are mainly caused by resonance and a lack of positive central viscosity in the applied numerical scheme; the latter most often results from the assumption of one-dimensional physics in the calculation of intercell fluxes.

Acknowledgements

I would like to thank Prof. G. Bader, whose probing questions gave rise to this study. Many thanks also to the participants of the Eighth Hirschegg Workshop

on Conservation Laws, especially to Manuel Torrilhon, for the deep inspiring discussions on the subject. I am also grateful to Felix Rieper for many interesting discussions and the proofreading of the manuscripts.

References

- [1] J. Balbás, 2008, personal communication at the 12th Conference on Hyperbolic Problems, University of Maryland.
- [2] J. Balbás and E. Tadmor, *Nonoscillatory central schemes for one- and two-dimensional magneto-hydrodynamics equations, II: High-order semidiscrete schemes*, SIAM J. Sci. Comput. **28** (2006), no. 2, 533–560. [MR 2007a:65116](#)
- [3] D. S. Balsara, *Divergence-free adaptive mesh refinement for magnetohydrodynamics*, J. Comput. Phys. **174** (2001), 614–648.
- [4] D. S. Balsara and D. S. Spicer, *A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations*, J. Comput. Phys. **149** (1999), no. 2, 270–292. [MR 99j:76080](#) [Zbl 0936.76051](#)
- [5] N. Besse and D. Kröner, *Convergence of locally divergence-free discontinuous-Galerkin methods for the induction equations of the 2D-MHD system*, M2AN Math. Model. Numer. Anal. **39** (2005), no. 6, 1177–1202. [MR 2006k:65262](#) [Zbl 1084.76046](#)
- [6] J. U. Brackbill and D. C. Barnes, *The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations*, J. Comput. Phys. **35** (1980), no. 3, 426–430. [MR 81f:65068](#) [Zbl 0429.76079](#)
- [7] M. Breuss, *The correct use of the Lax–Friedrichs method*, M2AN Math. Model. Numer. Anal. **38** (2004), no. 3, 519–540. [MR 2006e:65137](#) [Zbl 1077.65089](#)
- [8] R. K. Crockett, P. Colella, R. T. Fisher, R. J. Klein, and C. I. McKee, *An unsplit, cell-centered Godunov method for ideal MHD*, J. Comput. Phys. **203** (2005), no. 2, 422–448. [MR 2005j:76065](#) [Zbl 1143.76599](#)
- [9] C. M. Dafermos, *Quasilinear hyperbolic systems with involutions*, Arch. Rational Mech. Anal. **94** (1986), no. 4, 373–389. [MR 87h:35204](#) [Zbl 0614.35057](#)
- [10] ———, *Hyperbolic conservation laws in continuum physics*, Grundlehren der Math. Wissenschaften, no. 325, Springer, Berlin, 2000. [MR 2001m:35212](#) [Zbl 0940.35002](#)
- [11] H. De Sterck, *Multi-dimensional upwind constrained transport on unstructured grids for “shallow water” magnetohydrodynamics*, 15th AIAA Computational Fluid Dynamics Conference, AIAA, no. 2001-2623, 2001.
- [12] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, and M. Wesenberg, *Hyperbolic divergence cleaning for the MHD equations*, J. Comput. Phys. **175** (2002), no. 2, 645–673. [MR 2002k:76139](#) [Zbl 1059.76040](#)
- [13] B. Einfeldt, *On Godunov-type methods for gas dynamics*, SIAM J. Numer. Anal. **25** (1988), no. 2, 294–318. [MR 89e:65086](#) [Zbl 0642.76088](#)
- [14] C. R. Evans and J. F. Hawley, *Simulation of general relativistic magnetohydrodynamic flows: A constrained transport method*, Astrophys. J. **332** (1988), 659–677.
- [15] F. G. Fuchs, S. Mishra, and N. H. Risebro, *Splitting based finite volume schemes for ideal MHD equations*, J. Comput. Phys. **228** (2009), no. 3, 641–660. [MR 2010c:76087](#) [Zbl 05506587](#)
- [16] F. G. Fuchs, K. H. Karlsen, S. Mishra, and N. H. Risebro, *Stable upwind schemes for the magnetic induction equation*, M2AN Math. Model. Numer. Anal. **43** (2009), no. 5, 825–852. [MR 2010i:65150](#) [Zbl 1177.78057](#)

- [17] P. A. Gilman, *Magnetohydrodynamic “shallow water” equations for the solar tachocline*, *Astrophys. J. Lett.* **544** (2000), no. 2, L79.
- [18] S. K. Godunov, *Non-unique “blurrings” of discontinuities in solutions of quasilinear systems*, *Dokl. Akad. Nauk SSSR* **2** (1961), 43–44, in Russian; translated in *Sov. Math. Dokl.* **2** (1961), 947–949. [MR 22 #6936](#)
- [19] ———, *The problem of a generalized solution in the theory of quasi-linear equations and in gas dynamics*, *Uspehi Mat. Nauk* **17** (1962), no. 3, 147–158, in Russian; translated in *Russ. Math. Surv.* **17** (1962), no. 3, 145–156. [MR 27 #5445](#) [Zbl 0107.20003](#)
- [20] ———, *Symmetric form of the magnetohydrodynamic equation*, *Chislennye Metody Mekh. Sploshnoi Sredy* **3** (1972), no. 1, 26–34, in Russian.
- [21] A. Harten, *High resolution schemes for hyperbolic conservation laws*, *J. Comput. Phys.* **49** (1983), no. 3, 357–393. [MR 84g:65115](#) [Zbl 0565.65050](#)
- [22] A. Harten and J. M. Hyman, *Self-adjusting grid methods for one-dimensional hyperbolic conservation laws*, *J. Comput. Phys.* **50** (1983), no. 2, 235–269. [MR 85g:65111](#) [Zbl 0565.65049](#)
- [23] A. Harten, P. D. Lax, and B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, *SIAM Rev.* **25** (1983), no. 1, 35–61. [MR 85h:65188](#) [Zbl 0565.65051](#)
- [24] F. Kemm, *A carbuncle free Roe-type solver for the Euler equations*, *Hyperbolic problems: theory, numerics, applications* (S. Benzoni-Gavage et al., eds.), Springer, Berlin, 2008, pp. 601–608. [MR 2549194](#) [Zbl 1138.65072](#)
- [25] ———, *Discrete involutions, resonance, and the divergence problem in MHD*, *Hyperbolic problems: theory, numerics and applications* (E. Tadmor, J.-G. Liu, and A. E. Tzavaras, eds.), *Proc. Sympos. Appl. Math.*, no. 67, Amer. Math. Soc., Providence, RI, 2009, pp. 725–735. [MR 2011b:35406](#) [Zbl 1191.35174](#)
- [26] ———, *A comparative study of TVD-limiters—well-known limiters and an introduction of new ones*, *Internat. J. Numer. Methods Fluids* **67** (2011), no. 4, 404–440. [MR 2012h:65178](#) [Zbl 05975627](#)
- [27] F. Kemm, Y.-J. Lee, C.-D. Munz, and R. Schneider, *Divergence cleaning in finite-volume computations for electromagnetic wave propagations*, *Finite volumes for complex applications, III* (R. Herbin and D. Kröner, eds.), Hermes, Paris, 2002, pp. 561–568. [MR 2008982](#)
- [28] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge University Press, Cambridge, 2002. [MR 2003h:65001](#) [Zbl 1010.65040](#)
- [29] R. J. LeVeque et al., *Clawpack (conservation laws package)*, software.
- [30] B. Marder, *A method incorporating Gauss’ law into electromagnetic pic codes*, *J. Comput. Phys.* **68** (1987), 48–55.
- [31] S. Mishra and E. Tadmor, *Constraint preserving schemes using potential-based fluxes, I: Multidimensional transport equations*, *Commun. Comput. Phys.* **9** (2011), no. 3, 688–710. [MR 2011m:65203](#)
- [32] ———, *Constraint preserving schemes using potential-based fluxes, II: Genuinely multidimensional systems of conservation laws*, *SIAM J. Numer. Anal.* **49** (2011), no. 3, 1023–1045. [MR 2012h:65161](#)
- [33] ———, *Constraint preserving schemes using potential-based fluxes, III: Genuinely multidimensional schemes for the MHD equations*, *ESAIM Math. Model. Numer. Anal.* **46** (2012), 661–680. [MR 2877370](#)

- [34] C.-D. Munz, P. Omnes, R. Schneider, E. Sonnendrücker, and U. Voß, *Divergence correction techniques for Maxwell solvers based on a hyperbolic model*, J. Comput. Phys. **161** (2000), no. 2, 484–511. MR 2001c:78034 Zbl 0970.78010
- [35] C.-D. Munz, R. Schneider, and U. Voß, *A finite-volume method for the Maxwell equations in the time domain*, SIAM J. Sci. Comput. **22** (2000), no. 2, 449–475. MR 2001d:78033 Zbl 1039.78012
- [36] C.-D. Munz, R. Schneider, E. Sonnendrücker, and U. Voss, *Maxwell's equations when the charge conservation is not satisfied*, C. R. Acad. Sci. Paris Sér. I Math. **328** (1999), no. 5, 431–436. MR 99k:35170 Zbl 0937.78005
- [37] H. Nessyahu and E. Tadmor, *Nonoscillatory central differencing for hyperbolic conservation laws*, J. Comput. Phys. **87** (1990), no. 2, 408–463. MR 91i:65157
- [38] K. G. Powell, P. L. Roe, R. S. Myong, T. Gombosi, and D. de Zeeuw, *An upwind scheme for magnetohydrodynamics*, Workshop Méthodes numériques pour la M.H.D.
- [39] K. G. Powell, P. L. Roe, T. J. Linde, T. I. Gombosi, and D. L. De Zeeuw, *A solution-adaptive upwind scheme for ideal magnetohydrodynamics*, J. Comput. Phys. **154** (1999), no. 2, 284–309. MR 2000e:76094 Zbl 0952.76045
- [40] J. A. Rossmannith, *An unstaggered, high-resolution constrained transport method for magnetohydrodynamic flows*, SIAM J. Sci. Comput. **28** (2006), no. 5, 1766–1797. MR 2008d:76085 Zbl 05194927
- [41] M. Torrilhon and M. Fey, *Constraint-preserving upwind methods for multidimensional advection equations*, SIAM J. Numer. Anal. **42** (2004), no. 4, 1694–1728. MR 2005j:65090 Zbl 1146.76621
- [42] M. Torrilhon, *Zur Numerik der idealen Magnetohydrodynamik*, Ph.D. thesis, ETH Zürich, 2003.
- [43] ———, *Locally divergence-preserving upwind finite volume schemes for magnetohydrodynamic equations*, SIAM J. Sci. Comput. **26** (2005), no. 4, 1166–1191. MR 2006f:76054 Zbl 1149.76693
- [44] G. Tóth, *The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes*, J. Comput. Phys. **161** (2000), no. 2, 605–652. MR 2001a:76151 Zbl 0980.76051
- [45] K. Waagan, *A positive MUSCL-Hancock scheme for ideal magnetohydrodynamics*, J. Comput. Phys. **228** (2009), no. 23, 8609–8626. MR 2010m:76159 Zbl 05634510
- [46] A. L. Zachary, A. Malagoli, and P. Colella, *A higher-order Godunov method for multidimensional ideal magnetohydrodynamics*, SIAM J. Sci. Comput. **15** (1994), no. 2, 263–284. MR 95d:76081 Zbl 0797.76063

Received October 20, 2010. Revised May 14, 2012.

FRIEDEMANN KEMM: kemm@math.tu-cottbus.de

Institute for Applied Mathematics and Scientific Computing, Brandenburg University of Technology Cottbus, Platz der Deutschen Einheit 1, D-03046 Cottbus, Germany

RENORMALIZED REDUCED MODELS FOR SINGULAR PDES

PANOS STINIS

We present a novel way of constructing reduced models for systems of ordinary differential equations. In particular, the approach combines the concepts of renormalization and effective field theory developed in the context of high energy physics and the Mori–Zwanzig formalism of irreversible statistical mechanics. The reduced models we construct depend on coefficients which measure the importance of the different terms appearing in the model and need to be estimated. The proposed approach allows the estimation of these coefficients on the fly by enforcing the equality of integral quantities of the solution as computed from the original system and the reduced model. In this way we are able to construct stable reduced models of higher order than was previously possible. The method is applied to the problem of computing reduced models for ordinary differential equation systems resulting from Fourier expansions of singular (or near-singular) time-dependent partial differential equations. Results for the 1D Burgers and the 3D incompressible Euler equations are used to illustrate the construction. Under suitable assumptions, one can calculate the higher order terms by a simple and efficient recursive algorithm.

1. Introduction

Spatial discretizations or Fourier expansions of the solutions of time-dependent partial differential equations (PDEs) lead to systems of ordinary differential equations (ODEs). The most difficult case arises when the solution of a PDE becomes singular in finite time. At such instants the solution of the PDE develops activity down to the zero length scale. A brute force numerical simulation (no matter how large) of such a solution is bound to fail because the simulation has a finite resolution and thus will be unable to resolve all the length scales down to the zero scale. When the solution develops activity at a scale smaller than the smallest scale available to the simulation, the numerically computed solution becomes underresolved. This leads to a rapid deterioration of the accuracy of the simulation.

The notion of propagation of activity to smaller and smaller scales depends on the physical context of the PDE. In some cases, like the 3D Euler or Navier–Stokes

MSC2010: 65M99, 35B44, 35D30.

Keywords: model reduction, Mori–Zwanzig, renormalization, singularity, partial differential equations.

equations [14], this could mean a cascade of energy to smaller and smaller scales. In other cases, such as the nonlinear focusing Schrödinger equation [24], this could mean a cascade of mass to smaller and smaller scales. Irrespective of the specific physical context, the problem facing the numerical analyst is how to use a *finite* simulation and yet prevent the computed solution from suffering a loss of accuracy. In other words, how to construct a numerical method which reproduces correctly the features of the solution of the original equation at the length scales that are available numerically. This is the motivation behind the construction of reduced models (see [16; 10], for example).

By construction, a reduced model must allow for energy (mass) to escape from the scales that are accessible to the simulation (called resolved scales or modes) to the inaccessible scales (called unresolved). The main difficulty in constructing an accurate reduced model is the need to estimate the *correct* rate at which activity is propagated from the resolved to the unresolved scales. The Mori–Zwanzig (MZ) formalism [8; 9] proceeds by dividing the available resolution into resolved and unresolved parts. Then, it constructs a reduced model for the resolved scales and uses the unresolved scales to effect the drain of energy (mass) out of the resolved scales.

Although the MZ formalism allows for the construction, in principle, of an exact reduced model it has two drawbacks (which are also shared by *any* other reduction formalism). First, the reduced model can be, in general, prohibitively expensive to calculate. The reason is that one must obtain an accurate representation of the behavior of the unresolved scales before they can be safely eliminated. Obtaining this representation can be rather costly.

The second drawback is more subtle and has not been adequately appreciated by the scientific computing community. It is specific to the case of constructing reduced models for singular PDEs or in general for systems of ordinary differential equations which are larger than any available numerical resolution. Suppose that you have to construct a reduced model of a full system which is larger than any available numerical simulation. Let us call this system S1. Exactly because S1 is larger than any available numerical simulation, if we want to construct a reduced model we have to use as a starting point a system, call it S2, whose size is smaller than the size of S1. Suppose that you start with S2 and use the MZ formalism (or any other reduction formalism for that matter) and construct an *exact* reduced model S3 for a subset of S2. An exact reduced model means that if one evolves S2 and S3 separately, then the behavior of the scales resolved by S3 will be the same as the behavior of the scales in S3 predicted by the simulation of the system S2. However, and this is the heart of the problem, since S2 itself will become eventually underresolved, the exact reduced model S3 will also become underresolved. In other words, the predictions of the exact reduced model S3 can only be trusted for as long as the predictions of the system S2 can be trusted. As a result, *any reduced*

model that has any chance of being accurate for longer times cannot be exact.

There are examples of *inexact* reduced models, such as the t -model [9; 3; 20], coming from the MZ formalism, which have been applied to singular PDEs and shown numerically to be relatively accurate for long time intervals. However, the t -model’s accuracy is difficult to assess beforehand and the reason for its relative success has remained a mystery (see also [6] for an application to the 3D Navier–Stokes equations which shows that the t -model, while not bad, is in need of some modification). In order to construct better reduced models we need to incorporate dynamic information from the full system which will help us decide which of the terms appearing in the exact reduced model are the ones that are most important. In this way, we can construct an *inexact* but accurate reduced model by keeping the important terms and disregarding the unimportant ones.

The way we propose to address the problem of constructing better reduced models is to embed the MZ reduced models in a larger class of reduced models which share the same functional form as the MZ reduced models but have different coefficients in front of the various terms that appear in the reduced models. Then, one can estimate these coefficients on the fly while the original system of equations is still valid. The estimation of the coefficients is achieved by requiring that certain integral quantities (e.g., L_p norms) involving only resolved scales, should acquire the same values when computed from the original system and the reduced model. Before the original system ceases to be valid, one reverts to the reduced model with the various coefficients having their estimated values. We call the proposed approach the renormalized Mori–Zwanzig (rMZ) algorithm. Note that the constraints used to obtain the coefficients are the analog of the “matching conditions” used in effective field theory [15]. Also, the approach is the time-dependent analog of the process of renormalization used in high energy and condensed matter physics [11; 17].

A special case of the proposed method which utilized only the t -model term was first presented by the author in [23]. The goal there was to construct a mesh refinement scheme to allow us to reach the singularity instant more efficiently. For that purpose the use of only the t -model term was adequate. In the current work, we not only want to reach the singularity instant but also follow the solution accurately for later times. This requires the use of higher order terms than the t -model term. Under suitable assumptions (see Section 2.3.2) we are able to calculate recursively and efficiently (and with minimal storage requirements) the higher order terms (see also Sections 3.1 and 3.2).

It is interesting to see to what extent the values (or at least the form) of the renormalized coefficients for the reduced model can be deduced from analytical considerations. In Section 3.4 we include some numerical results which hint that the value of the renormalized coefficients depends on the structure of the initial condition and the scaling symmetries of the PDE.

2. Renormalization of Mori–Zwanzig reduced models

In [Section 2.1](#) we set up the notation for the original system and the reduced model in an abstract way which does not make reference to any specific method for obtaining the reduced model. In [Section 2.2](#) we show how to obtain the coefficients for the reduced model. In [Section 2.3](#) we give a brief presentation of the MZ formalism which allows us to obtain the functional form of the terms appearing in the reduced model. In [Section 2.4](#) we combine the ideas in [Section 2.2](#) with the MZ formalism from [Section 2.3](#) to derive the proposed algorithm for computing renormalized MZ reduced models.

2.1. Full and reduced systems. Suppose that we want to construct a reduced model for the partial differential equation (PDE)

$$v_t + H(t, x, v, v_x, \dots) = 0,$$

where H is a operator (in general nonlinear) and $x \in D \subseteq \mathbb{R}^d$ (the construction extends readily to the case of systems of partial differential equations). After spatial discretization or expansion of the solution in series, the PDE transforms into a system of ordinary differential equations (ODEs). For simplicity we restrict ourselves to the case of periodic boundary conditions, so that a Fourier expansion of the solution leads to system of ODEs for the Fourier coefficients. To simulate the system for the Fourier coefficients we need to truncate at some point the Fourier expansion. Let $F \cup G$ denote the set of Fourier modes retained in the series, where we have split the Fourier modes in two sets, F and G . We call the modes in F resolved and the modes in G unresolved. The reduced model involving only the resolved modes F will be called the reduced system and the system involving both the resolved *and* unresolved modes $F \cup G$ will be called the full system.

The main idea behind the algorithm is that the evolution of moments of the reduced set of modes, for example l_p norms of the modes in F , should be the same whether computed from the full or the reduced system. This requirement will eventually allow us to compute the coefficients appearing in the reduced model (see [Section 2.2](#)).

The full system of equations for the modes $F \cup G$ is given by

$$\frac{du(t)}{dt} = R(t, u(t)),$$

where $u = (\{u_k\})$, $k \in F \cup G$ is the vector of Fourier coefficients of u and R is the Fourier transform of the operator H . The system should be supplemented with an initial condition $u(0) = u_0$. The vector of Fourier coefficients can be written as $u = (\hat{u}, \tilde{u})$, where \hat{u} are the resolved modes (those in F) and \tilde{u} the unresolved ones (those in G). Similarly, for the right-hand sides (RHS) we have

$R(t, u) = (\hat{R}(t, u), \tilde{R}(t, u))$. Note that the RHS of the resolved modes involves both resolved and unresolved modes. In anticipation of the construction of a reduced model we can rewrite the RHS as $R(t, u) = R^{(0)}(t, u) = (\hat{R}^{(0)}(t, u), \tilde{R}^{(0)}(t, u))$.

In general, when one constructs a reduced model, additional terms appear on the RHS of the equations of the reduced model (see [Section 2.3](#) for more details). The role of these additional terms is to account for the interactions between the resolved and unresolved modes, since the unresolved modes no longer appear explicitly in the reduced model. As is standard in renormalization theory [4], one can augment the RHS of the equations in the full system by including such additional terms. That is accomplished by multiplying each of these additional terms by a zero coefficient. In this way, the reduced and full systems' RHSs have the same functional form. In particular, for each mode u_k , $k \in F \cup G$, we can rewrite $R_k^{(0)}(t, u)$ as

$$R_k^{(0)}(t, u(t)) = \sum_{i=1}^m a_i^{(0)} R_{ik}^{(0)}(t, u(t)),$$

where $R_{1k}^{(0)}(t, u(t)) = R_k^{(0)}(t, u(t))$ and the $R_{ik}^{(0)}(t, u(t))$, for $i = 2, \dots, m$ are of the same functional form as the additional terms which appear in the reduced model. This is easy to do by taking $a_1^{(0)} = 1$ and $a_i^{(0)} = 0$, for $i = 2, \dots, m$. Thus, the equation for the mode u_k , $k \in F \cup G$ is written as

$$\frac{du_k(t)}{dt} = R_k(t, u) = R_k^{(0)}(t, u(t)) = \sum_{i=1}^m a_i^{(0)} R_{ik}^{(0)}(t, u(t)) \quad (1)$$

Correspondingly, the reduced model for the mode u'_k , $k \in F$, is given by

$$\frac{du'_k(t)}{dt} = R_k^{(1)}(t, \hat{u}'(t)) = \sum_{i=1}^m a_i^{(1)} R_{ik}^{(1)}(t, \hat{u}'(t)) \quad (2)$$

with initial condition $u'_k(0) = u_{0k}$.

Define m quantities \hat{E}_i , $i = 1, \dots, m$ involving only modes in F . For example, these could be l_p norms of the reduced set of modes. To proceed we require that these quantities' rates of change are the same when computed from (1) and (2):

$$\frac{d\hat{E}_i(\hat{u})}{dt} = \frac{d\hat{E}_i(\hat{u}')}{dt}, \quad i = 1, \dots, m. \quad (3)$$

Similar conditions, albeit time-independent, lie at the heart of the renormalization group theory for equilibrium systems [4, p. 154]. Also, the conditions (3) are the analog of the ‘‘matching conditions’’ underlying the construction of effective field theories [15].

2.2. How to compute the coefficients of the reduced model. When we only know the functional form of the terms appearing in the reduced model but not their coefficients it is not possible to evolve a reduced system. We present a way of actually computing the coefficients of the reduced model as needed. If the quantities \hat{E}_i , $i = 1, \dots, m$ are, for example, l_p norms of the Fourier modes, then we can multiply Equations (2) with appropriate quantities and combine with Equations (3) to get

$$\frac{d\hat{E}_i(\hat{u})}{dt} = \sum_{j=1}^m a_j^{(1)} B_{ij}(t, \hat{u}(t)),$$

where

$$B_{ij} = \frac{\partial}{\partial a_j^{(1)}} \frac{d\hat{E}_i(\hat{u}')}{dt}, \quad i, j = 1, \dots, m$$

are the new RHS functions that appear. Note that the RHS of the equations above does not involve primed quantities. The reason is that here the reduced quantities are computed by using the values of the resolved modes from the full system. The above system of equations is a linear system of equations for the vector of coefficients $a^{(1)}$. The linear system can be written as

$$B a^{(1)} = \mathbf{e}, \quad (4)$$

where $\mathbf{e} = (d\hat{E}_1(\hat{u})/dt, \dots, d\hat{E}_m(\hat{u})/dt)$. This system of equations can provide us with the time evolution of the vector $a^{(1)}$.

The determination of coefficients for the reduced model through the system (4) is a time-dependent version of the method of moments. We specify the coefficients of the reduced model so that the reduced model reproduces the rates of change of a finite number of moments of the solution of the original system. This ensures that each term in the model is properly weighted so that the resulting reduced model reproduces, at the scales accessible to the reduced model, the dynamics (see (3)) of the original system.

By construction, the entry B_{ij} , $i, j = 1, \dots, m$, of the matrix B measures the contribution of the j -th term of the reduced model to the rate of change of \hat{E}_i . In fact, the j -th column of the matrix B is comprised of all the contributions of the j -th term in the reduced model to the rates of change of the different \hat{E}_i . While the reduced system has no need to transfer activity from the resolved to the unresolved scales, the columns of B corresponding to the activity-transferring terms will be zero (to the numerical precision used). Thus, the matrix B will be singular. This can be monitored by estimating the rank of the matrix through the Singular Value Decomposition (SVD) [18]. When the smallest singular value becomes nonzero for the numerical precision used the reduced system starts transferring activity to

the unresolved scales. After that instant we can use the system (4) to estimate the coefficient vector $a^{(1)}$ (see [23] for more details).

2.3. The Mori–Zwanzig formalism. We have presented in the previous section an abstract way of writing the reduced system which does not make any reference to a specific method for obtaining the functions $R_k^{(1)}(t, \hat{u}'(t))$, $k \in F$, appearing on the RHS of (2). In order to proceed we need to specify the functions $R_k^{(1)}(t, \hat{u}'(t))$. We will do that through the Mori–Zwanzig formalism [8; 9].

Suppose we are given the full system

$$\frac{du(t)}{dt} = R(t, u(t)), \quad (5)$$

where $u = (\{u_k\})$, $k \in F \cup G$ with initial condition $u(0) = u_0$. The system of ordinary differential equations we are asked to solve can be transformed into a system of linear partial differential equations

$$\frac{\partial \phi_k}{\partial t} = L\phi_k, \quad \phi_k(u_0, 0) = u_{0k}, \quad k \in F \cup G, \quad (6)$$

where $L = \sum_{k \in F \cup G} R_k(u_0) \partial / \partial u_{0k}$. The solution of (6) is given by $u_k(u_0, t) = \phi_k(u_0, t)$. Using semigroup notation we can rewrite (6) as

$$\frac{\partial}{\partial t} e^{tL} u_{0k} = L e^{tL} u_{0k}$$

Suppose that the vector of initial conditions can be divided as $u_0 = (\hat{u}_0, \tilde{u}_0)$, where \hat{u}_0 is the vector of the resolved variables and \tilde{u}_0 is the vector of the unresolved variables. Let P be an orthogonal projection on the space of functions of \hat{u}_0 and $Q = I - P$.

Equation (6) can be rewritten as

$$\frac{\partial}{\partial t} e^{tL} u_{0k} = e^{tL} P L u_{0k} + e^{tQL} Q L u_{0k} + \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds, \quad k \in F, \quad (7)$$

where we have used Dyson's formula

$$e^{tL} = e^{tQL} + \int_0^t e^{(t-s)L} P L e^{sQL} ds. \quad (8)$$

Equation (7) is the Mori–Zwanzig identity. Note that this relation is exact and is an alternative way of writing the original PDE. The first term in (7) is usually called Markovian since it depends only on the values of the variables at the current instant, the second is called “noise” and the third “memory”. Note that $P e^{tQL} Q L u_{0k} = 0$ and the operator e^{tQL} is called the orthogonal dynamics operator [8].

We can project the Mori–Zwanzig equation (7) and find

$$\frac{\partial}{\partial t} P e^{tL} u_{0k} = P e^{tL} P L u_{0k} + P \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds. \quad (9)$$

In order to proceed we need to compute the Markovian term and the memory term. For the specific projection P we will be using the Markovian term is straightforward to compute (see the definition of the operator P in Sections 3.1 and 3.2). On the other hand, the memory term computation is rather involved due to the presence of the evolution operator e^{tQL} . In fact, it is the presence of this operator which makes, in general, the computation of MZ reduced models prohibitively expensive (see [10] for a thorough discussion). One can start from (9) and based on assumptions derive simplified reduced models that are easier to calculate [9; 3; 20; 22].

The memory term integrand in (9) contains two operators evolving on their own time scales. The full dynamics operator $e^{(t-s)L}$ evolving on a time scale τ_f and the orthogonal dynamics operator e^{sQL} evolving on the time-scale τ_o . There are three major cases: $\tau_f \gg \tau_o$, $\tau_f \sim \tau_o$, and $\tau_f \ll \tau_o$. The first and last correspond to very short and very long memory respectively. The case of $\tau_f \sim \tau_o$ corresponds to absence of time-scale separation between the full dynamics and the orthogonal dynamics. For the problem of constructing reduced models for singular PDEs, it is plausible to assume absence of time-scale separation between the resolved and unresolved variables and thus we expect this case to be of relevance.

If we assume that $\tau_f \sim \tau_o$ and that both $e^{(t-s)L}$ and e^{sQL} are analytic, we can expand the expression $e^{(t-s)L} P L e^{sQL}$ in Taylor series around $s = 0$. We have

$$\begin{aligned} P \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds \\ = t P e^{tL} P L Q L u_{0k} + \frac{1}{2} t^2 P e^{tL} (P L Q L Q L u_{0k} - L P L Q L u_{0k}) \\ + \frac{1}{6} t^3 P e^{tL} (L^2 P L Q L u_{0k} - 2 L P L Q L Q L u_{0k} \\ + P L Q L Q L Q L u_{0k}) + O(t^4). \end{aligned} \quad (10)$$

The terms in the Taylor expansion of $e^{(t-s)L} P L e^{sQL}$ beyond the first order (in t) involve *both* resolved and unresolved variables. In order to construct a reduced model which is closed in the resolved variables these terms need to be modified while retaining the order of accuracy of the model (a way to achieve that is presented in Section 2.3.3). However, there is a special case for which all the terms in (10) are closed in the resolved variables. The simplification, if possible, is due to the small value of the commutator $[PL, QL]$ (see Section 2.3.2).

2.3.1. The commutative case. If $[PL, QL] = 0$ the only term that remains is the Markovian one. This can be seen by observing that $[PL, QL] = 0$ implies that

$[L, QL] = 0$. We have

$$\begin{aligned} \frac{\partial}{\partial t} P e^{tL} u_{0k} &= P L e^{tL} u_{0k} = P e^{tL} L u_{0k} = P e^{tL} P L u_{0k} + P e^{tL} Q L u_{0k} \\ &= P e^{tL} P L u_{0k} + P Q L e^{tL} u_{0k} = P e^{tL} P L u_{0k}, \end{aligned}$$

where in the last equation we have used the fact that $PQ = 0$.

2.3.2. The almost commutative case. We examine the case when $[PL, QL]$ is small. To see how this affects the computation of the memory term, we proceed by rewriting the expression for the memory $\int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds$. Through Dyson's formula (8) and the linearity of e^{tL} the memory term can be written as

$$\int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds = e^{tL} (Q L u_{0k} - e^{-tL} e^{tQL} Q L u_{0k})$$

By the identity $I = P + Q$ and the Baker–Campbell–Hausdorff (BCH) series for $e^{-tL} e^{tQL}$ (see [2], for instance), the above equation can be rewritten as

$$\int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds = e^{tL} (Q L u_{0k} - e^{C(t, u_0)} Q L u_{0k}), \quad (11)$$

where $C(t, u_0) = -tPL + \frac{1}{2}[-tL, tQL] + \dots$ with all the higher terms involving the commutator $[-tL, tQL] = -tL tQL - tQL(-tL)$. Note that we also have $[-tL, tQL] = [tL, tPL] = [tQL, tPL] = -[tPL, tQL]$. Thus

$$C(t, u_0) = -tPL - \frac{1}{2}[tPL, tQL] + \dots$$

We want to examine when the approximation $C(t, u_0) \approx -tPL$ is acceptable. From the BCH series we have

$$e^{-tL} e^{tQL} - e^{-tPL} = -\frac{1}{2}[tPL, tQL] + O(t^3). \quad (12)$$

Depending on the initial conditions, $[PL, QL]$ may be small and thus allow the simplification of the memory term expression. In Section 3, where we present numerical results for the 1D Burgers and 3D Euler equations, we comment briefly on the form of initial conditions that make the commutator $[PL, QL]$ small. However, a more detailed analysis of the magnitude of $[PL, QL]$ will be presented in a future publication.

If we assume that $[PL, QL] \approx 0$ and thus $C(t, u_0) \approx -tPL$, we get from (11)

$$\int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds \approx e^{tL} (Q L u_{0k} - e^{-tPL} Q L u_{0k})$$

Expansion of the operator e^{-tPL} in Taylor series around $t = 0$ gives

$$P \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds \approx \sum_{j=1}^{\infty} (-1)^{j+1} \frac{t^j}{j!} P e^{tL} (PL)^j Q L u_{0k}. \quad (13)$$

One can obtain different simplified models by truncating the series in (13) after different values of j . In particular, if we omit all the terms after the first one we obtain the t -model which has been studied thoroughly [9; 3; 20]. Note that, if we make the assumption $[PL, QL] = 0$, then the expansion in (10) reduces to the expansion in (13).

As will be explained in Section 2.4, even if $[PL, QL]$ is very small but still finite, these simplified models are *not* guaranteed to be stable. This is reminiscent of singular perturbation problems where there is change in the qualitative behavior of the solution when the perturbation parameter changes from zero to nonzero [1].

2.3.3. The noncommutative case. For the sake of completeness, we comment briefly on the case when $[PL, QL] \neq 0$ and not small. In this case we need to modify the terms in the memory expansion to make them closed in the resolved variables while retaining the accuracy of the model. For example, from (10), if we keep terms up to the second order (in t) we have

$$P \int_0^t e^{(t-s)L} P L e^{sQL} Q L u_{0k} ds = t P e^{tL} P L Q L u_{0k} + \frac{1}{2} t^2 P e^{tL} (P L Q L Q L u_{0k} - L P L Q L u_{0k}) + O(t^3). \quad (14)$$

The term $e^{tL} L P L Q L u_{0k}$ can be written as $L e^{tL} P L Q L u_{0k}$. This term depends on *all* the variables, resolved and unresolved. Thus we need to approximate it with a term that depends only on the resolved variables and still keeps the $O(t^3)$ accuracy of the approximation. To do that we observe that $L e^{tL} P L Q L u_{0k}$ is the RHS of the equation for the evolution of the quantity $e^{tL} P L Q L u_{0k}$. We have

$$\frac{\partial}{\partial t} e^{tL} P L Q L u_{0k} = L e^{tL} P L Q L u_{0k}.$$

We can apply the (projected) Mori–Zwanzig formalism to this equation and get

$$\begin{aligned} \frac{\partial}{\partial t} P e^{tL} P L Q L u_{0k} &= P L e^{tL} P L Q L u_{0k} = P e^{tL} L P L Q L u_{0k} \\ &= P e^{tL} P L P L Q L u_{0k} + P \int_0^t e^{(t-s)L} P L e^{sQL} Q L P L Q L u_{0k} ds \\ &= P e^{tL} P L P L Q L u_{0k} + O(t). \end{aligned} \quad (15)$$

The fact that the memory term is $O(t)$ can be seen by expanding (as before) the memory integrand $e^{(t-s)L} P L e^{sQL} Q L P L Q L u_{0k}$ in Taylor series around $s = 0$. The difference is that now, we retain *only* the Markovian term in the equation

for the evolution of $e^{tL} PLQLu_{0k}$. Thus if we substitute in (14) the expression $Pe^{tL} PLPLQLu_{0k} + O(t)$ for $Pe^{tL} LPLQLu_{0k}$ we have

$$P \int_0^t e^{(t-s)L} PL e^{sQL} QL u_{0k} ds \\ = t P e^{tL} PLQLu_{0k} + \frac{1}{2} t^2 P e^{tL} (PLQLQLu_{0k} - PLPLQLu_{0k}) + O(t^3). \quad (16)$$

The last equation results from the multiplication of the $O(t)$ term in (15) with t^2 which gives a $O(t^3)$ term. What we have gained is that we have expressed the RHS of the evolution equation for $Pe^{tL}u_{0k}$ as a function only of the resolved variables while retaining $O(t^3)$ accuracy. Similar constructions can be carried out for higher order terms. Numerical results for this approach will be presented elsewhere (see also discussion at the end of Section 3.3).

2.4. The renormalized Mori–Zwanzig algorithm. We focus on the case on Mori–Zwanzig reduced models corresponding to the almost commutative case (see Section 2.3.2).

As we have already mentioned, the computational advantage of (13) is that it contains expressions which depend *only* on the resolved variables. The series representation of the memory term in (13) is based on the assumption of analyticity in time of the operator e^{-tPL} . This assumption may be true for small t but it does not have to hold for larger t . In other words, the Taylor expansion of the operator e^{-tPL} has, in general, only a *finite* radius of convergence. Insisting on using the Taylor expansion of the operator e^{-tPL} as is for later times is dangerous and can lead to the instability of the reduced model (see also Section 3.3). In fact, when dealing with full systems coming from discretizations of singular PDEs, the breakdown of the Taylor expansion of the operator e^{-tPL} is related to the onset of underresolution on the part of the full system.

To proceed we put the MZ model given by (9) and (13) in the framework of Section 2.1. To do that we set

$$R_{1k}^{(1)} = P e^{tL} PLu_{0k}, \quad (17)$$

$$R_{jk}^{(1)} = (-1)^j \frac{t^{j-1}}{(j-1)!} P e^{tL} (PL)^{j-1} QL u_{0k}, \quad j = 2, \dots \quad (18)$$

With this identification we have, in essence, embedded the reduced models derived through the MZ formalism in a larger class of reduced models which share the same functional form with the MZ models but which are allowed to have different coefficients. In the notation of Section 2.1, the original MZ models correspond to the coefficient vector $a^{(1)} = (1, 1, 1, \dots)$.

While the original MZ models may suffer from instabilities (see also Section 3.3), the new models can be made stable by *assigning to each term in the reduced model*

the appropriate coefficient. The magnitude of the coefficient of a term reflects the importance of the term in the reduced model. The values of the coefficients can now be determined by solving the linear algebraic system (4). This ensures that the coefficient of each term in the model is properly redefined (renormalized) so that the resulting reduced model reproduces, at the scales accessible to the reduced model, the dynamics (see (3)) of the original system.

We are now in a position to state the renormalized Mori–Zwanzig algorithm, which constructs a reduced model with the necessary coefficients computed on the fly.

Renormalized Mori–Zwanzig (rMz) algorithm.

- (1) Choose a number of terms, say m , to keep at the Taylor expansion of the memory term.
- (2) Evolve the full system and compute, at every step, using the SVD, the rank of the $(m + 1) \times (m + 1)$ matrix B .
- (3) When the smallest singular value σ_{m+1} reaches a value larger than a prescribed tolerance ε (we assume that the singular values are indexed from largest to smallest), solve the system (4) for the coefficients.
- (4) For the remaining simulation time, switch from the full system to the reduced model with the estimated values of the coefficients.

To apply the algorithm, we need to specify the quantities \hat{E}_i , $i = 1, \dots, m$. Also, we need to compute the expression for the Markovian term, as well as the expressions for the terms in the Taylor expansion of the memory term.

3. Application of rMZ to 1D Burgers and 3D Euler equations

In this section we present results of the rMZ algorithm for the 1D Burgers and the 3D Euler equations.

3.1. 1D Burgers equation.

3.1.1. Setup of the reduced model. We use the 1D inviscid Burgers equation as an instructive example for the constructions presented in this section. The equation is given by

$$u_t + uu_x = 0. \tag{19}$$

Equation (19) should be supplemented with an initial condition $u(x, 0) = u_0(x)$ and boundary conditions. We solve (19) in the interval $[0, 2\pi]$ with periodic boundary conditions. This allows us to expand the solution in Fourier series

$$u_M(x, t) = \sum u_k(t)e^{ikx},$$

where $F \cup G = [-M/2, M/2 - 1]$. We have written the set of Fourier modes as the union of two sets in anticipation of the construction of the reduced model comprising only of the modes in $F = [-N/2, N/2 - 1]$, where $N < M$. The equation of motion for the Fourier mode u_k becomes

$$\frac{du_k}{dt} = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F \cup G}} u_p u_q. \quad (20)$$

To conform with the Mori–Zwanzig formalism we set

$$R_k(u) = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F \cup G}} u_p u_q$$

and we have

$$\frac{du_k}{dt} = R_k(u) \quad (21)$$

for $k \in F \cup G$. The system (21) is supplemented by the initial condition $u_0 = (\hat{u}_0, \tilde{u}_0) = (\hat{u}_0, 0)$. We focus on initial conditions where the unresolved Fourier modes are set to zero. We also define L by

$$L = \sum_{k \in F \cup G} R_k(u_0) \frac{\partial}{\partial u_{0k}}.$$

Note that $Lu_{0k} = R_k(u_0)$.

We also need to define a projection operator P . For a function $h(u_0)$ of all the variables, the projection operator we will use is defined by

$$P(h(u)) = P(h(\hat{u}_0, \tilde{u}_0)) = h(\hat{u}_0, 0);$$

that is, it replaces the value of the unresolved variables \tilde{u}_0 in any function $h(u_0)$ by zero. Note that this choice of projection is consistent with the initial conditions we have chosen. Also, we define the Markovian term

$$\hat{R}_1^{(1)}k(\hat{u}_0) = PLu_{0k} = PR_k(u_0) = -\frac{ik}{2} \sum_{\substack{p+q=k \\ p,q \in F}} \hat{u}_{0p} \hat{u}_{0q}.$$

The Markovian term has the same functional form as the RHS of the full system but is restricted to a sum over only the resolved modes in F . The full system conserves the energy $\frac{1}{2} \sum_{k \in F \cup G} |u_k|^2$ contained in all the modes. Similarly, the Markovian term of the reduced model does *not* alter the energy content of the resolved modes. The necessary energy transfer out of the resolved modes rests on the memory terms. Based on our choice of projection operator and the scaling symmetries of the Burgers equation we set $N = M/2$.

With the definition of P given above, we find for QLu_{0k}

$$QLu_{0k} = -\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} u_{0p}u_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} u_{0p}u_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in G}} u_{0p}u_{0q}.$$

The expression for QLu_{0k} contains three terms which involve at least one wavenumber in the unresolved range G . The terms in the Taylor expansion of the memory term are given by

$$R_{jk}^{(1)} = (-1)^j \frac{t^{j-1}}{(j-1)!} P e^{tL} (PL)^{j-1} QLu_{0k}, \quad j = 2, \dots$$

For the j -th term we have

$$\begin{aligned} & (PL)^{j-1} QLu_{0k} \\ &= (PL)^{j-1} \left(-\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} u_{0p}u_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} u_{0p}u_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in G}} u_{0p}u_{0q} \right). \end{aligned} \quad (22)$$

3.1.2. Recursive computation of the memory terms. The expression in (22) for $(PL)^{j-1} QLu_{0k}$ for the j -th term ($j = 2, \dots$) can be computed recursively using a simple construction based on a Pascal triangle. Note that for our choice of projection operator P , we have

$$(PL)^{j-1} \left(-\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in G}} u_{0p}u_{0q} \right) = 0.$$

We begin with the (first-order) term for $j = 2$, which is $PLQLu_{0k}$. We find

$$PLQLu_{0k} = -2\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} Pu_{0p}PLu_{0q}. \quad (23)$$

The first order term can be computed by convolving the resolved part of Pu_{0p} with the unresolved part of PLu_{0q} . In practice, all the convolutions sums can be computed using Fast Fourier Transforms [5]. Note that the expression Pu_{0p} is linear in the Fourier modes while PLu_{0q} is quadratic. Thus, the convolution sum in $PLQLu_{0k}$ (including the factor $-ik/2$) can be denoted by $(1r * 2u)$, where $*$ stands for convolution while r and u stand for the resolved and unresolved parts. This notation facilitates the recognition of the pattern for the higher order terms.

With this notation, the first-order term can be written as

$$\mathbf{PLQLu}_{0k} = 2 \times \mathbf{1}(1r * 2u), \quad (24)$$

where we have used boldface to denote the coefficient. We continue with the second order term $\mathbf{PLPLQLu}_{0k}$. We find

$$\mathbf{PLPLQLu}_{0k} = 2 \left(-\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} P u_{0p} \mathbf{PLPLu}_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} \mathbf{PLP}u_{0p} \mathbf{PLu}_{0q} \right). \quad (25)$$

The convolution sums in this term can be denoted by $(1r * 3u)$ and $(2r * 2u)$. The second order term can be written as

$$\mathbf{PLPLQLu}_{0k} = 2 \times \left(\mathbf{1}(1r * 3u) + \mathbf{1}(2r * 2u) \right) \quad (26)$$

To see the pattern more clearly we need one more term:

$\mathbf{PLPLPLQLu}_{0k}$

$$= 2 \left(-\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} P u_{0p} \mathbf{PLPLPLu}_{0q} - 2\frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} \mathbf{PLP}u_{0p} \mathbf{PLPLu}_{0q} - \frac{ik}{2} \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} \mathbf{PLPLP}u_{0p} \mathbf{PLu}_{0q} \right). \quad (27)$$

The terms in the parenthesis can be denoted by $(1r * 4u)$, $(2r * 3u)$ and $(3r * 2u)$. The third order term can be written as

$$\mathbf{PLPLPLQLu}_{0k} = 2 \times \left(\mathbf{1}(1r * 4u) + \mathbf{2}(2r * 3u) + \mathbf{1}(3r * 2u) \right). \quad (28)$$

By examining the expressions in (24)–(28) we see that the memory terms can be computed as weighted sums of convolution sums where the weights are given by appropriate Pascal triangle coefficients (the boldface numbers). This was to be expected since we started with a convolution sum (involving products) of two functions and each new term in the Taylor series involves a differentiation. Moreover, the number of convolution sums that need to be added is equal to the order of the memory term in the Taylor expansion. Also, for each term, the convolution sums involve expressions whose degree (in Fourier modes) follows an easily discernible pattern. For the l -th order term in the Taylor series we need the convolution sums $(1r * (l + 1)u)$, $(2r * lu)$, \dots , $(lr * 2u)$.

Finally, the expressions entering the convolution sums can also be computed by a Pascal triangle construction. For example, in order to calculate the third

order term, as can be seen from (27), one needs to compute and store only the quantities Pu_{0p} , PLu_{0p} , $PLPLu_{0p}$, $PLPLPLu_{0p}$ for $p \in F \cup G$. Note that the quantities Pu_{0p} , $PLPLu_{0p}$ and $PLPLPLu_{0p}$ which are also needed are the same as Pu_{0p} , PLu_{0p} and $PLPLu_{0p}$ for the *resolved* modes and zero for the *unresolved* modes. So, they do not need to be stored. They can be quickly constructed when needed. We see that the storage requirements for the calculation of the memory terms grows only linearly in the order of the Taylor expansion. Also, the ability to calculate the needed expressions through FFTs speeds up significantly the calculation of the various memory terms.

The recursive estimation of the memory terms allows us to calculate memory terms of very high order efficiently, without having to write down explicitly the analytical expressions which become very complicated after the first few orders in the expansion.

3.1.3. Results using rMZ. The construction of the renormalized MZ reduced models in the previous section assume that the commutator $[PL, QL]$ is small. It is not easy to estimate the commutator in general. However, from (11) and (12), we see that we are interested in the magnitude of the quantity $e^{tL}[PL, QL]QLu_{0k}$. For this quantity to be zero for all time, we must have $[PL, QL]QLu_{0k} \equiv 0$; that is, $[PL, QL]QLu_{0k}$ must be the zero function. This is not possible unless $[PL, QL] \equiv 0$. However, we can look for initial conditions u_0 such that $[PL, QL]QLu_{0k}$ is small. The expression for $[PL, QL]QLu_{0k}$ is

$$\begin{aligned}
 [PL, QL]QLu_{0k} &= 2 \left(-\frac{ik}{2} \right) \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} [PL, QL]u_{0p}u_{0q} \\
 &+ 2 \left(-\frac{ik}{2} \right) \sum_{\substack{p+q=k \\ q \in G \\ p \in F \cup G}} PLu_{0p}PLu_{0q} - 2 \left(-\frac{ik}{2} \right) \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} PLu_{0p}QLu_{0q}, \quad (29)
 \end{aligned}$$

where

$$[PL, QL]u_{0p} = 2 \left(-\frac{ip}{2} \right) \sum_{\substack{r+s=p \\ s \in F \\ r \in G}} PLu_{0r}u_{0s} - 2 \left(-\frac{ip}{2} \right) \sum_{\substack{r+s=p \\ s \in F \\ r \in F}} QL u_{0r}u_{0s}.$$

It is straightforward to see from these expressions that if the initial condition is smooth, in the sense that it contains only a few small wavenumber Fourier modes, the value of $e^{tL}[PL, QL]QLu_{0k}$ is small. The reason for that is the polynomial nonlinearity which allows only a *finite* rate of propagation of activity to higher wavenumbers. We have used the smoothest possible nontrivial initial condition which is $u_0(x) = \sin x$. This leads to the formation of a standing shock at $T = 1$.

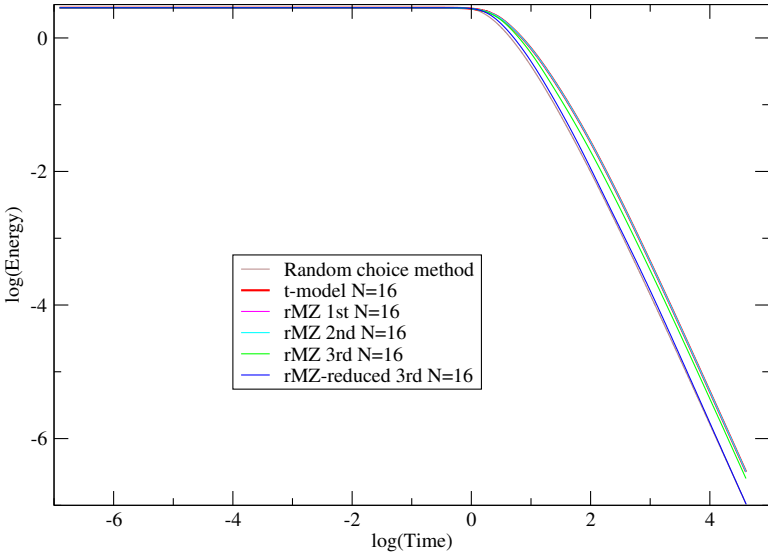


Figure 1. 1D Burgers equation: comparison of the evolution of the energy in the resolved modes computed by the random choice method, the t -model and various rMZ models.

Of course, due to the formation of the shock, the value of $e^{tL}[PL, QL]QLu_{0k}$ will eventually stop being small. However, it will allow us to renormalize the first few terms in the memory term expansion and obtain a finite result (see also the discussion in [Section 3.3](#)).

[Figure 1](#) shows the evolution of $\frac{1}{2} \sum_{k \in F} |u_k|^2$ (energy) of the resolved modes computed by reduced models of different orders and the random choice method [\[7\]](#). All the reduced models use $N = 16$ Fourier modes while the full system has $M = 32$ modes. The results of the reduced models are compared to a converged solution of the random choice method with $N = 4096$ points. The energy of the random choice method solution was computed using only $N = 16$ modes. However, note that practically all the energy of the random choice method solution is concentrated in the first few Fourier modes, so even if we had computed the energy for all $N = 4096$ Fourier modes the results would not have changed. This is to be expected, since for the initial condition we are using, a standing shock forms at time $T = 1$ and, thus, by time $T = 100$ the only Fourier modes having some energy left in them are the first few.

The quantities \hat{E}_i used to set up the linear algebraic system needed to compute the coefficients of the reduced system are l_p norms of the solution. In particular, for the first order model we use $\hat{E}_i = \sum_{k \in F} |u_k|^{2i}$, $i = 1, 2$. For the second order model $\hat{E}_i = \sum_{k \in F} |u_k|^{2i}$, $i = 1, 2, 3$ and for the third order model $\hat{E}_i = \sum_{k \in F} |u_k|^{2i}$, $i = 1, 2, 3, 4$. In general, for the reduced model of order λ we need $\lambda + 1$ quantities

because we also have to compute the coefficient of the Markovian term. All the calculations are done in double precision. The tolerance ε used to decide when it is time to switch to the reduced model is set to 10^{-12} . The systems of ordinary differential equations for the different reduced models were solved using the Runge–Kutta–Fehlberg method with the step-size control tolerance set to 10^{-10} [19].

The numerical problem of solving the linear system for the coefficients is hard because the resulting system has very large condition number and very small determinant. This happens for three reasons. First, the dominant contribution to the linear system matrix comes from the Markovian term (except for the contribution to the rate of change of the $\hat{E}_1 = \sum_{k \in F} |u_k|^2$ which is zero). This means that the coefficient of the Markovian term is practically 1. Second, the contributions of each memory term to the rates of change of the different \hat{E}_i vary dramatically. Third, the contributions to each \hat{E}_i by the different memory terms also varies substantially. Of course, this situation is exacerbated if we use more terms in the expansion. For the case when we retain up to the third order term in the memory expansion, we have to deal with condition numbers of the order 10^{11} and determinant values of order 10^{-20} . Inevitably, even the use of double precision cannot provide us with an accurate estimate of the coefficients. Since the linear system matrix is practically singular (for the numerical precision used) we have chosen to solve the linear system using the SVD algorithm [18].

A partial remedy to the problem comes from a slight modification in the way of estimating the coefficients. Since we know that the Markovian term coefficient is practically 1, we can set it to 1, and subtract the column of contributions of the Markovian term from the RHS of the linear system. This allows us to reduce the dimensionality of the linear system to be solved from $(\lambda + 1) \times (\lambda + 1)$ to $\lambda \times \lambda$. This practice of subtracting almost equal numbers is not advisable in general because it leads to loss of significant digits [18]. However, in our case it helps to improve the results by lowering the condition number of the matrix from about 10^{11} to about 10^5 . In , the estimation of the coefficients for the third order model using the reduced dimension matrix is denoted by “rMZ-reduced 3rd”.

As shown in Figure 1, the rMZ models of first and second order give practically the same results as the t -model. The third order model gives a slight improvement. However, when the reduced dimension matrix is used, the energy evolution predicted by the third order model is practically identical to the correct energy evolution of the resolved modes predicted by the random choice method. If we increase the resolution of the reduced model, the numerical problems for the calculation of higher order coefficients become even more pronounced. This is to be expected, since a larger resolution means that the renormalized coefficients of the reduced model will be smaller. Thus, computing them with accuracy is more difficult.

We have to note that the computation of the higher order coefficients is more difficult for our choice of initial condition since it only involves one active Fourier mode. Initial conditions with more active Fourier modes will transfer activity to the unresolved scales at a higher rate and thus the corresponding renormalized coefficients will be larger. A detailed study of the behavior of the coefficients for different initial conditions will be presented elsewhere. For the first order model, we have already presented in [23] a detailed study about the change of the value of the renormalized coefficient with resolution up to the order of 10^5 Fourier modes. In that work, the renormalized coefficient calculation was used to determine, in a fixed point analysis, the blow-up exponent.

3.2. Incompressible Euler equations in 3D. Consider the incompressible Euler equations in 3D with periodic boundary conditions in the cube $[0, 2\pi]^3$:

$$u_t + u \cdot \nabla u = -\nabla p, \quad \nabla \cdot u = 0, \quad (30)$$

where $u(x, t) = (u_1(x_1, x_2, x_3, t), u_2(x_1, x_2, x_3, t), u_3(x_1, x_2, x_3, t))$ is the velocity, p is the pressure and $\nabla = (\partial/\partial x_1, \partial/\partial x_2, \partial/\partial x_3)$. The system in 3.2 is supplemented with the initial condition $u(x, 0) = u_0(x)$ which is also periodic and incompressible and $x = (x_1, x_2, x_3)$. Since we are working with periodic boundary conditions, we expand the solution in Fourier series keeping N modes in each spatial direction,

$$u_M(x, t) = \sum u_k(t) e^{ikx},$$

where $F \cup G = [-M/2, M/2 - 1] \times [-M/2, M/2 - 1] \times [-M/2, M/2 - 1]$. Also $k = (k_1, k_2, k_3)$ and $u_k(t) = (u_k^1(t), u_k^2(t), u_k^3(t))$.

The equation of motion for the Fourier mode u_k becomes

$$\frac{du_k}{dt} = -i \sum_{\substack{p+q=k \\ p, q \in F \cup G}} k \cdot u_p A_k u_q, \quad (31)$$

where $A_k = I - kk^T/|k|^2$ is the incompressibility projection matrix and I is the 3×3 identity matrix. The symbol \cdot denotes inner product in \mathbb{R}^3 . The system (31) is supplemented by the initial condition $u_0 = \{u_k(0)\} = \{u_{0k}\}$, $k \in F \cup G$, where u_{0k} are the Fourier coefficients of the initial condition $u_0(x)$.

To conform with the MZ formalism we set

$$R_k(u) = -i \sum_{\substack{p+q=k \\ p, q \in F \cup G}} k \cdot u_p A_k u_q$$

and we have

$$\frac{du_k}{dt} = R_k(u) \quad (32)$$

for $k \in F \cup G$. The system (32) is supplemented by the initial condition $u_0 = (\hat{u}_0, \tilde{u}_0) = (\hat{u}_0, 0)$. Note that we focus on initial conditions where the unresolved Fourier modes are set to zero. We also define L by

$$L = \sum_{k \in F \cup G} R_k(u_0) \frac{\partial}{\partial u_{0k}}.$$

Note that $Lu_{0k} = R_k(u_0)$. Consider the subset

$$F = [-N/2, N/2 - 1] \times [-N/2, N/2 - 1] \times [-N/2, N/2 - 1]$$

for $N < M$. We will construct the reduced models for the Fourier modes u_k with $k \in F$.

We need to define a projection operator P . For a function $h(u_0)$ of all the variables, the projection operator we will use is defined by $P(h(u)) = P(h(\hat{u}_0, \tilde{u}_0)) = h(\hat{u}_0, 0)$, i.e., it replaces the value of the unresolved variables \tilde{u}_0 in any function $h(u_0)$ by zero. Note that this choice of projection is consistent with the initial conditions we have chosen. Based on our choice of projection operator and the scaling symmetries of the Euler equations we set $N = \frac{M}{2}$.

Define

$$\hat{R}_k(\hat{u}_0) = P R_k(u_0) = -i \sum_{\substack{p+q=k \\ p, q \in F}} k \cdot \hat{u}_{0p} A_k \hat{u}_{0q}.$$

The Markovian term has the same functional form as the RHS of the full system but is restricted to a sum over only the resolved modes in F . The full system conserves the energy $\frac{1}{2} \sum_{k \in F \cup G} |u_k|^2$ contained in all the modes. Similarly, the Markovian term of the reduced model does *not* alter the energy content of the resolved modes. The necessary energy transfer out of the resolved modes rests on the memory terms.

With the definition of P given above, we find for QLu_{0k}

$$QLu_{0k} = -i \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} k \cdot u_{0p} A_k u_{0q} - \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} k \cdot u_{0p} A_k u_{0q} - i \sum_{\substack{p+q=k \\ q \in G \\ p \in G}} k \cdot u_{0p} A_k u_{0q}.$$

The expression for QLu_{0k} contains three terms which involve at least one wavenumber in the unresolved range G . The terms in the Taylor expansion of the memory term are given by

$$R_{jk}^{(1)} = (-1)^j \frac{t^{j-1}}{(j-1)!} P e^{tL} (PL)^{j-1} QLu_{0k}, \quad j = 2, \dots$$

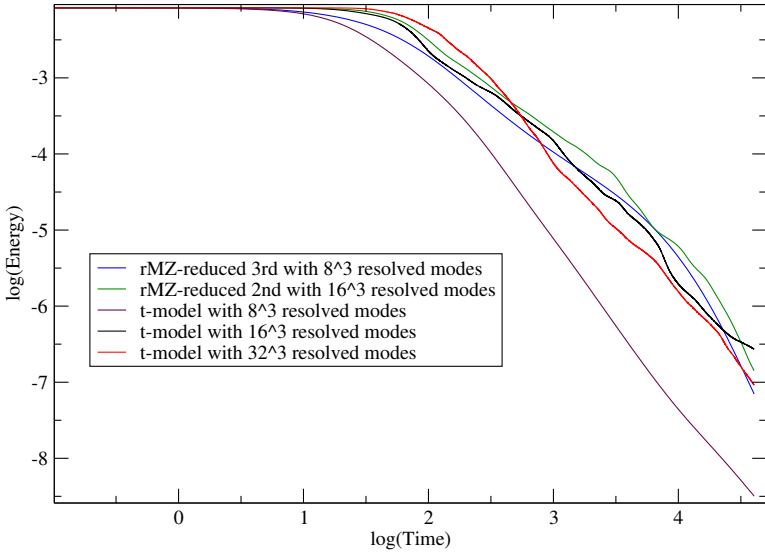


Figure 2. 3D Euler equation. Comparison of the evolution of the energy in the resolved modes computed by the t -model and various rMZ models.

For the j -th term we have

$$(PL)^{j-1} QLu_{0k} = (PL)^{j-1} \left(-i \sum_{\substack{p+q=k \\ q \in G \\ p \in F}} k \cdot u_{0p} A_k u_{0q} - i \sum_{\substack{p+q=k \\ q \in F \\ p \in G}} k \cdot u_{0p} A_k u_{0q} - i \sum_{\substack{p+q=k \\ q \in G \\ p \in G}} k \cdot u_{0p} A_k u_{0q} \right). \quad (33)$$

The different terms in the memory expansion can be computed recursively as in the case of Burgers. However, there is a slight complication because the presence of the incompressibility operator and of the inner product on the RHS destroys the commutativity which allowed us in Burgers to group terms (the factor 2 which appears outside every parenthesis there). This problem can be addressed by a construction which uses 2 Pascal triangles instead of 1 used in the case of Burgers. For each order, one adds up the corresponding terms from the 2 Pascal triangles and obtains the desired memory term. Other than that, the recursive algorithm remains the same and we omit the details. Also, note that all the higher order terms are divergence-free by construction.

We have used the same quantities \hat{E}_i as in the case of Burgers, with the obvious generalizations, since for 3D Euler we have a 3-dimensional velocity vector instead of the scalar velocity in Burgers. Also, even though for 3D Euler we have a 3-dimensional vector, we have assumed that the reduced model renormalized coefficients are the same for all 3 velocities. This is a simplifying assumption. Of course,

one can use different renormalized coefficients for the different velocities at the expense of having to solve a larger linear system for the renormalized coefficients. A detailed study of that case will be presented in future work.

We have used the Taylor–Green initial condition (see [21], for example) which is given by

$$\begin{aligned} u_1(x, 0) &= \sin x_1 \cos x_2 \cos x_3, \\ u_2(x, 0) &= -\cos x_1 \sin x_2 \cos x_3, \\ u_3(x, 0) &= 0. \end{aligned}$$

Figure 2 shows the evolution of the energy $\frac{1}{2} \sum_{k \in F} |u_k|^2$ of the resolved modes for different resolutions computed by rMZ reduced models of different orders and the t -model. We have presented results for the rMZ reduced models using the reduced linear system matrix approach discussed above to tame the condition number of the matrix. Based on these results, we make two observations.

First, for 8^3 resolved modes, the rMZ third order model dissipates energy at a slower rate than the t -model with 8^3 resolved modes. This is true not only for the third order model but also for the first and second order models (we have omitted those results to avoid cluttering the figure). This slower rate of energy dissipation compared to the t -model holds also for the case of 16^3 resolved modes.

The second observation is that the rate of energy dissipation of the rMZ models is consistent with the rate predicted by the t -model with *higher* resolution. This is to be expected since a higher order model should result in a more accurate prediction of the energy dissipation rate.

The reader may be concerned about the small resolutions used in the numerical experiments. There are two reasons for that. First, if one keeps several terms in the memory expansion, then, for a very smooth initial condition like the one we use, the matrix B becomes even more ill-conditioned for large resolutions. However, this is not a severe problem. On the contrary, it signifies that most of the higher order terms should have small coefficients and thus can be safely removed from the model.

The second reason we have used small resolutions both for Burgers and Euler is because an accurate reduced model should be able to reproduce the correct energy content for its resolved scales no matter how small the resolution. For example, for Burgers, where we know what the energy content should be after the singularity, we see that the rMZ model with a small resolution (16^3) indeed reproduces the correct energy content for this resolution.

3.3. rMZ vs MZ. We show that the renormalized version of the MZ formalism is advantageous with respect to the original MZ formalism. In particular, we show that for the same order in the Taylor expansion of the memory term, the renormalized

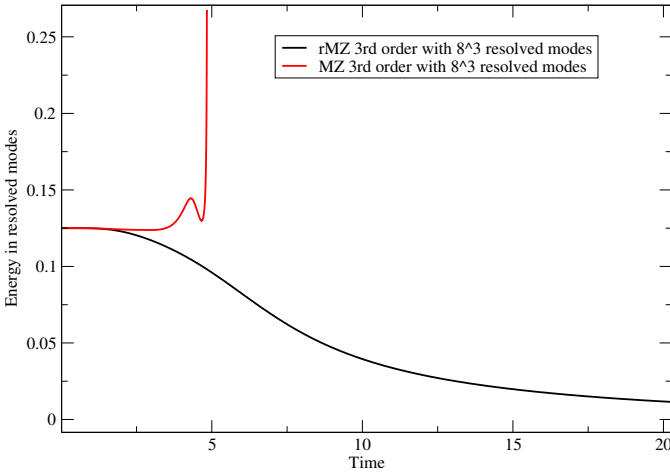


Figure 3. Comparison, for 3D Euler, of the value of the energy content of the resolved modes for the third order renormalized and unrenormalized MZ models.

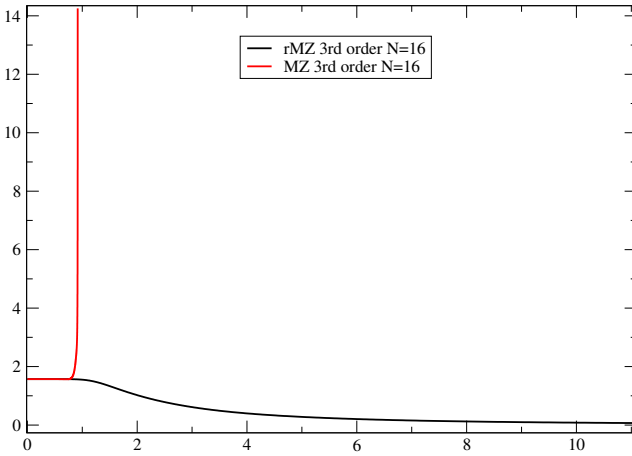


Figure 4. Comparison, for 1D Burgers, of the value of the energy content of the resolved modes for the third order renormalized and unrenormalized MZ models.

algorithm leads to the stabilization of the reduced model. [Figure 3](#) compares, for the 3D Euler equations, the energy $\frac{1}{2} \sum_{k \in F} |u_k|^2$ for 8^3 resolved modes for the renormalized and unrenormalized third order models. The unrenormalized model quickly becomes unstable and loses all predictive ability. [Figure 4](#) compares the behavior of the renormalized and unrenormalized third order models for Burgers.

The unrenormalized expansion leads to divergence of the predicted energy content of the resolved modes. This is analogous to the divergences that plagued perturbative calculations in quantum field theory (QFT) before the advent of renormalization

[13]. In QFT, the reason for the divergences was that the perturbation expansion was performed in a quantity (the bare mass, charge etc.) which turns out to be ill-defined. The process of renormalization replaces the perturbation expansion in powers of the ill-defined quantity with a perturbation expansion in powers of the experimentally determined values of this quantity. This allows the subtraction of the terms that cause divergences and leads to finite results.

In our case, the expansion of the memory of the MZ formalism is ill-defined because the Taylor expansion at $t = 0$ breaks down after some time. On the other hand, the renormalized MZ formalism takes into account dynamic information from the evolution of the full system (while this system is still valid) and prescribes to each term in the memory expansion an appropriate coefficient. The coefficient measures how important this term is. In this way, the divergences are averted and the results become finite.

We should emphasize that the reason the renormalization of the MZ model works is the smoothness of the initial condition which renders higher order terms less and less important. This is reflected in the values of the renormalized coefficients which decrease with the order of the memory expansion. However, if we attempt to renormalize the MZ model for an initial condition where *all* the resolved Fourier modes are initially nonzero, we find that all the coefficients remain of $O(1)$ as in the nonrenormalized (and unstable) MZ model. This means that in this case renormalization cannot help with the stabilization of the reduced model.

The last observation suggests that the road to stable reduced models for the case when the initial condition has many nonzero Fourier modes may lie in a different expansion than in a Fourier series. In particular, one may have to expand the solution in a basis of appropriate collective degrees of freedom so that the initial condition contains only a few nonzero collective modes. For the case of incompressible flows these could be vortices or even Beltrami flows [12]. If one can do that, then the framework presented in the current work will remain applicable.

3.4. Universality of the renormalized coefficients. In the introduction, we hinted at the possibility that the renormalized coefficients may be determined by two factors: the ratio of the smallest active scale in the initial condition to the smallest resolvable scale, and the scaling symmetries of the equation under investigation. Even though the numerical difficulties with the ill-conditioned linear system matrix do not allow us at present to study accurately the higher order renormalized coefficients, we have enough accuracy to study the first order renormalized coefficient both for 1D Burgers and 3D Euler. Note that the two equations share the same scaling symmetries. Also, we have chosen for Burgers the initial condition $u_0(x) = \sin x$ which has only one active Fourier mode (for $k = \pm 1$) and for 3D Euler, the Taylor–Green initial condition which also has only active Fourier modes for $k_i = \pm 1, i = 1, 2, 3$.

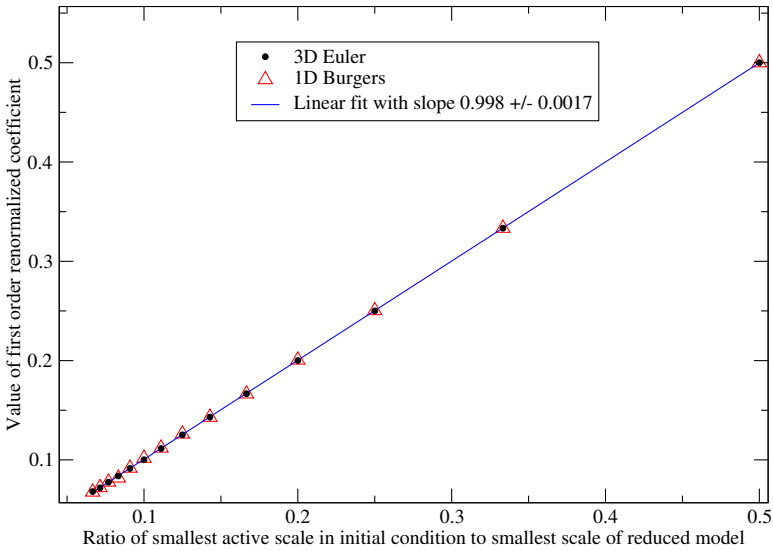


Figure 5. Comparison, for 1D Burgers and 3D Euler, of the value of the renormalized coefficient for the first order rMZ model for different resolutions. The initial conditions for Burgers and Euler have only 1 active Fourier mode in each direction.

Figure 5 shows the comparison of the value of the renormalized first-order coefficient for Burgers and 3D Euler as a function of the ratio of the smallest scale active in the initial condition to the smallest scale of the reduced model. We make two observations. First, the values of the renormalized coefficient for the two equations are in remarkable agreement. Second, from the slope of the linear fit, we see that the value of the coefficient is practically equal to the ratio of the smallest scale active in the initial condition to the smallest scale of the reduced model.

Needless to say that one example is not enough to infer the generality of the result for arbitrary initial conditions. A theoretical explanation of this result is lacking at the moment. Note that due to the way we have defined the terms in the expansion of the memory, all the terms have the same dimensions as the Markovian term and the left-hand side of the equation for each Fourier mode. So, the corresponding coefficients have to be dimensionless. Thus, we expect the coefficients to depend on ratios of quantities with the same dimensions. Here we have investigated the possibility that this ratio is that of the smallest active scale in the initial condition to the smallest active scale of the reduced model.

We should comment here on the behavior of the rMZ algorithm for the 2D Euler equations for which the 2D version of the Taylor–Green initial condition is an exact solution, i.e., a steady state. Exactly because it is a steady state there is no need for a reduced model. Application of the rMZ algorithm agrees with this. There is never any need to transfer energy to the unresolved scales and thus, no need to

switch to a reduced model. The contributions of the different memory terms to the matrix B are all well below the double precision threshold. This allows the freedom to assign to the renormalized coefficient the values shown in [Figure 5](#) without incurring any trouble. In other words, the behavior of the solution of the 2D Euler for the Taylor–Green initial condition does not contradict the agreement for the renormalized coefficient of the 1D Burgers and 3D Euler equations shown in [Figure 5](#).

4. Conclusions and future work

We have presented a new way of computing reduced models for systems of ordinary differential equations. The approach combines renormalization and effective field theory techniques with the Mori–Zwanzig formalism. The constructed reduced models are stable because they transfer activity out of the resolved scales at a rate which is dictated by the full system. The consistency between the rate of transfer activity of the reduced model and the rate of transfer activity dictated by the full system is the analog of the matching conditions employed in effective field theory. The matching conditions lead to a redefinition (renormalization) of the coefficients of a reduced model originally constructed through the Mori–Zwanzig formalism.

The results we have obtained for the 1D Burgers and 3D Euler equations are rather encouraging. However, we have to deal with the ill-conditioning of the linear system for the coefficients. We plan to address the problem through various techniques designed to deal with ill-conditioned matrices. Also, it is very interesting to study more to what extent the renormalized coefficients are determined by the structure of the initial condition and the scaling symmetries of the PDE.

We note that the proposed approach can also be applied to the Navier–Stokes equations [\[14\]](#). The viscosity starts contributing from the second order memory term. Also, the inclusion of viscosity does not complicate considerably the recursive algorithm for the calculation of the higher order terms. The expressions needed to compute the viscosity contributions can be estimated through terms already computed in the construction of the inviscid terms.

The approach presented in the current work opens new possibilities for the construction of accurate and stable reduced models for (large) systems of ordinary differential equations. It also highlights the affinity between problems of model reduction in scientific computing and the construction of effective field theories in high energy physics. We hope that this connection will benefit the problem of constructing reduced models and will be of use in tackling real world problems which are impossible to address through brute force calculations.

In conclusion, as Steven Weinberg once put it [\[25\]](#), renormalization is indeed a good thing.

Acknowledgements

I am grateful to Profs. G. I. Barenblatt, A. J. Chorin, O. H. Hald and V. Sverak for their ongoing guidance and support.

References

- [1] G. I. Barenblatt, *Scaling*, Cambridge University Press, 2003. [MR 2005e:00011](#) [Zbl 1094.00006](#)
- [2] R. Bellman, *Perturbation techniques in mathematics, engineering and physics*, Holt, Rinehart and Winston, New York, 1964, Reprinted Dover, Mineola, NY, 2003. [MR 28 #4212](#) [Zbl 0274.34001](#)
- [3] D. Bernstein, *Optimal prediction of Burgers's equation*, *Multiscale Model. Simul.* **6** (2007), no. 1, 27–52. [MR 2008b:76034](#) [Zbl 1135.65373](#)
- [4] J. Binney, N. Dowrick, A. Fisher, and M. Newman, *The theory of critical phenomena: An introduction to the renormalization group*, Clarendon Press, Oxford, 1992.
- [5] J. P. Boyd, *Chebyshev and Fourier spectral methods*, 2nd ed., Dover Publications, Mineola, NY, 2001. [MR 2002k:65160](#) [Zbl 0994.65128](#)
- [6] A. J. Chandy and S. H. Frankel, *The τ -model as a large eddy simulation model for the Navier–Stokes equations*, *Multiscale Model. Simul.* **8** (2009), no. 2, 445–462. [MR 2010m:76089](#) [Zbl 05719773](#)
- [7] A. J. Chorin, *Random choice solution of hyperbolic systems*, *J. Computational Phys.* **22** (1976), no. 4, 517–533. [MR 57 #11077](#) [Zbl 0354.65047](#)
- [8] A. J. Chorin, O. H. Hald, and R. Kupferman, *Optimal prediction and the Mori–Zwanzig representation of irreversible processes*, *Proc. Natl. Acad. Sci. USA* **97** (2000), no. 7, 2968–2973. [MR 2000m:82045](#) [Zbl 0968.60036](#)
- [9] ———, *Optimal prediction with memory*, *Phys. D* **166** (2002), no. 3–4, 239–257. [MR 2003e:62150](#) [Zbl 1017.60046](#)
- [10] A. J. Chorin and P. Stinis, *Problem reduction, renormalization, and memory*, *Commun. Appl. Math. Comput. Sci.* **1** (2006), 1–27. [MR 2007f:82092](#) [Zbl 1108.82023](#)
- [11] J. C. Collins, *Renormalization: An introduction to renormalization, the renormalization group, and the operator-product expansion*, Cambridge University Press, 1984. [MR 86k:81093](#) [Zbl 1094.53505](#)
- [12] P. Constantin and A. Majda, *The Beltrami spectrum for incompressible fluid flows*, *Comm. Math. Phys.* **115** (1988), no. 3, 435–456. [MR 89g:35089](#) [Zbl 0669.76049](#)
- [13] B. Delamotte, *A hint of renormalization*, *Am. J. Phys.* **72** (2004), no. 2, 170–184.
- [14] C. R. Doering and J. D. Gibbon, *Applied analysis of the Navier–Stokes equations*, Cambridge University Press, 1995. [MR 96a:76024](#)
- [15] H. Georgi, *Effective field theory*, *Annu. Rev. Nucl. Part. Sci.* **43** (1993), 209–252.
- [16] D. Givon, R. Kupferman, and A. Stuart, *Extracting macroscopic dynamics: model problems and algorithms*, *Nonlinearity* **17** (2004), no. 6, R55–R127. [MR 2006i:82081](#) [Zbl 1073.82038](#)
- [17] N. Goldenfeld, *Lectures on phase transitions and the renormalization group*, Perseus Books, Reading, MA, 1992.
- [18] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996. [MR 97g:65006](#) [Zbl 0865.65009](#)
- [19] E. Hairer, S. P. Nørsett, and G. Wanner, *Solving ordinary differential equations, I: Nons-tiff problems*, Springer Series in Computational Mathematics, no. 8, Springer, Berlin, 1987. [MR 87m:65005](#) [Zbl 0838.76016](#)

- [20] O. H. Hald and P. Stinis, *Optimal prediction and the rate of decay for solutions of the Euler equations in two and three dimensions*, Proc. Natl. Acad. Sci. USA **104** (2007), no. 16, 6527–6532. MR 2008e:76100 Zbl 1155.76036
- [21] C.-W. Shu, W.-S. Don, D. Gottlieb, O. Schilling, and L. Jameson, *Numerical convergence study of nearly incompressible, inviscid Taylor–Green vortex flow*, J. Sci. Comput. **24** (2005), no. 1, 1–27. MR 2006m:76028 Zbl 1161.76535
- [22] P. Stinis, *Higher order Mori–Zwanzig models for the Euler equations*, Multiscale Model. Simul. **6** (2007), no. 3, 741–760. MR 2008m:76012 Zbl 1151.65344
- [23] ———, *A phase transition approach to detecting singularities of partial differential equations*, Commun. Appl. Math. Comput. Sci. **4** (2009), 217–239. MR 2011c:35015 Zbl 1180.35014
- [24] C. Sulem and P.-L. Sulem, *The nonlinear Schrödinger equation: Self-focusing and wave collapse*, Applied Mathematical Sciences, no. 139, Springer, New York, 1999. MR 2000f:35139 Zbl 0928.35157
- [25] S. Weinberg, *Why the renormalization group is a good thing*, Asymptotic realms of physics: Essays in honor of Francis E. Low, MIT Press, Cambridge, MA, 1983.

Received November 26, 2012. Revised April 17, 2013.

PANOS STINIS: stinis@umn.edu

Department of Mathematics, University of Minnesota, 206 Church St. SE, Minneapolis, MN 55455, United States

LEGENDRE SPECTRAL-COLLOCATION METHOD FOR VOLTERRA INTEGRAL DIFFERENTIAL EQUATIONS WITH NONVANISHING DELAY

YANPING CHEN AND ZHENDONG GU

The main purpose of this paper is to propose the Legendre spectral-collocation method to solve the Volterra integral differential equations with nonvanishing delay which arise in many problems, such as modeling in biosciences and population. In our method we divide the definition domain of the solution into several subintervals where the solution is sufficiently smooth. Then we can use the spectral-collocation method for these equations in each subinterval. We provide convergence analysis for this method, which shows that the numerical errors decay exponentially. Numerical examples are presented to confirm these theoretical results.

1. Introduction

VIDEs (Volterra integral differential equations) with delay arise in many problems, for example, ecological competition systems [54], modeling in biosciences and population [1; 9; 26] and models for transmission of disease with immigration of infectives [10]. Nonlinear Volterra integral and integrodifferential equations with nonvanishing delay have been used since the 1920s as mathematical models of population growth and related phenomena in biology. Volterra [46] refined his earlier predator-prey model to include situations where “historical actions cease after a certain interval of time” [47]. This leads to a system of nonlinear Volterra

Yanping Chen is the corresponding author.

This work is supported by National Science Foundation of China (11271145), Foundation for Talent Introduction of Guangdong Provincial University, Specialized Research Fund for the Doctoral Program of Higher Education (20114407110009), the Project of Department of Education of Guangdong Province (2012KJCX0036), Hunan Provincial Innovation Foundation for Postgraduate (CX2012B241).

PACS: primary 65M70; secondary 45D05, 45J05.

Keywords: Volterra integral differential equations, nonvanishing delay, Legendre spectral-collocation method, convergence analysis.

integro-differential equations with constant delay $T_0 > 0$ (using Volterra's notation):

$$\begin{aligned} N_1'(t) &= N_1(t) \left(\varepsilon_1 - \gamma_1 N_2(t) - \int_{t-T_0}^t F_1(t-\tau) N_1(\tau) d\tau \right), \\ N_2'(t) &= N_2(t) \left(-\varepsilon_2 + \gamma_2 N_1(t) + \int_{t-T_0}^t F_2(t-\tau) N_2(\tau) d\tau \right), \end{aligned}$$

with $\varepsilon_i > 0$, $\gamma_i \geq 0$ and continuous $F_i(t) \geq 0$, where $N_i(t) = \phi_i(t)$, $t \leq 0$, $i = 1, 2$. $N_1(t)$ and $N_2(t)$ represent the sizes of two populations (prey and predator) at time $t \geq 0$. These equations can be extended naturally to describe the dynamics of multispecies ecological systems. A further development of such population models based on VIDEs can be found in [20].

There exist many numerical methods for the VIDEs with delay, for example, general linear methods [53], linear multistep methods [5], block-by-block methods [32], Runge–Kutta methods [6; 7; 21; 29; 38], Petrov–Galerkin methods [31], piecewise polynomial collocation methods [13; 14; 36; 37]. Brunner investigated the numerical solution of nonlinear VIDEs with infinite delay in [11] and neutral VIDEs with constant delay in [12]. The superconvergence of the collocation method for VIDEs with nonvanishing delay is investigated in [13; 37; 39]. The monograph by Brunner [13] contains a wealth of material on the theory and numerical methods for VIDEs, with the focus being on the basic theory of Volterra equations with delay and the collocation methods and their convergence analysis.

Without the integral terms in VIDEs we obtain DDEs (delay differential equations). DDE models arise in many problems, such as the growth of tumors [45], population dynamics [28], hepatitis B virus infection [23], harmful algal blooms in the presence of toxic substances [16]. More applications of DDEs are described in [28]. Numerically solving DDEs has many of the same difficulties discussed for delay VIDEs. Many numerical methods are investigated for DDEs [6; 22; 34; 55]. The monograph by Bellen and Zennaro [8] gives a comprehensive account of numerical methods for DDEs, with the focus being on (classical and continuous) Runge–Kutta methods and their asymptotic stability properties which were also investigated by Baker and Tang [7]. There are some well-developed softwares for delay differential equations or systems. The popular solver developed by Shampine and Thompson [40; 44] for DDEs is well tested and user-friendly.

Spectral methods receive considerable attention mainly due to their high accuracy. Tang, Xu and Cheng [43] proposed a Legendre spectral-collocation method to solve VIEs (Volterra integral equations) of the second kind whose kernel and solutions are sufficiently smooth. Chen and Tang [17; 18; 19] proposed and analyzed a Jacobi spectral-collocation approximation for linear VIEs of the second kind with weakly singular kernels provided that the underlying solutions of the VIEs are sufficiently smooth. Then, in [30], the Jacobi spectral-collocation method was extended to

solve VIEs with Abel-type kernel. Recently, another spectral method, i.e., the Legendre spectral Galerkin method, was investigated in [48; 52] for VIEs. The spectral-collocation methods also attract the interest of those people who study the Volterra-type integral and related functional differential equations (see, e.g., [2; 3; 4; 27; 42; 49; 50; 51]).

However, there is very little literature about the spectral method to solve VIDEs with nonvanishing delay. The main difficulty in applying the spectral method to VIDEs with nonvanishing delay is that the solutions of these equations are not smooth enough at the primary discontinuous points associated with the delay function. In this paper, we overcome this difficulty and propose a Legendre spectral-collocation method to solve these equations. In our method we divide the definition domain into several subintervals according to the primary discontinuous points associated with the nonvanishing delay function. In each subinterval, where the solution is smooth enough, we can apply the Legendre spectral-collocation method to approximate the solution. We provide convergence analysis to show that the numerical errors decay exponentially. Numerical examples are presented to confirm this theoretical prediction.

The VIDEs with nonvanishing delay considered in this paper are as follows:

$$y'(t) = a(t)y(t) + b(t)y(\theta(t)) + g(t) + \int_0^t K_1(t, s)y(s) ds + \int_0^{\theta(t)} K_2(t, s)y(s) ds, \quad t \in (0, T],$$

$$y(t) = \phi(t), \quad t \in [\theta(0), 0]. \tag{1}$$

In population models, $y(t)$ means the population size at time t . The delay $\theta(t)$ means that the growth of population size depends on the historical action. We assume that the functions describing the above equation all possess continuous derivatives of at least order $m \geq 1$ on their respective domains; i.e.,

$$a(t), b(t), g(t) \in C^m([0, T]), \quad \phi(t) \in C^m([\theta(0), 0]),$$

$$K_1(t, s) \in C^m(\Omega_1), \quad \Omega_1 := \{(t, s) : 0 \leq s \leq t \leq T\},$$

$$K_2(t, s) \in C^m(\Omega_2), \quad \Omega_2 := \{(t, s) : \theta(0) \leq s \leq \theta(t), 0 \leq t \leq T\}, \tag{2}$$

and the delay function θ will be subject to the following conditions:

$$\theta(t) := t - \tau(t), \quad \tau \in C^m([0, T]),$$

$$\tau(t) \geq \tau_0 > 0 \quad \text{for all } t \in [0, T],$$

$$\theta \text{ is strictly increasing on } [0, T]. \tag{3}$$

The nonvanishing delay θ gives rise to the primary discontinuity points $\{\xi_\mu\}$ for the solution of (1): they are determined by the recursion

$$\theta(\xi_\mu) = \xi_{\mu-1}, \quad \mu \geq 0 \quad (\xi_{-1} := \theta(0), \xi_0 = 0).$$

These points have the uniform separation property

$$\xi_\mu - \xi_{\mu-1} \geq \tau_0 > 0 \quad \text{for all } \mu \geq 0.$$

For ease of notation we will assume that

$$T = \xi_{M+1} \quad \text{for some } M \geq 1.$$

Theorem 4.1.9 in [13] states that the unique solution of (1) is in $C^{m+1}(\xi_\mu, \xi_{\mu+1}]$ for each $\mu = 0, 1, \dots, M$ and is bounded on $Z_M := \{\xi_\mu : \mu = 0, 1, \dots, M\}$ and hence on $[0, T]$. At $t = \xi_\mu$ ($\mu = 1, \dots, \min\{m, M\}$),

$$\lim_{t \rightarrow \xi_\mu^-} y^{(\mu)}(t) = \lim_{t \rightarrow \xi_\mu^+} y^{(\mu)}(t),$$

while the $(\mu + 1)$ -th derivative of y is in general not continuous at ξ_μ . In addition, if $\min\{m, M\} = m < M$, the solution also lies in $C^{m+1}[\xi_m, T]$. This motivates us to apply the spectral-collocation method to approximate the solution on the subinterval $(\xi_\mu, \xi_{\mu+1}]$, $\mu = 0, 1, \dots, M$.

This paper is organized as follows. In Section 2, we introduce the Legendre spectral-collocation method for VIDEs with nonvanishing delay. Some useful lemmas for the convergence analysis will be provided in Section 4, and the convergence analysis, in both L^∞ and L^2 , will be given in Section 5. Numerical experiments are carried out in Section 6. Finally, in Section 7, we end with the conclusion and future work.

2. Legendre spectral-collocation method

For ease of analysis we change the interval $[0, T]$ to the standard interval $[-1, 1]$. Precisely we use the variable transformation

$$t(x) = \frac{1}{2}T(x + 1), \quad s(z) = \frac{1}{2}T(z + 1). \quad (4)$$

Then (1) can be written as

$$\begin{aligned} u'(x) &= A(x)u(x) + B(x)u(\vartheta(x)) + f(x) \\ &\quad + \int_{-1}^x R_1(x, z)u(z) dz + \int_{-1}^{\vartheta(x)} R_2(x, z)u(z) dz, \quad x \in (-1, 1], \end{aligned} \quad (5)$$

$$u(x) = \psi(x), \quad x \in [\vartheta(-1), -1],$$

where

$$\begin{aligned} u(x) &:= y(t(x)), \quad A(x) := \frac{1}{2}Ta(t(x)), \quad B(x) := \frac{1}{2}Tb(t(x)), \\ f(x) &:= \frac{1}{2}Tg(t(x)), \quad \vartheta(x) := \frac{2}{T}\theta(t(x)) - 1, \quad \psi(x) := \phi(t(x)), \\ R_1(x, z) &:= \left(\frac{1}{2}T\right)^2 K_1(t(x), s(z)), \quad R_2(x, z) := \left(\frac{1}{2}T\right)^2 K_2(t(x), s(z)). \end{aligned} \quad (6)$$

The primary discontinuity point ξ_μ becomes

$$\eta_\mu := (2\xi_\mu/T) - 1, \quad \mu = -1, 0, 1, \dots, M.$$

Define

$$\delta_\mu := (\eta_\mu, \eta_{\mu+1}], \quad \mu = -1, 0, \dots, M.$$

Set the collocation points as follows:

$$X_N := \bigcup_{\mu=0}^M X^\mu, \quad X^\mu := \{x_n^\mu : \eta_\mu = x_0^\mu < x_1^\mu < \dots < x_N^\mu = \eta_{\mu+1}\}, \quad (7)$$

where

$$x_i^\mu := \frac{\eta_{\mu+1} - \eta_\mu}{2} x_i + \frac{\eta_{\mu+1} + \eta_\mu}{2}; \quad (8)$$

here x_i , $i = 0, 1, \dots, N$, are the $N + 1$ Legendre Gauss–Lobatto points in the standard interval $[-1, 1]$. Then (5) holds at x_i^μ , $i = 0, 1, \dots, N$, $\mu = 0, 1, \dots, M$:

$$\begin{aligned} u'(x_i^\mu) &= A(x_i^\mu)u(x_i^\mu) + B(x_i^\mu)u(\vartheta(x_i^\mu)) + f(x_i^\mu) \\ &\quad + \int_{-1}^{x_i^\mu} R_1(x_i^\mu, z)u(z) dz + \int_{-1}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z)u(z) dz. \end{aligned} \quad (9)$$

We use u_i^μ to approximate $u(x_i^\mu)$, v_i^μ to approximate $u(\vartheta(x_i^\mu))$, ρ_i^μ to approximate $u'(x_i^\mu)$. Then we can use

$$u_\mu(x) := \sum_{j=0}^N u_j^\mu F_j^\mu(x), \quad x \in [\eta_\mu, \eta_{\mu+1}]$$

to approximate $u|_{\delta_\mu}(x)$, i.e., the restriction of $u(x)$ to the interval $[\eta_\mu, \eta_{\mu+1}]$. $F_j^\mu(x)$, $x \in [\eta_\mu, \eta_{\mu+1}]$, is the j -th Lagrange interpolation basic function associated with the collocation points $x_0^\mu, x_1^\mu, \dots, x_N^\mu$ in the interval $[\eta_\mu, \eta_{\mu+1}]$. Similarly, we use

$$\rho_\mu(x) := \sum_{j=0}^N \rho_j^\mu F_j^\mu(x), \quad x \in [\eta_\mu, \eta_{\mu+1}]$$

to approximate $u'|_{\delta_\mu}(x)$, i.e., the restriction of $u'(x)$ to the subinterval $[\eta_\mu, \eta_{\mu+1}]$. Eventually $u(x)$ can be approximated by

$$u^N(x) := u_\mu(x) \quad \text{if } x \in [\eta_\mu, \eta_{\mu+1}], \quad \mu = 0, 1, \dots, M,$$

and $u'(x)$ can be approximated by

$$\rho^N(x) := \rho_\mu(x) \quad \text{if } x \in [\eta_\mu, \eta_{\mu+1}], \quad \mu = 0, 1, \dots, M.$$

Then (9) can be approximated by

$$\begin{aligned} \rho_i^\mu &\approx A(x_i^\mu)u_i^\mu + B(x_i^\mu)v_i^\mu + f(x_i^\mu) \\ &\quad + \int_{-1}^{x_i^\mu} R_1(x_i^\mu, z)u^N(z) dz + \int_{-1}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z)u^N(z) dz, \end{aligned} \quad (10)$$

which can be written as

$$\begin{aligned} \rho_i^\mu &\approx A(x_i^\mu)u_i^\mu + B(x_i^\mu)v_i^\mu + f(x_i^\mu) \\ &\quad + \sum_{r=0}^{\mu-1} \int_{\eta_r}^{\eta_{r+1}} R_1(x_i^\mu, z)u_r(z) dz + \int_{\eta_\mu}^{x_i^\mu} R_1(x_i^\mu, z)u_\mu(z) dz \\ &\quad + \sum_{r=0}^{\mu-2} \int_{\eta_r}^{\eta_{r+1}} R_2(x_i^\mu, z)u_r(z) dz + \int_{\eta_{\mu-1}}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z)u_{\mu-1}(z) dz. \end{aligned} \quad (11)$$

In order to compute the integral term by the Gauss quadrature rule, we change the integration interval to the standard interval $[-1, 1]$. Note that the variable transformation

$$z(a, b, v) := \frac{b-a}{2}v + \frac{b+a}{2}, \quad v \in [-1, 1] \quad (12)$$

can change the interval $[a, b]$ to $[-1, 1]$. For simplicity, we define

$$z_r(v) := z(\eta_r, \eta_{r+1}, v), \quad v \in [-1, 1], \quad r \geq 0. \quad (13)$$

Using the Gauss quadrature formula to approximate the integration term in (11) we obtain

$$\begin{aligned} \rho_i^\mu &= A(x_i^\mu)u_i^\mu + B(x_i^\mu)v_i^\mu + f(x_i^\mu) + \sum_{r=0}^{\mu-1} \frac{\eta_{r+1} - \eta_r}{2} \sum_{k=0}^N R_1(x_i^\mu, z_r(v_k))u_r(z_r(v_k))\omega_k \\ &\quad + \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \sum_{k=0}^N R_1(x_i^\mu, z_\mu(z(-1, x_i, v_k)))u_\mu(z_\mu(z(-1, x_i, v_k)))\omega_k \\ &\quad + \sum_{r=0}^{\mu-2} \frac{\eta_{r+1} - \eta_r}{2} \sum_{k=0}^N R_2(x_i^\mu, z_r(v_k))u_r(z_r(v_k))\omega_k \\ &\quad + \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \sum_{k=0}^N R_2(x_i^\mu, z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v_k))) \\ &\quad \quad \quad \times u_{\mu-1}(z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)))\omega_k, \end{aligned} \quad (14)$$

where $v_k, k = 0, 1, \dots, N$, are the $N + 1$ Legendre Gauss–Lobatto points in the standard interval $[-1, 1]$, corresponding to the weights $\omega_k, k = 0, 1, \dots, N$, and

$$\tilde{\vartheta}(x_i^\mu) := \frac{2}{\eta_\mu - \eta_{\mu-1}} \vartheta(x_i^\mu) - \frac{\eta_\mu + \eta_{\mu-1}}{\eta_\mu - \eta_{\mu-1}}, \quad \mu > 0. \quad (15)$$

Note that, for $j, k = 0, 1, \dots, N$, $r = 0, 1, \dots, M$,

$$F_j^r(z_r(v_k)) = F_j(v_k) = \begin{cases} 1, & k = j, \\ 0, & k \neq j, \end{cases}$$

$$F_j^r(z_r(z(-1, x, v))) = F_j(z(-1, x, v)), \quad (16)$$

where $F_j(v)$ is the j -th Lagrange interpolation basic function associated with the $N + 1$ Legendre Gauss–Lobatto points in the standard interval $[-1, 1]$. Then (14) can be simplified to

$$\rho_i^\mu = A(x_i^\mu)u_i^\mu + B(x_i^\mu)v_i^\mu + f(x_i^\mu) + \beta(x_i^\mu) + \lambda(x_i^\mu),$$

$$\mu = 0, 1, \dots, M, \quad i = 0, 1, \dots, N, \quad (17)$$

where

$$\beta(x_i^\mu) := \sum_{r=0}^{\mu-1} \frac{\eta_{r+1} - \eta_r}{2} \beta_1^r(x_i^\mu) + \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \beta_3(x_i^\mu),$$

$$\beta_1^r(x_i^\mu) := \sum_{k=0}^N R_1(x_i^\mu, z_r(v_k)) u_k^r \omega_k, \quad r = 0, 1, \dots, \mu - 1,$$

$$\beta_3(x_i^\mu) := \sum_{j=0}^N u_j^\mu \sum_{k=0}^N R_1(x_i^\mu, z_\mu(z(-1, x_i, v_k))) F_j(z(-1, x_i, v_k)) \omega_k,$$

and

$$\lambda(x_i^\mu) := \begin{cases} \frac{\vartheta(x_i^0) + 1}{2} \lambda_2(x_i^0), & \mu = 0, \\ \sum_{r=0}^{\mu-2} \frac{\eta_{r+1} - \eta_r}{2} \lambda_1^r(x_i^\mu) + \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \lambda_3(x_i^\mu), & \mu > 0, \end{cases}$$

$$\lambda_1^r(x_i^\mu) := \sum_{j=0}^N u_j^r R_2(x_i^\mu, z_r(v_j)) \omega_j, \quad r = 0, 1, \dots, \mu - 2,$$

$$\lambda_2(x_i^0) := \sum_{k=0}^N R_2(x_i^0, z(-1, \vartheta(x_i^0), v_k)) \psi(z(-1, \vartheta(x_i^0), v_k)) \omega_k,$$

$$\lambda_3(x_i^\mu) := \sum_{j=0}^N u_j^{\mu-1} \sum_{k=0}^N R_2(x_i^\mu, z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v_k))) F_j(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)) \omega_k.$$

However, the linear systems (17) alone are not enough to find out the unknown elements. We need two other linear systems associated with u_i^μ , v_i^μ , ρ_i^μ ,

$i = 0, 1, \dots, N, \mu = 0, 1, \dots, M$. Note that

$$\begin{aligned} u(x_i^\mu) &= u(-1) + \int_{-1}^{x_i^\mu} u'(z) dz \\ &= \psi(-1) + \sum_{r=0}^{\mu-1} \frac{\eta_{r+1} - \eta_r}{2} \int_{-1}^1 u'(z_r(v)) dv \\ &\quad + \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \int_{-1}^1 u'(z_\mu(z(-1, x_i, v))) dv. \end{aligned} \quad (18)$$

Then we can approximate the above equation by

$$u_i^\mu = \psi(-1) + \alpha_1(x_i^\mu), \quad \mu = 0, 1, \dots, M, \quad i = 0, 1, \dots, N, \quad (19)$$

where

$$\begin{aligned} \alpha_1(x_i^\mu) &= \sum_{r=0}^{\mu-1} \frac{\eta_{r+1} - \eta_r}{2} \sum_{k=0}^N \rho_k^r \omega_k \\ &\quad + \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \sum_{j=0}^N \rho_j^\mu \sum_{k=0}^N F_j(z(-1, x_i, v_k)) \omega_k. \end{aligned} \quad (20)$$

Similarly, the equation

$$\begin{aligned} u(\vartheta(x_i^\mu)) &= u(-1) + \int_{-1}^{\vartheta(x_i^\mu)} u'(z) dz \\ &= \psi(-1) + \sum_{r=0}^{\mu-2} \frac{\eta_{r+1} - \eta_r}{2} \int_{-1}^1 u'(z_r(v)) dv \\ &\quad + \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \int_{-1}^1 u'(z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v))) dv \end{aligned} \quad (21)$$

can be approximated by

$$v_i^\mu = \psi(-1) + \alpha_2(x_i^\mu), \quad \mu = 0, 1, \dots, M, \quad i = 0, 1, \dots, N, \quad (22)$$

where

$$\alpha_2(x_i^\mu) = \begin{cases} \psi(\vartheta(x_i^0)) - \psi(-1), & \mu = 0, \\ \sum_{r=0}^{\mu-2} \frac{\eta_{r+1} - \eta_r}{2} \sum_{k=0}^N \rho_k^r \omega_k \\ \quad + \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \sum_{j=0}^N \rho_j^{\mu-1} \sum_{k=0}^N F_j(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)) \omega_k, & \mu > 0. \end{cases}$$

Equations (19) and (22) are two linear systems we want to find.

The Legendre spectral-collocation method is to find $\rho_i^\mu, u_i^\mu, i = 0, 1, \dots, N, \mu = 0, 1, 2, \dots, M$, which satisfy (17), (19) and (22). The approximation to $y(t)$

is $u^N(2t/T - 1)$; the approximation to $y'(t)$ is $(2/T)\rho^N(2t/T - 1)$. An efficient computation of $F_j(s)$ can be found in [15] or [43].

3. The existence of the solution to the discrete system

In this section we will discuss the existence of the solution to the discrete system (17), (19) and (22). We write the linear system (17), (19) and (22) into matrix form:

$$\begin{cases} U^{(\mu)} = \Phi_1^{(\mu)} + A^{(\mu)}U^{(\mu)} + R_1^{(\mu)}U^{(\mu)} + B^{(\mu)}V^{(\mu)}, \\ U^{(\mu)} = \Phi_2^{(\mu)} + \frac{\eta_{\mu+1} - \eta_\mu}{2}R_3^{(\mu)}U'^{\mu}, \\ V^{(\mu)} = \Phi_3^{(\mu)} + R_4^{(\mu)}U'^{\mu-1}, \end{cases} \quad (23)$$

where

$$U^{(\mu)} := [\rho_0^\mu, \rho_1^\mu, \dots, \rho_N^\mu]',$$

$$U^{(\mu)} := [u_0^\mu, u_1^\mu, \dots, u_N^\mu]',$$

$$V^{(\mu)} := [v_0^\mu, v_1^\mu, \dots, v_N^\mu]', \quad \mu > 0,$$

$$V^{(0)} := [\psi(\vartheta(x_0^0)), \psi(\vartheta(x_1^0)), \dots, \psi(\vartheta(x_N^0))]', \quad \mu = 0,$$

$$\Phi_1^{(\mu)} := F^{(\mu)} + \sum_{r=0}^{\mu-1} R_1^{(r)}U^{(r)} + \sum_{r=0}^{\mu-2} R_2^{(r)}U^{(r)} + R_2^{(\mu)}U^{(\mu-1)}, \quad \mu > 0,$$

$$\Phi_1^{(0)}(i) := F^{(0)}(i) + \frac{\vartheta(x_i^0) + 1}{2} \sum_{k=0}^N R_2(x_i^0, z(-1, \vartheta(x_i^0), v_k)) \times \psi(z(-1, \vartheta(x_i^0), v_k))\omega_k,$$

$$\Phi_2^{(\mu)} := \psi(-1)[1, 1, \dots, 1]' + \sum_{r=0}^{\mu-1} R_3^{(r)}U^{(r)},$$

$$\Phi_3^{(\mu)} := \psi(-1)[1, 1, \dots, 1]' + \sum_{r=0}^{\mu-2} R_3^{(r)}U^{(r)},$$

$$F^{(\mu)} := [f(x_0^\mu), f(x_1^\mu), \dots, f(x_N^\mu)]',$$

$$A^{(\mu)} := \text{diag}[A(x_0^\mu), A(x_1^\mu), \dots, A(x_N^\mu)],$$

$$B^{(\mu)} := \text{diag}[B(x_0^\mu), B(x_1^\mu), \dots, B(x_N^\mu)],$$

$$R_j^{(r)}(i, k) := \frac{\eta_{r+1} - \eta_r}{2} R_j(x_i^\mu, z_r(v_k))\omega_k, \quad j = 1, 2, r = 0, 1, \dots, \mu - 1,$$

$$R_3^{(r)}(i, k) := \frac{\eta_{r+1} - \eta_r}{2} \omega_k, \quad r = 0, 1, \dots, \mu - 1,$$

$$R_1^{(\mu)}(i, j) := \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \sum_{k=0}^N R_1(x_i^\mu, z_\mu(z(-1, x_i, v_k))) F_j(z(-1, x_i, v_k))\omega_k,$$

$$\begin{aligned}
R_2^{(\mu)}(i, j) &:= \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \sum_{k=0}^N R_2(x_i^\mu, z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v_k))) \\
&\quad \times F_j(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)) \omega_k, \\
R_3^{(\mu)}(i, j) &:= \frac{x_i + 1}{2} \sum_{k=0}^N F_j(z(-1, x_i, v_k)) \omega_k, \\
R_4^{(\mu)}(i, j) &:= \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \sum_{k=0}^N F_j(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)) \omega_k.
\end{aligned}$$

Plugging the second equation in (23) into the first one we obtain

$$\begin{cases}
U^{(\mu)} = \Phi_1^{(\mu)} + \frac{\eta_{\mu+1} - \eta_\mu}{2} (A^{(\mu)} + R_1^{(\mu)}) R_3^{(\mu)} U^{(\mu)} \\
\quad + (A^{(\mu)} + R_1^{(\mu)}) \Phi_2^{(\mu)} + B^{(\mu)} V^{(\mu)}, \\
V^{(\mu)} = \Phi_3^{(\mu)} + R_4^{(\mu)} U^{(\mu-1)}.
\end{cases} \quad (24)$$

This discrete system is based on the interval $[\eta_\mu, \eta_{\mu+1}]$. The existence of the solution to (24) depends on the existence of the solution to the first matrix equation of (24). Since $A(t)$, $R_1(x, z)$, $F_j(z)$ are continuous on their definition domains, the elements of the matrices $A^{(\mu)}$, $R_1^{(\mu)}$ and $R_3^{(\mu)}$, $\mu = 0, 1, \dots, M$, are all bounded. The Neumann lemma (see [35, p. 26] or [13, p. 87]) then shows that the inverse of the matrix

$$\mathcal{B}^{(\mu)} := I - \frac{\eta_{\mu+1} - \eta_\mu}{2} (A^{(\mu)} + R_1^{(\mu)}) R_3^{(\mu)}$$

exists whenever

$$\frac{\eta_{\mu+1} - \eta_\mu}{2} \|(A^{(\mu)} + R_1^{(\mu)}) R_3^{(\mu)}\| < 1$$

for some matrix norm. This clearly holds whenever $\eta_{\mu+1} - \eta_\mu$ is sufficiently small. For this aim, we divide the interval $[\eta_\mu, \eta_{\mu+1}]$ into $M_\mu + 1$ subintervals $[\tau_i^\mu, \tau_{i+1}^\mu] \subseteq [\eta_\mu, \eta_{\mu+1}]$, $i = 0, 1, \dots, M_\mu$, $\tau_0^\mu = \eta_\mu$, $\tau_{M_\mu+1}^\mu = \eta_{\mu+1}$. The exact solution of (1) still possesses continuous derivatives of order $m+1$ on the subinterval $[\tau_i^\mu, \tau_{i+1}^\mu]$, $i = 0, 1, \dots, M_\mu$, $\mu = 0, 1, \dots, M$. Applying the method in Section 2 we use Legendre spectral-collocation method to approximate the exact solution in the basic subinterval $[\tau_i^\mu, \tau_{i+1}^\mu]$.

Observing each step of the proof for convergence analysis in Section 5, we conclude that the numerical errors decay at an exponential rate no matter how many basic subintervals $[\tau_i^\mu, \tau_{i+1}^\mu]$, $i = 0, 1, \dots, M_\mu$, $\mu = 0, 1, \dots, M$, we divide the interval $[-1, 1]$ into. Therefore there exists a constant h_0 such that, for all $[\tau_i^\mu, \tau_{i+1}^\mu]$ with $\tau_{i+1}^\mu - \tau_i^\mu < h_0$, each matrix

$$\mathcal{B}^{(i\mu)} := I - \frac{\tau_{i+1}^\mu - \tau_i^\mu}{2} (A^{(i\mu)} + R_1^{(i\mu)}) R_3^{(i\mu)}, \quad i = 0, 1, \dots, M_\mu, \mu = 0, 1, \dots, M,$$

has a uniformly bounded inverse. This ensures the corresponding discrete system based on the interval $[\tau_i^\mu, \tau_{i+1}^\mu]$ possesses a unique solution.

4. Some useful lemmas

In this section, we will provide some elementary lemmas, which are important for the derivation of error estimates in [Section 5](#). In order to give the subsequent lemmas conveniently, we first introduce some spaces. For simplicity, we denote by $\partial_x^k u(x)$ the k -th derivative of u ; i.e., $\partial_x^k u(x) := (d^k u/dx^k)(x)$.

Let (a, b) be a bounded interval of the real line. We denote by $L^2(a, b)$ the space of measurable functions $u : (a, b) \rightarrow \mathbb{R}$ such that $\int_a^b |u(x)|^2 dx < +\infty$. It is a Hilbert space for the inner product

$$(u, v) := \int_a^b u(x)v(x) dx,$$

which induces the norm

$$\|v\|_{L^2(a,b)} := \left(\int_a^b |v(x)|^2 dx \right)^{1/2}.$$

Let $m \geq 1$ be an integer. We define $H^m(a, b)$ to be the vector space of the functions $v \in L^2(a, b)$ such that all the distributions of v of order up to m can be represented by functions in $L^2(a, b)$. In short,

$$H^m(a, b) := \{v \in L^2(a, b) : \text{for } 0 \leq k \leq m, \partial_x^k v(x) \in L^2(a, b)\}.$$

$H^m(a, b)$ is endowed with the inner product

$$(u, v)_m = \sum_{k=0}^m \int_a^b \partial_x^k u(x) \partial_x^k v(x) dx$$

for which $H^m(a, b)$ is a Hilbert space. The associated norm is

$$\|v\|_{H^m(a,b)} := ((v, v)_m)^{1/2}.$$

In bounding the approximation error from above, only some of the L^2 norms appearing on the right-hand side of the above norm enter into play. Thus, for a nonnegative integer N , it is convenient to introduce the seminorm

$$|v|_{H^{m,N}(a,b)} := \left(\sum_{k=\min(m,N+1)}^m \|\partial_x^k v(x)\|_{L^2(a,b)}^2 \right)^{1/2},$$

which implies that if $N \geq m - 1$ then $|v|_{H^{m,N}(a,b)} = \|\partial_x^m v\|_{L^2(a,b)}$.

Let Λ_h denote the collection of subintervals δ_μ , $\mu = 0, 1, \dots, M$. Referring to [\[25\]](#), we define the broken Sobolev space $H^m(\Lambda_h)$ as

$$H^m(\Lambda_h) := \{u : u|_{\delta_\mu} \in H^m(\delta_\mu), \mu = 0, 1, \dots, M\}.$$

The associated norm is

$$\|u\|_{H^m(\Lambda_h)} := \left(\sum_{k=0}^m \|u^{(k)}\|_{L^2(\Lambda_h)}^2 \right)^{1/2},$$

where

$$\|u^{(k)}\|_{L^2(\Lambda_h)}^2 := \sum_{\mu=0}^M \|\partial_x^k(u|_{\sigma_\mu})\|_{L^2(\delta_\mu)}^2, \quad k = 0, 1, \dots, m.$$

For a nonnegative integer N , the associated seminorm is

$$|u|_{H^{m:N}(\Lambda_h)} := \left(\sum_{k=\min(m, N+1)}^m \|u^{(k)}\|_{L^2(\Lambda_h)}^2 \right)^{1/2}.$$

If $N \geq m - 1$ then $|u|_{H^{m:N}(\Lambda_h)} = \|u^{(m)}\|_{L^2(\Lambda_h)}$.

The space $L^\infty(a, b)$ is the Banach space of measurable functions u that are bounded outside a set of measure zero, equipped with the norm

$$\|u\|_{L^\infty(a, b)} := \operatorname{ess\,sup}_{x \in (a, b)} |u(x)|.$$

We denote by $C([a, b])$ the space of continuous functions on the interval $[a, b]$.

We define an interpolation operator I_N associated with the collocation points X_N as follows: for any continuous function $v \in C([-1, 1])$,

$$I_N v(x) := I_N^\mu(v|_{\delta_\mu})(x) \quad \text{if } x \in (\eta_\mu, \eta_{\mu+1}], \quad 0 \leq \mu \leq M, \quad (25)$$

where $v|_{\delta_\mu}(x)$ is the restriction of $v(x)$ to the subinterval $[\eta_\mu, \eta_{\mu+1}]$, and I_N^μ is the interpolation operator associated with the collocation points X^μ in the subinterval $[\eta_\mu, \eta_{\mu+1}]$; i.e.,

$$I_N^\mu(v|_{\delta_\mu})(x) := \sum_{j=0}^N v|_{\delta_\mu}(x_j^\mu) F_j^\mu(x), \quad x \in [\eta_\mu, \eta_{\mu+1}].$$

Hereafter, C denotes a generic positive constant that is independent of N .

Lemma 1. *Assume that $u \in H^m(-1, 1)$, $m \geq 1$, $v(x)$ is a bounded function. Then there exists a constant C independent of u and v such that, for $N \geq m - 1$,*

$$\|u - J_N u\|_{L^2(-1, 1)} \leq C N^{-m} \|\partial_x^m u\|_{L^2(-1, 1)}, \quad (26)$$

$$\|u - J_N u\|_{L^\infty(-1, 1)} \leq C N^{1/2-m} \|\partial_x^m u\|_{L^2(-1, 1)}, \quad (27)$$

$$\sup_N \|J_N v\|_{L^2(-1, 1)} \leq C \|v\|_{L^\infty(-1, 1)}, \quad (28)$$

$$\|J_N\|_{L^\infty(-1, 1)} \leq (2/\pi) \log(N+1) + 0.685, \quad (29)$$

where J_N is the interpolation operator associated with the $N+1$ Legendre Gauss-Lobatto points in the interval $[-1, 1]$.

Proof. Inequalities (26) and (28) can be found in [15; 33; 43], and (29) can be found in [24]. We only prove (27). Using the Sobolev inequality [15, p. 490], we have

$$\|u - J_N u\|_{L^\infty(-1,1)} \leq C \|u - J_N u\|_{L^2(-1,1)}^{1/2} \|u - J_N u\|_{H^1(-1,1)}^{1/2}.$$

Applying the result (26) to $\|u - J_N u\|_{L^2(-1,1)}^{1/2}$ makes the above inequality become

$$\|u - J_N u\|_{L^\infty(-1,1)} \leq C N^{-m/2} \|\partial_x^m u\|_{L^2(-1,1)}^{1/2} \|u - J_N u\|_{H^1(-1,1)}^{1/2}, \quad (30)$$

which leads to (27) because $\|u - J_N u\|_{H^1(-1,1)}^{1/2}$ can be estimated as follows [15, p. 289]:

$$\|u - J_N u\|_{H^1(-1,1)}^{1/2} \leq C N^{(1-m)/2} \|\partial_x^m u\|_{L^2(-1,1)}^{1/2}. \quad \square$$

Lemma 2. *Assume that $u \in C([-1, 1]) \cap H^m(\Lambda_h)$. Let $I_N u$ be the interpolation function defined in (25) where $N + 1$ is the number of collocation points in the intervals $[\eta_\mu, \eta_{\mu+1}]$, $\mu = 0, 1, \dots, M$. Then the following estimates hold for $N \geq m - 1$:*

$$\|u - I_N u\|_{L^2(-1,1)} \leq C N^{-m} \|u^{(m)}\|_{L^2(\Lambda_h)}, \quad (31)$$

$$\|u - I_N u\|_{L^\infty(-1,1)} \leq C N^{1/2-m} \|u^{(m)}\|_{L^2(\Lambda_h)}, \quad (32)$$

$$\|I_N\|_{L^\infty(-1,1)} \leq C \log(N + 1), \quad (33)$$

$$\sup_N \|I_N u\|_{L^2(-1,1)} \leq C \|u\|_{L^\infty(-1,1)}. \quad (34)$$

Proof. By the definition of I_N^μ we know that the $(I_N^\mu(u|_{\delta_\mu}))(z)$ is a function defined on the subinterval $[\eta_\mu, \eta_{\mu+1}]$. The variable transformation $z = z_\mu(v)$ changes it to be a function valued on the standard interval $[-1, 1]$; i.e., for $v \in [-1, 1]$,

$$(I_N^\mu(u|_{\delta_\mu}))(z_\mu(v)) = \sum_{j=0}^N u|_{\delta_\mu}(x_j^\mu) F_j^\mu(z_\mu(v)) = \sum_{j=0}^N u|_{\delta_\mu}(x_j^\mu) F_j(v). \quad (35)$$

The result (16) is used in the derivation of the second equality above. On the other hand, we note that $u|_{\delta_\mu}(z_\mu(v))$ is a function defined on the interval $[-1, 1]$. Its interpolation polynomial associated with the Legendre Gauss–Lobatto points v_j , $j = 0, 1, \dots, N$, in the interval $[-1, 1]$ is

$$J_N(u|_{\delta_\mu}(z_\mu(v))) = \sum_{j=0}^N u|_{\delta_\mu}(z_\mu(v_j)) F_j(v), \quad v \in [-1, 1]. \quad (36)$$

Note that $v_j = x_j$; then

$$z_\mu(v_j) = x_j^\mu, \quad j = 0, 1, \dots, N.$$

Plugging this into the right-hand side of (36) yields

$$J_N(u|_{\delta_\mu}(z_\mu(v))) = \sum_{j=0}^N u|_{\delta_\mu}(x_j^\mu) F_j(v), \quad v \in [-1, 1]. \quad (37)$$

Combining (35) with (37) yields

$$(I_N^\mu(u|_{\delta_\mu}))(z_\mu(v)) = J_N(u|_{\delta_\mu}(z_\mu(v))), \quad v \in [-1, 1]. \quad (38)$$

By (26), we have

$$\begin{aligned} \int_{\eta_\mu}^{\eta_{\mu+1}} (u|_{\delta_\mu}(z) - I_N^\mu(u|_{\delta_\mu})(z))^2 dz \\ &= \frac{\eta_{\mu+1} - \eta_\mu}{2} \int_{-1}^1 (u|_{\delta_\mu}(z_\mu(v)) - J_N(u|_{\delta_\mu}(z_\mu(v))))^2 dv \\ &\leq CN^{-2m} \left(\frac{\eta_{\mu+1} - \eta_\mu}{2} \right)^{2m+1} \|\partial_v^m(u|_{\delta_\mu}(z_\mu(\cdot)))\|_{L^2(-1,1)}^2 \\ &\leq CN^{-2m} \|\partial_z^m(u|_{\delta_\mu}(\cdot))\|_{L^2(\delta_\mu)}^2. \end{aligned} \quad (39)$$

This helps to deduce that

$$\begin{aligned} \|u - I_N u\|_{L^2(-1,1)}^2 &= \sum_{\mu=0}^M \int_{\eta_\mu}^{\eta_{\mu+1}} (u|_{\delta_\mu}(z) - I_N^\mu(u|_{\delta_\mu})(z))^2 dz \\ &\leq CN^{-2m} \sum_{\mu=0}^M \|\partial_z^m(u|_{\delta_\mu}(\cdot))\|_{L^2(\delta_\mu)}^2 = CN^{-2m} \|u^{(m)}\|_{L^2(\Lambda_h)}^2, \end{aligned} \quad (40)$$

which leads to (31).

Using (27), we have

$$\begin{aligned} \|u - I_N u\|_{L^\infty(-1,1)} &= \max_{0 \leq \mu \leq M} \left\{ \|u|_{\delta_\mu}(z_\mu(\cdot)) - J_N(u|_{\delta_\mu}(z_\mu(\cdot)))\|_{L^\infty(-1,1)} \right\} \\ &\leq CN^{1/2-m} \max_{0 \leq \mu \leq M} \left\{ \|\partial_v^m(u|_{\delta_\mu}(z_\mu(\cdot)))\|_{L^2(-1,1)} \right\} \\ &\leq CN^{1/2-m} \|u^{(m)}\|_{L^2(\Lambda_h)}. \end{aligned} \quad (41)$$

This is (32).

Now we begin to prove (33). It is evident that

$$\|I_N u\|_{L^\infty(-1,1)} = \max_{0 \leq \mu \leq M} \|I_N^\mu(u|_{\delta_\mu})\|_{L^\infty(\sigma_\mu)}. \quad (42)$$

We use (29) to estimate $\|I_N^\mu(u|_{\delta_\mu})\|_{L^\infty(\delta_\mu)}$:

$$\begin{aligned} \|I_N^\mu(u|_{\delta_\mu})\|_{L^\infty(\delta_\mu)} &= \|(I_N^\mu(u|_{\delta_\mu}))(z_\mu(\cdot))\|_{L^\infty(-1,1)} = \|J_N(u|_{\delta_\mu}(z_\mu(\cdot)))\|_{L^\infty(-1,1)} \\ &\leq C \log(N+1) \|u|_{\delta_\mu}(z_\mu(\cdot))\|_{L^\infty(-1,1)} = C \log(N+1) \|u|_{\delta_\mu}\|_{L^\infty(\delta_\mu)} \\ &\leq C \log(N+1) \|u\|_{L^\infty(-1,1)}, \end{aligned} \quad (43)$$

which together with (42) give that

$$\|I_N u\|_{L^\infty(-1,1)} \leq C \log(N+1) \|u\|_{L^\infty(-1,1)}.$$

This leads to the desired result (33).

Now we begin to prove (34). The result (28) is useful in the following derivation:

$$\begin{aligned} \|I_N u\|_{L^2(-1,1)}^2 &= \sum_{\mu=0}^M \|I_N^\mu(u|_{\delta_\mu})\|_{L^2(\delta_\mu)}^2 = \sum_{\mu=0}^M \frac{\eta_{\mu+1} - \eta_\mu}{2} \|(I_N^\mu(u|_{\delta_\mu}))(z_\mu(\cdot))\|_{L^2(-1,1)}^2 \\ &= \sum_{\mu=0}^M \frac{\eta_{\mu+1} - \eta_\mu}{2} \|J_N(u|_{\delta_\mu}(z_\mu(\cdot)))\|_{L^2(-1,1)}^2 \\ &\leq C \sum_{\mu=0}^M \frac{\eta_{\mu+1} - \eta_\mu}{2} \|u|_{\delta_\mu}(z_\mu(\cdot))\|_{L^\infty(-1,1)}^2 \\ &\leq C \sum_{\mu=0}^M \frac{\eta_{\mu+1} - \eta_\mu}{2} \|u\|_{L^\infty(-1,1)}^2 \leq C \|u\|_{L^\infty(-1,1)}^2, \end{aligned} \quad (44)$$

which leads to the desired result (34). Now we have completed the whole proof for this lemma. \square

Lemma 3 [15; 41]. *Assume that $u \in H^m(-1, 1)$ for some $m \geq 1$ and $\varphi \in \mathcal{P}_N$, which denotes the space of all polynomials of degree not exceeding N . Then there exists a constant C independent of $N \geq m - 1$ such that*

$$\left| \int_{-1}^1 u(x)\varphi(x) dx - \sum_{j=0}^N u(x_j)\varphi(x_j)\omega_j \right| \leq C N^{-m} \|\partial_x^m u\|_{L^2(-1,1)} \|\varphi\|_{L^2(-1,1)},$$

where x_j are the $N+1$ Legendre Gauss–Lobatto points, with corresponding weights ω_j , $j = 0, 1, \dots, N$.

Lemma 4 [43]. *Suppose $0 \leq M < +\infty$. If a nonnegative integrable function $e(x)$ satisfies*

$$e(x) \leq v(x) + M \int_{-1}^x e(z) dz,$$

where $v(x)$ is also a nonnegative integrable function, then

$$\|e(x)\|_{L^p(-1,1)} \leq C \|v(x)\|_{L^p(-1,1)}, \quad p = 2, +\infty.$$

5. Convergence analysis

This section is devoted to providing a convergence analysis for the numerical scheme. The goal is to show that the rate of convergence is exponential; i.e., the spectral

accuracy can be obtained for the proposed approximations. Firstly, we will carry out convergence analysis in the $L^\infty(-1, 1)$ space.

Theorem 1. *Let $u(x)$ be the exact solution to (5), $u^N(x)$ be the approximate solution, and $\rho^N(x)$ be the approximate derivative obtained by using the spectral-collocation schemes (17), (19) and (22). Then, for $N \geq m - 1$ sufficiently large,*

$$\|e_i(x)\|_{L^\infty(-1,1)} \leq CN^{1/2-m} (R\|u\|_{L^\infty(-1,1)} + \|u^{(m+1)}\|_{L^2(\Lambda_h)}), \quad i = 0, 1, \quad (45)$$

where

$$e_0(x) := \begin{cases} 0, & x \in [\vartheta(-1), -1], \\ u(x) - u^N(x), & x \in (-1, 1], \end{cases}$$

$$e_1(x) := \begin{cases} 0, & x \in [\vartheta(-1), -1], \\ u'(x) - \rho^N(x), & x \in (-1, 1], \end{cases}$$

and R is a constant dependent on the m -order derivatives of $R_j(x, z)$, $\psi(z)$, $j = 1, 2$.

Proof. In each subinterval $(\eta_\mu, \eta_{\mu+1}]$, $\mu = 0, 1, \dots, M$, the degree of the polynomial $\rho^N(s)$ does not exceed N . Then

$$\alpha_1(x_i^\mu) = \int_{-1}^{x_i^\mu} \rho^N(z) dz \quad \text{and} \quad \alpha_2(x_i^\mu) = \int_{-1}^{\vartheta(x_i^\mu)} \rho^N(z) dz, \quad (46)$$

which implies that

$$u(x_i^\mu) - u_i^\mu = \int_{-1}^{x_i^\mu} e_1(z) dz, \quad (47)$$

$$u(\vartheta(x_i^\mu)) - v_i^\mu = \int_{-1}^{\vartheta(x_i^\mu)} e_1(z) dz.$$

Subtracting (17) from (9) yields

$$u'(x_i^\mu) - \rho_i^\mu = A(x_i^\mu) \int_{-1}^{x_i^\mu} e_1(z) dz + B(x_i^\mu) \int_{-1}^{\vartheta(x_i^\mu)} e_1(z) dz + \int_{-1}^{x_i^\mu} R_1(x_i^\mu, z) e_0(z) dz$$

$$+ \int_{-1}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z) e_0(z) dz + \sum_{j=0}^1 E_j(x_i^\mu), \quad (48)$$

where, for $x \in [-1, 1]$,

$$E_1(x) := \int_{-1}^x R_1(x, z) u^N(z) dz - \beta(x), \quad E_0(x) := \int_{-1}^{\vartheta(x)} R_2(x, z) u^N(z) dz - \lambda(x),$$

Multiplying by $F_i^\mu(x)$ on both sides of (48) and summing from $i = 0$ to N , we obtain that

$$\begin{aligned}
 & \sum_{i=0}^N u'(x_i^\mu) F_i^\mu(x) - \sum_{i=0}^N \rho_i^\mu F_i^\mu(x) \\
 &= \sum_{i=0}^N \left(A(x_i^\mu) \int_{-1}^{x_i^\mu} e_1(z) dz \right) F_i^\mu(x) + \sum_{i=0}^N \left(B(x_i^\mu) \int_{-1}^{\vartheta(x_i^\mu)} e_1(z) dz \right) F_i^\mu(x) \\
 &+ \sum_{i=0}^N \left(\int_{-1}^{x_i^\mu} R_1(x_i^\mu, z) e_0(z) dz \right) F_i^\mu(x) + \sum_{i=0}^N \left(\int_{-1}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z) e_0(z) dz \right) F_i^\mu(x) \\
 &+ \sum_{j=0}^1 \sum_{i=0}^N E_j(x_i^\mu) F_i^\mu(x), \quad x \in [\eta_\mu, \eta_{\mu+1}]. \tag{49}
 \end{aligned}$$

By the definitions of I_N and $\rho^N(x)$, we have

$$\begin{aligned}
 I_N u'(x) - \rho^N(x) &= I_N \left(A(x) \int_{-1}^x e_1(z) dz \right) + I_N \left(B(x) \int_{-1}^{\vartheta(x)} e_1(z) dz \right) \\
 &+ I_N \left(\int_{-1}^x R_1(x, z) e_0(z) dz \right) + I_N \left(\int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz \right) \\
 &+ \sum_{j=0}^1 I_N E_j(x), \quad x \in [-1, 1]. \tag{50}
 \end{aligned}$$

This leads to

$$\begin{aligned}
 e_1(x) &= \sum_{j=0}^1 I_N E_j(x) + \sum_{j=2}^6 E_j(x) + A(x) \int_{-1}^x e_1(z) dz + B(x) \int_{-1}^{\vartheta(x)} e_1(z) dz \\
 &+ \int_{-1}^x R_1(x, z) e_0(z) dz + \int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz, \tag{51}
 \end{aligned}$$

where

$$\begin{aligned}
 E_2(x) &:= (I - I_N) u'(x), \\
 E_3(x) &:= (I_N - I) \int_{-1}^x R_1(x, z) e_0(z) dz, \\
 E_4(x) &:= (I_N - I) \int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz, \\
 E_5(x) &:= (I_N - I) \left(A(x) \int_{-1}^x e_1(z) dz \right), \\
 E_6(x) &:= (I_N - I) \left(B(x) \int_{-1}^{\vartheta(x)} e_1(z) dz \right). \tag{52}
 \end{aligned}$$

Applying the Dirichlet formula to the last two terms in the right-hand side of (51) yields

$$\int_{-1}^x R_1(x, z) e_0(z) dz = \int_{-1}^x \left[\int_s^x R_1(x, z) dz \right] e_1(s) ds, \quad (53)$$

$$\int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz = \int_{-1}^{\vartheta(x)} \left[\int_s^{\vartheta(x)} R_2(x, z) dz \right] e_1(s) ds, \quad (54)$$

which help to deduce that there exist constants $C_1, C_2, C > 0$ such that

$$\left| A(x) \int_{-1}^x e_1(z) dz + B(x) \int_{-1}^{\vartheta(x)} e_1(z) dz + \int_{-1}^x R_1(x, z) e_0(z) dz + \int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz \right| \leq C_1 \int_{-1}^x e_1(z) dz + C_2 \int_{-1}^{\vartheta(x)} e_1(z) dz \leq C \int_{-1}^x e_1(z) dz. \quad (55)$$

Then, by Lemma 4, $e_1(x)$ in (51) can be estimated as follows:

$$\|e_1(x)\|_{L^\infty(-1,1)} \leq C \left(\sum_{j=0}^1 \|I_N E_j(x)\|_{L^\infty(-1,1)} + \sum_{j=2}^6 \|E_j(x)\|_{L^\infty(-1,1)} \right). \quad (56)$$

We estimate each term of the right-hand side of the above inequality one by one.

First we estimate $\|I_N E_0(x)\|_{L^\infty(-1,1)}$. By (33) we have

$$\|I_N E_0(x)\|_{L^\infty(-1,1)} \leq C \log(N+1) \|E_0(x)\|_{L^\infty(-1,1)}. \quad (57)$$

We estimate $\|E_0(x)\|_{L^\infty(-1,1)}$. Note that $E_0(x)$ can be written as

$$E_0(x) = \begin{cases} \int_{-1}^{\vartheta(x)} R_2(x, z) \psi(z) dz - \frac{\vartheta(x)+1}{2} \lambda_2(x), & x \in \delta_0, \\ \sum_{r=0}^{\mu-2} \left(\int_{\eta_r}^{\eta_{r+1}} R_2(x, z) u_r(z) dz - \frac{\eta_{r+1} - \eta_r}{2} \lambda_1^r(x) \right) \\ \quad + \int_{\eta_{\mu-1}}^{\vartheta(x)} R_2(x, z) u_{\mu-1}(z) dz - \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x)+1}{2} \lambda_3(x), & x \in \delta_\mu, \mu > 0. \end{cases} \quad (58)$$

Lemma 3 helps to deduce that

$$\begin{aligned} & \left| \int_{-1}^{\vartheta(x)} R_2(x, z) \psi(z) dz - \frac{\vartheta(x)+1}{2} \lambda_2(x) \right| \\ & \leq C N^{-m} \left\| \partial_v^m (R_2(x, z(-1, \vartheta(x), \cdot)) \psi(z(-1, \vartheta(x), \cdot))) \right\|_{L^2(-1,1)} \\ & \leq C N^{-m} \left| \frac{\vartheta(x)+1}{2} \right|^m \left\| \partial_z^m (R_2(x, z) \psi(z)) \Big|_{z=z(-1, \vartheta(x), \cdot)} \right\|_{L^2(-1,1)} \\ & \leq C N^{-m} \left\| \partial_z^m (R_2(x, \cdot) \psi(\cdot)) \right\|_{L^2(\vartheta(x), -1)}, \quad x \in \delta_0, \end{aligned} \quad (59)$$

and, for $x \in \delta_\mu$, $\mu > 0$,

$$\begin{aligned}
 & \left| \int_{\eta_r}^{\eta_{r+1}} R_2(x, z) u_r(z) dz - \frac{\eta_{r+1} - \eta_r}{2} \lambda_1^r(x) \right| \\
 & \leq CN^{-m} \left| \frac{\eta_{r+1} - \eta_r}{2} \right| \left\| \partial_v^m (R_2(x, z_r(x, \cdot))) \right\|_{L^2(-1,1)} \|u_r(z_r(\cdot))\|_{L^2(-1,1)} \\
 & \leq CN^{-m} \left| \frac{\eta_{r+1} - \eta_r}{2} \right|^{m+1/2} \left\| \partial_z^m (R_2(x, z)) \Big|_{z=z_r(\cdot)} \right\|_{L^2(-1,1)} \|u_r\|_{L^2(\delta_r)} \\
 & \leq CN^{-m} \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(\delta_r)} \|u_r\|_{L^2(\delta_r)}.
 \end{aligned} \tag{60}$$

Similarly,

$$\begin{aligned}
 & \left| \int_{\eta_{\mu-1}}^{\vartheta(x)} R_2(x, z) u_{\mu-1}(z) dz - \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}_{\mu-1}(x) + 1}{2} \lambda_3(x) \right| \\
 & \leq CN^{-m} \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(\eta_{\mu-1}, \vartheta(x))} \|u_{\mu-1}\|_{L^2(\delta_{\mu-1})}.
 \end{aligned} \tag{61}$$

By the Cauchy inequality, which states that

$$\sum_{r=0}^{\mu-1} a_r b_r \leq \left(\sum_{r=0}^{\mu-1} a_r^2 \right)^{1/2} \left(\sum_{r=0}^{\mu-1} b_r^2 \right)^{1/2},$$

in which we let

$$\begin{aligned}
 a_r &= \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(\delta_r)}, \quad b_r = \|u_r\|_{L^2(\delta_r)}, \quad r = 0, 1, \dots, \mu - 2, \\
 a_{\mu-1} &= \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(\eta_{\mu-1}, \vartheta(x))}, \quad b_{\mu-1} = \|u_{\mu-1}\|_{L^2(\delta_{\mu-1})},
 \end{aligned}$$

we have, for $x \in \delta_\mu$, $\mu > 0$,

$$\begin{aligned}
 |E_0(x)| &\leq CN^{-m} \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(-1, \vartheta(x))} \|u^N\|_{L^2(-1,1)} \\
 &\leq CN^{-m} \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(-1, \vartheta(x))} (\|e_0\|_{L^\infty(-1,1)} + \|u\|_{L^\infty(-1,1)}).
 \end{aligned} \tag{62}$$

Then

$$\|E_0(x)\|_{L^\infty(-1,1)} \leq CN^{-m} \tilde{R}_2 (\|e_0\|_{L^\infty(-1,1)} + \|u\|_{L^\infty(-1,1)}), \tag{63}$$

where

$$\tilde{R}_2 := \max \left\{ \max_{x \in \delta_0} \left\| \partial_z^m (R_2(x, \cdot)) \psi(\cdot) \right\|_{L^2(\vartheta(x), -1)}, \max_{x \in [\eta_1, 1]} \left\| \partial_z^m (R_2(x, \cdot)) \right\|_{L^2(-1, \vartheta(x))} \right\}.$$

Therefore, combining (63) with (57) gives

$$\|I_N E_0(x)\|_{L^\infty(-1,1)} \leq CN^{-m} \log(N+1) \tilde{R}_2 (\|e_0\|_{L^\infty(-1,1)} + \|u\|_{L^\infty(-1,1)}). \tag{64}$$

Using the same analysis as for $\|I_N E_0(x)\|_{L^\infty(-1,1)}$, we can obtain the estimate

$$\|I_N E_1(x)\|_{L^\infty(-1,1)} \leq CN^{-m} \log(N+1) \tilde{R}_1 (\|e_0\|_{L^\infty(-1,1)} + \|u\|_{L^\infty(-1,1)}), \tag{65}$$

where

$$\tilde{R}_1 := \max_{x \in [-1, 1]} \|\partial_z^m (R_1(x, \cdot))\|_{L^2(-1, x)}.$$

Now we begin to estimate $\|E_j(x)\|_{L^\infty(-1, 1)}$, $j = 2, 3, 4, 5, 6$. Note that, in each subinterval δ_μ , $\mu = 0, 1, \dots, M$, $u'|_{\delta_\mu}(x) \in H^m(\delta_\mu)$. Applying (32) to $u'(x)$, we have

$$\|E_2(x)\|_{L^\infty(-1, 1)} \leq CN^{1/2-m} \|u^{(m+1)}\|_{L^2(\Lambda_h)}. \quad (66)$$

Now we begin to estimate $\|E_4(x)\|_{L^\infty(-1, 1)}$. For simplicity of notation, we set

$$b(x) := \int_{-1}^{\vartheta(x)} R_2(x, z) e_0(z) dz.$$

Applying (32) with $m = 1$ to $b(x)$ yields

$$\|(I_N - I)b(x)\|_{L^\infty(-1, 1)} \leq CN^{-1/2} \|\partial_x^1 b\|_{L^2(-1, 1)}. \quad (67)$$

Note that

$$\begin{aligned} |\partial_x^1 b(x)| &= \left| R_2(x, \vartheta(x)) e_0(\vartheta(x)) \vartheta'(x) + \int_{-1}^{\vartheta(x)} \frac{\partial R_2}{\partial x}(x, z) e_0(z) dz \right| \\ &\leq \|e_0\|_{L^\infty(-1, 1)} \left| \left(R_2(x, \vartheta(x)) \vartheta'(x) + \int_{-1}^{\vartheta(x)} \frac{\partial R_2}{\partial x}(x, z) dz \right) \right| \\ &\leq C \|e_0\|_{L^\infty(-1, 1)}, \end{aligned} \quad (68)$$

which, together with (67), yields

$$\|E_4(x)\|_{L^\infty(-1, 1)} = \|(I_N - I)b(x)\|_{L^\infty(-1, 1)} \leq CN^{-1/2} \|e_0\|_{L^\infty(-1, 1)}. \quad (69)$$

Similarly,

$$\begin{aligned} \|E_3(x)\|_{L^\infty(-1, 1)} &\leq CN^{-1/2} \|e_0\|_{L^\infty(-1, 1)}, \\ \|E_5(x)\|_{L^\infty(-1, 1)} &\leq CN^{-1/2} \|e_1\|_{L^\infty(-1, 1)}, \\ \|E_6(x)\|_{L^\infty(-1, 1)} &\leq CN^{-1/2} \|e_1\|_{L^\infty(-1, 1)}. \end{aligned} \quad (70)$$

Combining (56) with (64), (65), (66), (69) and (70) yields that

$$\begin{aligned} \|e_1(x)\|_{L^\infty(-1, 1)} &\leq CN^{-m} (\log(N+1)R \|u\|_{L^\infty(-1, 1)} + N^{1/2} \|u^{(m+1)}\|_{L^2(\Lambda_h)}) \\ &\quad + CN^{-1/2} \|e_0\|_{L^\infty(-1, 1)}, \end{aligned} \quad (71)$$

where

$$R := \max\{\tilde{R}_1, \tilde{R}_2\}.$$

Now we need another relation between $\|e_1(x)\|_{L^\infty(-1, 1)}$ and $\|e_0\|_{L^\infty(-1, 1)}$. Multiplying by $F_i^\mu(x)$ on both sides of (47) and summing from $i = 0$ to N for

$\mu = 0, 1, \dots, M$, we obtain that

$$e_0(x) = E_7(x) + (I_N - I) \left(\int_{-1}^x e_1(s) ds \right) + \int_{-1}^x e_1(s) ds, \quad (72)$$

where

$$E_7(x) := (I - I_N)u(x).$$

Then

$$\begin{aligned} & \|e_0\|_{L^\infty(-1,1)} \\ & \leq C \left(\|E_7(x)\|_{L^\infty(-1,1)} + \left\| (I_N - I) \left(\int_{-1}^x e_1(s) ds \right) \right\|_{L^\infty(-1,1)} + \|e_1\|_{L^\infty(-1,1)} \right). \end{aligned}$$

Using (32) for $E_7(x)$, and applying (32) with $m = 1$ to the middle term of the right-hand side of the above inequality, we have

$$\|e_0\|_{L^\infty(-1,1)} \leq CN^{-m-1/2} \|u^{(m+1)}\|_{L^2(\Lambda_h)} + C \|e_1\|_{L^\infty(-1,1)}. \quad (73)$$

Plugging the above result into the last term of (71) yields the desired estimate (45) for e_1 , which, in turn, substituted into the last term of (73), yields the estimate (45) for e_0 . \square

Next, we will give the error estimate in the $L^2(-1, 1)$ space.

Theorem 2. *Let $u(x)$ be the exact solution to (5). Let $u^N(x)$ be the approximate solution, and $\rho^N(x)$ be the approximate derivative obtained by using the spectral-collocation schemes (17), (19) and (22). Then, for $N \geq m - 1$ sufficiently large,*

$$\|e_i\|_{L^2(-1,1)} \leq CN^{-m} R(R+1) (\|u\|_{L^\infty(-1,1)} + \|u^{(m+1)}\|_{L^2(\Lambda_h)}), \quad i = 0, 1. \quad (74)$$

Proof. By Lemma 4, it follows from (51) and (55) that

$$\|e_1(x)\|_{L^2(-1,1)} \leq C \left(\sum_{j=0}^1 \|I_N E_j(x)\|_{L^2(-1,1)} + \sum_{j=2}^6 \|E_j(x)\|_{L^2(-1,1)} \right). \quad (75)$$

We estimate each term on the right of the above inequality one by one. Applying (34) to $E_0(x)$ yields

$$\|I_N E_0(x)\|_{L^2(-1,1)} \leq C \|E_0(x)\|_{L^\infty(-1,1)}. \quad (76)$$

Recalling the result (63) and using the result of Theorem 1, we obtain that

$$\|I_N E_0(x)\|_{L^2(-1,1)} \leq CN^{-m} R(R+1) (\|u\|_{L^\infty(-1,1)} + \|u^{(m+1)}\|_{L^2(\Lambda_h)}). \quad (77)$$

Similarly,

$$\|I_N E_1(x)\|_{L^2(-1,1)} \leq CN^{-m} R(R+1) (\|u\|_{L^\infty(-1,1)} + \|u^{(m+1)}\|_{L^2(\Lambda_h)}). \quad (78)$$

Note that, in each subinterval δ_μ , $\mu = 0, 1, \dots, M$, $u'|_{\delta_\mu}(x) \in H^m(\delta_\mu)$. Applying (31) to $u'(x)$, we have

$$\|E_2(x)\|_{L^2(-1,1)} \leq CN^{-m} \|u^{(m+1)}\|_{L^2(\Lambda_h)}. \quad (79)$$

Applying the analysis from (67)–(69), using (31) in Lemma 2 with $m = 1$ for $b(x)$ yields

$$\|E_4\|_{L^2(-1,1)} = \|(I - I_N)b(x)\|_{L^2(-1,1)} \leq CN^{-1} \|e_0\|_{L^\infty(-1,1)}. \quad (80)$$

Using the estimate for e_0 in Theorem 1 makes the above inequality become

$$\|E_4\|_{L^2(-1,1)} \leq CN^{-m-1/2} (R\|u\|_\infty + \|u^{(m+1)}\|_{L^2(\Lambda_h)}). \quad (81)$$

Similarly,

$$\|E_3\|_{L^2(-1,1)} \leq CN^{-m-1/2} (R\|u\|_\infty + \|u^{(m+1)}\|_{L^2(\Lambda_h)}). \quad (82)$$

Using the same analysis from (67)–(69), using (31) in Lemma 2 with $m = 1$ we obtain

$$\|E_i\|_{L^2(-1,1)} \leq CN^{-1} \|e_1\|_{L^\infty(-1,1)}, \quad i = 5, 6. \quad (83)$$

Combining (75) with (77), (78), (79), (81), (82) and (83) we obtain the estimate (74) for e_1 .

Now we begin to estimate $\|e_0\|_{L^2(-1,1)}$. From (72) we have

$$\|e_0\|_{L^2(-1,1)} \leq C \left(\|E_7(x)\|_{L^2(-1,1)} + \left\| (I_N - I) \int_{-1}^x e_1(s) ds \right\|_{L^2(-1,1)} + \|e_1\|_{L^2(-1,1)} \right).$$

Using (31) for $E_7(x)$, and applying (31) with $m = 1$ to the middle term of the right-hand side of the above inequality, we have

$$\|e_0\|_{L^2(-1,1)} \leq CN^{-m-1} \|u^{(m+1)}\|_{L^2(\Lambda_h)} + C \|e_1\|_{L^2(-1,1)}, \quad (84)$$

which leads to the estimate (74) for e_0 by plugging the result (74) for e_1 into the last term of (84). \square

6. Numerical examples

In this section, we give four numerical examples. The first one is the linear case with smooth solution. The second one is the linear case with solution unsmooth at the primary discontinuous points. The third one is the nonlinear case. The fourth one is the case in which the delay is a function of the solution to the equations. These examples confirm the theoretical results obtained in the previous section.

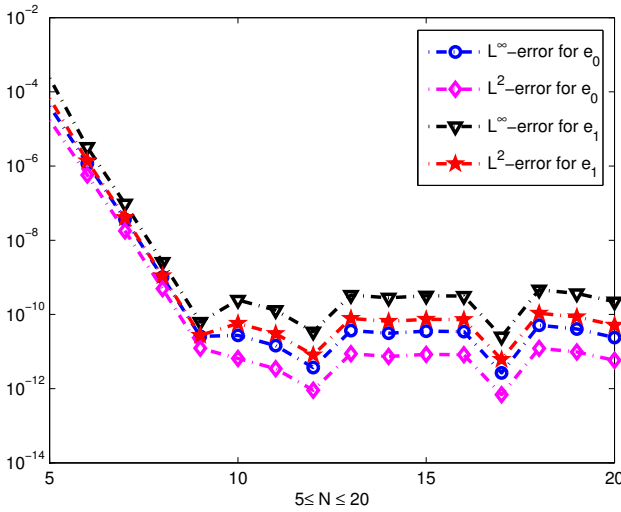


Figure 1. Example 1: Errors versus N in L^∞ and L^2 norms.

Example 1. Consider (1) with

$$T = 4, \quad a(t) = t, \quad b(t) = t^2, \quad K_1(t, s) = \sin(t + s), \quad (85)$$

$$K_2(t, s) = \cos(t + s), \quad \theta(t) = t - 1, \quad \phi(t) = e^t,$$

$$g(t) = e^t - te^t - t^2e^{t-1} + \sin t - \frac{1}{2}[e^t(\sin 2t - \cos 2t) + e^{t-1}(\cos(2t-1) + \sin(2t-1))].$$

The corresponding exact solution is $y(t) = e^t, t \in (0, T]$.

Figure 1 plots the errors for $5 \leq N \leq 20$ in both L^∞ and L^2 norms. Moreover, the corresponding errors versus several values of N are displayed in Table 1. As expected, the errors decay exponentially, which confirms our theoretical predictions. This example shows that our method is also valid for the nonvanishing delay VIDES with smooth solution.

Example 2. Consider (1) with

$$T = 3, \quad a(t) = 0, \quad b(t) = g(t) = e^t, \quad K_1(t, s) = 0,$$

$$K_2(t, s) = e^{t+s}, \quad \theta(t) = t - \left(\frac{1}{2} + \frac{1}{2}t\right), \quad \phi(t) = 1.$$

N	5	8	11	14	17	20
L^∞ -error for e_0	$4.04 \cdot 10^{-05}$	$1.01 \cdot 10^{-09}$	$1.45 \cdot 10^{-11}$	$3.11 \cdot 10^{-11}$	$2.66 \cdot 10^{-12}$	$2.38 \cdot 10^{-11}$
L^2 -error for e_0	$1.71 \cdot 10^{-05}$	$4.92 \cdot 10^{-10}$	$3.44 \cdot 10^{-12}$	$7.37 \cdot 10^{-12}$	$6.87 \cdot 10^{-13}$	$5.81 \cdot 10^{-12}$
L^∞ -error for e_1	$2.32 \cdot 10^{-04}$	$2.58 \cdot 10^{-09}$	$1.30 \cdot 10^{-10}$	$2.81 \cdot 10^{-10}$	$2.53 \cdot 10^{-11}$	$2.20 \cdot 10^{-10}$
L^2 -error for e_1	$6.65 \cdot 10^{-05}$	$1.10 \cdot 10^{-09}$	$3.03 \cdot 10^{-11}$	$6.53 \cdot 10^{-11}$	$6.09 \cdot 10^{-12}$	$5.12 \cdot 10^{-11}$

Table 1. Example 1: Errors versus N in L^∞ and L^2 norms.

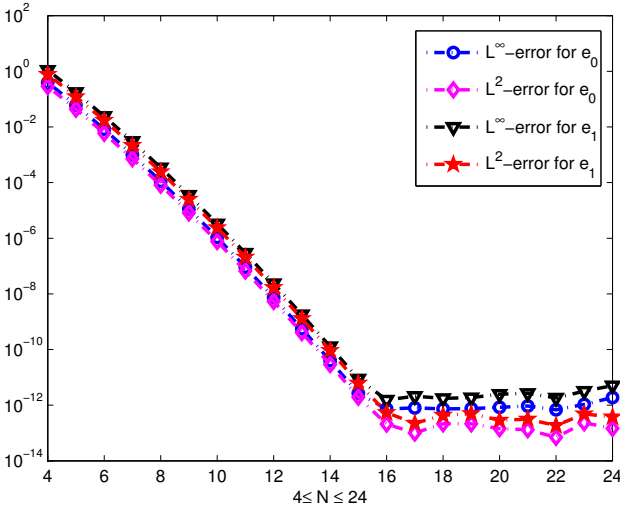


Figure 2. Example 2: Errors versus N in L^∞ and L^2 norms.

The corresponding exact solution is

$$y(t) = \begin{cases} e^t + \frac{2}{3}e^{-1/2}(e^{3t/2} - 1), & t \in (0, 1], \\ y_1(t), & t \in (1, 3], \end{cases}$$

where

$$y_1(t) := \frac{2}{3}e^{(3t-1)/2} + \frac{8}{21}e^{(7t-5)/4} + \frac{1}{2}e^t + \frac{1}{4}e^{2t-1} + \frac{16}{135}e^{(9t-7)/4} - \frac{4}{15}e^{t-1/2} - \frac{4}{9}e^{(3t-2)/2} + \frac{1}{4}e - \frac{40}{189}e^{1/2} - \frac{2}{3}e^{-1/2}. \quad (86)$$

It is worth noting that the solution of this example possesses primary discontinuous points $t = 0, 1$, where

$$0 = y^{(k)}(0-) \neq y^{(k)}(0+) \quad \text{and} \quad y^{(k)}(1-) \neq y^{(k)}(1+), \quad k \geq 1.$$

Figure 2 plots the errors for $4 \leq N \leq 24$ in both L^∞ and L^2 norms. The corresponding errors versus several values of N are displayed in Table 2. The spectral accuracy is obtained although the solution for the equation is unsmooth at the primary discontinuous points.

N	4	8	12	16	20	24
L^∞ -error for e_0	$3.99 \cdot 10^{-01}$	$1.12 \cdot 10^{-04}$	$7.43 \cdot 10^{-09}$	$7.39 \cdot 10^{-13}$	$8.38 \cdot 10^{-13}$	$1.89 \cdot 10^{-12}$
L^2 -error for e_0	$2.86 \cdot 10^{-01}$	$8.06 \cdot 10^{-05}$	$5.37 \cdot 10^{-09}$	$2.10 \cdot 10^{-13}$	$1.42 \cdot 10^{-13}$	$1.46 \cdot 10^{-13}$
L^∞ -error for e_1	$1.10 \cdot 10^{-00}$	$3.42 \cdot 10^{-04}$	$2.37 \cdot 10^{-08}$	$1.53 \cdot 10^{-12}$	$2.50 \cdot 10^{-12}$	$5.06 \cdot 10^{-12}$
L^2 -error for e_1	$7.65 \cdot 10^{-01}$	$2.44 \cdot 10^{-04}$	$1.69 \cdot 10^{-08}$	$5.39 \cdot 10^{-13}$	$2.86 \cdot 10^{-13}$	$3.75 \cdot 10^{-13}$

Table 2. Example 2: Errors versus N in L^∞ and L^2 norms.

For the nonlinear VIDes with nonvanishing delay of the form

$$\begin{aligned}
 y'(t) &= d(t, y(t), y(\theta(t))) + \int_0^t K_1(t, s, y(s)) ds + \int_0^{\theta(t)} K_2(t, s, y(s)) ds, \\
 & \hspace{25em} t \in (0, T], \\
 y(t) &= \phi(t), \quad t \in [\theta(0), 0],
 \end{aligned} \tag{87}$$

we can design a spectral-collocation method similar to the linear case. Equation (87) can be written as

$$\begin{aligned}
 u'(x) &= h(x, u(x), u(\vartheta(x))) + \int_{-1}^x R_1(x, z, u(z)) dz + \int_{-1}^{\vartheta(x)} R_2(x, z, u(z)) dz, \\
 & \hspace{25em} z \in (-1, 1], \\
 u(x) &= \psi(x), \quad x \in [\vartheta(-1), -1],
 \end{aligned} \tag{88}$$

where

$$\begin{aligned}
 u(x) &:= y(t(x)), \quad \vartheta(x) := \frac{2}{T}\theta(t(x)) - 1, \quad \psi(x) := \phi(t(x)), \\
 h(x, u(x), u(\vartheta(x))) &= \frac{T}{2}d(t(x), y(t(x)), y(\theta(t(x)))), \\
 R_1(x, z, u(z)) &:= \left(\frac{T}{2}\right)^2 K_1(t(x), s(z), y(s(z))), \\
 R_2(x, z, u(z)) &:= \left(\frac{T}{2}\right)^2 K_2(t(x), s(z), y(s(z))).
 \end{aligned} \tag{89}$$

We assume that (88) holds at the collocation points x_i^μ , where $i = 0, 1, \dots, n$ and $\mu = 0, 1, \dots, M$:

$$\begin{aligned}
 u'(x_i^\mu) &= h(x_i^\mu, u(x_i^\mu), u(\vartheta(x_i^\mu))) + \int_{-1}^{x_i^\mu} R_1(x_i^\mu, z, u(z)) dz \\
 & \quad + \int_{-1}^{\vartheta(x_i^\mu)} R_2(x_i^\mu, z, u(z)) dz \\
 &= h(x_i^\mu, u(x_i^\mu), u(\vartheta(x_i^\mu))) + \sum_{r=0}^{\mu-1} \int_{\eta_r}^{\eta_{r+1}} R_1(x_i^\mu, z, u(z)) dz \\
 & \quad + \int_{\eta_\mu}^{x_i^\mu} R_1(x_i^\mu, z, u(z)) dz \\
 & \quad + \sum_{r=0}^{\mu-2} \int_{\eta_r}^{\eta_{r+1}} R_2(x_i^\mu, z, u(z)) dz + \int_{\eta_{\mu-1}}^{\vartheta(x_i^\mu)} R_2(x_i, z, u(z)) dz.
 \end{aligned} \tag{90}$$

We use u_i^μ to approximate $u(x_i^\mu)$, v_i^μ to approximate $u(\vartheta(x_i^\mu))$, ρ_i^μ to approximate $u'(x_i^\mu)$, $i = 0, 1, \dots, N$, $\mu = 0, 1, \dots, M$, and use $u^N(x)$ to approximate $u(x)$,

$\rho^N(x)$ to approximate $u'(x)$. Similarly to (17), the numerical scheme for (88) is

$$\rho_i^\mu = h(x_i^\mu, u_i^\mu, v_i^\mu) + \varpi(x_i^\mu) + \gamma(x_i^\mu), \quad (91)$$

where

$$\begin{aligned} \varpi(x_i^\mu) &= \sum_{r=0}^{\mu-1} \frac{\eta_{r+1} - \eta_r}{2} \varpi_1^r(x_i^\mu) + \frac{\eta_{\mu+1} - \eta_\mu}{2} \frac{x_i + 1}{2} \varpi_3(x_i^\mu), \quad \mu \geq 0, \\ \varpi_1^r(x_i^\mu) &:= \sum_{k=0}^N R_1(x_i^\mu, z_r(v_k), u_k^r) \omega_k, \quad r = 0, 1, \dots, \mu - 1, \\ \varpi_3(x_i^\mu) &:= \sum_{k=0}^N R_1\left(x_i^\mu, z_\mu(z(-1, x_i, v_k)), \sum_{j=0}^N u_j^\mu F_j(z(-1, x_i, v_k))\right) \omega_k, \end{aligned}$$

and

$$\begin{aligned} \gamma(x_i^\mu) &= \begin{cases} \frac{\vartheta(x_i^0) + 1}{2} \gamma_2(x_i^0), & \mu = 0, \\ \sum_{r=0}^{\mu-2} \frac{\eta_{r+1} - \eta_r}{2} \gamma_1^r(x_i^\mu) + \frac{\eta_\mu - \eta_{\mu-1}}{2} \frac{\tilde{\vartheta}(x_i^\mu) + 1}{2} \gamma_3(x_i^\mu), & \mu > 0, \end{cases} \\ \gamma_1^r(x_i^\mu) &:= \sum_{k=0}^N R_2(x_i^\mu, z_r(v_k), u_k^r) \omega_k, \quad r = 0, 1, \dots, \mu - 2, \\ \gamma_2(x_i^0) &:= \sum_{k=0}^N R_2(x_i^0, z(-1, \vartheta(x_i^0), v_k), \psi(z(-1, \vartheta(x_i^0), v_k))) \omega_k, \\ \gamma_3(x_i^\mu) &:= \sum_{k=0}^N R_2\left(x_i^\mu, z_{\mu-1}(z(-1, \tilde{\vartheta}(x_i^\mu), v_k)), \sum_{j=0}^N u_j^{\mu-1} F_j(z(-1, \tilde{\vartheta}(x_i^\mu), v_k))\right) \omega_k. \end{aligned}$$

Combining (19) with (22) and (91), we obtain the numerical scheme for the nonlinear VIDEs (87).

Example 3. Consider (87) with

$$\begin{aligned} T = 2, \quad \phi(t) = 1, \quad \theta(t) = t - 1, \quad d(t, y(t), y(\theta(t))) &= y^2(\theta(t)) - 1 - e^{2t-1} + 2e^t, \\ K_1(t, s, y(s)) &= 0, \quad K_2(t, s, y(s)) = e^{t+s} y^2(s). \end{aligned} \quad (92)$$

The corresponding exact solution is

$$y(t) = \begin{cases} e^t, & t \in (0, 1], \\ \frac{1}{2}e^{2t-2} - \frac{1}{2}e^{2t-1} + \frac{5}{3}e^t + \frac{1}{12}e^{4t-3} - t + \frac{1}{4}e + \frac{1}{2}, & t \in (1, 2]. \end{cases}$$

In this example, the primary discontinuous points are $t = 0, 1$, where

$$0 = y^{(k)}(0-) \neq y^{(k)}(0+) \quad \text{and} \quad y^{(k)}(1-) \neq y^{(k)}(1+), \quad k \geq 1.$$

Numerical errors versus several values of N are displayed in Table 3 and Figure 3. These results indicate that the desired spectral accuracy is obtained.

N	4	8	12	16	20	24
L^∞ -error for e_0	$5.67 \cdot 10^{-02}$	$1.56 \cdot 10^{-05}$	$8.55 \cdot 10^{-10}$	$1.15 \cdot 10^{-13}$	$1.28 \cdot 10^{-13}$	$3.06 \cdot 10^{-13}$
L^2 -error for e_0	$3.51 \cdot 10^{-02}$	$9.75 \cdot 10^{-06}$	$5.36 \cdot 10^{-10}$	$3.00 \cdot 10^{-14}$	$2.23 \cdot 10^{-14}$	$2.24 \cdot 10^{-14}$
L^∞ -error for e_1	$2.12 \cdot 10^{-01}$	$6.13 \cdot 10^{-05}$	$3.40 \cdot 10^{-09}$	$3.55 \cdot 10^{-13}$	$3.48 \cdot 10^{-13}$	$9.02 \cdot 10^{-13}$
L^2 -error for e_1	$1.28 \cdot 10^{-01}$	$3.80 \cdot 10^{-05}$	$2.11 \cdot 10^{-09}$	$8.71 \cdot 10^{-14}$	$6.06 \cdot 10^{-14}$	$6.39 \cdot 10^{-14}$

Table 3. Example 3: Errors versus N in L^∞ and L^2 norms.

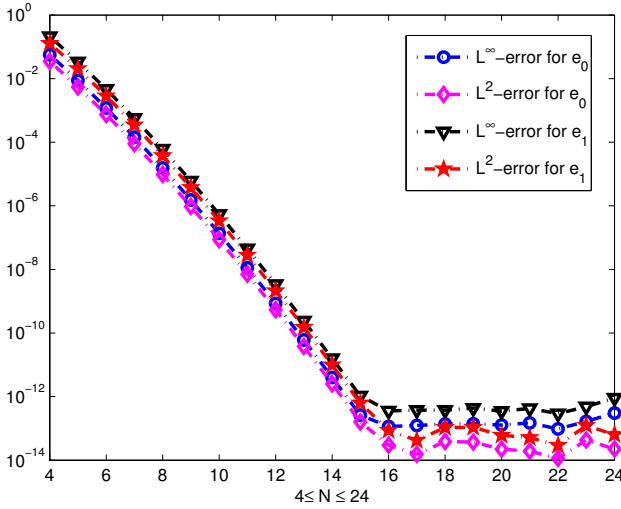


Figure 3. Example 3: Errors versus N in L^∞ and L^2 norms.

Example 4. Now we consider the case where the delay is a function of the solution; i.e.,

$$\begin{aligned}
 y'(t) &= d(t, y(t), y(\theta(t, y(t)))) + \int_0^t K_1(t, s, y(s)) ds + \int_0^{\theta(t, y(t))} K_2(t, s, y(s)) ds, \\
 & \quad t \in [0, T], \\
 y(t) &= \phi(t), \quad t \in [\theta(0, y(0)), 0].
 \end{aligned} \tag{93}$$

If we take

$$\begin{aligned}
 T &= 2, \quad \phi(t) = 2.5, \quad \theta(t, y(t)) = t - y(t), \\
 d(t, y(t), y(\theta(t, y(t)))) &= y^2(t) + y^2(\theta(t, y(t))) + g(t), \\
 g(t) &= -\frac{1}{2}e^{-t} - \frac{1}{4}(e^{-t} + 4)^2 \\
 & \quad - \frac{1}{4}e^t \left(-\frac{1}{3}e^{-3t} - 4e^{-2t} - 16e^{-t} + \frac{1}{3} + 20 \right) - (2.5)^2 \left(t - \frac{1}{2}e^{-t} - 1 \right), \\
 K_1(t, s, y(s)) &= e^{t-s} y^2(s), \quad K_2(t, s, y(s)) = y^2(s),
 \end{aligned} \tag{94}$$

N	2	4	6	8	10	12
L^∞ -error for e_0	$3.32 \cdot 10^{-02}$	$5.33 \cdot 10^{-04}$	$8.78 \cdot 10^{-07}$	$2.87 \cdot 10^{-09}$	$6.54 \cdot 10^{-12}$	$2.22 \cdot 10^{-14}$
L^2 -error for e_0	$3.05 \cdot 10^{-02}$	$2.50 \cdot 10^{-04}$	$7.65 \cdot 10^{-07}$	$2.62 \cdot 10^{-09}$	$5.91 \cdot 10^{-12}$	$1.24 \cdot 10^{-14}$
L^∞ -error for e_1	$1.57 \cdot 10^{-01}$	$1.14 \cdot 10^{-03}$	$1.56 \cdot 10^{-06}$	$3.99 \cdot 10^{-09}$	$8.29 \cdot 10^{-12}$	$3.42 \cdot 10^{-14}$
L^2 -error for e_1	$8.73 \cdot 10^{-02}$	$6.41 \cdot 10^{-04}$	$1.06 \cdot 10^{-06}$	$3.16 \cdot 10^{-09}$	$6.75 \cdot 10^{-12}$	$1.66 \cdot 10^{-14}$

Table 4. Example 4: Errors versus N in L^∞ and L^2 norms.

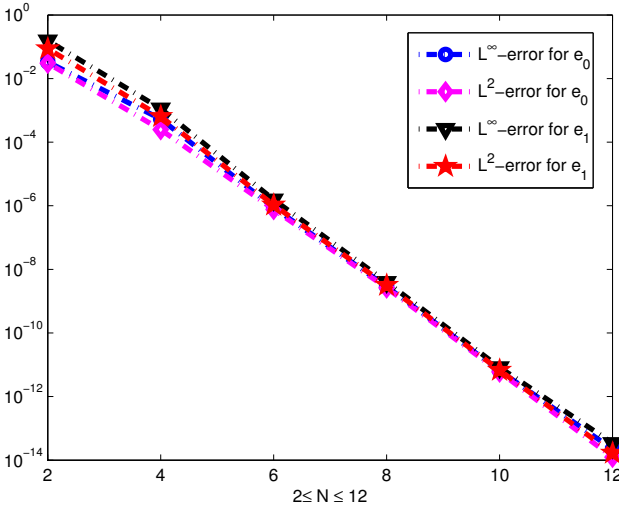


Figure 4. Example 4: Errors versus N in L^∞ and L^2 norms.

then the corresponding exact solution is

$$y(t) = \begin{cases} 2.5, & t \in [-1, 0], \\ \frac{1}{2}(e^{-t} + 4), & t \in (0, 2]. \end{cases}$$

This solution possesses a primary discontinuous point $t = 0$ where $y^{(k)}(0-) = 0$ while $y^{(k)}(0+) = (-1)^k \frac{1}{2}$, $k \geq 1$.

We use the Newton iterative method to solve the nonlinear discrete system corresponding to this example. Errors versus N are listed in Table 4 and plotted in Figure 4 from which we can see that the spectral accuracy is obtained. This example shows that our method can also handle the case where the delay is a function of the solution.

7. Conclusion and future work

We propose the Legendre spectral-collocation method to solve VIDEs with nonvanishing delay, and provide convergence analysis for the proposed method. Numerical

examples are provided to confirm the theoretical results that the numerical errors decay exponentially. The main difficulty in applying the spectral method to VIDES with nonvanishing delay is the solution of this equation possesses primary discontinuous points associated with the nonvanishing delay function. We overcome this difficulty by dividing the global definition domain of the solution into several subintervals where the solution is smooth enough. Then spectral method can be used to approximate the solution in each subinterval.

Our future work will focus on the spectral method for the Volterra functional integral and differential integral equation with nonvanishing delay.

References

- [1] K. Al-Khaled, *Numerical approximations for population growth models*, Appl. Math. Comput. **160** (2005), no. 3, 865–873. MR 2005i:65113 Zbl 1062.65142
- [2] I. Ali, *Convergence analysis of spectral methods for integro-differential equations with vanishing proportional delays*, J. Comput. Math. **29** (2011), no. 1, 49–60. MR 2012b:65214 Zbl 1249.65281
- [3] I. Ali, H. Brunner, and T. Tang, *A spectral method for pantograph-type delay differential equations and its convergence analysis*, J. Comput. Math. **27** (2009), no. 2-3, 254–265. MR 2010f:65197 Zbl 1212.65308
- [4] I. Ali, H. Brunner, and T. Tang, *Spectral methods for pantograph-type differential and integral equations with multiple delays*, Front. Math. China **4** (2009), no. 1, 49–61. MR 2009m:65099 Zbl 05567165
- [5] C. T. H. Baker and N. J. Ford, *Asymptotic error expansions for linear multistep methods for a class of delay integro-differential equations*, Bull. Soc. Math. Grèce (N.S.) **31** (1990), 5–18. MR 92e:65170 Zbl 0746.65097
- [6] C. T. H. Baker and C. A. H. Paul, *Parallel continuous Runge–Kutta methods and vanishing lag delay differential equations*, Adv. Comput. Math. **1** (1993), no. 3-4, 367–394. MR 94g:65140 Zbl 0824.65055
- [7] C. T. H. Baker and A. Tang, *Stability analysis of continuous implicit Runge–Kutta methods for Volterra integro-differential systems with unbounded delays*, Appl. Numer. Math. **24** (1997), no. 2-3, 153–173. MR 99a:45021 Zbl 0878.65122
- [8] A. Bellen and M. Zennaro, *Numerical methods for delay differential equations*, Clarendon/Oxford University Press, New York, 2003. MR 2004i:65001 Zbl 1038.65058
- [9] G. A. Bocharov and F. A. Rihan, *Numerical modelling in biosciences using delay differential equations*, J. Comput. Appl. Math. **125** (2000), no. 1-2, 183–199. MR 1803191 Zbl 0969.65124
- [10] F. Brauer and P. van den Driessche, *Models for transmission of disease with immigration of infectives*, Math. Biosci. **171** (2001), no. 2, 143–154. MR 2002b:92035 Zbl 0995.92041
- [11] H. Brunner, *Collocation methods for nonlinear Volterra integro-differential equations with infinite delay*, Math. Comp. **53** (1989), no. 188, 571–587. MR 90m:65227 Zbl 0681.65105
- [12] ———, *The numerical solution of neutral Volterra integro-differential equations with delay arguments*, Ann. Numer. Math. **1** (1994), no. 1-4, 309–322. MR 96m:45014 Zbl 0828.65146
- [13] ———, *Collocation methods for Volterra integral and related functional differential equations*, Cambridge Monographs on Applied and Computational Mathematics, no. 15, Cambridge University Press, 2004. MR 2005k:65002 Zbl 1059.65122

- [14] ———, *Recent advances in the numerical analysis of Volterra functional differential equations with variable delays*, J. Comput. Appl. Math. **228** (2009), no. 2, 524–537. MR 2010c:65251 Zbl 1170.65103
- [15] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral methods: fundamentals in single domains*, Springer, Berlin, 2006. MR 2007c:65001 Zbl 1093.76002
- [16] J. Chattopadhyay, R. R. Sarkar, and A. el Abdllaoui, *A delay differential equation model on harmful algal blooms in the presence of toxic substances*, Math. Med. Biol. **19** (2002), no. 2, 137–161. Zbl 1013.92046
- [17] Y. Chen, X. Li, and T. Tang, *A note on Jacobi spectral-collocation methods for weakly singular Volterra integral equations with smooth solutions*, J. Comput. Math. **31** (2013), no. 1, 47–56. MR 3021419
- [18] Y. Chen and T. Tang, *Spectral methods for weakly singular Volterra integral equations with smooth solutions*, J. Comput. Appl. Math. **233** (2009), no. 4, 938–950. MR 2010k:65304 Zbl 1186.65161
- [19] ———, *Convergence analysis of the Jacobi spectral-collocation methods for Volterra integral equations with a weakly singular kernel*, Math. Comp. **79** (2010), no. 269, 147–167. MR 2011c:65298 Zbl 1207.65157
- [20] J. M. Cushing, *Integrodifferential equations and delay models in population dynamics*, Lecture Notes in Biomathematics, no. 20, Springer, Berlin, 1977. MR 58 #15300 Zbl 0363.92014
- [21] W. H. Enright and M. Hu, *Continuous Runge–Kutta methods for neutral Volterra integro-differential equations with delay*, Appl. Numer. Math. **24** (1997), no. 2-3, 175–190. MR 98g:65126 Zbl 0876.65089
- [22] D. J. Evans and K. R. Raslan, *The Adomian decomposition method for solving delay differential equation*, Int. J. Comput. Math. **82** (2005), no. 1, 49–54. MR 2159285 Zbl 1069.65074
- [23] S. A. Gourley, Y. Kuang, and J. D. Nagy, *Dynamics of a delay differential equation model of hepatitis B virus infection*, J. Biol. Dyn. **2** (2008), no. 2, 140–153. MR 2009g:92047 Zbl 1140.92014
- [24] J. S. Hesthaven, *From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex*, SIAM J. Numer. Anal. **35** (1998), no. 2, 655–676. MR 99b:65009 Zbl 0933.41004
- [25] J. S. Hesthaven and T. Warburton, *Nodal discontinuous Galerkin methods: algorithms, analysis, and applications*, Texts in Applied Mathematics, no. 54, Springer, New York, 2008. MR 2008k:65002 Zbl 1134.65068
- [26] D. Jiang, J. Wei, and B. Zhang, *Positive periodic solutions of functional differential equations and population models*, Electron. J. Differential Equations **2002** (2002), Article ID #71. MR 2003i:34161 Zbl 1010.34065
- [27] Y.-J. Jiang, *On spectral methods for Volterra-type integro-differential equations*, J. Comput. Appl. Math. **230** (2009), no. 2, 333–340. MR 2010d:45011 Zbl 1202.65170
- [28] Y. Kuang, *Delay differential equations with applications in population dynamics*, Mathematics in Science and Engineering, no. 191, Academic Press, Boston, 1993. MR 94f:34001 Zbl 0777.34002
- [29] S. Li, *High order contractive Runge–Kutta methods for Volterra functional differential equations*, SIAM J. Numer. Anal. **47** (2010), no. 6, 4290–4325. MR 2011a:65167 Zbl 1221.65166
- [30] X. Li and T. Tang, *Convergence analysis of Jacobi spectral collocation methods for Abel–Volterra integral equations of second kind*, Front. Math. China **7** (2012), no. 1, 69–84. MR 2876899 Zbl 1260.65111

- [31] T. Lin, Y. Lin, M. Rao, and S. Zhang, *Petrov–Galerkin methods for linear Volterra integro-differential equations*, SIAM J. Numer. Anal. **38** (2000), no. 3, 937–963. MR 2001e:65227 Zbl 0983.65138
- [32] A. Makroglou, *A block-by-block method for the numerical solution of Volterra delay integro-differential equations*, Computing **30** (1983), no. 1, 49–62. MR 85f:65125 Zbl 0499.65064
- [33] P. Nevai, *Mean convergence of Lagrange interpolation, III*, Trans. Amer. Math. Soc. **282** (1984), no. 2, 669–698. MR 85c:41009 Zbl 0577.41001
- [34] H. J. Oberle and H. J. Pesch, *Numerical treatment of delay differential equations by Hermite interpolation*, Numer. Math. **37** (1981), no. 2, 235–255. MR 83a:65077 Zbl 0469.65057
- [35] J. M. Ortega, *Numerical analysis: a second course*, Academic Press, New York, 1972. MR 53 #6967 Zbl 0248.65001
- [36] B. V. Riley, *The numerical solution of Volterra integral equations with nonsmooth solutions based on sinc approximation*, Appl. Numer. Math. **9** (1992), no. 3-5, 249–257. MR 93b:65210 Zbl 0757.65148
- [37] M. Shakourifar and M. Dehghan, *On the numerical solution of nonlinear systems of Volterra integro-differential equations with delay arguments*, Computing **82** (2008), no. 4, 241–260. MR 2009j:45019 Zbl 1154.65098
- [38] M. Shakourifar and W. H. Enright, *Reliable approximate solution of systems of Volterra integro-differential equations with time-dependent delays*, SIAM J. Sci. Comput. **33** (2011), no. 3, 1134–1158. MR 2012e:65325 Zbl 1233.65101
- [39] ———, *Superconvergent interpolants for collocation methods applied to Volterra integro-differential equations with delay*, BIT **52** (2012), no. 3, 725–740. MR 2965299 Zbl 1255.65250
- [40] L. F. Shampine and S. Thompson, *Solving DDEs in MATLAB*, Appl. Numer. Math. **37** (2001), no. 4, 441–458. MR 2002c:65106 Zbl 0983.65079
- [41] J. Shen and T. Tang, *Spectral and high-order methods with applications*, Mathematics Monograph Series, no. 3, Science Press, Beijing, 2006. MR 2012b:65001 Zbl 1234.65005
- [42] T. Tang and X. Xu, *Accuracy enhancement using spectral postprocessing for differential equations and integral equations*, Commun. Comput. Phys. **5** (2009), no. 2-4, 779–792. MR 2010d:65178
- [43] T. Tang, X. Xu, and J. Cheng, *On spectral methods for Volterra integral equations and the convergence analysis*, J. Comput. Math. **26** (2008), no. 6, 825–837. MR 2010c:65256 Zbl 1174.65058
- [44] S. Thompson and L. F. Shampine, *A friendly Fortran DDE solver*, Appl. Numer. Math. **56** (2006), no. 3-4, 503–516. MR 2207606 Zbl 1089.65062
- [45] M. Villasana and A. Radunskaya, *A delay differential equation model for tumor growth*, J. Math. Biol. **47** (2003), no. 3, 270–294. MR 2004j:92012 Zbl 1023.92014
- [46] V. Volterra, *Variazioni e fluttuazioni del numero d'individui in specie animali conviventi*, Memorie del Regio Comitato Talassografico Italiano **131** (1927). JFM 52.0450.06
- [47] ———, *The general equations of biological strife in the case of historical actions*, Proc. Edinburgh Math. Soc. (2) **6** (1939), no. 1, 4–10. Zbl 0021.34003
- [48] Z. Wan, Y. Chen, and Y. Huang, *Legendre spectral Galerkin method for second-kind Volterra integral equations*, Front. Math. China **4** (2009), no. 1, 181–193. MR 2009k:65285 Zbl 05567172
- [49] Y. Wei and Y. Chen, *Convergence analysis of the Legendre spectral collocation methods for second order Volterra integro-differential equations*, Numer. Math. Theory Methods Appl. **4** (2011), no. 3, 419–438. MR 2012k:65177 Zbl 1265.65278

- [50] ———, *Convergence analysis of the spectral methods for weakly singular Volterra integro-differential equations with smooth solutions*, Adv. Appl. Math. Mech. **4** (2012), no. 1, 1–20. MR 2876648 Zbl 1262.45005
- [51] ———, *Legendre spectral collocation methods for pantograph Volterra delay-integro-differential equations*, J. Sci. Comput. **53** (2012), no. 3, 672–688. MR 2996451 Zbl 1264.65218
- [52] Z. Xie, X. Li, and T. Tang, *Convergence analysis of spectral Galerkin methods for Volterra type integral equations*, J. Sci. Comput. **53** (2012), no. 2, 414–434. MR 2983100 Zbl 06198196
- [53] C. Zhang and S. Vandewalle, *General linear methods for Volterra integro-differential equations with memory*, SIAM J. Sci. Comput. **27** (2006), no. 6, 2010–2031. MR 2006k:65394 Zbl 1104.65133
- [54] W. Zhang and M. Fan, *Periodicity in a generalized ecological competition system governed by impulsive differential equations with delays*, Math. Comput. Modelling **39** (2004), no. 4-5, 479–493. MR 2046535 Zbl 1065.92066
- [55] H. ZivariPiran and W. H. Enright, *An efficient unified approach for the numerical solution of delay differential equations*, Numer. Algorithms **53** (2010), no. 2-3, 397–417. MR 2011j:65133 Zbl 1184.65071

Received November 17, 2012. Revised August 28, 2013.

YANPING CHEN: yanpingchen@scnu.edu.cn

School of Mathematics Science, South China Normal University, Guangzhou 510631, China

ZHENDONG GU: guzhd@qq.com

School of Mathematics and Computational Science, Xiangtan University, Xiangtan 411105, Hunan, China

A CARTESIAN GRID EMBEDDED BOUNDARY METHOD FOR THE COMPRESSIBLE NAVIER–STOKES EQUATIONS

DANIEL T. GRAVES, PHILLIP COLELLA, DAVID MODIANO,
JEFFREY JOHNSON, BJORN SJOGREEN AND XINFENG GAO

We present an unsplit method for the time-dependent compressible Navier–Stokes equations in two and three dimensions. We use a conservative, second-order Godunov algorithm. We use a Cartesian grid, embedded boundary method to resolve complex boundaries. We solve for viscous and conductive terms with a second-order semiimplicit algorithm. We demonstrate second-order accuracy in solutions of smooth problems in smooth geometries and demonstrate robust behavior for strongly discontinuous initial conditions in complex geometries.

1. Introduction

In this paper, we present an unsplit method for the time-dependent compressible Navier–Stokes equations in two and three dimensions. This algorithm is an extension of the algorithm in [9] to flows with viscous and thermal diffusion. The Navier–Stokes equations contain parabolic terms that arise from conductivity and viscosity. There are several methods to advance these terms. In [10], for example, a kinetic energy equation is evolved to get a stable approximation to the viscous term in the energy equation. This solution is elegant but also difficult to extend to multiple dimensions. We use a conservative, semiimplicit method in which the hyperbolic terms are advanced explicitly and the parabolic terms advanced implicitly. This approach to the compressible Navier–Stokes equations has been used without embedded boundaries [3; 30; 16; 14; 11]. Our algorithm follows the basic outline in the mapped grid algorithm presented in [30], in which the velocity and temperature evolution are split. They use a Crank–Nicolson time evolution with the energy-momentum coupling term treated explicitly. We use a hybrid approach to energy-momentum coupling. Also, since Crank–Nicolson has been shown to be marginally

Colella’s research at Lawrence Livermore National Laboratory (LBNL) was supported financially by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract DE-AC02-05CH11231. Sjogreen’s participation funded under the auspices of the U.S. Department of Energy by LBNL under contract DE-AC52-07NA27344.

MSC2010: primary 76-04; secondary 35-04.

Keywords: Cartesian grid embedded boundaries, compressible Navier–Stokes, block-structured adaptive mesh refinement.

stable in certain cases [23], we use the L_0 -stable algorithm presented in [31] for elliptic coupling. We present our changes to the [31] algorithm that were necessary to make the linear equations tractable in the presence of small cells. This algorithm has been implemented with adaptive mesh refinement (AMR) as described in [4; 2]. All our cut cells are refined to the finest level, reducing all coarse-fine interactions (such as refluxing and coarse-fine interpolation) to exactly those described in [30].

Dragojlovic et al. [12] present a two-dimensional algorithm for viscous, conducting compressible flow with embedded boundaries. They use a split hyperbolic scheme, explicit updates of the viscous state and the (formally inconsistent) extended state algorithm developed in [24]. Our algorithm uses an unsplit scheme (as seen in [8; 25; 1]) and works in two and three dimensions. Ghias et al. [13] present an immersed boundary method to solve the same set of equations for subsonic applications. Hartmann et al. [15] present a cut-cell method that uses a form of cell merging to achieve small-cell stability. Berger et al. [5] survey a wide variety of these algorithmic permutations. We use redistribution (first presented by Chern et al. [7]) for small-cell stability. We use the (formally consistent) approach in [9] to construct extended states. To evaluate viscous fluxes at the embedded boundary we use the ray-casting algorithm developed in [18] for Poisson's equation. Also, for increased stability, we treat the viscous stress and conductivity terms implicitly.

This algorithm is suitable for use in applications where compressibility is important and the geometries are complex. Our target application is flow inside of capillary tubes in laser wakefield particle accelerators. In these accelerators, the pressure and temperature is driven very high along the axis of a capillary tube. The resulting flow produces a low density core through which lasers are shot. The capillary is connected to fill tubes which are used to fill the capillary with gas [27; 19; 20; 29]. We present a simplified version of this problem as our example to demonstrate robustness while acknowledging that other physics in these problems (such as ionization and magnetization) are very important. We drive a capillary tube with a large pressure pulse to demonstrate the stability of the algorithm under extreme conditions. The geometric configuration is derived from the experimental set-up described in [29].

There are of course many regimes for which the compressible Navier Stokes equations are relevant. The regime of interest for this algorithm has substantial compressibility effects (including shocks) as well as substantial viscous effects. We are also interested in time-accurate (as opposed to steady state calculations). For algorithm validation, we run several examples which demonstrate the efficacy of the algorithm in this regime.

First we present convergence tests demonstrating second-order solution error accuracy in two and three dimensions. For these tests, we use a smooth, subsonic ($M = 0.5$) flow inside a sphere. This demonstrates that, even with compressibility

effects, we get the expected convergence rate for smooth problems.

Next, for more a quantitative validations, we present a boundary layer calculation and a viscous shock reflection calculation. Charest et al. [6] present a low-Mach-number algorithm for steady state calculations. They present a boundary layer calculation that reproduces the behavior of the similarity solution which emerges from analysis (a Blasius boundary layer profile). We present a similar run which also reproduces Blasius behavior. This demonstrates that the algorithm has correct boundary layer behavior.

Glaz et al. [14] present a comparison between inviscid calculations of shock reflections and experimental results. They show a case where viscous effects cause substantial changes in the reflection pattern. We present both viscous and inviscid calculations of the same problem and show good agreement with their results. This demonstrates, that even in this very complex, time-dependent flow, we compare well with experiment.

2. Notation

Cartesian grids with embedded boundaries are useful to describe finite-volume representations of solutions to partial differential equations in the presence of irregular boundaries. In Figure 1, the gray area represents the region excluded from the solution domain. The underlying description of space is given by rectangular control volumes on a Cartesian mesh $\Upsilon_i = [(\mathbf{i} - \frac{1}{2}\mathbf{v})h, (\mathbf{i} + \frac{1}{2}\mathbf{v})h]$, $\mathbf{i} \in \mathbb{Z}^D$, where D is the dimensionality of the problem, h is the mesh spacing, and \mathbf{v} is the vector whose entries are all one. Given an irregular domain Ω , we obtain control volumes $V_i = \Upsilon_i \cap \Omega$ and faces $A_{i \pm e^d/2}$ which are the intersection of the boundary of ∂V_i with the coordinate planes $\{\mathbf{x} : x_d = (i_d \pm \frac{1}{2})h\}$. We also define A_i^B to be the intersection of the boundary of the irregular domain with the Cartesian control volume: $A_i^B = \partial\Omega \cap \Upsilon_i$. For ease of exposition, we will assume here that there is only one control volume per Cartesian cell. The algorithm described here has been generalized to allow for boundaries whose width is less that the mesh spacing.

To construct finite-volume methods using this description, we will need several quantities derived from these geometric objects.

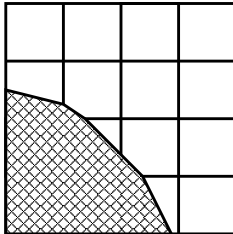


Figure 1. Illustration of cut cells. The shaded area is outside the solution domain.

- Volume fractions κ and area fraction α :

$$\kappa_i = \frac{|V_i|}{h^D}, \quad \alpha_{i+e_s/2} = \frac{|A_{i+e_s^d/2}|}{h^{D-1}}, \quad \alpha_i^B = \frac{|A_i^B|}{h^{D-1}}.$$

- The centroids of the faces and of A_i^B ; and \mathbf{n} , the average of outward normal of $\partial\Omega$ over A_i^B .

$$\mathbf{x}_{i+e^d/2} = \frac{1}{|A_{i+\frac{1}{2}e^d}|} \int_{A_{i+e^d/2}} \mathbf{x} dA - (\mathbf{i} + \mathbf{e}^d/2)h,$$

$$\mathbf{x}_i^B = \frac{1}{|A_i^B|} \int_{A_i^B} \mathbf{x} dA - \mathbf{i}h, \quad \mathbf{n}_i = \frac{1}{|A_i^B|} \int_{A_i^B} \mathbf{n} dA.$$

Here D is the dimension of space and $1 \leq d \leq D$. We assume we can compute all derived quantities to $O(h^2)$. With just these geometric descriptors, we can define a conservative discretization of the divergence operator. Let $\vec{F} = (F^1 \dots F^D)$ be a function of \mathbf{x} , then

$$\nabla \cdot \vec{F} \approx \frac{1}{|V_i|} \int_{V_i} \vec{F} dV = \frac{1}{|V_i|} \int_{\partial V_i} \vec{F} \cdot \mathbf{n} dA.$$

We discretize the divergence of the flux as

$$\kappa D(F)_i = \frac{1}{h} \left(\sum_{d=1}^D \sum_{\pm=+,-} \pm \alpha_{i \pm e^d/2} F^d(\mathbf{x}_{i \pm e^d/2}) + \alpha_i^B \mathbf{n}_i \cdot \vec{F}(\mathbf{x}_i^B) \right), \quad (1)$$

where (1) is obtained by replacing the normal components of the vector field \vec{F} with the values at the centroids. This converges to the exact divergence by the relation $D(F)_i = \nabla \cdot F + O(h/\kappa_i)$ in cells which intersect the embedded boundary and converges to $O(h^2)$ away from the boundary. The elliptic operators in this calculation all take the form

$$L(\phi) = a(\mathbf{x})\phi + D(F(\phi)).$$

We refer to a in this context as the identity coefficient.

3. System of equations

We are solving the compressible Navier–Stokes equations, given here in conservation form with hyperbolic terms to the left and elliptic terms to the right.

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0, \\
\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u} + p \mathbf{I}) &= \nabla \cdot \boldsymbol{\sigma}, \\
\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho \mathbf{u} E + \mathbf{u} p) &= \nabla \cdot (\boldsymbol{\sigma} \mathbf{u}) + \nabla \cdot (\boldsymbol{\xi} (\nabla T)).
\end{aligned} \tag{2}$$

In these equations, ρ is the mass density, \mathbf{u} is the velocity, $\boldsymbol{\xi}$ is the thermal conductivity, p is the pressure, and T is the temperature. The shear stress tensor $\boldsymbol{\sigma}$ is given by

$$\boldsymbol{\sigma} = \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \lambda (\nabla \cdot \mathbf{u}) \mathbf{I},$$

where μ and λ are the viscosity coefficients (typically $\lambda = -\frac{2}{3}\mu$). The total energy is given by $E = e + \frac{1}{2}|\mathbf{u}|^2$; the internal energy is given by $e = C_v T$ (where C_v is the specific heat at constant volume). The fluid is assumed to be an ideal gas ($p = C_v(\gamma - 1)\rho T$).

4. Algorithm description

We define $U = (\rho, \rho \mathbf{u}, \rho E)$ and we define $L^H(U) = \nabla \cdot F$, the divergence of the hyperbolic flux. The flux is given by

$$F = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} \mathbf{u} + p \mathbf{I} \\ \rho \mathbf{u} E + \mathbf{u} p \end{pmatrix}.$$

The divergence and the fluxes are computed in the same way as in [9]. To summarize, a Taylor series extrapolation is done to produce second-order (in both space and time) approximations to the fluxes at the centroids of the faces. A conservative approximation to the divergence ($D^c(F)$) is computed using (1). Ideally, we would use $D^c(F)$ for our hyperbolic divergence. The difficulty with this approach is that the CFL (Courant–Friedrichs–Lewy) stability constraint on the time step for an algorithm using the conservative divergence for an explicit update is at best

$$\Delta t = O\left(\frac{h}{v_i^{\max}} (\kappa_i)^{1/D}\right),$$

where v_i^{\max} is the magnitude of the maximum wave speed for the i -th control volume. This is the well-known small-cell problem for embedded boundary methods. Instead, we compute a stable, nonconservative approximation to the divergence ($D^{nc}(F)$) using an extended state where necessary and ignoring the embedded boundary. This extended state is extrapolated from the interior. The effective divergence is

$$L^H(U) = \kappa D^c(F) + (1 - \kappa) D^{nc}(F).$$

The mass difference (δM) between using L^H and using only the conservative divergence $D^c(F)$ is given by

$$\delta M = \kappa(1 - \kappa)(D^c(F) - D^{nc}(F)).$$

This mass difference is redistributed to neighboring cells. The redistribution algorithm is described in [7]. This hybrid formulation preserves conservation and allows this algorithm to be stable using a time step constraint based on full cells. We compute our time step as follows:

$$\Delta t = \frac{C_F h}{W^{\max}}, \quad (3)$$

where W^{\max} is the maximum wave speed in the problem and C_F is the Courant number ($0 < C_F < 1$).

Define L^v to be the elliptic terms in the system of equations

$$L^v(U)_i = \begin{pmatrix} 0 \\ L^m(\mathbf{u})_i \\ L^k(T)_i + L^d(\mathbf{u})_i \end{pmatrix}.$$

The term L^m is the discretization of the viscous stress term ($L^m \nabla \cdot \sigma$) and is described in Section 5.3. The term $L^k \approx \nabla \cdot \xi \nabla T$ is a discretization of the heat conduction term and is described in Section 5.2. The term $L^d \approx \nabla \cdot (\sigma \mathbf{u})$ is the viscous heating term and is described in Section 5.1.

4.1. Outline. We begin with the state at time $U^n = U(n\Delta t)$, we advance the solution as follows.

1. Compute U^* , the solution advanced explicitly using only hyperbolic terms.

$$U_i^* = U_i^n - \Delta t L_i^H(U^n).$$

This produces the final value of density ($\rho^{n+1} = \rho^*$). From U^* , we compute \mathbf{u}^* and T^* , the intermediate values of velocity and temperature (which exclude the effects of conduction and viscosity).

2. Compute L_0 -stable approximations to the momentum diffusion $L^m(U) = \nabla \cdot \sigma$ by advancing the diffusion equation

$$\rho \frac{\partial \mathbf{u}}{\partial t} = L^m(\mathbf{u})$$

using the method described in Section 5:

$$\mathbf{u}^{n+1} = G_{L^m}(\rho^{n+1})\mathbf{u}^*.$$

The symbol G is defined in (6). The stable approximation to $L^m(\mathbf{u})$ is calculated as

$$(L^m(\mathbf{u}))^{n+\frac{1}{2}} = \rho^{n+1} \frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t},$$

giving us the final value of momentum:

$$(\rho\mathbf{u})^{n+1} = (\rho\mathbf{u})^* + \Delta t(L^m(\mathbf{u}))^{n+\frac{1}{2}}.$$

The operator L^m is described in [Section 5.3](#).

3. Using the value of \mathbf{u} calculated above, calculate the viscous dissipation of energy ($L^d \approx \nabla \cdot (\sigma\mathbf{u})$) as described in [Section 4.2](#). We then update the energy with the term

$$(\rho E)^{**} = (\rho E)^* + \Delta t L^d(\mathbf{u}^{n+1}).$$

From E^{**} , we compute the intermediate value of temperature T^{**} .

4. Compute L_0 -stable approximations to the conduction term

$$L^k(T) = \nabla \cdot \xi \nabla T$$

by advancing the diffusion equation

$$\rho C_v \frac{\partial T}{\partial t} = L^k(T)$$

using the method described in [Section 5](#):

$$T^{n+1} = G_{L^m}(\rho^{n+1} C_v) T^{**},$$

where G is described in [Equation \(6\)](#). The stable approximation to $L^k(T)$ is computed by

$$(L^k(T))^{n+\frac{1}{2}} = \rho^{n+1} C_v \frac{T^{n+1} - T^{**}}{\Delta t},$$

giving us the final value of energy:

$$(\rho E)^{n+1} = (\rho E)^{**} + \Delta t (L^k(T))^{n+\frac{1}{2}}.$$

The operator $L^k(T)$ is described in [Section 5.2](#).

4.2. Viscous dissipation calculation. To avoid small-cell instabilities, we split up the $L^d(U)$ into conservative and nonconservative approximations much as we did with L^H . The conservative approximation to $L^{d,c} = \nabla \cdot (\sigma\mathbf{u})$ is described in [Section 5.1](#). The nonconservative form of the operator is given by the volume-weighted average of the neighbor's conservative operator evaluations. Define $N(i)$ to be the set of cells reachable from i by a unit monotone path. The nonconservative approximation of L^d is

$$L^{d,nc}(\mathbf{u})_i = \frac{\sum_{j \in N(i)} (\kappa L^{d,c}(\mathbf{u}))_j}{\sum_{j \in N(i)} \kappa_j}.$$

We use a linear combination of conservative and nonconservative versions of the divergence to advance the solution:

$$L^d(U) = \kappa L^{d,c}(\mathbf{u}) + (1 - \kappa)(L^{d,nc}(\mathbf{u})).$$

To preserve conservation, we compute the energy difference between this version and the conservative version:

$$\delta E = \Delta t \kappa (1 - \kappa) (L^{d,c} - L^{d,nc}).$$

We push this energy correction δE into the solution implicitly. First we set a right hand side $R = 0$ and redistribute δE into the cells of R that can be reached by a unit monotone path (as described in [7]). We then solve for a temperature difference that can account for this energy using the conduction operator

$$(\rho^{n+1} C_v I - \Delta t L^k) \delta T = \Delta t R.$$

This change in temperature is interpreted as an increment to the energy as follows:

$$(\delta E)^{**} = \rho^* C_v \delta T.$$

We add $(\delta E)^{**}$ into E^{**} .

5. Stable parabolic discretizations

Twizell et al. [31] present a second-order L_0 -stable algorithm to advance the constant coefficient heat equation. Given the equation

$$\frac{\partial \phi}{\partial t} = \nu L \phi, \tag{4}$$

their time advance takes the form

$$\phi^{n+1} = (I - \mu_1 L)^{-1} (I - \mu_2 L)^{-1} (I + \mu_3 L) \phi^n, \tag{5}$$

where μ_1, μ_2, μ_3 are constants. In the present algorithm we have two parabolic equations of the form

$$a \frac{\partial \phi}{\partial t} = L(\phi),$$

where $a = a(\mathbf{x}) > 0$ is the identity coefficient. Define the operator $M(\phi) = L(\phi)/a$. In the case of our viscous operator (Section 5.3) $M^m = L^m(\mathbf{u})/\rho$ and the case of conduction (Section 5.2), $M^k(T) = L^k(T)/(\rho C_v)$. In both cases, the denominators are positive and restricted away from zero. In each case, a naive interpretation of (5) yields

$$\phi^{n+1} = (I - \mu_1 M)^{-1} (I - \mu_2 M)^{-1} (I + \mu_3 M) \phi^n.$$

This is problematic in the presence of small cells because this would involve dividing by the volume fraction to evaluate M (see (1)) and volume fractions here can be arbitrarily close to zero. Using the matrix identity $(AB)^{-1} = B^{-1}A^{-1}$, we refactor the preceding equation, obtaining

$$\phi^{n+1} = G_L(a)\phi = (\kappa a I - \mu_1 \kappa L)^{-1}(\kappa a)(\kappa a I - \mu_2 \kappa L)^{-1}(\kappa a I + \mu_3 \kappa L)\phi^n. \quad (6)$$

This is the implicit advance we use for stable discretizations of $L^m(\mathbf{u})$ and $L^k(T)$.

5.1. Viscous heating operator. The viscous heating operator flux is an approximation to the shear stress dotted with the velocity ($F^h = \sigma \cdot \mathbf{u}$):

$$F^h = (\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \lambda I \nabla \cdot \mathbf{u}) \cdot \mathbf{u}. \quad (7)$$

We compute the shear stress as described in Section 5.3. To get face-centered velocities, we average from neighboring cells:

$$\mathbf{u}_{i+e^d/2} = \frac{1}{2}(\mathbf{u}_{i+e^d} + \mathbf{u}_i).$$

At embedded boundaries and domain boundaries we set this flux to zero because the no slip condition requires that $\mathbf{u}|_{\partial\Omega} = 0$. We then can find the conservative discretization of the operator $L^{d,c}$ as given by (1).

5.2. Conductivity operator. Our operator for heat conduction

$$L^k(T) = \nabla \cdot (\xi \nabla T)$$

is an extension to variable coefficients of the operator described by Schwartz et al. [28]. The flux at face centers for the discretization in (1) is given by

$$F_{i+e^d/2}^T = \xi_{i+e^d/2} \frac{T_{i+e^d} - T_i}{\Delta x}.$$

Since we are representing thermally insulated embedded boundaries, $F_B^T = 0$. Given these fluxes, discretization of the operator is given by (1).

5.3. Viscous stress operator. For viscous diffusion, we first calculate the cell-centered gradient of the solution using centered differences:

$$\frac{\partial u^{d1}}{\partial x_{d2}} = \frac{u_{i+e^{d2}}^{d1} - u_{i-e^{d2}}^{d1}}{2\Delta x}.$$

The face centered gradient uses this gradient for tangential gradients and differences normal gradients directly:

$$(\nabla \mathbf{u})_{i+e^{d'}/2}^{d'} = \begin{cases} (1/h)(\mathbf{u}_{i+e^d} - \mathbf{u}_i) & \text{if } d = d', \\ \frac{1}{2}((\nabla \mathbf{u})_{i+e^d}^{d'} + (\nabla \mathbf{u})_i^{d'}) & \text{if } d \neq d', \end{cases}$$

where

$$(\nabla \mathbf{u})_i^d = \frac{1}{2h}(\mathbf{u}_{i+e^d} - \mathbf{u}_{i-e^d}).$$

We then construct the flux at the face using the appropriate gradients:

$$F^v = \mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) + \lambda I \nabla \cdot \mathbf{u}. \quad (8)$$

At the embedded boundary, we have a physical boundary condition that $\mathbf{u} = 0$. Define a local coordinate system rotated to align with the normal to the embedded boundary \hat{n} and the tangent plane (\hat{t}^1, \hat{t}^2) . The Jacobian J of this rotational transformation is given by

$$J = \begin{pmatrix} \hat{n} \\ \hat{t}^1 \\ \hat{t}^2 \end{pmatrix}.$$

The transformation between a vector in Cartesian space (v) and a vector in rotated space (v^R) is given by

$$v^R = Jv.$$

We start by treating each component of the velocity as a scalar ϕ . To create our boundary flux, we use the Johansen extrapolation [18] to compute the normal gradient of ϕ , $(\nabla \phi^{R,n})$. We set the tangential components of the gradient of ϕ to zero (a consequence of the no-slip condition). So, in the rotated frame $(\nabla \phi)^R = (\nabla \phi^{R,n}, 0, 0)$. We then compute the Cartesian gradient of ϕ :

$$\nabla \phi = J^{-1}(\nabla \phi)^R.$$

We then construct the boundary flux using (8). Given these fluxes, discretization of the operator is given by (1).

5.4. Performance implications of implicit parabolic discretization. The time step constraint for the present algorithm is given by (3). Since we are advancing our elliptic terms implicitly, this adds no additional time step constraint. Suppose we were to advance (4) explicitly:

$$\phi^{n+1} = \phi^n + \nu \Delta t_{\text{exp}} L(\phi^n). \quad (9)$$

In the absence of cut cells, the stability constraint on this method is

$$\Delta t_{\text{exp}}^{\text{noeb}} < \frac{\Delta x^2}{2D\nu}.$$

where D is the dimensionality of the problem. For the conductivity operator at constant density with constant coefficients, this relationship is exact with $\nu = \xi/(\rho C_v)$.

To illustrate the performance tradeoff in the design decision to use the implicit discretization, we compare the number of operator evaluations required to advance the solution. Define N_{exp} to be the number of operator evaluations needed to advance the solution to a fixed time t_f using (9):

$$N_{\text{exp}} = \frac{t_f}{\Delta t_{\text{exp}}}.$$

Define N_{imp} to be the number of operator evaluations needed to advance the solution to a fixed time t_f using (6). We solve our elliptic equations using multigrid and we measure how many times the operators are applied. This puts the implicit method in the worst light possible because coarse and fine applications of the operator (through multigrid) are counted the same.

The problem in Section 7 is the target application for this algorithm. When we run this problem with 4 levels of refinement for a final time of $0.7 \mu\text{s}$ (which accounts for 3 steps at the coarsest level and 48 total steps at all levels), the conductivity operator is called $N_{\text{imp}} = 600$ times. For these parameters, the time step restriction for the explicit advance is $\Delta t_{\text{exp}}^{\text{noeb}} = 2.63 \cdot 10^{-10}$, so an explicit advance would call the operator $N_{\text{exp}} = 2665$ times. For problems with less resolution or lower viscosity, this performance tradeoff can easily flip and make the explicit method more efficient. In the shock-boundary layer calculation presented in Section 9, for example, $\Delta t_{\text{exp}}^{\text{noeb}} > \Delta t$, which means that the explicit parabolic advance for this case presents no addition time step constraint in the absence of embedded boundaries.

With embedded boundaries, however, the time step constraint for the explicit advance (Equation (9)) is far more severe. If κ_{\min} is the smallest volume fraction in the domain, the true time step constraint for the explicit advance is given by

$$\Delta t_{\text{exp}} < \frac{\Delta x^2 (\kappa_{\min})^{2/D}}{2D\nu}.$$

In this context, let us reconsider the shock-boundary layer calculation for a final time of $0.57 \mu\text{s}$ (and all other parameters described in Section 9), which is one time step at the coarsest level and 97 time steps at all levels. The smallest volume fraction at the finest level of this calculation is $\kappa_{\min} = 3.83 \cdot 10^{-7}$, which means that $\Delta t_{\text{exp}} = 4.26 \cdot 10^{-15}$ and the number of operator evaluations required for stability is given by $N_{\text{exp}} = 1.34 \cdot 10^8$. The number of operator evaluations we count for our implicit algorithm is $N_{\text{imp}} = 37536$. Clearly, in the presence of small cells, the implicit advance is the more efficient algorithm to advance our elliptic terms.

6. Convergence tests

To test the convergence rate of the algorithm we start with an initial condition of flow within a sphere (or a circle in two dimensions). All tests are done using

Richardson extrapolation which means that an average of a finer solution is used as an exact solution. Define A^{h-2h} to be a volume-weighted averaging operator. Given S_f to the set of fine volumes which cover a coarse volume i ,

$$A^{h-2h}(f)_i = \frac{\sum_{i_f \in S_f} \kappa_{i_f} f_{i_f}}{\sum_{i_f \in S_f} \kappa_{i_f}}.$$

U_h is defined to be our solution on a grid with resolution h . For an exact solution U^e , we use $U_{2h}^e = A^{h-2h}(U_h)$ and the error is given by

$$\epsilon^h = U^h(t) - U^e(t). \quad (10)$$

The order of convergence ϖ is estimated by

$$\varpi = \frac{\log(\|\epsilon^{2h}\|/\|\epsilon^h\|)}{\log 2}. \quad (11)$$

We compute the convergence rates using compute using L_∞ , L_1 , and L_2 norms (all these norms are defined in [9]). The geometry of the test is a sphere with radius in the center of a domain of length L . The initial condition of the tests is given by an axisymmetric Gaussian disturbance $f(r) = \exp(-30(r/r_0 - 0.5)^2)$. The maximum Mach number is set to $M = 0.5$. Define (x, y, z) to be Cartesian coordinates in a coordinate system whose origin is the sphere center. Define the distance $r = (x^2 + y^2 + z^2)^{1/2}$. The velocity is given by $\mathbf{u} = (-Mf(r)y/r_0, Mf(r)x/r_0)$ in two dimensions and $\mathbf{u} = (Mf(r)(z - y)/r_0, Mf(r)(x - z)/r_0, Mf(r)(y - x)/r_0)$ in three dimensions. Define v to be the magnitude of the velocity vector. The density and pressure are given by $\rho = \gamma(1 + v^2/r)$, $p = (1 + v^2/r)$. See Table 1 for other solution parameters.

Solution error is a measure of the convergence rate of the solution run to a fixed time. All refinements were advanced to a fixed time $t_f = 32 \mu\text{s}$. The finest solution was advanced 64 time steps with $\Delta t = 0.5 \mu\text{s}$. Each successively coarser solution was advanced half as many steps with twice as big a time step. This results in a Courant number (C_F , see (3)) of approximately 0.1 for full cells. The results of the solution error test are given in Tables 2 and 3. We demonstrate second-order accuracy in all norms.

$$\begin{aligned} \mu &= 2.1 \cdot 10^{-5} \text{ kg/(m s)} & L &= 1.0 \cdot 10^{-2} \text{ m} \\ \lambda &= -1.4 \cdot 10^{-5} \text{ kg/(m s)} & r_0 &= 4.5 \cdot 10^{-3} \text{ m} \\ C_v &= 3.00 \cdot 10^2 \text{ J/(kg K)} & \gamma &= \frac{7}{5} \\ \xi &= 1.7 \cdot 10^{-2} \text{ W/(m K)} \end{aligned}$$

Table 1. Initial condition set-up for the convergence tests. See text for variable definitions.

norm	variable	$e_{4h \rightarrow 2h}$	ϖ	$e_{2h \rightarrow h}$
L_∞	ρ	$1.103 \cdot 10^{-2}$	1.980	$2.796 \cdot 10^{-3}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y$	$2.740 \cdot 10^{-3}$	1.822	$7.748 \cdot 10^{-4}$
	ρE	$2.006 \cdot 10^{-2}$	1.978	$5.092 \cdot 10^{-3}$
L_1	ρ	$2.519 \cdot 10^{-3}$	1.982	$6.377 \cdot 10^{-4}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y$	$3.645 \cdot 10^{-4}$	1.822	$1.031 \cdot 10^{-4}$
	ρE	$4.560 \cdot 10^{-3}$	1.978	$1.158 \cdot 10^{-3}$
L_2	ρ	$3.900 \cdot 10^{-3}$	1.978	$9.903 \cdot 10^{-4}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y$	$6.795 \cdot 10^{-4}$	1.824	$1.920 \cdot 10^{-4}$
	ρE	$7.066 \cdot 10^{-3}$	1.973	$1.801 \cdot 10^{-3}$

Table 2. Solution error convergence rates in two dimensions using the L_∞ -, L_1 - and L_2 -norms for $h = \frac{1}{1024}$ cm.

norm	variable	$e_{4h \rightarrow 2h}$	ϖ	$e_{2h \rightarrow h}$
L_∞	(ρ)	$3.536 \cdot 10^{-2}$	1.979	$8.968 \cdot 10^{-3}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y, (\rho \mathbf{u})_z$	$7.406 \cdot 10^{-3}$	1.814	$2.107 \cdot 10^{-3}$
	(ρE)	$6.887 \cdot 10^{-2}$	1.978	$1.748 \cdot 10^{-2}$
L_1	(ρ)	$4.167 \cdot 10^{-3}$	1.986	$1.053 \cdot 10^{-3}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y, (\rho \mathbf{u})_z$	$4.503 \cdot 10^{-4}$	1.805	$1.289 \cdot 10^{-4}$
	(ρE)	$7.767 \cdot 10^{-3}$	1.983	$1.965 \cdot 10^{-3}$
L_2	(ρ)	$7.931 \cdot 10^{-3}$	1.982	$2.007 \cdot 10^{-3}$
	$(\rho \mathbf{u})_x, (\rho \mathbf{u})_y, (\rho \mathbf{u})_z$	$1.060 \cdot 10^{-3}$	1.810	$3.024 \cdot 10^{-4}$
	(ρE)	$1.495 \cdot 10^{-2}$	1.980	$3.790 \cdot 10^{-3}$

Table 3. Solution error convergence rates in three dimensions using the L_∞ -, L_1 - and L_2 -norms for $h = \frac{1}{1024}$ cm.

7. Capillary tube simulation

Our target application is the flow inside of capillary tubes in laser wakefield particle accelerators. We present a simplified version of this problem as our robustness calculation while acknowledging that other physics in these problems (such as ionization and magnetization) are very important. Refer to [Figure 2](#). The main tube (C) and the fill tubes (A) are filled with gas. The experimentalists drive the core pressure p_{core} along the axis of the tube to a high value using electrical charge, leaving the density constant. The resulting flow causes the core to expand and create a low density, high energy core. In the experiment, the laser (B) is shot through

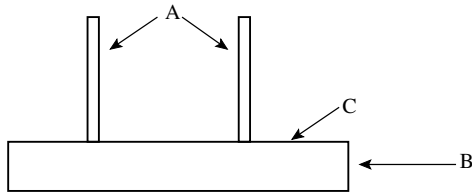


Figure 2. Illustration of wakefield accelerator. The main tube (C) and fill tubes (A) are filled with gas. The pressure and temperature are initialized to high values up along the axis of the tube. The resulting flow causes this region to expand and create a low density, high-energy core. A laser (B) is shot through this core.

this low density core. Ideally this core should be cylindrical and have a relatively flat density profile. There is some concern in the community, however, that the fill tubes can alter the core shape before the laser is shot.

We present both two- and three-dimensional runs that are meant to approximate to this problem. For a computational geometry we intersect a 200 micron diameter main tube with a perpendicular fill tube 50 microns in diameter. [Figure 3](#) shows the

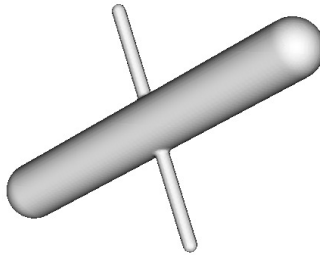


Figure 3. Geometric configuration of the three-dimensional example. The core tube's diameter is 200 microns; the filler tube's diameter is 50 microns. The core tube's length is 1.2 mm; the filler tube's length is 0.85 mm.

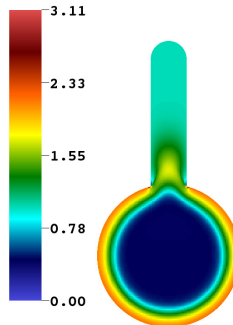


Figure 4. Two-dimensional plot of $\log \rho$ after $35 \mu\text{s}$. The base grid is 256^2 and there are 2 levels of refinement, all by a factor of 2. This means the effective grid resolution is 1024^2 . Though the density profile in the core is relatively flat, the core shape is no longer circular.

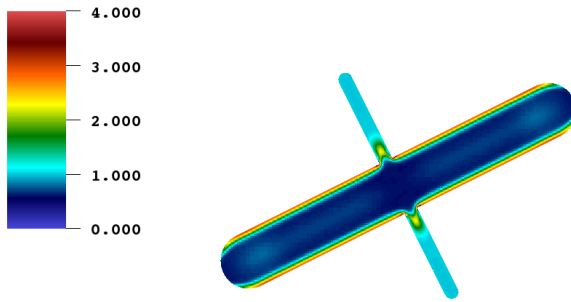


Figure 5. Two-dimensional plot of $\log \rho$ after $35 \mu\text{s}$. The base grid is 128×64 and there are 3 levels of refinement, all by a factor of 2. This means the effective grid resolution is 1024×512 . Though the density profile in the core is relatively flat, the filler tube has distorted the profile.

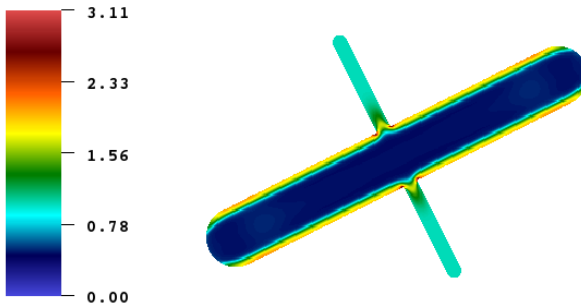


Figure 6. Axial slice through three-dimensional run plot of $\log \rho$ after $50 \mu\text{s}$. This is a one-level calculation with resolution $256 \times 128 \times 128$. Though the density profile in the core is relatively flat, the filler tube has distorted the profile.

geometric configuration. Both are filled with argon at 1 Pa, 1 kg/m^3 . We initialize the core pressure to be $p_{\text{core}} = 20 \text{ Pa}$, leave the density constant and initialize the velocity everywhere to zero. The core diameter is 100 microns. [Figure 4](#) (on the previous page) shows a two-dimensional run of the plane normal to the central tube cutting through a filler tube. We plot the logarithm of density after $35 \mu\text{s}$. Though the density profile in the core is relatively flat, the core shape is no longer circular. [Figure 5](#) shows a two-dimensional run of the plane along the central tube cutting through a filler tube. We plot the logarithm of density after $35 \mu\text{s}$. Though the density profile in the core is relatively flat, the core shape is once again distorted by the presence of the filler tube. [Figure 6](#) shows an axial slice through a three-dimensional run after $50 \mu\text{s}$ and shows a similar result. To be clear, since we do not include any source terms for the effects of ionization or magnetization, this is greatly simplified approximation. We have, however, managed to show that purely hydrodynamic effects can distort the shape of the low density core.

$$\begin{aligned}
 p_{\text{outside}} &= 1.0 \text{ Pa} & \mu &= 2.1 \cdot 10^{-5} \text{ kg/(m s)} \\
 p_{\text{core}} &= 20.0 \text{ Pa} & \lambda &= -1.4 \cdot 10^{-5} \text{ kg/(m s)} \\
 \rho_{\text{outside}} &= 1.0 \text{ kg/m}^3 & C_v &= 3.00 \cdot 10^2 \text{ J/(kg K)} \\
 \rho_{\text{core}} &= 1.0 \text{ kg/m}^3 & \xi &= 1.7 \cdot 10^{-2} \text{ W/(m K)} \\
 & & \gamma &= \frac{5}{3}
 \end{aligned}$$

Table 4. Initial condition set-up for capillary tube problem. The initial velocity is zero.

8. Boundary layer similarity solution

Given a semiinfinite flat plate in a flow field at zero incidence to the flow, the velocity profile over the plate can be calculated using a similarity solution in the absence of thermal and compressibility effects. Define x to be the distance along the plate and y to be the distance from the surface of the plate and ν to be the kinematic viscosity and $U = Mc$ is the incident velocity (refer to [Figure 7](#)). The similarity variable η is given by

$$\eta = y \sqrt{\frac{U}{\nu x}}. \quad (12)$$

This reduces the equations to a nonlinear ordinary differential equation, the solution of which is the familiar Blasius boundary layer. See Schlichting [26] or White [32] for a full exposition of this derivation. Charest et al. [6] present a low-Mach-number algorithm for steady state calculations. In this calculation, they present a boundary layer calculation that reproduces the behavior Blasius layer. Berger et al. [5] present a wide variety of these calculations.

We cut a rectangular grid with a wedge of angle θ . Refer to [Figure 8](#) and [Table 5](#) for the initial and boundary conditions. The density and temperature are set to constants $\rho = \rho_0$, $T = T_0 = P_0/(RT_0)$. The velocity is set to $(U \cos \theta, U \sin \theta)$ everywhere. The velocity boundary conditions are inflow-outflow left to right (the

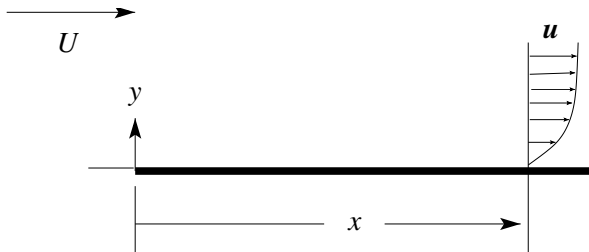


Figure 7. Formulation of semiinfinite flat plate boundary layer problem. U is the (constant) inflow velocity, x is the distance along the plate and y is the distance above the plate.

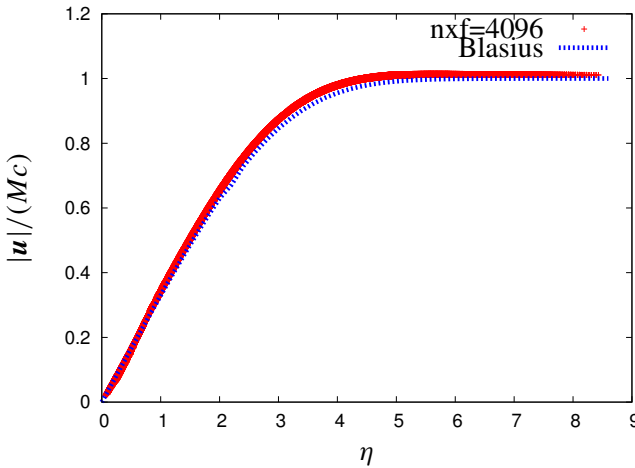


Figure 9. Results of boundary layer calculation as compared with the Blasius solution. This is a scatter plot of normalized velocity versus the similarity variable η at two different resolutions. The magnitude of solution velocity is $|\mathbf{u}|$, c is the sound speed and the similarity variable η is defined in (12). The Blasius solution is in blue. The solution with effective resolution of 4096×4096 is in red. We cast a ray from every point along the boundary where the local Reynolds number is in the range $5000 < \text{Re}_x < 15000$. We plot every point along every ray. The rays are 30 points long.

9. Shock reflection

Define M to be the Mach number of a shock propagating into a gas at rest. Glaz et al. [14] present a comparison between inviscid calculations of shock reflections and experimental results. They show a case where viscous effects cause substantial changes in the reflection pattern. For $M = 7.1$ shock reflection from a 49 degree wedge, they show that the Mach stem is much shorter in the experiment than in an inviscid calculation. The reason cited for this difference is that the viscosity of argon varies strongly with temperature and the temperature behind the shock is quite high (the initial temperature behind the diaphragm is 10265 K). The viscosity is approximated to vary with the Sutherland's power law (dynamic viscosity varies with $T^{3/2}$). We use the viscosity shown in Table 6. We compute this viscosity using the highest value given in [22] and extrapolating to the initial high temperature. The specific heat and conductivity of argon are left at the room temperature values. These approximations are sufficient to illustrate the phenomenon.

Refer to Figure 10 for an illustration of the initial conditions. Table 6 has the numerical values of the inputs. Both calculations have a 128×64 base grid with seven levels of adaptive mesh refinement, all by a factor of 2. This makes the effective resolution 16384×8192 . All embedded boundary cells are refined to the finest level. This gives resolution at the boundary layer $h = 9.1$ microns.

$$\begin{aligned}
 p_0 &= 1.95 \cdot 10^3 \text{ Pa} & Y_3 &= 7.50 \cdot 10^{-2} \text{ m} \\
 p_1 &= 7.42 \cdot 10^5 \text{ Pa} & \mu &= 1.21 \cdot 10^{-3} \text{ kg/(m s)} \\
 \rho_0 &= 3.29 \cdot 10^{-2} \text{ kg/m}^3 & \lambda &= -8.08 \cdot 10^{-4} \text{ kg/(m s)} \\
 \rho_1 &= 3.61 \cdot 10^{-1} \text{ kg/m}^3 & C_v &= 3.00 \cdot 10^2 \text{ J/(kg K)} \\
 X_1 &= 9.0 \cdot 10^{-2} \text{ m} & \xi &= 1.7 \cdot 10^{-2} \text{ W/(m K)} \\
 X_2 &= 1.0 \cdot 10^{-1} \text{ m} & \theta &= 49^\circ \\
 X_3 &= 1.5 \cdot 10^{-1} \text{ m} & \gamma &= \frac{5}{3}
 \end{aligned}$$

Table 6. Initial condition set-up for shock reflection problem. The initial velocity is zero. See [Figure 10](#) for variable definitions.

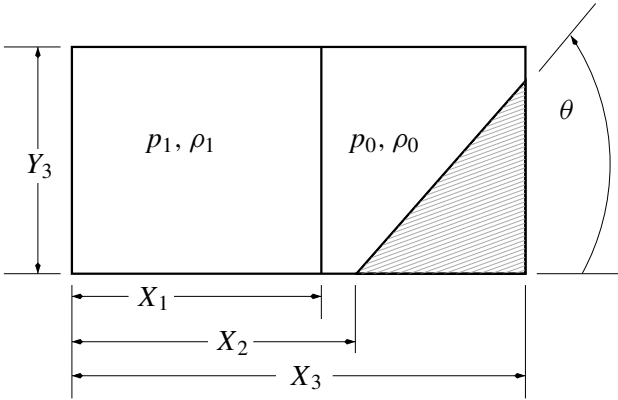


Figure 10. Shock tube set-up. The initial velocity is zero. The initial pressures and densities are tailored to make a $M = 7.1$ shock. See [Table 6](#) for details.

[Figure 11](#) illustrates the Mach reflection problem. [Figure 12](#) shows the viscous and inviscid calculations at the same scale after $9.61 \mu\text{s}$. The viscous calculation shows an interesting shock-boundary layer interaction, which is magnified in [Figure 14](#). The shock reflects off of the boundary layer, creating a separation bubble. This is followed by a compression (from the reflected shock) and boundary layer reattachment. For steady shocks interacting with laminar boundary layers, this is the classical lambda shock phenomenon. Both Schlichting [26] and Liepmann et al. [21] explain this in detail and include a wealth of experimental images. This is also observed (albeit barely) in the experiment presented in [14]. The interferogram they show has only two or three density contours in that region which makes the feature difficult to see.

[Figure 12](#) clearly shows that the viscous boundary layer has reduced the Mach stem substantially and a density stratification on the left. Recall that the problem is configured as a shock tube. The initial conditions are zero velocity with a discontinuity in pressure and density. As the shock moves to the right, a rarefaction

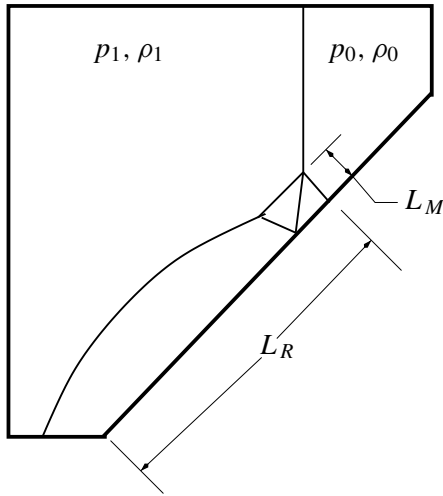


Figure 11. Shock reflection illustration. The ratio of the Mach stem length L_M to the shock distance L_R is the quantity of interest.

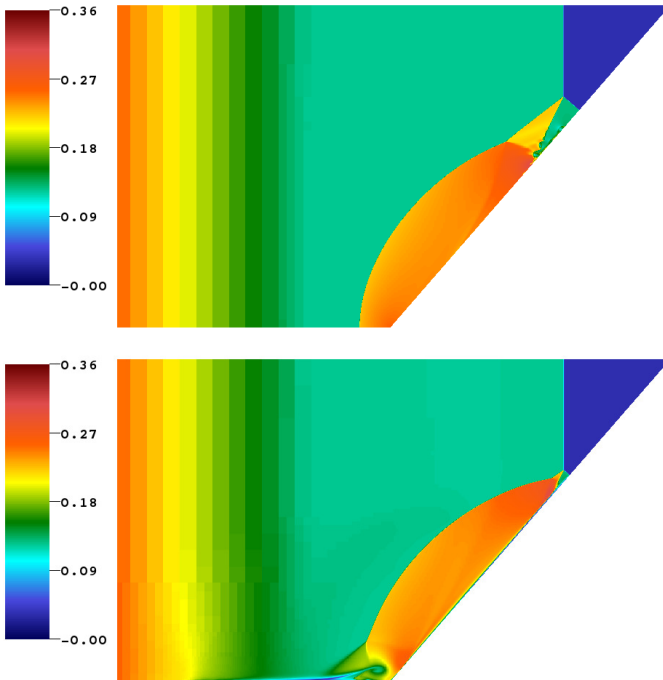


Figure 12. Mass density (kg/m^3) in the inviscid (top) and viscous (bottom) calculations of $M = 7.1$, with 49° shock reflection at $9.6 \mu\text{s}$. These calculations were run 128×64 base grid with seven levels of adaptive mesh refinement, all by a factor of 2. This makes the effective resolution 16384×8192 . All embedded boundary cells are refined to the finest level. This gives resolution at the boundary layer $h = 9.1$ microns.

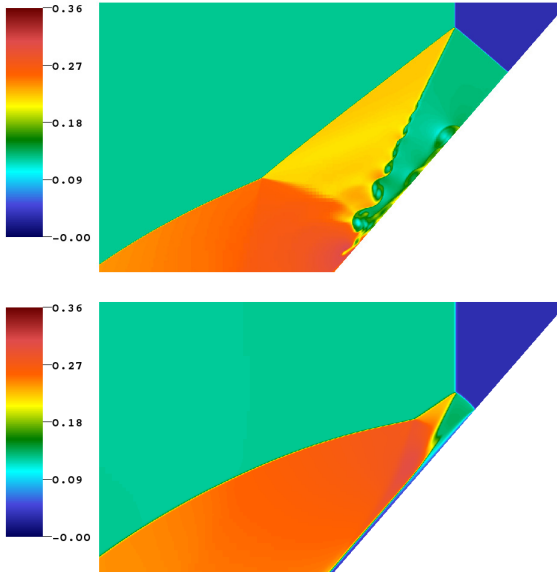


Figure 13. Mass density (kg/m^3) in the inviscid (top) and viscous (bottom) calculation of $M = 7.1$, with 49° shock reflection at $9.6 \mu\text{s}$, zoomed in to show the reflection pattern.

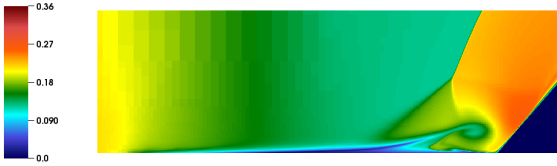


Figure 14. Magnification of the lambda shock-boundary layer pattern in the viscous calculation. Density (in kg/m^3) shown here.

fan moves to the left, producing this density variation. We show the two shock reflection patterns more closely in [Figure 13](#). For a quantitative look at this reduction, we refer to the experimental and computational results in [\[14\]](#). See [Figure 11](#) for an illustration of the relevant lengths. The ratio of the Mach stem length L_M to the shock distance L_R is the quantity of interest:

$$R_m = \frac{L_M}{L_R}.$$

Glaz et al. report a value of $R_m = 0.07$ in their inviscid calculation and $R_m = 0.038$ for an experimental result (see [Figure 10](#) in [\[14\]](#)). Our inviscid calculation has $R_m = 0.072$ and our viscous calculation has $R_m = 0.03$. We believe that our agreement is reasonable since not all the experimental set-up information is available (the time at which the interferogram is taken, for example, is not available). For more examples of this viscous effect, see [Henderson et al. \[17\]](#).

10. Conclusion

We have presented a stable, second-order method for solving the two- and three-dimensional compressible Navier–Stokes equations in the presence of complex geometries. This semiimplicit method advances parabolic terms implicitly and hyperbolic terms explicitly. This allows a time step controlled by the CFL constraint associated with the hyperbolic wave speeds. We demonstrate second-order accuracy for smooth initial conditions in smooth geometric configurations and robust behavior in the presence of strong discontinuities and geometric complexity that mimic the conditions in a plasma wakefield accelerator in the absence of magnetic or ionization effects. We also show good quantitative agreement with experimental results in a viscous shock reflection problem and a boundary layer problem.

Acknowledgements

The authors would like to thank Drs. Ann Almgren, John Bell, Marc Day, Cameron Geddes, Wim Leemans, and Daniel Martin for valuable technical discussions.

References

- [1] J. Bell, P. Colella, and M. Welcome, *Conservative front-tracking for inviscid compressible flow*, AIAA 10th Computational Fluid Dynamics Conference, 1991, pp. 814–822.
- [2] J. Bell, M. Berger, J. Saltzman, and M. Welcome, *Three-dimensional adaptive mesh refinement for hyperbolic conservation laws*, SIAM J. Sci. Comput. **15** (1994), no. 1, 127–138. MR 95d:65070
- [3] J. B. Bell, P. Colella, J. A. Greenough, and D. L. Marcus, *A multi-fluid algorithm for compressible, reacting flow*, AIAA 95-1720 — Proceedings of the 12th AIAA Computational Fluid Dynamics Conference, 1995.
- [4] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys. **82** (1989), no. 1, 64–84.
- [5] M. Berger and M. J. Aftosmis, *Progress toward a Cartesian cut-cell method for compressible viscous flow*, 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, 2012, With an appendix by Steven Allmaras, pp. 2012–1301.
- [6] M. R. J. Charest, C. P. T. Groth, and P. Q. Gauthier, *High-order CENO finite-volume scheme for low-speed viscous flows on three-dimensional unstructured mesh*, Seventh International Conference on Computational Fluid Dynamics (ICCFD7), 2012, p. 1002.
- [7] I. L. Chern and P. Colella, *A conservative front-tracking method for hyperbolic conservation laws*, technical report UCRL-97200, Lawrence Livermore National Laboratory, 1987.
- [8] P. Colella, *Multidimensional upwind methods for hyperbolic conservation laws*, J. Comput. Phys. **87** (1990), no. 1, 171–200. MR 91c:76087
- [9] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys. **211** (2006), no. 1, 347–366. MR 2006i:65142
- [10] P. Colella, A. Majda, and V. Roytburd, *Theoretical and numerical structure for reacting shock waves*, SIAM J. Sci. Statist. Comput. **7** (1986), no. 4, 1059–1080. MR 87i:76037

- [11] M. S. Day and J. B. Bell, *Numerical simulation of laminar reacting flows with complex chemistry*, *Combust. Theory Modelling* **4** (2000), 535–556.
- [12] Z. Dragojlovic, F. Najmabadi, and M. Day, *An embedded boundary method for viscous, conducting compressible flow*, *J. Comput. Phys.* **216** (2006), no. 1, 37–51. MR 2223435
- [13] R. Ghias, R. Mittal, and H. Dong, *A sharp interface immersed boundary method for compressible viscous flows*, *J. Comput. Phys.* **225** (2007), no. 1, 528–553. MR 2008g:76094
- [14] H. Glaz, P. Colella, and R. Deschambault, *A numerical study of oblique shock-wave reflections with experimental comparisons*, *Proc. R. Soc. London* **398** (1985), 117–149.
- [15] D. Hartmann, M. Meinke, and W. Schröder, *A strictly conservative Cartesian cut-cell method for compressible viscous flows on adaptive grids*, *Comput. Methods Appl. Mech. Engrg.* **200** (2011), no. 9–12, 1038–1052. MR 2011m:76119
- [16] L. F. Henderson, K. Takayama, W. Y. Crutchfield, and S. Itabashi, *The persistence of regular reflection during strong shock diffraction over rigid ramps*, *Journal of Fluid Mechanics* **431** (2001), 273–296.
- [17] L. F. Henderson, K. Takayama, W. Y. Crutchfield, and R. J. Virgona, *The effects of thermal conductivity and viscosity of argon on shock waves diffracting over rigid ramps*, *Journal of Fluid Mechanics* **331** (1997), 1–36.
- [18] H. Johansen and P. Colella, *A Cartesian grid embedded boundary method for Poisson’s equation on irregular domains*, *J. Comput. Phys.* **147** (1998), no. 1, 60–85. MR 99m:65231
- [19] M. Kim, D. G. Jang, H. Uhm, S. W. Hwang, I. W. Lee, and H. Suk, *Discharge characteristics of a gas-filled capillary plasma for laser wakefield acceleration*, *IEEE Transactions on Plasma Science* **39** (2011), no. 8, 1638–1643.
- [20] W. P. Leemans, B. Nagler, A. J. Gonsalves, C. S. Toth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, and S. M. Hooker, *GeV electron beams from a centimetre-scale accelerator*, *Nature Physics* **2** (2006), 696–699.
- [21] H. W. Liepmann, A. Roshko, and S. Dhawan, *On reflection of shock waves from boundary layers*, technical report NACA-1100, National Advisory Committee for Aeronautics, 1952.
- [22] M. N. Macrossan and C. R. Lilley, *Viscosity of argon at temperatures greater than 2000 K from measured shock thickness*, *Phys. Fluids A* **15** (2003), 1363–1371.
- [23] D. F. Martin, P. Colella, and D. Graves, *A cell-centered adaptive projection method for the incompressible Navier–Stokes equations in three dimensions*, *J. Comput. Phys.* **227** (2008), no. 3, 1863–1886. MR 2009g:76085
- [24] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, *An adaptive Cartesian grid method for unsteady compressible flow in irregular regions*, *J. Comput. Phys.* **120** (1995), no. 2, 278–304. MR 96d:76081
- [25] J. Saltzman, *An unsplit 3D upwind method for hyperbolic conservation laws*, *J. Comput. Phys.* **115** (1994), no. 1, 153–168. MR 1300337
- [26] H. Schlichting, *Boundary layer theory*, McGraw-Hill, New York, 1955. MR 17,912d
- [27] C. B. Schroeder, E. Esarey, C. Geddes, C. Benedetti, and W. P. Leemans, *Physics considerations for laser-plasma linear colliders*, *Phys. Rev. ST Accel. Beams* **13** (2010), no. 10, 101301.
- [28] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, *A Cartesian grid embedded boundary method for the heat equation and Poisson’s equation in three dimensions*, *J. Comput. Phys.* **211** (2006), no. 2, 531–550. MR 2006e:65194
- [29] D. J. Spence and S. M. Hooker, *Investigation of a hydrogen plasma waveguide*, *Physical Review E* **63** (2000), 015401.

- [30] E. Steinhörsson, D. Modiano, W. Crutchfield, J. Bell, and P. Colella, *An adaptive semi-implicit scheme for simulations of unsteady viscous compressible flow*, AIAA 95-1727 — Proceedings of the 12th AIAA CFD Conference, 1995, pp. 95–1727–CP.
- [31] E. H. Twizell, A. B. Gumel, and M. A. Arigu, *Second-order, L_0 -stable methods for the heat equation with time-dependent boundary conditions*, Adv. Comput. Math. **6** (1996), no. 3-4, 333–352. MR 97m:65164
- [32] F. M. White, *Viscous fluid flow*, McGraw-Hill, New York, NY, 1974.

Received January 23, 2013. Revised September 30, 2013.

DANIEL T. GRAVES: DTGraves@lbl.gov

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148, Berkeley, CA 94720, United States

PHILLIP COLELLA: PColella@lbl.gov

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148, Berkeley, CA 94720, United States

DAVID MODIANO: dave@alum.mit.edu

Sanzaru Games, Inc., 323B Vintage Park Drive, Foster City, CA 94404, United States

JEFFREY JOHNSON: jjphatt@gmail.com

Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 50A-1148, Berkeley, CA 94720, United States

BJORN SJOGREEN: sjogreen2@llnl.gov

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808, L-422, Livermore, CA 94551-0808, United States

XINFENG GAO: Xinfeng.Gao@colostate.edu

Department of Mechanical Engineering, Colorado State University, Fort Collins, CO 80523, United States

SECOND-ORDER ACCURACY OF VOLUME-OF-FLUID INTERFACE RECONSTRUCTION ALGORITHMS II: AN IMPROVED CONSTRAINT ON THE CELL SIZE

ELBRIDGE GERRY PUCKETT

In a previous article in this journal the author proved that, given a square grid of side h covering a two times continuously differentiable simple closed curve z in the plane, one can construct a pointwise second-order accurate piecewise linear approximation \tilde{z} to z from just the volume fractions due to z in the grid cells. In the present article the author proves a sufficient condition for \tilde{z} to be a second-order accurate approximation to z in the max norm is h must be bounded above by $2/(33\kappa_{\max})$, where κ_{\max} is the maximum magnitude of the curvature κ of z . This constraint on h is solely in terms of an intrinsic property of the curve z , namely κ_{\max} , which is invariant under rotations and translations of the grid. It is also far less restrictive than the constraint presented in the previous article. An important consequence of the proof in the present article is that the max norm of the difference $z - \tilde{z}$ depends linearly on κ_{\max} .

1. Introduction

The topic of this article is the *interface reconstruction problem* for a volume-of-fluid method in two space dimensions. This problem can be described as follows. Let $\Omega \subset \mathbb{R}^2$ denote a closed and bounded rectangular region in the plane, and let Ω_1 and Ω_2 be disjoint, connected (but not necessary simply connected) relatively open regions such that $\overline{\Omega}_1 \cup \overline{\Omega}_2 = \Omega$ and that $\overline{\Omega}_1 \cap \overline{\Omega}_2$ is the image of a twice continuously differentiable simple closed curve in Ω , denoted by $z(s) = (x(s), y(s))$, where s is arc length. The regions Ω_1 and Ω_2 contain “material 1” and “material 2”, respectively, where each material may be a thought of as a gas, fluid or solid and z is the boundary or *interface* between these two materials.

Let L be a characteristic length of the problem domain Ω and cover Ω with a grid Ω^h consisting of square cells, each of side $h \ll L$. Given integers i and j ,

Sponsored by the US Department of Energy Mathematical, Information, and Computing Sciences Division contracts DE-FC02-01ER25473 and DE-FG02-03ER25579.

MSC2010: primary 65M12, 76T99; secondary 65M06, 76M12, 76M25.

Keywords: volume-of-fluid, piecewise linear interface reconstruction, fronts, front reconstruction, interface reconstruction, two-phase flow, multiphase systems, under-resolved computations, computational fluid dynamics.

let $x_i = ih$ (resp. $y_j = jh$) denote the location of the i -th vertical (resp. j -th horizontal) grid line and let (x_i, y_j) denote the lower left hand corner of the ij -th cell

$$C_{ij} \stackrel{\text{def}}{=} [x_i, x_{i+1}] \times [y_j, y_{j+1}] \quad (1)$$

in the grid.

Denote the fraction of material 1 in the ij -th cell by Λ_{ij} . For each i, j the number Λ_{ij} satisfies $0 \leq \Lambda_{ij} \leq 1$ and is called the *volume fraction* (of material 1) in the ij -th cell. (Even though in two dimensions Λ_{ij} is technically an *area* fraction, the convention is to refer to it as a volume fraction.) Thus $0 < \Lambda_{ij} < 1$ if and only if a portion of the interface $\mathbf{z}(s)$ lies in the ij -th cell and $\Lambda_{ij} = 1$ (resp. $\Lambda_{ij} = 0$) if and only if the ij -th cell contains only material 1 (resp. material 2). In the volume-of-fluid interface reconstruction problem one is asked to determine an approximation $\tilde{\mathbf{z}}(s)$ to $\mathbf{z}(s)$ in Ω given only the volume fractions Λ_{ij} .

Suppose the interface $\mathbf{z}(s)$ passes through the ij -th cell C_{ij} and can be written as a single-valued function of x in C_{ij} ; that is, for $x \in [x_i, x_{i+1}]$ the interface can be written in the form $\mathbf{z}(s) = (x(s), y(s)) = (x(s), g(x(s)))$. Let $\tilde{g}_{ij}(x)$ denote an approximation to the interface in C_{ij} . Then the max norm of the difference between the interface $(x, g(x))$ and the approximate interface $(x, \tilde{g}_{ij}(x))$ in C_{ij} is defined in the usual way,

$$\|g - \tilde{g}_{ij}\|_{\infty(ij)} \stackrel{\text{def}}{=} \max_{x \in [x_i, x_{i+1}]} |g(x) - \tilde{g}_{ij}(x)|. \quad (2)$$

In the event the interface in the ij -th cell can only be expressed as a single-valued function $G(y)$ of $y \in [y_j, y_{j+1}]$ the max norm of the difference between the interface $(G(y), y)$ and the approximate interface $(\tilde{G}_{ij}(y), y)$ is defined analogously.

By [Theorem A.1](#) in the [Appendix](#), if the interface $\mathbf{z}(s) \in C^2(\mathbb{R})$ passes through the ij -th cell C_{ij} and the constraint in [\(5\)–\(6\)](#) below is satisfied, then it is possible to represent $\mathbf{z}(s)$ as either a single-valued function $y = g(x)$ or $x = G(y)$ of the independent variable x (resp. y) in the 3×3 block of cells B_{ij} centered on C_{ij} . For convenience, in all of the following the interface is assumed to be of the form $y = g(x)$ in the block B_{ij} with material 1 lying *below* the graph of g in B_{ij} ; it being understood that all of the definitions, results, etc. in this article also apply to the case in which the interface can only be expressed as a single-valued function $x = G(y)$ in B_{ij} . In [Section 2.1](#) I will present an algorithm for determining which of the four standard rotations of B_{ij} about its center, 0, 90, 180, or 270 degrees, will orient the block B_{ij} so the interface can be expressed as either $y = g(x)$ or $x = G(y)$ with material 1 lying below the interface.

Let $\kappa(s)$ denote the curvature of the interface $\mathbf{z}(s)$ and let

$$\kappa_{\max} \stackrel{\text{def}}{=} \max_s |\kappa(s)| \quad (3)$$

denote the maximum of the magnitude of $\kappa(s)$ in Ω . The main result of this article is as follows. If conditions (I)–(V) below hold, then the piecewise linear volume-of-fluid approximation $\tilde{g}_{ij}(x)$ defined in equations (7)–(10) below will approximate the true interface $z(s) = (x(s), g(x(s)))$ to second-order in h in the max norm,

$$\|g - \tilde{g}_{ij}\|_{\infty(ij)} \leq C_m \kappa_{\max} h^2 \quad \text{for all } i, j \text{ such that } 0 < \Lambda_{ij} < 1, \quad (4)$$

where the constant C_m , defined in (59) below, is independent of h and κ_{\max} . Note the linear dependence of the bound in (4) on κ_{\max} .

The following conditions are sufficient to ensure that (4) holds. Note that (II)–(IV) constitute an algorithm for constructing the piecewise linear approximation \tilde{z} to z . This algorithm is described in detail in [24].

(I) The interface $z = (x(s), y(s))$ is a two times continuously differentiable simple closed curve in Ω .

(II) The grid size h and the maximum magnitude κ_{\max} of the curvature of the interface satisfy the following inequality with respect to one another,

$$h \leq \frac{C_h}{\kappa_{\max}}, \quad (5)$$

where

$$C_h \stackrel{\text{def}}{=} \frac{2}{33}. \quad (6)$$

(III) In each cell C_{ij} that contains a portion of the interface $(x, g(x))$ the piecewise linear approximation

$$\tilde{g}_{ij}(x) \stackrel{\text{def}}{=} m_{ij}x + b_{ij} \quad (7)$$

to g in C_{ij} has the same volume fraction $\Lambda_{ij}(\tilde{g})$ in C_{ij} as does the interface,

$$\Lambda_{ij}(\tilde{g}) = \Lambda_{ij}(g). \quad (8)$$

See Figure 1 for an example. Note that, once the slope m_{ij} in (7) is given, the constraint in (8) uniquely determines b_{ij} .

(IV) In each cell C_{ij} that contains a portion of the interface, the slope m_{ij} of the piecewise linear approximation $\tilde{g}_{ij}(x)$ defined in (7) is given by

$$m_{ij} \stackrel{\text{def}}{=} \frac{S_{i+\alpha} - S_{i+\beta}}{\alpha - \beta} \quad \text{for some } \alpha, \beta = -1, 0, 1 \text{ with } \alpha \neq \beta, \quad (9)$$

where

$$S_{i+\alpha} \stackrel{\text{def}}{=} \sum_{j'=j-1}^{j+1} \Lambda_{i+\alpha, j'} \quad \text{and} \quad S_{i+\beta} \stackrel{\text{def}}{=} \sum_{j'=j-1}^{j+1} \Lambda_{i+\beta, j'} \quad (10)$$

denote two distinct *column sums* of volume fractions from the 3×3 block of cells $B_{ij} = [x_{i-1}, x_{i+2}] \times [y_{j-1}, y_{j+2}]$ centered on the ij -th cell C_{ij} .

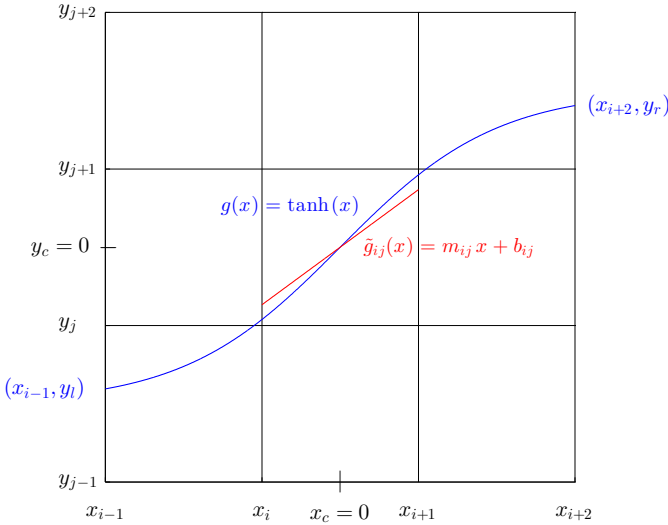


Figure 1. In this example the interface is $g(x) = \tanh(x)$ and all three of the column sums S_{i-1} , S_i , and S_{i+1} are *exact*. The linear approximation $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ in the center cell C_{ij} is also plotted, where the slope m_{ij} is given by (9) with $\alpha = 1$ and $\beta = -1$ and b_{ij} is determined by the constraint $\Lambda_{ij}(\tilde{g}) = \Lambda_{ij}(g)$ in (8).

For $\alpha = -1, 0, 1$ the column sum $S_{i+\alpha}$ is said to be *exact* if

$$S_{i+\alpha} = \frac{1}{h^2} \int_{x_{i+\alpha}}^{x_{i+\alpha+1}} (g(x) - y_{j-1}) dx. \quad (11)$$

and *exact to $O(h)$* if

$$\left| S_{i+\alpha} - \frac{1}{h^2} \int_{x_{i+\alpha}}^{x_{i+\alpha+1}} (g(x) - y_{j-1}) dx \right| \leq \bar{C} \kappa_{\max} h, \quad (12)$$

where $\bar{C} > 0$ is a constant, defined in (53) below, which is independent of h and κ_{\max} . Column sums are discussed in greater detail in Section 2.2.

(V) Each of the two column sums $S_{i+\alpha}$ and $S_{i+\beta}$ in (9), where $\alpha \neq \beta$, is either exact or exact to $O(h)$. Thus, by Theorem 23 of [23] the slope m_{ij} defined in (9) is a first-order accurate approximation to $g'(x_c)$,

$$|m_{ij} - g'(x_c)| \leq \bar{C} \kappa_{\max} h, \quad (13)$$

where x_c denotes the center of the interval $[x_i, x_{i+1}]$. It then follows from Theorem 4 on page 152 below that the approximation \tilde{g} defined in (7)–(10) is a second-order accurate approximation to g in C_{ij} ; i.e., the bound in (4) holds in C_{ij} .

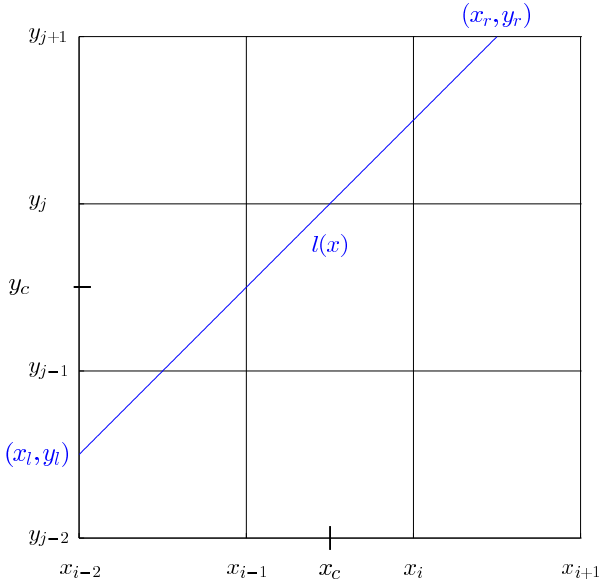


Figure 2. In this example the interface is a line $l(x) = mx + b$ that has two exact column sums, S_{i-1} and S_i , in the first and second columns of the 3×3 block of cells B_{ij} centered on the cell C_{ij} . In this case the slope m_{ij} defined in (9) with $\alpha = 0$ and $\beta = -1$ is *exactly* equal to the slope m of the interface: $m_{ij} = m$. It is always the case if the true interface is a line; then one of the four standard rotations of B_{ij} about its center will orient the block so at least one of the divided differences of the column sums in (9) is exact and hence, the approximation \tilde{g}_{ij} to the interface in the center cell C_{ij} defined in (7)–(10) will exactly equal the interface in that cell, $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij} = mx + b = l(x)$. In other words, the approximation \tilde{g}_{ij} defined in (7)–(10) will *always* reconstruct a linear interface exactly.

1.1. Remarks concerning conditions (I)–(V).

(1) The proof of (4) is based on showing if the constraint in (5)–(6) holds, then for all cells C_{ij} that contain a portion of the interface, there are at least two distinct column sums $S_{i+\alpha}$ and $S_{i+\beta}$, with $\alpha \neq \beta$, which are either exact or exact to $O(h)$ in one or more of the four standard rotations of the 3×3 block of cells B_{ij} centered on C_{ij} . An algorithm for determining which of the four standard rotations of the block B_{ij} has this property is described in Section 2.1.

(2) The constraint on h in (5) may be viewed as dictating the number of cells required to produce a pointwise second-order accurate approximation to a circle of radius r on a grid with cell size h . To see this, note the curvature of the circle is $\kappa_{\max} = r^{-1}$ and hence, by (5) and (6), one must have

$$16.5h = C_h^{-1}h \leq r. \tag{14}$$

This implies one needs a 35×35 square block of cells covering the circle (this includes a border one cell wide outside the circle) in order to ensure the piecewise

linear approximation defined in (7)–(10) is a pointwise second-order accurate approximation to the circle in each cell C_{ij} that contains a portion of the circle.

This could be an overestimate of the number of cells required to achieve pointwise second-order accuracy. However, if this is so, then it is likely one will need to employ ideas other than the ones presented in this article, and in [23], in order to obtain a better result; that is, a larger value for C_h , thereby implying fewer cells are required to reconstruct a circle of radius r to pointwise second-order accuracy in h . In other words, the constant of proportionality C_h in (6) appears to be optimal in the sense that it is about as large as one can obtain with the ideas and techniques presented here and in [23].

(3) In [23] the constraint that corresponds to (5) is

$$h \leq \min\{\tilde{C}_h \kappa_{\max}^{-1}, \kappa_{\max}^{-2}\}, \quad (15)$$

where

$$\tilde{C}_h \stackrel{\text{def}}{=} \bar{C}_h[4] = \frac{\sqrt{4} - \sqrt{2}}{4\sqrt{2}\sqrt{4-1}} = \frac{\sqrt{2} - 1}{4\sqrt{3}}, \quad (16)$$

where $\bar{C}_h[a]$ is defined in Equation (A.1) in the Appendix. The principal new result of this article is the elimination of the much more restrictive (and dimensionally inconsistent) constraint

$$h \leq \kappa_{\max}^{-2} \quad (17)$$

in (15). Thus, for a given interface z , one can reconstruct z to second-order in h using a larger value of h than dictated by (17). A notable consequence of this new proof is that the bound on the error in (4) depends linearly on κ_{\max} .

A minor change from [23] is the very slight increase in the value of C_h from $C_h = \tilde{C}_h \approx (16.73)^{-1}$ to $C_h = 2/33 = (16.5)^{-1}$. The reason for this change is solely for the purpose of presenting the example in item (2) above in terms of an integral number of grid cells. The details concerning how C_h and \tilde{C}_h are chosen appear in the Appendix.

The majority of the work in this article is concerned with proving the more restrictive constraint in (17) is unnecessary. This involves replacing the arguments in Sections 3.2–3.4 of [23] with those in Section 4 here. Sections 2.2.2 and 3 of this article contain a more detailed discussion of the modifications to the argument in [23] required to eliminate the constraint in (17).

Although it is not necessary to modify the argument in [23] in order to *increase* the value of C_h from $(\sqrt{2} - 1)/(4\sqrt{3}) \approx (16.73)^{-1}$ to $C_h = 2/33 = (16.5)^{-1}$, a more general version of Theorem 6 from [23] is presented as Theorem A.1 in the Appendix in order to clearly show the considerations that influence the choice of C_h .

(4) **Figure 1** contains an example of the volume-of-fluid approximation $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ defined in (7)–(10) to the interface $g(x) = \tanh(x)$ in the center cell C_{ij} of a 3×3 block of cells in which all three column sums are exact. Hence, \tilde{g}_{ij} is a pointwise second-order accurate approximation to g for any choice of $\alpha, \beta = 1, 0, -1$ with $\alpha \neq \beta$ in (9) provided b_{ij} is chosen so (8) holds, where $\Lambda_{ij}(g)$ is the volume fraction in C_{ij} lying under the curve g .

Figure 2 contains an example of a linear interface $l(x) = mx + b$ in which only two of the column sums, namely, S_{i-1} and S_i , are exact, yet the approximate interface $\tilde{g}_{ij} = m_{ij}x + b_{ij}$, *exactly reproduces the line l* if m_{ij} is given by (9) with $\alpha = 0$ and $\beta = -1$ and b_{ij} is chosen so $\Lambda_{ij}(\tilde{g}) = \Lambda_{ij}(l)$ where $\Lambda_{ij}(l)$ is the volume fraction in C_{ij} lying below the line l . (See **Example 1** on page 136 for additional details.)

Figure 3 contains an example of the arc of a circle, $c_\epsilon(x)$ that passes through the center cell C_{ij} of the 3×3 block B_{ij} , but for which the center column sum

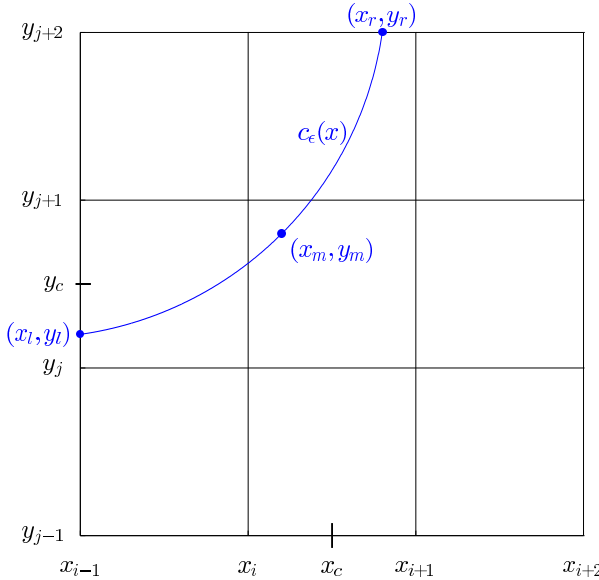


Figure 3. This figure contains an example of an interface $c_\epsilon(x)$, which is a circle that satisfies (5)–(6), but for which the center column sum is not exact in any of the four standard rotations of the 3×3 block of cells B_{ij} centered on the cell C_{ij} . Consequently, the only reasonable approximation m_{ij} to $c'_\epsilon(x_c)$ of the form (9) is with $\alpha = 0$ and $\beta = -1$, which must necessarily have a nonexact center column sum S_i . By **Theorem 3** below, which is the basis for the principal result of this article, if the constraint in (5) and (6) is satisfied, then the center column sum S_i *must be exact to $O(h)$* . In other words, in this case the constraint in (5)–(6) implies (12) (with $\alpha = 0$) must hold. This is sufficient to prove (13), namely, $|m_{ij} - c'_\epsilon(x_c)| \leq \bar{C}\kappa_{\max}h$, which is Theorem 23 of [23]. Finally, by **Theorem 4**, the approximate interface $\tilde{g}_{ij}(x)$ with the slope m_{ij} as given above must be a second-order accurate approximation to $c_\epsilon(x)$ in the max norm.

S_i is not exact. However, by [Theorem 3](#) below, S_i is exact to $O(h)$, as defined in [\(12\)](#). It follows from [Theorem 23](#) of [\[23\]](#) and [Theorem 4](#) in this article that the approximation to the interface $\tilde{g}_{ij}(x)$, with the slope m_{ij} given by [\(9\)](#) with $\alpha = 0$ and $\beta = -1$, will still be pointwise second-order accurate in h , even though S_i is only exact to $O(h)$.

The convention followed in each of these examples is that material 1 *lies below* the interface. However, in practice the 3×3 block B_{ij} centered on a cell C_{ij} containing a portion of the interface can have material 1 lying above, below, to the right or to the left of the interface. In [Section 2.1](#) I present an algorithm for determining which of the four standard rotations of the 3×3 block of cells B_{ij} , namely, rotation clockwise by 0, 90, 180, or 270 degrees, will orient the block B_{ij} so material 1 lies below the interface.

[Theorem 4](#), which is the main result of this article, follows from proving that if [\(5\)–\(6\)](#) holds, then in at least one of these four standard orientations of the block B_{ij} there will always exist at least one column sum that is exact and a second column sum that is either exact or exact to $O(h)$. A more detailed discussion of these issues is contained in [Section 3](#).

[Figure 1](#) contains an example in which one orientation of B_{ij} contains three exact column sums. [Figure 2](#) contains an example in which in two different orientations of B_{ij} contain two exact column sums. [Figure 3](#) contains an example in which in two different orientations B_{ij} contain one exact column sum and one column sum that is exact to $O(h)$. (Note: rotation of the block B_{ij} in [Figure 3](#) by 180 or 270 degrees clockwise results in a configuration in which material 1 lies *above* the interface and therefore, neither [\(11\)](#) nor [\(12\)](#) is true.)

(5) The constraint in [\(5\)–\(6\)](#) is sufficient to ensure *filaments* or *fingers* of the type shown in [Figure 4](#) will not occur on a grid with cell size h where κ_{\max} is the maximum magnitude of curvature of the filament. In this article I have not attempted to catalog all of the ways in which a filament of width $w < h$ can occur in an arbitrary C^2 simple closed curve lying in the domain Ω . It could be that the constraint in [\(5\)–\(6\)](#) is sufficient to ensure if the interface is a simple closed curve in Ω , then all such filaments will be resolved to pointwise second-order accuracy in h . However, I have not attempted to prove this here. The result in this article concerning filaments of the type illustrated in [Figure 4](#) is only a *local* result. In other words, in all of what follows I am explicitly excluding interfaces \mathbf{z} such that for two disjoint intervals (s_l, s_r) and $(\tilde{s}_l, \tilde{s}_r)$ the two *separate* portions of the interface $\mathbf{z}(s) = (x(s), y(s))$ for $s_l < s < s_r$ and $\mathbf{z}(\tilde{s}) = (x(\tilde{s}), y(\tilde{s}))$ for $\tilde{s}_l < \tilde{s} < \tilde{s}_r$, occupy the same 3×3 block of cells B_{ij} .

1.2. The volume-of-fluid interface reconstruction problem. Consider the following problem. Given only the collection of volume fractions Λ_{ij} in the grid Ω^h

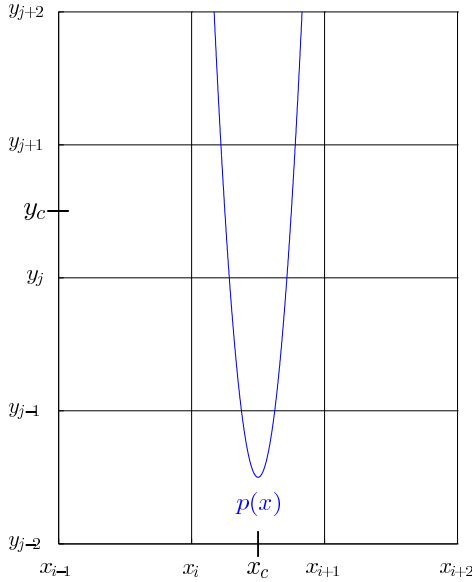


Figure 4. The interface $p(x)$ is a model of a *filament* of material 2 contained entirely within the center column of the 3×4 block of cells $[x_{i-1}, x_{i+2}] \times [y_{j-2}, y_{j+1}]$. In this example $h = 1$, the interface is the parabola $p(x) = 32(x - x_c)^2 + (y_{j-1} - h/2)$ and hence, it follows that κ_{\max} does not satisfy the constraint in (5)–(6), since $\kappa_{\max} = 64 > 2/(33h) = C_h/h$. This indicates the filament is underresolved on this grid. In general, the constraint in (5)–(6) ensures the interface does not have sharp or “hairpin” turns on the scale h of the cell in which one wants to reconstruct the interface.

covering Ω , *reconstruct* $\mathbf{z}(s)$ in the following way. For each cell C_{ij} in Ω^h for which $0 < \Lambda_{ij} < 1$, find a piecewise linear approximation $\tilde{\mathbf{z}}$ to \mathbf{z} as shown, for example, in Figure 1. Furthermore, the approximate interface $\tilde{\mathbf{z}}$ must have the property that the volume fractions $\tilde{\Lambda}_{ij}$ due to $\tilde{\mathbf{z}}$ are identical to the volume fractions Λ_{ij} due to \mathbf{z} ,

$$\tilde{\Lambda}_{ij} = \Lambda_{ij} \quad \text{for all cells } C_{ij} \text{ in } \Omega^h. \tag{18}$$

An algorithm for finding such an approximation is known as a *piecewise linear* volume-of-fluid interface reconstruction method. More generally, there are volume-of-fluid interface reconstruction methods that produce other types of approximations to the interface, such as with piecewise constant [18; 19] and piecewise parabolic [26] functions. However, this article is only concerned with piecewise linear approximations to the interface of the form (7).

Although these algorithms have historically been known as “volume-of-fluid” methods, one can use them to model the interface between any two (or more) materials, including two gases [7], a gas (or vacuum) and a solid [25], a liquid and a solid [14], two solids and vacuum [15; 16], or any other combination of materials.

The design of volume-of-fluid interface reconstruction methods for reconstructing multiple interfaces in problems with more than two materials, especially a large number of distinct materials, is currently a very active area of research.

The property (18) that $\tilde{\Lambda}_{ij} = \Lambda_{ij}$ in all cells in Ω^h is the principal feature that distinguishes volume-of-fluid interface reconstruction methods from other interface reconstruction or tracking methods. It ensures the computational value of the total volume of each material is *exact* to within machine precision. In other words, all volume-of-fluid interface reconstruction methods are conservative in that they conserve the volume of each material in the computation. This is essential if the interface reconstruction method is part of a conservative finite difference method designed to approximate solutions of a system of hyperbolic conservation laws since, for example, in order to obtain the correct shock speed it is necessary for all of the conserved quantities to be conserved by the underlying numerical method (e.g., see [12]). More generally, a necessary condition for the numerical method to converge to the correct weak solution of a system of hyperbolic conservation laws is all of the quantities conserved in the system of conservation laws must also be conserved by the numerical method [11; 13].

Volume-of-fluid methods have been used by researchers to track material interfaces since at least the mid 1970s (e.g., [18; 19]). Researchers have developed a variety of volume-of-fluid algorithms for modeling everything from flame propagation [4] to curvature and solidification [5]. In particular, the problem of developing high-order accurate volume-of-fluid methods for modeling the curvature and surface tension of an interface has received a lot of attention [1; 3; 5; 6; 9; 32; 22; 26]. Volume-of-fluid methods were among the first algorithms to be implemented in codes developed at national laboratories, both in the US [8; 10; 17; 18; 30; 31] and elsewhere [20; 33; 34; 35], for tracking interfaces in a variety of difficult fluid flow and material deformation problems.

The present article is only concerned with the accuracy one can obtain using a volume-of-fluid interface reconstruction algorithm to approximate a given *stationary* interface $z(s)$. The related problem of approximating the movement of the interface in time, for which one would use a *volume-of-fluid advection algorithm* is not addressed here. See, for example, [2; 21; 26; 27; 28] for a description and analysis of several such algorithms.

2. Essential background material

2.1. Rotation and/or reflection of the 5×5 block of cells \tilde{B}_{ij} . Given a cell C_{ij} that contains a portion of the interface it is expedient to consider the 5×5 block of cells \tilde{B}_{ij} centered on C_{ij} rotated clockwise by 0, 90, 180, or 270 degrees about (x_c, y_c) and/or reflected about the vertical line $x = x_c$ or the horizontal line $y = y_c$.

where (x_c, y_c) is the center of the cell C_{ij} as shown in Figures 2, 3 and 4. (Figure 1 on page 203 of [24] contains an illustration of the 5×5 block of cells \tilde{B}_{ij} .) This is because, with the proper choice of one of these four rotations, one can orient the block \tilde{B}_{ij} so the interface can either be written as a single-valued function $y = g(x)$ or $x = G(y)$ of the independent variable x (resp. y) such that in this new coordinate frame the column sum S_i corresponds to the integral of $g(x) - y_{j-1}$ (resp. $G(y) - x_{i-1}$) over the interval $[x_i, x_{i+1}]$ (resp. $[y_j, y_{j+1}]$) and similarly for the column sums S_{i-1} and S_{i+1} . I use the reflection about the line $x = x_c$ to transform cases such as the reflection of the case shown in Figure 3 about the line $x = x_c$ into the case shown in Figure 3, and similarly for reflections about the line $y = y_c$. This enables one to reduce all of the various ways the interface can enter the 3×3 block of cells B_{ij} , pass through the center cell C_{ij} , and leave the block B_{ij} to two canonical cases, namely, Configuration A and Configuration B below.

It is important to note one does not need to perform these coordinate transformations in order to *prove* the piecewise linear volume-of-fluid interface reconstruction algorithm defined in (7)–(10) produces a second-order accurate approximation to the exact interface. Rather, these coordinate transformations are simply an expedient that allows one to reduce consideration of all of the various ways the interface can enter B_{ij} , pass through C_{ij} , and then leave B_{ij} to two canonical cases. This is a consequence of the symmetry lemma on page 119 of [24], from which it follows that all such configurations of the interface with respect to the 3×3 block of cells B_{ij} are equivalent to one of the following two cases.¹

Configuration A: The interface enters B_{ij} across its left edge and exits across its right edge as shown, for example, in Figure 1. In this case the best slope for one to use is m_{ij} defined by $\alpha = 1$ and $\beta = -1$ in (9), although either of the other two slopes given by $\alpha = 0$ and $\beta = -1$ or $\alpha = 1$ and $\beta = 0$ will also furnish a pointwise second-order accurate approximation of the form (7)–(10) to the interface in the center cell C_{ij} .

Configuration B: The interface enters B_{ij} across its left edge and exits across its top edge as shown, for example, in Figures 2 and 3. In this case one *must* use the slope m_{ij} in (9) with $\alpha = 0$ and $\beta = -1$ in order to produce a pointwise second-order accurate approximation of the form (7)–(10) to the interface in the center cell C_{ij} .

¹The symmetry lemma in [23] ensures that if (5)–(6) holds, then each of the ways the interface can enter the block B_{ij} , pass through the center cell C_{ij} , and exit B_{ij} is equivalent to one of four canonical cases: I–IV. By Lemma 11 in [23] Case I cannot occur and a rotation of the block B_{ij} clockwise by 90° transforms Case III into Case II, thereby leaving only Case II, which is Configuration A, and Case IV, which is Configuration B.

Note that each 5×5 block of cells \tilde{B}_{ij} centered on a cell C_{ij} containing a portion of the interface will have its own rotation and/or reflection; that is, the rotation and/or reflection is only performed *locally*, solely for the purpose of determining the slope m_{ij} of the approximate interface in the ij -th cell C_{ij} . Different rotations and/or reflections will, in general, be required for different 5×5 blocks of cells centered on different cells C_{ij} that contain parts of the interface. Furthermore, one only uses these coordinate transformations to determine a first-order accurate approximation m_{ij} to $g'(x_c)$ in the center cell. The grid Ω^h covering the domain Ω always remains the same. Thus, if one is using the interface reconstruction algorithm as part of a numerical method to solve a more complex problem than the one posed here, e.g., the movement of a fluid interface where the underlying fluid flow is a solution of the Euler or Navier–Stokes equations, it is not necessary to perform a coordinate transformation on the underlying numerical fluid flow solver.

There are a variety of techniques for determining which of the four rotations and which reflection, if any, will orient the 3×3 block B_{ij} so the interface can be written as a single-valued function of one of the independent variables x or y , such that in the rotated coordinates the column sum S_i corresponds to the integral of $g(x) - y_{j-1}$ (resp. $G(y) - x_{i-1}$) over the interval $[x_i, x_{i+1}]$ (resp. $[y_j, y_{j+1}]$) and similarly for the column sums S_{i-1} and S_{i+1} . The simplest technique is probably the algorithm described in Section 3 of [24], a variation of which I will now describe.

Step I: Given a cell C_{ij} that contains a portion of the interface $\mathbf{z}(s)$, or equivalently, a cell C_{ij} in which $0 < \Lambda_{ij} < 1$, rotate the 5×5 block of cells \tilde{B}_{ij} centered on C_{ij} together with their associated volume fractions by 0, 90, 180, or 270 degrees so in the *rotated coordinate frame* the bottom row of cells in the 5×5 block \tilde{B}_{ij} satisfy

$$\Lambda_{i-2,j-2} = 1, \quad \Lambda_{i-1,j-2} = 1, \quad \Lambda_{i,j-2} = 1, \quad \Lambda_{i+1,j-2} = 1, \quad \Lambda_{i+2,j-2} = 1.$$

This ensures that the interface does not cross the bottom edge of the 3×3 block of cells B_{ij} .

Step II: Now examine the left and right edges of the 5×5 block of cells \tilde{B}_{ij} . If

$$\Lambda_{i-2,j-2} = 1, \quad \Lambda_{i-2,j-1} = 1, \quad \Lambda_{i-2,j} = 1, \quad \Lambda_{i-2,j+1} = 1, \quad \Lambda_{i-2,j+2} = 1,$$

then the interface must cross the top and right-hand edges of the 3×3 block of cells B_{ij} . In this case reflect the cells together with their associated volume fractions about the vertical line $x = x_c$ in order to orient the block \tilde{B}_{ij} so the interface only crosses the left-hand and top edges of the 3×3 block B_{ij} as shown in Figures 2 and 3. (Lemma 11 of [23] ensures any interface of the form $y = g(x)$ on the interval $[x_{i-1}, x_{i+2}]$ or $x = G(y)$ on the interval $[y_{j-1}, y_{j+2}]$ that satisfies the constraint

in (5)–(6) cannot enter the block B_{ij} across a given edge, pass through the center cell C_{ij} , and then exit B_{ij} across the same edge.)

Not only does this procedure reduce the number of cases one must consider during the course of proving the results in this article and those in [23], it also reduces the number of cases one must consider in the implementation of the algorithm described in [24]. In all of what follows I will express the interface as $y = g(x)$ and, unless noted otherwise, the coordinates of the edges of the cells in the 3×3 block B_{ij} centered on a cell C_{ij} containing the interface will be denoted by $x_{i-1}, x_i, x_{i+1}, x_{i+2}$ and $y_{j-1}, y_j, y_{j+1}, y_{j+2}$, with it being understood that a transformation of the coordinate system as described above may have been performed in order for this representation of the interface to be valid, and that the names of the variables x and y might have been interchanged in order to write the interface as $y = g(x)$.

2.2. Column sums. Let C_{ij} be a cell such that $0 < \Lambda_{ij} < 1$ and assume the 3×3 block of cells B_{ij} centered on C_{ij} has been rotated by 0, 90, 180, or 270 degrees as described above, so the interface $z(s)$ can be expressed as a single-valued function $y = g(x)$ or $x = G(y)$ of the independent variable x (resp. y). Thus, in this new coordinate frame the column sum S_i corresponds to the integral of $g(x) - y_{j-1}$ (resp. $G(y) - x_{i-1}$) over the interval $[x_i, x_{i+1}]$ (resp. $[y_j, y_{j+1}]$) and similarly for the column sums S_{i-1} and S_{i+1} . The accuracy of the piecewise linear approximation to the interface in C_{ij} defined in (7)–(10) depends entirely on the accuracy with which the column sums S_{i-1} , S_i and S_{i+1} approximate the volume / area under the interface in their respective columns from the base $y = y_{j-1}$ of the block B_{ij} to the interface. The purpose of this section is to give the reader an understanding of why this must be so.

2.2.1. Exact column sums. Consider the three columns in the 3×3 block of cells B_{ij} centered on the cell C_{ij} . The column sums S_{i-1} , S_i , and S_{i+1} are a nondimensional way of storing the total volume / area of material 1 in these three columns. In order to approximate the portion of the interface $g(x)$ in the ij -th cell C_{ij} to second-order in h with the piecewise linear function $\tilde{g}_{ij}(x)$ defined in (7), one must use two of the three column sums in B_{ij} to compute the slope m_{ij} of $\tilde{g}_{ij}(x)$ as illustrated in the examples in Figures 1 and 2.

To see why this is so, consider an arbitrary column consisting of three cells with left edge $x = x_i$ and right edge $x = x_{i+1}$ and assume the interface can be written as a function $y = g(x)$ on the interval $[x_i, x_{i+1}]$. Assume also the interface enters the column through its left edge and exits the column through its right edge and does not cross the top or bottom edges of the column as, for example, is the case for each of the three columns in the 3×3 block of cells in Figure 1. Then the total volume / area of material 1 that occupies the three cells in this particular column and lies below the interface $g(x)$ is equal to the integral of $g(x) - y_{j-1}$ over the

interval $[x_i, x_{i+1}]$. Thus, (11) holds; in other words, the i -th column sum S_i is exact.

Exact column sums are the key to ensuring a volume-of-fluid interface reconstruction algorithm of the form defined in (7)–(10) is second-order accurate. Given the 3×3 block of cells B_{ij} centered on a cell C_{ij} that contains a portion of the interface $y = g(x)$, the main result in this article, [Theorem 4](#), is based on how well the column sums S_{i-1} , S_i and S_{i+1} approximate the *normalized* integral of g in that particular column,

This is because, by (9), the slope m_{ij} of the piecewise linear approximation \tilde{g}_{ij} to the interface g in C_{ij} will be the divided difference of two of these column sums. In other words, m_{ij} is chosen to be one of the following three quantities:

$$m_{ij}^l = (S_i - S_{i-1}), \quad (19a)$$

$$m_{ij}^c = \frac{(S_{i+1} - S_{i-1})}{2}, \quad (19b)$$

$$m_{ij}^r = (S_{i+1} - S_i). \quad (19c)$$

A consequence of [Theorem 23](#) in [23] is if two of the column sums $S_{i+\alpha}$ and $S_{i+\beta}$ for some $\alpha, \beta = 1, 0, -1$ with $\alpha \neq \beta$ are exact, then the slope m_{ij} in (9) must satisfy (13). Consequently, by [Theorem 4](#) below, which is a stronger version of [Theorem 24](#) in [23], the piecewise linear approximation $\tilde{g}_{ij}(x)$ defined in (7)–(10) will be a pointwise second-order accurate approximation to the true interface $g(x)$ for all $x \in [x_i, x_{i+1}]$. In fact, $\tilde{g}_{ij}(x)$ will be a pointwise second-order accurate approximation to $g(x)$ for all $x \in [x_{i-1}, x_{i+2}]$, albeit with a slightly larger constant multiplying $\kappa_{\max} h^2$.

Example 1. In order to see why the divided difference of two exact column sums must produce a slope m_{ij} that is a first-order accurate approximation to $g'(x_c)$, the slope of the interface at the center of the interval $[x_i, x_{i+1}]$, consider the case of a linear interface $l(x) = mx + b$ as shown in [Figure 2](#). In this particular orientation of the 3×3 block of cells B_{ij} the interface g has two exact column sums; namely, the first and second ones, S_{i-1} and S_i , where S_i denotes the column sum associated with the interval $[x_i, x_{i+1}]$ and S_{i-1} denotes the column sum associated with the interval $[x_{i-1}, x_i]$. It is easy to check that

$$\begin{aligned} m &= \frac{1}{h^2} \int_{x_i}^{x_{i+1}} (l(x) - y_{j-1}) dx - \frac{1}{h^2} \int_{x_{i-1}}^{x_i} (l(x) - y_{j-1}) dx \\ &= (S_i - S_{i-1}) = m_{ij}^l. \end{aligned}$$

In this example the divided difference m_{ij}^l in (19a) of the column sums S_{i-1} and S_i is *exactly* equal to the slope m of the linear interface $l(x) = mx + b$ and hence,

since (8) must also hold, the piecewise linear approximation $\tilde{g}_{ij}(x)$ defined in (7) coincides with the true interface $l(x)$ in C_{ij} ,

$$|l(x) - \tilde{g}_{ij}(x)| = 0 \quad \text{for all } x \in [x_i, x_{i+1}].$$

If the exact interface is a line, then it is always the case that in one of the four standard rotations of the 3×3 block of cells B_{ij} at least one of the divided differences of the column sums in (19) is exact. For example, note that in the case shown in Figure 2 one can rotate the 3×3 block of cells B_{ij} 90 degrees clockwise and in this new orientation the correct slope to use when forming the piecewise linear approximation $\tilde{g}_{ij}(x) = m_{ij} + b_{ij}$ to the line $l(x) = mx + b$ will be $m_{ij} = m_{ij}^r$ as defined in (19c), which again exactly equals the slope m of $l(x)$.

Of course, in general, the divided difference of two of the column sums will not be precisely equal to the slope of the interface at the midpoint x_c of the interval $[x_i, x_{i+1}]$ as in the preceding example. However, as a consequence of Theorem 3 below, and Theorem 23 of [23], if h satisfies (5)–(6), one can always find an orientation of the 3×3 block of cells B_{ij} such that at least two of the column sums are sufficiently accurate that one of the divided differences in (19) satisfies (13).

Once one has chosen an orientation of the 3×3 block of cells B_{ij} such that at least two of the column sums are sufficiently accurate that one of the divided differences in (19) satisfies (13), one uses the constraint in (8), namely, $\Lambda_{ij}(\tilde{g}) = \Lambda_{ij}(g)$, to form the piecewise linear approximation $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ to the interface. In other words, given m_{ij} , the constraint $\Lambda_{ij}(\tilde{g}) = \Lambda_{ij}(g)$ determines b_{ij} .

2.2.2. Column sums that are exact to $O(h)$. One might expect there exists a value of C_h that will ensure if the cell size h satisfies the constraint in (5)–(6), then after one of the four standard rotations of the 3×3 block B_{ij} about its center, the block will always have at least two exact column sums. Unfortunately, as the following example demonstrates, there is no bound of the form (5)–(6) which, for a fixed h , will ensure a C^2 interface will always have at least two exact column sums in one of the four standard orientations of the grid.

Example 2. Consider the curve $c_\epsilon(x)$ shown in Figure 3 where $0 < \epsilon < h$ is a small parameter. One can always find a circle $c_\epsilon(x)$ that passes through the three noncollinear points $(x_l, y_l) = (x_{i-1}, y_j + \epsilon)$, $(x_m, y_m) = (x_i + \epsilon, y_{j+1} - \epsilon)$ and $(x_r, y_r) = (x_{i+1} - \epsilon, y_{j+2})$ as shown in the figure. As $\epsilon \rightarrow 0$ the arc of the circle passing through (x_l, y_l) , (x_m, y_m) and (x_r, y_r) tends to the chord connecting (x_l, y_l) and (x_r, y_r) which, since the curvature of the chord is 0, implies the radius r^ϵ of $c_\epsilon(x)$ tends to ∞ . Therefore, no matter how small one chooses C_h there exists $\epsilon_0 > 0$, such that the radius r^ϵ satisfies $h \leq C_h r^\epsilon$, or equivalently, $h \leq C_h (\kappa_{\max}^\epsilon)^{-1}$ for all $\epsilon \leq \epsilon_0$. Hence, for $\epsilon \leq \epsilon_0$ the circle $c_\epsilon(x)$ satisfies (5)–(6). However, since by construction $y_j < y_l$ and $x_r < x_{i+1}$, the center column sum will not be exact

in any of the four standard orientations of the block B_{ij} . Consequently, if one wants to construct an approximation to $c_\epsilon(x)$ based solely on the volume fraction information contained in the 3×3 block B_{ij} centered on the cell C_{ij} that contains the point (x_m, y_m) on the interface $c_\epsilon(x)$, the best result one can hope for is that the center column sum S_i is exact to $O(h)$ in the sense defined in (12).

All of the work in Sections 4.1 to 4.3 of this article is devoted to proving if (5)–(6) holds, then in cases such as the one shown in Figure 3 the error between the column sum S_i and the normalized integral of the interface $g(x)$ in that column is $O(h)$; i.e., the inequality in (12) holds with $\alpha = 0$, where $\bar{C} > 0$, defined in (53) below, is a global constant independent of h and κ_{\max} .

In [23], in order to prove if the center cell S_i is not exact, but is exact to $O(h)$, then one of the divided differences m_{ij}^l or m_{ij}^r is still sufficiently accurate that (13) must hold, it was necessary to have a more stringent restriction on the cell size than one of the form (5)–(6).² This restriction was $h \leq \kappa_{\max}^{-2}$, which for κ_{\max} large enough is more restrictive than the constraint in (5)–(6). In all of the other ways in which the interface enters the 3×3 block B_{ij} , passes through the center cell C_{ij} and exits the block B_{ij} , the constraint in (5)–(6) is sufficient to prove there is an orientation of B_{ij} such that at least two of the column sums are *exact*, and hence one of the divided differences in (19) satisfies (13). The primary purpose of this article is to prove if the center column sum S_i is not exact the more restrictive constraint $h \leq \kappa_{\max}^{-2}$ is not necessary. In other words, if the exact interface g satisfies (5)–(6), then for every cell C_{ij} that contains a portion of the interface, after one of the four standard rotations of the 3×3 block B_{ij} about its center (x_c, y_c) there are at least two column sums that are sufficiently accurate (meaning either exact or exact to $O(h)$), that one of the divided differences in (19) satisfies (13).

The purpose of this article is to show the constraint in (5)–(6) is sufficient to ensure in cases such as the one shown in Figure 3, the error between the center column sum S_i and the normalized integral of the interface g in the center column satisfies (12) with $\alpha = 0$ and hence, the error in the approximation m_{ij}^l to the slope $g'(x_c)$ is small enough that (13) still holds. Once this is done, by (13) the slope m_{ij}^l is a first-order accurate approximation to the first derivative $g'(x_c)$ of the interface at the center x_c of the interval $[x_i, x_{i+1}]$. One can then show the piecewise linear

²A consequence of the proof of Theorem 10 in [23] is if the interface satisfies (5)–(6) and passes through the center cell C_{ij} of the 3×3 block of cells B_{ij} , then after one of the four standard rotations of B_{ij} about its center, either the left or right column sum must be exact. If it is the right column sum S_{i+1} that is exact, then reflection of the block B_{ij} about the vertical line $x = x_c$ results in the block being oriented so the left column sum S_{i-1} is now exact as shown in Figure 3. Thus, it is only necessary to consider the case in which the center column sum S_i is exact to $O(h)$ and the left column sum S_i is exact, as illustrated in Figure 3.

approximation $\tilde{g}_{ij}(x) = m_{ij}^l x + b_{ij}$ is a second-order accurate approximation to the interface $g(x)$ on the interval $[x_i, x_{i+1}]$. This is [Theorem 4](#).

3. An overview of the structure of the proof

Sections 3 and 4 of [\[23\]](#) contain a proof of the following. If h satisfies the constraint in [\(15\)](#), then by rotating the 3×3 block of cells B_{ij} centered on C_{ij} by 0, 90, 180, or 270 degrees clockwise, one can find a coordinate frame in which there are at least two distinct column sums $S_{i+\alpha}$ and $S_{i+\beta}$ such that their divided difference [\(9\)](#) satisfies [\(13\)](#). Section 3.1 of [\[23\]](#) contains a proof that the constraint $h \leq \tilde{C}_h \kappa_{\max}^{-1}$, where \tilde{C}_h is defined in [\(A.6\)](#), is sufficient to ensure the interface has two *exact* column sums *in all but one of the ways* in which the interface g enters the 3×3 block of cells B_{ij} , passes through its center cell C_{ij} , and exits B_{ij} . The exception is the case in which the center column sum S_i is not exact, but only exact to $O(h)$, as illustrated in [Figure 3](#). Sections 3.2–3.4 of [\[23\]](#) are devoted to proving that, in this latter case, the center column sum S_i is exact to $O(h)$. However, the proof requires the second of the two constraints in [\(15\)](#) above, namely $h \leq \kappa_{\max}^{-2}$, to hold.

The purpose of [Section 4](#) is to prove the weaker constraint in [\(5\)](#), with C_h defined in [\(6\)](#), is sufficient to ensure that in cases such as the one described above, the center column sum S_i is exact to $O(h)$. This, together with the results from [Section 4](#) of [\[23\]](#), ensure the approximation $\tilde{g}_{ij}(x)$ in [\(7\)](#) is a second-order accurate approximation in the max norm to $g(x)$ on the interval $[x_i, x_{i+1}]$.

[Theorem 4](#) in [Section 5](#), which is the main result of this article, is a stronger version of [Theorem 24](#) of [\[23\]](#). Namely, if $h \leq C_h \kappa_{\max}^{-1}$, then [\(4\)](#) holds. This theorem is based on the results in [Section 4](#) below.

The terms in the error bound on the right-hand side of [\(4\)](#) that have changed from [Theorem 24](#) of [\[23\]](#) are the positive constants κ_{\max} and C_m . In particular, the linear dependence on κ_{\max} of the max norm of the difference $z - \tilde{z}$ is explicitly displayed in the present article. In [\[23\]](#) the constant C_m was of the form $50\kappa_{\max}/3 + C_S$, where C_S is a constant, which is independent of κ_{\max} and h . The new value of C_m is defined in [\(59\)](#) below.

4. The center column sum S_i is exact to $O(h)$

The purpose of the work in this section is to prove the constraint on h in [\(5\)](#)–[\(6\)](#) is sufficient to ensure that if the center column sum S_i is not exact, then it must be exact to $O(h)$. This is the case in which the center column sum is not exact in each of the four standard orientations of the block B_{ij} as shown, for example, in [Figure 3](#). The main result of this section is stated explicitly in [Theorem 3](#) below. Note that it is only necessary to prove this result in one of the four standard orientations of the grid, since the proof of the other three cases is essentially the same. Note also that

in this one case the interface $g(x)$ is monotonically increasing. In Lemma 13 of [23] it is proven if the interface is a nonmonotonically increasing function of x in B_{ij} , then the constraint in (5)–(6) is sufficient to ensure it has two exact column sums, regardless of the manner in which it enters the 3×3 block of cells B_{ij} , passes through the center cell C_{ij} and exits the block B_{ij} again. See Section 3.1 of [23] and, in particular, Lemma 13 for details.

Notation. For convenience, in this section the edges of the 3×3 block of cells B_{ij} will be denoted x_0, x_1, x_2, x_3 , and y_0, y_1, y_2, y_3 , as shown, for example, in Figures 5, 6, and 7. Thus, the 3×3 block B_{ij} will be identified with the 3×3 block $B_{1,1} = [x_0, x_3] \times [y_0, y_3]$ and the center cell C_{ij} will be identified with $C_{1,1} = [x_1, x_2] \times [y_1, y_2]$, the center cell of $B_{1,1}$. Furthermore, in Section 4.1 it will be convenient to translate the coordinate system so the origin $(0, 0)$ coincides with the point (x_0, y_1) . This results in the following relations, which will be used in several of the proofs below: $(x_0, y_1) = (0, 0)$, $(x_1, y_2) = (h, h)$, and $(x_2, y_3) = (2h, 2h)$. For example, see Figure 5.

4.1. The comparison circle $\tilde{z}(s)$. To begin, define the parameters γ and R by

$$\gamma \stackrel{\text{def}}{=} \frac{1}{5} \sqrt{\frac{h\kappa_{\max}}{C_h}}, \quad (20a)$$

$$R \stackrel{\text{def}}{=} 5 \sqrt{\frac{C_h h}{\kappa_{\max}}}, \quad (20b)$$

where C_h is defined in (6), and note that $R\gamma = h$ and, since $0 < h \leq C_h \kappa_{\max}^{-1}$,

$$0 < \gamma \leq \frac{1}{5}. \quad (21)$$

Now consider the *comparison circle* $\tilde{z}(s) = (\tilde{x}(s), \tilde{y}(s))$, which is defined by

$$\tilde{x}(s) = R \sin(\phi_0 + s/R) - R \sin \phi_0, \quad (22a)$$

$$\tilde{y}(s) = -R \cos(\phi_0 + s/R) + R \cos \phi_0, \quad (22b)$$

where ϕ_0 is a parameter defined by

$$\phi_0 = \frac{\pi}{4} - \sin^{-1} \frac{\gamma}{\sqrt{2}}. \quad (23)$$

Note that $\tilde{z}(s) = (\tilde{x}(s), \tilde{y}(s))$ is a circle with radius R , center $(-R \sin \phi_0, R \cos \phi_0)$ and that s is arc length along the circle. In what follows $(x, \tilde{c}(x))$ will sometimes be used to denote the graph of $\tilde{z}(s)$ reparametrized as a function of x , just as $(x, g(x))$ is sometimes used to denote the graph of the interface $z(s)$.

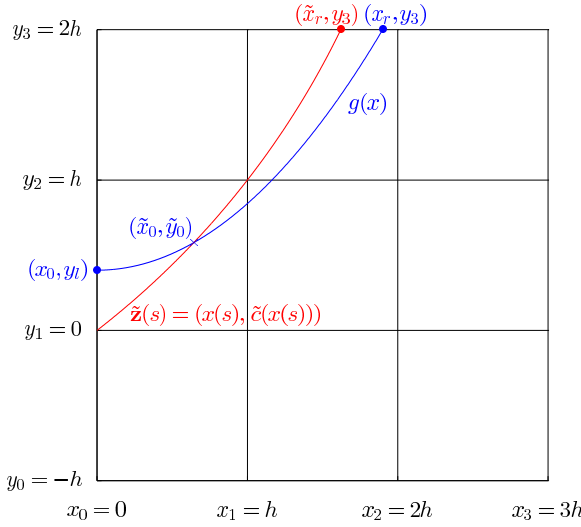


Figure 5. The interface g (shown in blue) is an arbitrary *strictly monotonically increasing* function that enters the 3×3 block B_{ij} through its left edge at the point (x_0, y_l) with $y_1 < y_l < y_2$, passes through the center cell C_{ij} , and exits B_{ij} through the top of its center column S_i at the point (x_r, y_3) , with $x_1 < x_r < x_2$. By [Corollary 2](#) in [Section 4.2](#) if the maximum magnitude κ_{\max} of the curvature κ^g of g satisfies $\kappa_{\max} \leq C_h h^{-1}$, then $x_2 - x_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$, where the constant \tilde{C} , defined in [\(32\)](#), is independent of h and κ_{\max} . The proof is based on forming a comparison function $\tilde{z}(x) = (x(s), \tilde{c}(x(s)))$ (shown in red), which is a circle that passes through the points $(x_0, y_1) = (0, 0)$ and $(x_1, y_2) = (h, h)$, and proving the abscissa \tilde{x}_r of the point $(\tilde{x}_r, \tilde{c}(\tilde{x}_r)) = (\tilde{x}_r, y_3)$ where \tilde{c} exits the 3×3 block B_{ij} satisfies $x_2 - \tilde{x}_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$. One then uses [Theorem 2](#) in [Section 4.2](#), the “comparison circle theorem”, to prove the interface g must eventually lie below the graph of \tilde{c} in the open interval (\tilde{x}_0, x_2) . This implies $\tilde{x}_r < x_r$ and hence, $x_2 - \tilde{x}_r < x_2 - x_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$.

Lemma 1. *Let*

$$s_1 = 2R \sin^{-1} \frac{\gamma}{\sqrt{2}}, \quad (24a)$$

$$s_2 = R \cos^{-1}(\cos \phi_0 - 2\gamma) - R\phi_0, \quad (24b)$$

$$s_3 = R \sin^{-1}(\sin \phi_0 + 2\gamma) - R\phi_0. \quad (24c)$$

Then

$$\tilde{z}(0) = (x_0, y_1) = (0, 0), \quad (25a)$$

$$\tilde{z}(s_1) = (x_1, y_2) = (h, h), \quad (25b)$$

$$\tilde{z}(s_2) = (\tilde{x}(s_2), y_3) = (\tilde{x}_r, 2h), \quad (25c)$$

$$\tilde{z}(s_3) = (x_2, \tilde{y}(s_3)) = (2h, \tilde{y}(s_3)). \quad (25d)$$

The proof is left to the reader.

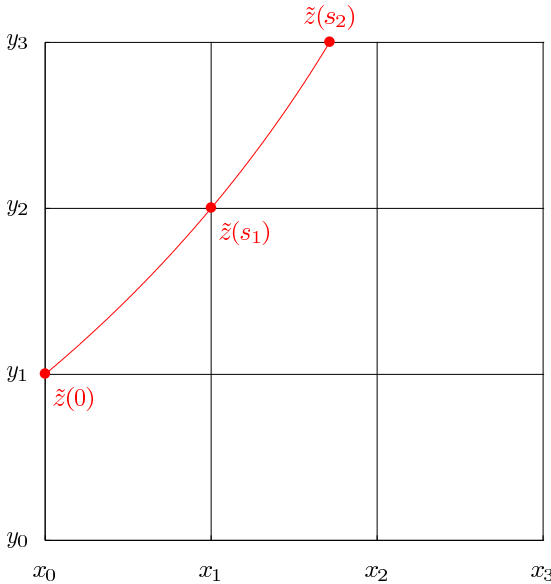


Figure 6. To better visualize the upper and lower bounds on the arc between $\tilde{z}(0) = (0, 0)$ and $\tilde{z}(s_2) = (\tilde{x}(s_2), 2h)$ in [Lemma 4](#), this figure contains an example of the comparison circle $\tilde{z}(s)$ in the 3×3 block $B_{ij} = B_{1,1}$ centered on the cell $C_{ij} = C_{1,1}$.

- Remarks.** (a) In [\(25c\)](#) the variable $\tilde{x}_r = \tilde{x}(s_2)$ is the x -coordinate of the point where the graph of the comparison circle $\tilde{z}(s) = (x, \tilde{c}(x))$ exits the top of the 3×3 block B_{ij} . It plays the same role with respect to the function $\tilde{z}(s)$ as the variable x_r plays with respect to the interface $z(s) = (x, g(x))$. In the problem considered in this section only the case $x_r < x_2 = 2h$ is relevant, for otherwise the center column sum S_i is exact. By [Theorem 2](#) “The Comparison Circle Theorem” below, the interface $(x, g(x))$ must lie below the comparison circle $(x, \tilde{c}(x))$ for $x \geq x_1 = h$ and hence, $x_r < x_2$ implies $\tilde{x}_r < x_2$.
- (b) Note also that [Equation \(25d\)](#) guarantees the comparison circle $(x, \tilde{c}(x))$ must extend all the way to the grid line $x = x_2$, thereby ensuring the comparison circle will lie above the interface $(x, g(x))$ for all $x \in [x_1, x_2]$. This is illustrated in [Figure 7](#).
- (c) Finally, note the comparison circle $\tilde{z}(s)$ is a monotonically increasing function of s for s in the interval $[0, s_3]$ and similarly, when written as a function of x , $(x, \tilde{c}(x))$ is a monotonically increasing function of x for x in the interval $[x_0, x_2] = [0, 2h]$.

The following three lemmas and one corollary concerning the quantities ϕ_0 and s_2 will be needed in the proof that $x_2 - \tilde{x}_r$ is $O(\sqrt{\kappa_{\max}}h^{3/2})$ and hence, $x_2 - x_r$ is also $O(\sqrt{\kappa_{\max}}h^{3/2})$.

Lemma 2. $(\cos \phi_0 - \sin \phi_0 = \gamma)$ Let ϕ_0 be defined as in (23):

$$\phi_0 = \frac{\pi}{4} - \sin^{-1} \frac{\gamma}{\sqrt{2}}.$$

Then

$$\cos \phi_0 - \sin \phi_0 = \gamma. \tag{26}$$

Proof. Define β by

$$\sin \beta \stackrel{\text{def}}{=} \frac{\gamma}{\sqrt{2}}$$

so that

$$\phi_0 = \frac{\pi}{4} - \sin^{-1} \frac{\gamma}{\sqrt{2}} = \frac{\pi}{4} - \beta.$$

Then (26) follows from writing ϕ_0 as $\pi/4 - \beta$ and applying the trigonometric identities for the sine and cosine of the difference of two angles:

$$\cos \phi_0 - \sin \phi_0 = \sqrt{2} \sin \beta = \gamma. \quad \square$$

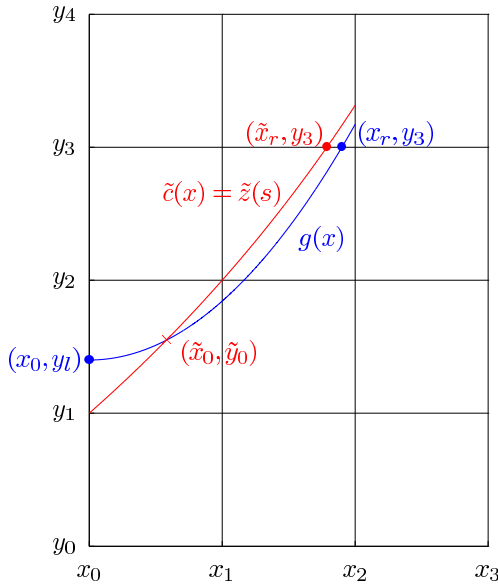


Figure 7. This figure includes the row of cells that lie above the standard 3×3 block of cells $B_{ij} = B_{1,1}$ centered on $C_{ij} = C_{1,1} = [x_1, x_2] \times [y_1, y_2]$ in which the approximation to the monotonically increasing interface g , shown in blue, will be constructed. The difference between the center column sum S_i and the exact volume (i.e., exact area) in B_{ij} under $g(x)$ is the region in the center column that lies under the graph of g and above the line $y = y_3$. By **Theorem 2** the comparison circle $\tilde{c}(x)$, shown in red, bounds $g(x)$ from above for all $x \in [\tilde{x}_0, \tilde{x}_2]$, and hence, allows one to bound the difference between S_i and the integral of $g(x) - y_0$ over $[x_1, x_2]$.

Lemma 3. ($\cos \phi_0 + \sin \phi_0 = \sqrt{2 - \gamma^2}$) Let ϕ_0 be defined as in (23). Then

$$\cos \phi_0 + \sin \phi_0 = \sqrt{2 - \gamma^2}. \quad (27)$$

Proof. As in the proof of the previous lemma let β be defined by $\sin \beta = \gamma / \sqrt{2}$. Then (27) is a consequence of the trigonometric identities for the sine and cosine of the difference of two angles, together with the trigonometric identity

$$\cos(\arcsin(x)) = \sqrt{1 - x^2},$$

as follows:

$$\cos \phi_0 + \sin \phi_0 = \sqrt{2} \cos \beta = \sqrt{2} \cos \left(\sin^{-1} \frac{\gamma}{\sqrt{2}} \right) = \sqrt{2 - \gamma^2}. \quad \square$$

Lemma 4. ($s_2 = O(h)$) Let s_2 be the parameter defined in (24b). Then s_2 satisfies

$$(1 + \sqrt{2})h < s_2 < 4h. \quad (28)$$

Proof. Recall that s is arc length along the comparison circle $\tilde{z}(s) = (\tilde{x}(s), \tilde{y}(s))$ starting at the point $\tilde{z}(0) = (\tilde{x}(0), \tilde{y}(0)) = (x_0, y_1) = (0, 0)$. The length of the arc of the comparison circle from $\tilde{z}(0)$ to $\tilde{z}(s_2)$ consists of two sections. The first section is the arc from $\tilde{z}(0)$ to $\tilde{z}(s_1) = (x_1, y_2) = (h, h)$ while the second section is the arc from $\tilde{z}(s_1)$ to $\tilde{z}(s_2) = (\tilde{x}(s_2), y_3) = (\tilde{x}_r, y_3)$.

The length of the first section is bounded below by the length of the diagonal joining $(0, 0)$ and (h, h) , which has length $\sqrt{2}h$, and is bounded above by the sum of the lengths of the bottom and right edges of the cell that has $(0, 0)$ and (h, h) as its opposite corners; i.e., the edge connecting $(0, 0) = (x_0, y_1)$ and the corner (x_1, y_1) and the edge connecting (x_1, y_1) and the corner $(h, h) = (x_1, y_2)$. Since both of these edges have length h , it follows that the portion of the arc joining $\tilde{z}(0)$ to $\tilde{z}(s_1)$ is bounded above by $2h$.

Since the point $\tilde{z}(s_2)$ lies on the top edge of the 3×3 block B_{ij} and, since $\tilde{z}(s)$ is a monotonically increasing function of s for $0 \leq s \leq s_3$, $\tilde{x}(s_2) = \tilde{x}_2$ must lie between x_1 and x_2 . It follows that a lower bound for the portion of the arc joining $\tilde{z}(s_1)$ to $\tilde{z}(s_2)$ is the length of the side of the cell joining the point $(h, h) = (x_1, y_2)$ and the point (x_1, y_3) on the top edge of $B_{ij} = B_{1,1}$. Since this edge has length h , it follows that a lower bound for the length of the arc joining $\tilde{z}(0)$ to $\tilde{z}(s_2)$, and hence a lower bound for s_2 , is $\sqrt{2}h + h = (\sqrt{2} + 1)h$ as shown in the inequality on the left in (28).

One can find an upper bound for the portion of the arc joining $\tilde{z}(s_1)$ to $\tilde{z}(s_2)$ by using reasoning that is identical to that used to obtain the upper bound on the portion of the arc joining $\tilde{z}(0)$ to $\tilde{z}(s_1)$. This yields an upper bound of $2h + 2h = 4h$ for the entire length of the arc joining $\tilde{z}(0)$ to $\tilde{z}(s_2)$, as shown in the inequality on the right in (28). \square

The proof of the next theorem depends on the following corollary.

Corollary 1. (θ is $O(\gamma)$) Let s_2 be the parameter defined in (24b) and define θ by

$$\theta \stackrel{\text{def}}{=} \frac{s_2}{R}. \quad (29)$$

Then θ satisfies

$$(1 + \sqrt{2})\gamma < \theta < 4\gamma. \quad (30)$$

Proof. One obtains the inequality in (30) by multiplying Equation (28) in Lemma 4 by R^{-1} and recalling that $h = R\gamma$. \square

The following theorem is the key step in the proof that $|x_2 - x_r| = O(\sqrt{\kappa_{\max}} h^{3/2})$.

Theorem 1 ($x_2 - \tilde{x}_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$). The difference $x_2 - \tilde{x}_r$ is bounded above by

$$x_2 - \tilde{x}_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}, \quad (31)$$

where $\tilde{x}_r = \tilde{x}(s_2)$ is defined in (25c) and

$$\tilde{C} \stackrel{\text{def}}{=} \frac{2\sqrt{66}}{3 \cdot 5^4} \{736\sqrt{2} - 349\} \approx 5.995421. \quad (32)$$

Remark 1. (a) One of the consequences of replacing the Lemmas, Theorems, and Corollaries in Sections 3.2–3.4 of [23] with those in Section 4 here is that the term bounding the difference $x_2 - \tilde{x}_r$ in (31) above now depends linearly on $\sqrt{\kappa_{\max}}$. As a result, the term on the right-hand side of (12) and (52) depends linearly on κ_{\max} , which in turn leads to a linear dependence of the bounds in (13) and (58) on κ_{\max} . The analogous bounds in Theorems 15, 23, and 25 of [23] do not depend linearly on κ_{\max} .

(b) As mentioned in Remark 1(a), it is possible that $\tilde{x}_r \geq x_2$. In this case, by the comparison circle theorem below, $x_r > x_2$ and hence, the center column sum S_i must be exact. Since the purpose of this section is to prove S_i must be exact to $O(h)$ if g satisfies (5)–(6) and S_i is not exact, the case in which S_i is exact, or equivalently, $\tilde{x}_r \geq x_2$, is not of interest here.

Proof. Since the coordinate system has been arranged so the origin $(0, 0)$ coincides with the point (x_0, y_1) and hence, $x_2 = 2h = y_3$ (e.g., see Figure 5), it follows that

$$x_2 = 2h = \tilde{y}(s_2).$$

Thus

$$\begin{aligned} x_2 - \tilde{x}_r &= \tilde{y}(s_2) - \tilde{x}(s_2) \\ &= R\{(\cos \phi_0 - \cos(\phi_0 + s_2/R)) - (-\sin \phi_0 + \sin(\phi_0 + s_2/R))\}. \end{aligned} \quad (33)$$

Since $R = 5\sqrt{C_h h / \kappa_{\max}}$, it suffices to show that the quantity inside the curly braces in (33) is $O(\gamma^2) = O(h\kappa_{\max}/C_h)$. One can rewrite (33) as

$$x_2 - \tilde{x}_r = R\{(\cos \phi_0 + \sin \phi_0) - (\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta))\} = RA, \quad (34)$$

where $\theta = s_2/R$ was defined in Corollary 1 above. Consider the quantity A obtained by dividing (34) by R ,

$$A = \{(\cos \phi_0 + \sin \phi_0) - (\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta))\}. \quad (35)$$

Now expand $\cos(\phi_0 + \theta)$ and $\sin(\phi_0 + \theta)$ in a Taylor series about $\cos \phi_0$ and $\sin \phi_0$, respectively, to obtain

$$\begin{aligned} A &= (\cos \phi_0 + \sin \phi_0) - (\cos(\phi_0 + \theta) + \sin(\phi_0 + \theta)) \\ &= -(\cos \phi_0 - \sin \phi_0)\theta + (\cos \phi_0 + \sin \phi_0)\frac{\theta^2}{2!} \\ &\quad + (\cos \phi_0 - \sin \phi_0)\frac{\theta^3}{3!} - (\cos \phi_0 + \sin \phi_0)\frac{\theta^4}{4!} \\ &\quad - (\cos \phi_0 - \sin \phi_0)\frac{\theta^5}{5!} + (\cos \phi_0 + \sin \phi_0)\frac{\theta^6}{6!} \\ &\quad + (\cos \phi_0 - \sin \phi_0)\frac{\theta^7}{7!} - (\cos \phi_0 + \sin \phi_0)\frac{\theta^8}{8!} + \dots \end{aligned}$$

This expression for A can be rewritten as

$$\begin{aligned} A &= -\left((\cos \phi_0 - \sin \phi_0) - (\cos \phi_0 + \sin \phi_0)\frac{\theta}{2}\right)\theta \\ &\quad + \left((\cos \phi_0 - \sin \phi_0) - (\cos \phi_0 + \sin \phi_0)\frac{\theta}{4}\right)\frac{\theta^3}{3!} \\ &\quad - \left((\cos \phi_0 - \sin \phi_0) - (\cos \phi_0 + \sin \phi_0)\frac{\theta}{6}\right)\frac{\theta^5}{5!} \\ &\quad + \left((\cos \phi_0 - \sin \phi_0) - (\cos \phi_0 + \sin \phi_0)\frac{\theta}{8}\right)\frac{\theta^7}{7!} + \dots \end{aligned} \quad (36)$$

Using Lemmas 2 and 3 one can rewrite this series in terms of θ and γ as

$$\begin{aligned} A &= -\left(\gamma - \frac{\theta}{2}\sqrt{2-\gamma^2}\right)\theta + \left(\gamma - \frac{\theta}{4}\sqrt{2-\gamma^2}\right)\frac{\theta^3}{3!} \\ &\quad - \left(\gamma - \frac{\theta}{6}\sqrt{2-\gamma^2}\right)\frac{\theta^5}{5!} + \left(\gamma - \frac{\theta}{8}\sqrt{2-\gamma^2}\right)\frac{\theta^7}{7!} - \dots \end{aligned} \quad (37)$$

The first term, A_1 , in this series is $O(\gamma^2)$. To see this note that the upper bound on θ in (30) implies

$$A_1 = \left(\frac{\theta}{2}\sqrt{2-\gamma^2} - \gamma\right)\theta < (2\sqrt{2}-1)4\gamma^2, \quad (38)$$

where the upper bound on $\sqrt{2-\gamma^2}$ follows from $0 < \gamma \leq 1/5$ in (21). Furthermore, $A_1 > 0$. To see this first note that since θ is always positive the sign of A_1 depends only on the terms in parentheses on the right-hand side of the equal sign in (38). Since $\theta > (1 + \sqrt{2})\gamma > 0$ by (30),

$$\begin{aligned} \frac{A_1}{\theta} &= \left(\frac{\theta}{2} \sqrt{2-\gamma^2} - \gamma \right) > \left(\frac{(1 + \sqrt{2})}{2} \sqrt{2 - \frac{1}{25}} - 1 \right) \gamma \\ &= \left(\frac{7(1 + \sqrt{2})}{10} - 1 \right) \gamma > 0, \end{aligned} \tag{39}$$

where the lower bound on $\sqrt{2-\gamma^2} > \sqrt{2-25^{-1}}$ also follows from (21).

In order to obtain an absolute upper bound on the entire series A in (37), and thus on $x_2 - \tilde{x}_r = RA$, begin by writing A in the form

$$A = A_1 + A_2 - B,$$

where

$$A_2 = \left(\gamma - \frac{\theta}{4} \sqrt{2-\gamma^2} \right) \frac{\theta^3}{3!},$$

and B is an alternating series of the form

$$B = b_1 - b_2 + b_3 - \dots ,$$

with the j -th term b_j of this series given by

$$b_j = \left(\gamma - \frac{\theta}{2j+4} \sqrt{2-\gamma^2} \right) \frac{\theta^{(2j+3)}}{(2j+3)!} \quad \text{for } j = 1, 2, 3, \dots \tag{40}$$

Using the same techniques one uses to derive the upper and lower bounds on A_1 in (38) and (39), respectively, one can derive the following upper and lower bounds for A_2 ,

$$\begin{aligned} (1 - \sqrt{2}) \frac{(1 + \sqrt{2})^3}{6} \gamma^4 &< A_2 = \left(\gamma - \frac{\theta}{4} \sqrt{2-\gamma^2} \right) \frac{\theta^3}{3!} \\ &< \left(1 - \frac{7(1 + \sqrt{2})}{20} \right) \frac{32}{3} \gamma^4. \end{aligned} \tag{41}$$

It is apparent from the bounds in (41) that A_2 may be either positive or negative, depending on the values of h and κ_{\max} .

Now note that each of the terms b_j defined in (40) of the series B are positive. To see this, first note that, since θ is positive by (30), the sign of b_j depends only on the terms in parentheses immediately to the right of the equal sign in (40). For

example,

$$b_1 = \left(\gamma - \frac{\theta}{6} \sqrt{2 - \gamma^2} \right) \frac{\theta^5}{5!} > 0,$$

since $0 < \gamma \leq 1/5$, $0 < \theta < 4\gamma$ and $\sqrt{2 - \gamma^2} < \sqrt{2}$, and hence,

$$\left(\gamma - \frac{\theta}{6} \sqrt{2 - \gamma^2} \right) > \left(1 - \frac{2}{3} \sqrt{2} \right) \gamma > 0. \quad (42)$$

Similarly, all of the subsequent terms in this series are also positive, since (42) implies the terms in parentheses immediately to the right of the equal sign in the definition of b_j for $j > 1$ in (40) must also be positive,

$$\left(\gamma - \frac{\theta}{2j+4} \sqrt{2 - \gamma^2} \right) > \left(\gamma - \frac{\theta}{6} \sqrt{2 - \gamma^2} \right) > 0 \quad \text{for all } j = 2, 3, 4, \dots$$

Furthermore, it is also the case that $b_j > b_{j+1}$ for all $j = 1, 2, 3, \dots$, since the terms b_j defined in (40) are (strictly) monotonically decreasing when viewed as a function of j . (To see this recall $\gamma \leq 1/5$ from (21) and hence, (30) implies $\theta < 4\gamma \leq 4/5$.) Finally, since $b_j > 0$ and $b_j > b_{j+1}$ for all $j = 1, 2, 3, \dots$, it follows that the entire series B is positive,

$$B = (b_1 - b_2) + (b_3 - b_4) + (b_5 - b_6) + \dots > 0.$$

This leads to an absolute upper bound on the series A in (37),

$$A = A_1 + A_2 - B < A_1 + A_2 < (2\sqrt{2} - 1)4\gamma^2 + \left(1 - \frac{7(1 + \sqrt{2})}{20} \right) \frac{32}{3} \gamma^4. \quad (43)$$

Finally, since $\gamma = \sqrt{h\kappa_{\max}}/5\sqrt{C_h}$, $R\gamma = h$ and, by (21), $\gamma^2 \leq 1/25$, the upper bound on $x_2 - \tilde{x}_r$ in (31) now follows from (34) and (43),

$$\begin{aligned} x_2 - \tilde{x}_r &= RA \leq R \left\{ (2\sqrt{2} - 1)4\gamma^2 + \left(1 - \frac{7(1 + \sqrt{2})}{20} \right) \frac{32}{3} \gamma^4 \right\} \\ &\leq \frac{\sqrt{33}}{5\sqrt{2}} \left\{ 8\sqrt{2} - 4 + \frac{8}{3 \cdot 25} \left(\frac{13}{5} - \frac{7\sqrt{2}}{5} \right) \right\} \sqrt{\kappa_{\max}} h^{3/2} \\ &= \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}. \quad \square \end{aligned}$$

4.2. The comparison circle theorem. Suppose the interface $(x, g(x))$ satisfies $h \leq C_h \kappa_{\max}^{-1}$ and $x_r < x_2$ where (x_r, y_3) is the point at which g exits the 3×3 block of cells B_{ij} . The following theorem states that once $g(x) < \tilde{c}(x)$ for some $x \in (x_0, x_2)$, then $g(x)$ must remain below $\tilde{c}(x)$ for all $x \in (\tilde{x}_0, x_2)$, where $(\tilde{x}_0, \tilde{y}_0)$ denotes the point where g initially crosses below \tilde{c} as illustrated in Figures 5 and 7. An immediate consequence of this theorem is $\tilde{x}_r < x_r$. Consequently, if

$x_r < x_2$, then $\tilde{x}_r < x_r < x_2$ and hence, $x_2 - x_r < x_2 - \tilde{x}_r$. Since by [Theorem 1](#), $x_2 - \tilde{x}_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$, it follows that $x_2 - x_r < \tilde{C} \sqrt{\kappa_{\max}} h^{3/2}$. This, together with the bound on $|g'(x)|$ in [\(A.5a\)](#), is sufficient to ensure the error in the center column sum S_i associated with g is $O(h)$.

Theorem 2 (the comparison circle theorem). *Let R as defined in [\(20b\)](#) be the radius of the comparison circle [\(22\)](#) and let $g \in C^2[x_0, x_2]$ be a strictly monotonic function that satisfies [\(5\)](#), where the constant C_h is defined in [\(6\)](#). Furthermore, assume the interface g enters the 3×3 block of cells B_{ij} across its left edge at the point (x_0, y_1) with $y_1 < y_l < y_2$, passes through the center cell C_{ij} , and exits B_{ij} through the top of its center column at the point (x_r, y_3) with $x_1 < x_r < x_2$. Let $(\tilde{x}_0, \tilde{y}_0)$ denote the first point at which the graph of g crosses the graph of \tilde{c} as shown, for example, in [Figures 5 and 7](#). Then*

$$g(x) < \tilde{c}(x) \quad \text{for all } x \in (\tilde{x}_0, x_r]. \tag{44}$$

Proof. First note that, since the interface g satisfies [\(5\)](#) where C_h is defined by [\(6\)](#), this ensures the maximum curvature κ_{\max} of g is bounded above by the curvature $\kappa^{\tilde{c}}$ of the comparison circle,

$$\kappa_{\max} < R^{-1} = \kappa^{\tilde{c}}.$$

The argument is as follows. Since the interface satisfies [\(5\)](#)–[\(6\)](#), it follows that $\kappa_{\max} \leq C_h h^{-1}$ and hence,

$$\sqrt{\kappa_{\max}} \leq \sqrt{\frac{C_h}{h}}. \tag{45}$$

Multiplying both sides of [\(45\)](#) by $\sqrt{\kappa_{\max}}$ yields

$$\kappa_{\max} \leq \sqrt{\frac{C_h \kappa_{\max}}{h}}. \tag{46}$$

Now, since $(5C_h)^{-1} = 3.3 > 1$, we can bound the right-hand side of [\(46\)](#) by

$$\kappa_{\max} \leq \sqrt{\frac{C_h \kappa_{\max}}{h}} < \frac{1}{5C_h} \sqrt{\frac{C_h \kappa_{\max}}{h}} \leq \frac{1}{5} \sqrt{\frac{\kappa_{\max}}{C_h h}} = R^{-1} = \kappa^{\tilde{c}}.$$

Thus, κ_{\max} , the maximum magnitude of the curvature of g , is bounded above by the curvature $\kappa^{\tilde{c}} = R^{-1}$ of the comparison circle and therefore,

$$\kappa^g(x) \leq \kappa_{\max} < \kappa^{\tilde{c}}(x) = R^{-1} \quad \text{for all } x \in [x_0, x_2]. \tag{47}$$

The inequality in [\(44\)](#) is proven by contradiction. One begins by assuming

$$g(\xi) = \tilde{c}(\xi) \quad \text{for some } \xi \in (\tilde{x}_0, x_r], \tag{48}$$

and then showing that this implies the maximum curvature κ_{\max} of g in (\tilde{x}_0, x_r) must exceed $\kappa^{\tilde{c}}$, thereby contradicting (47). The argument is as follows. Let ξ denote the first point in $(\tilde{x}_0, x_r]$ that satisfies (48). Since $g(x) > \tilde{c}(x)$ for $x_0 < x < \tilde{x}_0$ and $g(x) < \tilde{c}(x)$ for $\tilde{x}_0 < x < \xi$, it follows that

$$g'(\tilde{x}_0) < \tilde{c}'(\tilde{x}_0), \quad (49)$$

However, since, by assumption, $g(\xi) = \tilde{c}(\xi)$ for some $\xi > \tilde{x}_0$ (i.e., (48) holds), it must be the case that eventually $g'(x) \geq \tilde{c}'(x)$. Let $x^* \in (\tilde{x}_0, \xi)$ be the first x such that $g'(x^*) = \tilde{c}'(x^*)$ so that

$$g'(x^*) = g'(\tilde{x}_0) + \int_{\tilde{x}_0}^{x^*} g''(x) dx = \tilde{c}'(\tilde{x}_0) + \int_{\tilde{x}_0}^{x^*} \tilde{c}''(x) dx = \tilde{c}'(x^*).$$

By virtue of (49) this can only be true if $g''(x) > \tilde{c}''(x)$ on some subinterval of (\tilde{x}_0, x^*) . In particular,

$$g''(\eta) > \tilde{c}''(\eta), \quad (50)$$

for some $\eta \in (\tilde{x}_0, x^*)$.

Now recall the following three facts.

- (1) By assumption g is strictly monotonic and hence, $0 < g'(x)$ for all $x \in (x_0, \tilde{x}_r]$.
- (2) For all $x \in (\tilde{x}_0, x^*)$, $0 < g'(x) < \tilde{c}'(x)$.
- (3) For all $x \in [x_0, x_2]$, $\kappa^g(x) = g''(x)(\sqrt{1 + g'(x)^2})^{-3}$ (e.g., see [29]).

Equation (50) together with items (1)-(3) above imply

$$\kappa^g(\eta) = \frac{g''(\eta)}{(\sqrt{1 + g'(\eta)^2})^3} > \frac{\tilde{c}''(\eta)}{(\sqrt{1 + \tilde{c}'(\eta)^2})^3} = \kappa^{\tilde{c}}(\eta),$$

which contradicts (47). Therefore, g must be bounded above by the comparison circle as claimed. \square

Corollary 2 ($x_2 - x_r < \tilde{C} \sqrt{\kappa_{\max} h^{3/2}}$). *Let $g \in C^2[x_0, x_3]$ be a function that satisfies the assumptions stated in Theorem 2. Then*

$$x_2 - x_r < \tilde{C} \sqrt{\kappa_{\max} h^{3/2}}, \quad (51)$$

where \tilde{C} is defined in (32).

Proof. By the Comparison Circle Theorem (Theorem 2) there exists a point $\tilde{x}_0 \in (x_0, x_r)$ such that

$$g(x) < \tilde{c}(x) \quad \text{for all } x \in (\tilde{x}_0, x_r).$$

This implies $\tilde{x}_r < x_r$, and hence that $x_2 - x_r < x_2 - \tilde{x}_r$. Equation (51) follows immediately from (31) in Theorem 1. \square

4.3. The column sum S_i is exact to $O(h)$.

Theorem 3 (the column sum S_i is exact to $O(h)$). *Assume the interface $g \in C^2[x_0, x_3]$ and g is a strictly monotonically increasing function that satisfies the constraint in (5) with the constant C_h defined in (6). Furthermore, assume g enters the 3×3 block of cells $B_{ij} = B_{11} = [x_0, x_3] \times [y_0, y_3]$ across its left edge at the point (x_0, y_l) with $y_1 < y_l < y_2$, passes through the center cell $C_{ij} = C_{11} = [x_1, x_2] \times [y_1, y_2]$, and exits B_{ij} through the top of its center column at the point (x_r, y_3) with $x_1 < x_r < x_2$ as shown, for example, in Figure 7. Then the error between the normalized integral of g over the center column and the column sum S_i is $O(h)$:*

$$\left| \frac{1}{h^2} \int_{x_1}^{x_2} (g(x) - y_0) dx - S_i \right| < \bar{C} \kappa_{\max} h, \quad (52)$$

where

$$\bar{C} \stackrel{\text{def}}{=} \tilde{C}^2, \quad (53)$$

and \tilde{C} is defined in (32).

Proof. Since, by assumption,

$$\min_{[x_0, x_r]} g(x) = y_l > y_1 > y_0,$$

and the interface is a strictly monotonically increasing function of x on $[x_0, x_2]$, it follows that

$$S_i = h^{-2} \int_{x_1}^{x_2} (\min\{g(x), y_3\} - y_0) dx.$$

The error between the normalized volume (i.e., area) under the interface $y = g(x)$ in the center column and the center column sum S_i is therefore

$$h^{-2} \int_{x_1}^{x_2} (g(x) - y_0) dx - S_i = h^{-2} \int_{x_r}^{x_2} (g(x) - y_3) dx. \quad (54)$$

An example is shown in Figure 7. Thus, it suffices to show

$$\left| \int_{x_r}^{x_2} (g(x) - y_3) dx \right| \leq \bar{C} \kappa_{\max} h^3. \quad (55)$$

By (A.5a) $|g'(x)| < 2$, which implies

$$\left| \int_{x_r}^{x_2} (g(x) - y_3) dx \right| \leq \left| \int_{x_r}^{x_2} L(x) dx \right|, \quad (56)$$

where $L(x)$ is the line with slope 2 that passes through the point x_r . The region of integration on the right hand side of (56) is a right triangle with corners (x_r, y_3) ,

(x_2, y_3) , and $(x_2, y_3 + 2(x_2 - x_r))$, and hence the integral on the right-hand side of (56) is the area of this triangle, namely $(x_2 - x_r)^2$. Thus,

$$\left| \int_{x_r}^{x_2} (g(x) - y_3) dx \right| \leq \left| \int_{x_r}^{x_2} L(x) dx \right| \leq (x_2 - x_r)^2 < \tilde{C}^2 \kappa_{\max} h^3 = \bar{C} \kappa_{\max} h^3, \quad (57)$$

where the bound $(x_2 - x_r)^2 < \tilde{C}^2 \kappa_{\max} h^3$ between the third and fourth terms in (57) follows from Equation (51) in Corollary 2. Equation (52), and hence the theorem, now follows immediately from (54) and (57). \square

5. Second-order accuracy in the max norm

All of the results in Section 4 (“Second-order accuracy in the max norm”) of [23] now hold provided the interface is a C^2 simple closed curve, the constraint in (5)–(6) is satisfied, the constant C_S in the statement of Theorem 23 in [23] is replaced by the constant \bar{C} defined in (53), and the term $(50\kappa_{\max}/3 + C_S)$ that appears in the statement of Theorem 24 of [23] is replaced by $C_m \kappa_{\max}$, where the constant C_m is defined in (59) below.

The key theorem that has changed between the two papers is Theorem 15 of [23]. Theorem 3 above is a stronger version of this theorem. Theorem 3 ensures that in cases such as the one shown in Figure 3, if the interface is C^2 and the constraint in (5)–(6) is satisfied, then in some orientation of the 3×3 block of cells B_{ij} centered on the cell C_{ij} in which one wishes to reconstruct the interface, there is a parametrization of the interface of the form $y = g(x)$ or $x = G(y)$, such that the center column sum S_i in the new orientation of the 3×3 block is exact to $O(h)$. This result provides the basis for the main result of this article, namely Theorem 4 below, which is a stronger version of Theorem 24, the main result of [23]. As has been the case throughout this article, in the statement of Theorem 4 below the interface $\tilde{z}(s)$ is written in the form $y = g(x)$ with material 1 lying below the graph of g , with the understanding the theorem also holds in those cases in which one must instead express the interface in the form $x = G(y)$ with material 1 lying below the graph of G .

Theorem 4. *Assume the interface $g \in C^2[x_{i-1}, x_{i+2}]$ and the grid size h and the maximum magnitude of the curvature κ_{\max} of the interface in the 3×3 block of cells B_{ij} centered on the cell C_{ij} in which one wishes to reconstruct the interface satisfy*

$$h \leq C_h \kappa_{\max}^{-1} = \frac{2}{33} \kappa_{\max}^{-1}. \quad (5)$$

Then there exists $\alpha, \beta = 1, 0, -1$ with $\alpha \neq \beta$ such that the column sums $S_{i+\alpha}$ and $S_{i+\beta}$ in B_{ij} are either exact or exact to $O(h)$. Furthermore, let

$$\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$$

be a piecewise linear approximation to $g(x)$ for $x \in [x_i, x_{i+1}]$ such that $g(x)$ and $\tilde{g}_{ij}(x)$ have the same volume fraction in the center cell

$$\Lambda_{ij}(g) = \Lambda_{ij}(\tilde{g}) \quad \text{and} \quad m_{ij} = \frac{(S_{i+\alpha} - S_{i+\beta})}{(\alpha - \beta)}.$$

Then $\tilde{g}_{ij}(x)$ is a pointwise, second-order accurate approximation to $g(x)$ in the interval $[x_i, x_{i+1}]$,

$$|g(x) - \tilde{g}_{ij}(x)| \leq \frac{25}{12}\kappa_{\max}h^2 + \bar{C}\kappa_{\max}h^2 = C_m\kappa_{\max}h^2 \quad \text{for all } x \in [x_i, x_{i+1}], \quad (58)$$

where

$$C_m \stackrel{\text{def}}{=} \left\{ \frac{25}{12} + \bar{C} \right\}, \quad (59)$$

and the constant \bar{C} is defined in (53).

Proof. The proof of this theorem is identical to the proof of Theorem 24 in [23] after one replaces the constant C_S defined in equation (89) of [23] with $\bar{C}\kappa_{\max}$, where \bar{C} is defined in (53) above. \square

6. Conclusions

This article contains a proof of the following result. Suppose one is given a square grid with cells of side h covering a closed and bounded rectangle $\Omega \subset \mathbb{R}^2$ and a C^2 simple closed curve $z(s)$ in Ω . If

$$h \leq C_h(\kappa_{\max})^{-1} = \frac{2}{33}(\kappa_{\max})^{-1}, \quad (5)$$

where κ_{\max} is the maximum magnitude of the curvature $\kappa(s)$ of the interface z in Ω . Then in every cell $C_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$ that contains a portion of the interface there exists a piecewise linear function $\tilde{g}_{ij}(x) = m_{ij}x + b_{ij}$ that is a second-order accurate approximation to the portion of the interface $y = g(x)$ that lies in C_{ij} ,

$$|g(x) - \tilde{g}_{ij}(x)| \leq C_m\kappa_{\max}h^2 \quad \text{for all } x \in [x_i, x_{i+1}],$$

where C_m is a constant, defined in (59) above, which is independent of h and κ_{\max} . For convenience, the interface $z(s)$ has been written here as a function $y = g(x)$ of the independent variable x with it being understood that in some cells it may be necessary to express the interface as a function $x = G(y)$ of the independent variable y . Theorem A.1 in the Appendix ensures if h satisfies the constraint in (5)–(6), then the interface can be written as a single-valued function of at least one of the coordinate variables x or y in every 3×3 block of cells centered on every cell C_{ij} that contains a portion of the interface.

In an earlier paper [23] the author proved a similar result, but with a constraint on the cell size h that was more restrictive than the one in (5)–(6). In order to obtain the less restrictive constraint on h in (5)–(6) Sections 3.2–3.4 of [23] required extensive modification. These modifications constitute Section 4 of the present paper.

The algorithm described in [24] is an example of a volume-of-fluid interface reconstruction algorithm that satisfies conditions (I)–(V) on page 125 of this article and hence, by Theorem 4 above, produces a pointwise second-order accurate approximation to the interface $\mathbf{z}(s)$.

Future work in this area should include the analysis of fingers and other regions of large curvature in both stationary and moving interfaces in an effort to determine conditions such as (5)–(6) that will ensure all filaments and similar regions are accurately resolved on grids that satisfy these conditions. Future work should also include proving the volume-of-fluid interface reconstruction algorithm coupled to a volume-of-fluid advection algorithm produces a second-order accurate approximation to the solutions of the advection equation.

Appendix: Considerations that affect the value of C_h

Definition. Let $a > 2$ be a real-valued parameter and define $\bar{C}_h[a]$ by

$$\bar{C}_h[a] \stackrel{\text{def}}{=} \frac{\sqrt{a} - \sqrt{2}}{4\sqrt{2}\sqrt{a-1}}. \quad (\text{A.1})$$

The following theorem is a generalization of Theorem 6 of [23].

Theorem A.1. For $s_L \leq s \leq s_R$ with $\mathbf{z}(s_L) = \mathbf{z}(s_R)$ let s be arclength along the two times continuously differentiable simple closed curve $\mathbf{z}(s)$ in Ω . Given some $s_0 \in [s_L, s_R]$ such that

$$\dot{y}^2(s_0) \leq \frac{1}{2} \leq \dot{x}^2(s_0), \quad (\text{A.2})$$

suppose one wants to reconstruct the interface in a neighborhood of the point $\mathbf{z}(s_0) = (x(s_0), y(s_0))$. Let $a > 2$ be a real-valued parameter as in (A.1) above and let $s_l \geq s_L$ be the greatest number less than s_0 and $s_r \leq s_R$ be the smallest number greater than s_0 such that

$$\dot{x}^2(s_l) = \frac{1}{a} = \dot{x}^2(s_r), \quad (\text{A.3})$$

so $a^{-1} \leq \dot{x}^2(s) \leq 1$ for all $s \in [s_l, s_r]$. Let $x_0 = x(s_0)$ and let

$$h_{\max} \stackrel{\text{def}}{=} \bar{C}_h[a] \kappa_{\max}^{-1} \quad (\text{A.4})$$

where $\bar{C}_h[a]$ is defined in (A.1) above. Then one can represent the interface as a single-valued function $y = g(x)$ of x on the interval

$$[x_l, x_r] = [x_0 - 2h_{\max}, x_0 + 2h_{\max}].$$

In addition, for all $x \in [x_l, x_r]$,

$$\max_{x \in [x_l, x_r]} |g'(x)| \leq \sqrt{a-1}, \tag{A.5a}$$

$$\max_{x \in [x_l, x_r]} |g''(x)| \leq (\sqrt{a})^3 \kappa_{\max}. \tag{A.5b}$$

Furthermore, if the roles of \dot{x} and \dot{y} in (A.2) and (A.3) are reversed, then one can represent the interface as a single-valued function $x = G(y)$ of y on the interval $[y_l, y_r] = [y_0 - 2h_{\max}, y_0 + 2h_{\max}]$ and the bounds in (A.5) hold on the interval $[y_l, y_r]$ with the function $g(x)$ replaced by $G(y)$.

Proof. Let $a > 2$ be the parameter in the definition of $\bar{C}_h[a]$ in (A.1) above. The proof of this theorem is identical to the proof of Lemmas 3–5 and Theorem 6 in [23] after one replaces the constants $1/4$ and $3/4$ in equation (23) in Lemma 3 of [23] with $1/a$ and $(a-1)/a$, respectively, and makes similar substitutions in Lemmas 4–5 and Theorem 6 of the same. \square

Remarks. (1) Theorem 6 of [23] is the special case of Theorem A.1 with $a = 4$.

(2) If necessary, one can periodically extend the interval $[x_L, x_R] \stackrel{\text{def}}{=} [x(s_L), x(s_R)]$ to the interval $[x_L - D, x_R + D]$, where $D = x_R - x_L$, with

$$y(s \pm D) = y(s) \quad \text{for all } s \in [s_L, s_R],$$

in order to ensure one can find s_l and s_r with $s_L - D \leq s_l \leq s_0$ and $s_0 \leq s \leq s_R + D$ such that (A.3) holds.

(3) In the statement and proof of Lemmas 3–5 and Theorem 6 of [23] the value of a is $a = 4$, which yields a value for C_h , which is denoted \tilde{C}_h in this article in order to avoid confusion, of

$$\tilde{C}_h \stackrel{\text{def}}{=} \bar{C}_h[4] = \frac{\sqrt{4} - \sqrt{2}}{4\sqrt{2}\sqrt{4-1}} = \frac{\sqrt{2} - 1}{4\sqrt{3}}. \tag{A.6}$$

(4) The conclusions of Theorem A.1 remain valid if the assumption the interface $z(s)$ is a simple closed curve is replaced by the assumption $z(s_L)$ and $z(s_R)$ lie on the boundary $\partial\Omega$ of the computational domain Ω , subject to the assumptions stated in the second paragraph of (5) on page 130. In addition, one must modify the proof of Lemma 5 in [23], since in this case there may not be a point s_l such that $\dot{x}^2(s_l) = 1/a$ or s_r such that $\dot{x}^2(s_r) = 1/a$; i.e., (A.3), which is the analog of equation (37) in [23], may not hold. See the comments concerning Lemma 4 in item (2) on pages 109–110 of [23] for the reason, if $z(s_L)$ and $z(s_R)$ lie on $\partial\Omega$, then this does not change the conclusions of Theorem A.1.

(5) If one chooses $a = 4.053301$, then the constant $\bar{C}_h[a]$ in (A.1) becomes

$$C_h \stackrel{\text{def}}{=} \bar{C}_h[a] = \frac{\sqrt{a} - \sqrt{2}}{4\sqrt{2}\sqrt{a-1}} = \frac{2}{33},$$

and the bound on the first derivative of the interface in (A.5a) becomes

$$\max_{x \in [x_l, x_r]} |g'(x)| \leq \sqrt{a-1} \approx \sqrt{3.053301} < 2. \quad (\text{A.7})$$

The value of $\sqrt{a-1}$ in (A.7) is a deliberate overestimate, the purpose of which is to simplify the bound on the expression in (56) that appears on the right-hand side of (57), and subsequent expressions that depend on the bound in (52).

(6) [Theorem A.1](#) ensures h is small enough that the interface can always be written as a single-valued function of one of the independent variables x or y in any 3×3 block centered on a cell containing a portion of the interface. This places an upper bound on C_h through (A.3) and (A.4). In addition, C_h is constrained both from above and below by the need to show inequalities of the form

$$\kappa_{\max} \geq \frac{g''(x)}{(\sqrt{a})^3} > \frac{C_h}{h}, \quad (\text{A.8})$$

hold in Equations (61), (69), and (78) in the proofs of Lemmas 11–13 of [23], respectively, where (A.5b) has been used to bound κ_{\max} from below by $g''/(\sqrt{a})^3$. Since in each of Equations (61), (69), and (78) of [23] the bound on g'' is of the form

$$g''(x) > \frac{\tilde{M}}{h} \quad \text{for all } x \in [x_{i-1}, x_{i+2}], \quad (\text{A.9})$$

equations (A.1), (A.8), and (A.9) lead to the requirement that $a > 2$ must satisfy the following inequality,

$$4\sqrt{2}\tilde{M}\sqrt{a-1} \geq (\sqrt{a})^4 - \sqrt{2}(\sqrt{a})^3. \quad (\text{A.10})$$

A careful study of this inequality will reveal the range of permissible values for a and hence, for $C_h = \bar{C}_h[a]$, is quite narrow.

References

- [1] I. Aleinov and E. G. Puckett, *Computing surface tension with high-order kernels*, Proceedings of the 6th International Symposium on Computational Fluid Dynamics (K. Oshima, ed.), 1995, pp. 6–13.
- [2] E. Aulisa, S. Manservigi, R. Scardovelli, and S. Zaleski, *A geometrical area-preserving volume-of-fluid advection method*, *J. Comput. Phys.* **192** (2003), no. 1, 355–364.
- [3] J. U. Brackbill, D. B. Kothe, and C. Zemach, *A continuum method for modeling surface tension*, *J. Comput. Phys.* **100** (1992), no. 2, 335–354. [MR 93c:76008](#)

- [4] A. J. Chorin, *Flame advection and propagation algorithms*, J. Comput. Phys. **35** (1980), no. 1, 1–11. MR 81d:76061
- [5] ———, *Curvature and solidification*, J. Comput. Phys. **57** (1985), no. 3, 472–490. MR 86d:80001
- [6] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski, *Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flow*, J. Comput. Phys. **152** (1999), no. 2, 423–456.
- [7] L. F. Henderson, P. Colella, and E. G. Puckett, *On the refraction of shock waves at a slow-fast gas interface*, J. Fluid Mech. **224** (1991), 1–27.
- [8] C. W. Hirt and B. D. Nichols, *Volume of fluid (VOF) method for the dynamics of free boundaries*, J. Comput. Phys. **39** (1981), 201–225.
- [9] R. M. Hurst, *Numerical approximations to the curvature and normal of a smooth interface using high-order kernels*, MS Thesis, University of California, Davis, 1995.
- [10] D. B. Kothe, J. R. Baumgardner, S. T. Bennion, J. H. Cerutti, B. J. Daly, K. S. Holian, E. M. Kober, S. J. Mosso, J. W. Painter, R. D. Smith, and M. D. Torrey, *PAGOSA: a massively-parallel, multi-material hydro-dynamics model for three-dimensional high-speed flow and high-rate deformation*, technical report LA-UR-92-4306, Los Alamos National Laboratory, 1992.
- [11] P. Lax and B. Wendroff, *Systems of conservation laws*, Comm. Pure Appl. Math. **13** (1960), 217–237. MR 22 #11523
- [12] R. J. LeVeque, *Numerical methods for conservation laws*, Lectures in Mathematics ETH Zürich, no. 72, Birkhäuser, Basel, 1990. MR 91j:65142
- [13] ———, *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics, no. 31, Cambridge University Press, 2002. MR 2003h:65001
- [14] G. H. Miller and P. Colella, *A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing*, J. Comput. Phys. **183** (2002), no. 1, 26–82. MR 2003j:76080
- [15] G. H. Miller and E. G. Puckett, *Edge effects in molybdenum-encapsulated molten silicate shock wave targets*, J. Appl. Phys. **75** (1994), no. 3, 1426–1434.
- [16] ———, *A high-order Godunov method for multiple condensed phases*, J. Comput. Phys. **128** (1996), no. 1, 134–164.
- [17] B. D. Nichols, C. W. Hirt, and R. S. Hotchkiss, *SOLA-VOF: a solution algorithm for transient fluid flow with multiple free boundaries*, technical report LA-8355, Los Alamos National Laboratory, 1980.
- [18] W. F. Noh and P. R. Woodward, *SLIC (simple line interface calculation)*, technical report UCRL-77651, Lawrence Livermore National Laboratory, 1976.
- [19] ———, *SLIC (simple line interface calculation)*, Lecture Notes in Physics (A. I. van der Vooren and P. J. Zandbergen, eds.), no. 59, Springer, New York, 1976, pp. 330–340.
- [20] B. J. Parker and D. L. Youngs, *Two and three dimensional eulerian simulation of fluid flow with material interfaces*, technical report 01/92, UK Atomic Weapons Establishment, Aldermaston, Berkshire UK, 1992.
- [21] J. E. Pilliod, Jr. and E. G. Puckett, *Second-order accurate volume-of-fluid algorithms for tracking material interfaces*, J. Comput. Phys. **199** (2004), no. 2, 465–502. MR 2005d:65145
- [22] S. Popinet and S. Zaleski, *A front-tracking algorithm for the accurate representation of surface tension*, Int. J. Numer. Methods Fluids **30** (1999), no. 6, 775–793.
- [23] E. G. Puckett, *On the second-order accuracy of volume-of-fluid interface reconstruction algorithms: convergence in the max norm*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 1, 99–148. MR 2011c:76001

- [24] ———, *A volume-of-fluid interface reconstruction algorithm that is second-order accurate in the max norm*, Commun. Appl. Math. Comput. Sci. **5** (2010), no. 2, 199–220. MR 2012c:65130
- [25] E. G. Puckett and G. H. Miller, *The numerical computation of jetting impacts*, Proceedings of the 20th International Symposium on Shock Waves (New Jersey) (B. Sturtevant, J. E. Shepherd, and H. Hornung, eds.), vol. II, World Scientific, 1996, pp. 1467–1472.
- [26] Y. Renardy and M. Renardy, *PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method*, J. Comput. Phys. **183** (2002), no. 2, 400–421. MR 2003k:76102
- [27] W. J. Rider and D. B. Kothe, *Reconstructing volume tracking*, J. Comput. Phys. **141** (1998), no. 2, 112–152. MR 99a:65200
- [28] R. Scardovelli and S. Zaleski, *Direct numerical simulation of free-surface and interfacial flow*, Annual Review of Fluid Mechanics, no. 31, Annual Reviews, Palo Alto, CA, 1999, pp. 567–603. MR 99n:76002
- [29] S. K. Stein and A. Barcellos, *Calculus and analytic geometry*, 5th ed., McGraw-Hill, 1992.
- [30] M. D. Torrey, L. D. Cloutman, R. C. Mjolsness, and C. W. Hirt, *NASA-VOF2D: a computer program for incompressible flows with free surfaces*, technical report LA-10612-MS, Los Alamos National Laboratory, 1985.
- [31] M. D. Torrey, R. C. Mjolsness, and L. R. Stein, *NASA-VOF3D: a three-dimensional computer program for incompressible flows with free surfaces*, technical report LA-11009-MS, Los Alamos National Laboratory, 1987.
- [32] M. W. Williams, D. B. Kothe, and E. G. Puckett, *Accuracy and convergence of continuum surface-tension models*, Fluid dynamics at interfaces (W. Shyy and R. Narayanan, eds.), Cambridge University Press, 1999, pp. 294–305.
- [33] D. L. Youngs, *Time-dependent multi-material flow with large fluid distortion*, Numerical methods for fluid dynamics (K. W. Morton and M. J. Baines, eds.), Institute of Mathematics and Its Applications, Academic Press, 1982, pp. 273–285.
- [34] ———, *Numerical simulation of turbulent mixing by Rayleigh–Taylor instability*, Proceedings of the Third Annual International Conference on Fronts, Interfaces, and Patterns (A. R. Bishop, L. J. Campbell, and P. J. Channell, eds.), North-Holland, 1983, Reprinted from Physica D, **12D** (1984), nos. 1–3, pp. 32–44.
- [35] ———, *An interface tracking method for a 3D Eulerian hydrodynamics code*, technical report AWRE/44/92/35, UK Atomic Weapons Research Establishment, 1987.

Received September 7, 2010. Revised November 13, 2012.

ELBRIDGE GERRY PUCKETT: egpuckett@ucdavis.edu

Department of Mathematics, University of California, Davis, One Shields Avenue, Davis, CA 95616, United States

COMPUTATIONAL MODELS OF MATERIAL INTERFACES FOR THE STUDY OF EXTRACORPOREAL SHOCK WAVE THERAPY

KIRSTEN FAGNAN, RANDALL J. LEVEQUE AND THOMAS J. MATULA

Extracorporeal shock wave therapy (ESWT) is a noninvasive treatment for a variety of musculoskeletal ailments. A shock wave is generated in water and then focused using an acoustic lens or reflector so the energy of the wave is concentrated in a small treatment region where mechanical stimulation in principle enhances healing. In this work we have computationally investigated shock wave propagation in ESWT by solving a Lagrangian form of the isentropic Euler equations in the fluid and linear elasticity in the bone using high-resolution finite volume methods. We solve a full three-dimensional system of equations and use adaptive mesh refinement to concentrate grid cells near the propagating shock. We can model complex bone geometries, the reflection and mode conversion at interfaces, and the propagation of the resulting shear stresses generated within the bone. We discuss the validity of our simplified model and present results validating this approach.

1. Introduction

Extracorporeal shock wave therapy (ESWT) is a noninvasive treatment for musculoskeletal conditions such as bone fractures that fail to heal (nonunions), necrotic wounds, and strained tendons [55; 40]. In this treatment a shock wave is generated in water and then focused using an acoustic lens or reflector so that the energy of the wave is concentrated in a small treatment region. This technique has been used since the 1980's, more widely in Europe and Asia than in the US, where it is still considered experimental and has limited FDA approval.

Although the underlying biological mechanisms are not well understood [42], the mechanical compressional and/or shear stress caused by the propagating shock wave is thought to stimulate healing [42; 54; 26; 39; 53; 43; 44; 30; 13; 27; 11; 23]. In addition to stress, a number of other biological mechanisms potentially play a role in the body's response to ESWT. The focus of this study, however, is on mechanical stress deposition and computational tools for studying this phenomenon.

MSC2010: 92-08, 92C50, 65M08.

Keywords: high-resolution finite volume methods, computational biology, shock wave therapy.

Computational models for shock wave propagation and focusing can aid in the study of ESWT. In particular, there are many open questions concerning the interaction of shock waves with complex three-dimensional geometries such as bone embedded in tissue. In this paper we present a new method for studying ESWT that incorporates the fluid and solid materials in a set of coupled, nonlinear partial differential equations that are solved using high-resolution finite volume methods. In order to model the wave interaction with complex three-dimensional geometries, we employed adaptive mesh refinement to concentrate the finest grid around then propagating shock wave.

Because of the difference in material properties, a wave hitting the tissue/bone interface will be partially reflected, and the transmitted wave will have a modified strength and direction of propagation. This can greatly affect the location and size of the focal region and the peak pressure amplitude. Also, although the shock wave is primarily a pressure wave in soft tissue (which has a very small shear modulus), at a bone interface mode conversion takes place and shear waves as well as compressional waves are transmitted into the bone, generating a dynamically applied load.

The medical shock wave devices are similar to those used for extracorporeal shock wave lithotripsy (ESWL), a widely used nonsurgical treatment for kidney stones in which the focused shock waves have sufficient amplitude to pulverize the kidney stone. In shock wave therapy the amplitudes are generally smaller and the goal is mechanical stimulation rather than destruction, although in some applications such as the treatment of heterotopic ossifications (HO) (see [Section 5.4](#)) larger amplitudes may be used.

[Figure 1](#) shows the geometry of a laboratory shock wave device modeled on the

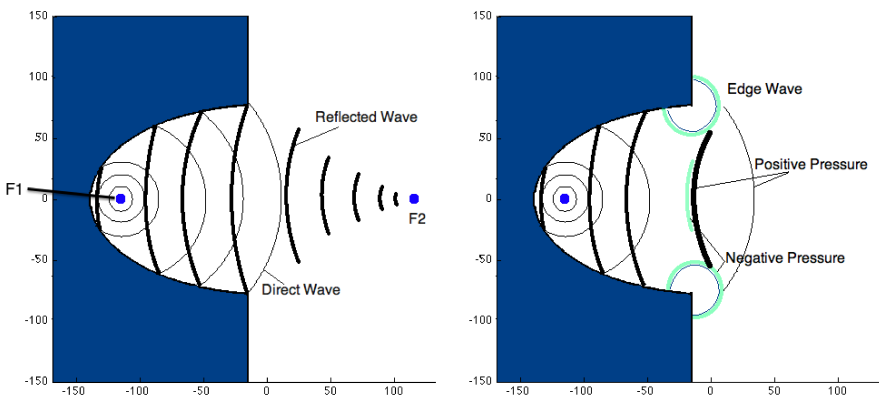


Figure 1. Cartoon of the Dornier HM3 Lithotripter. Left: the spherical wave is generated at F1, reflects off the ellipsoid and the reflected wave focuses at F2. Right: the creation of the edge waves at the corner of the ellipsoid and the contribution of negative pressure to the tail of the ESWT pressure wave.

clinical Dornier HM3 lithotripter. The three-dimensional axisymmetric geometry consists of an ellipsoidal reflector made out of metal and a cavity filled with water. A spark plug at the focus of the ellipse marked F1 generates a bubble which collapses and creates a spherical shock wave that reflects and focuses at F2. The major and minor axes of the ellipsoid in the HM3 are $a = 140$ mm and $b = 79.8$ mm, respectively. The foci of this ellipse are at $(\pm 115, 0, 0)$ and the reflector is truncated at 100 mm from F1, or $(-10, 0, 0)$.

In the laboratory, this reflector is immersed in a bath of water and objects can be placed at the second focus of the ellipsoid, F2. This device is in use at the Center for Industrial and Medical Ultrasound (CIMU) at the University of Washington Applied Physics Laboratory and we have used this geometry in order to compare directly with some laboratory experiments. Some preliminary comparisons were presented in [19].

Computationally, we use this geometry to calculate the initial condition by solving two-dimensional axisymmetric Euler equations with the Tammann equation of state (see Section 2). These initial conditions are then fed into a full three-dimensional calculation near the focus at F2.

In addition to the HM3, we have also used the geometry of the hand-held Sanywave device used in clinical studies by our collaborator Dr. Michael Chang. Some sample calculations related to the study of HOs are presented in Section 5.4.

In each case, the ESWT pressure wave form that is generated has a similar shape. There is a sharp increase in pressure from atmospheric pressure (~ 0.1 MPa) to a peak pressure ranging from 35 to 100 MPa over a very short rise time (~ 10 ns), followed by a decrease in pressure to ~ -10 MPa over ~ 5 μ s. The negative fluid pressure in the tail can lead to cavitation bubbles, as discussed below.

Bone healing is thought to be regulated in part by mechanical factors [39; 53; 45; 26; 23; 53]. Several studies have shown that the application of cyclic compressive and shear displacements can enhance healing through increased callus formation and ossification [39; 45; 46; 50; 43; 56]. The results also indicate that treatment is also dependent upon the rate, mode and magnitude of the stress deposition [39], as well as the gap size [14].

Carter et al. [11], as well as Claes and Heigele [13], proposed a model for skeletal tissue development based on hydrostatic pressure and tensile displacements [13]. Other research has proposed a different model for skeletal tissue formation based on shear strain and fluid flow [44; 30]. Augat et al. [2] found that tensile displacements are not effective in enhancing bone formation. This was further validated when Isaksson et al. [27] investigated the models in [13; 11; 44; 30] and found that shear strain and fluid flow, were more accurate predictors of bone growth. However, no single model was able to predict certain features of the bone formation and healing process [39], highlighting the need for further research in this area.

The shear waves generated at the fluid/solid interface have also been shown to be important in the effective break up of kidney stones [49; 20]. An additional effect of ESWT is the formation and collapse of cavitation bubbles that can cause tissue damage. While the shock wave is a compression wave, it is followed by a rarefaction wave of expansion, and in the tail the fluid pressure typically drops to negative values. Reflection at interfaces can lead to enhanced regions of expansion and to sufficiently negative pressures that cavitation bubbles can form [38; 51; 17].

To better understand all of these effects, it is desirable to have a three-dimensional computational model that can simulate the focusing of nonlinear shock waves and their interaction with arbitrarily complex interfaces between different materials.

In this paper we present an approach to this problem that has allowed the study of some of these issues in a simplified context. In particular, we consider an idealized situation in which soft tissue is replaced by water, ignoring its viscoelastic properties, and modeled by the nonlinear compressible Euler equations with the Tammann or Tait equation of state. This has been used for prior ESWT work in water as well as biological-like materials [28; 41]. Bone is modeled as an isotropic and homogeneous linear elastic material [21; 29].

In reality, soft tissue and bone are very complex multiscale materials with microstructures, inhomogeneities, and anisotropic properties. Any attempt to model the biological effect of shock wave propagation through such materials may require a more sophisticated and detailed model than used here. However, we believe that many of the macroscale shock propagation issues discussed above can be adequately and most efficiently studied with a simplified model of the form considered here, since the dominant effect we hope to capture is the reflection and transmission of waves at interfaces between materials.

The compressible Euler equations with the Tammann equation of state (see [Section 2.1](#)) in two-dimensional axisymmetric geometry is used to model the initial formation of the focusing shock wave. These initial conditions are then fed into a code that uses a simpler nonlinear model, the Tait equation of state, in a three-dimensional simulation of the fluid. The compressible fluid equations are written using a Lagrangian formulation that easily couples to the isotropic linear elasticity equations used in the bone-like material. The resulting equations have the same form everywhere, with a different stress-strain relationship in the different materials.

A high-resolution finite volume method is used to solve these equations. We use the wave-propagation algorithms described in [35] and implemented in Clawpack [15]. These are Godunov-type methods for the hyperbolic system that use solutions to the Riemann problem between adjacent grid cells to determine a set of waves used to update the solution, and second-order correction terms with slope limiters are added to resolve the nearly discontinuous shock waves with minimal smearing or nonphysical oscillation.

These methods are used on a purely rectangular Cartesian grid. Each grid cell has associated with it a set of material parameters determining the material in the cell, in a unified manner so that both fluid and solid can be modeled. Complex geometry is handled by using appropriate averaged values of these parameters in cells that are cut by the interface. This is described further in [Section 4.4](#). Averaging across the interface works quite well when the material properties are sufficiently similar and in [Section 4.4](#) we show that this is the case even for fluid/solid boundaries of the type we consider.

We also use patch-based adaptive mesh refinement (AMR) to concentrate grid cells in regions where they are most needed to resolve features of interest. The Clawpack software contains AMR software in both two and three space dimensions and this software has been used directly for the two-dimensional axisymmetric computations of the initial shock wave described in [Section 5](#). For the three-dimensional problem we have used ChomboClaw [\[10\]](#), an interface between Clawpack and the Chombo code [\[1\]](#) developed at the Lawrence Berkeley National Laboratory (LBL), which provides an implementation of AMR on parallel machines using MPI. Using ChomboClaw, the code originally developed using Clawpack was easily converted into a code that was run on an NSF TeraGrid machine at Texas Advanced Computing Center (TACC) and tested using up to 128 processors.

Extensive laboratory experiments have been performed on shock wave devices to measure the wave form of shock waves produced by various devices, the shape of the focal region, the peak amplitudes of pressure observed in these regions, and other related quantities. Most of these experiments have been done in a water tank where the shock wave propagates and focuses in a homogeneous medium where measurements are easily done, or with phantoms (acrylic objects with well understood photoelastic properties) that are placed in the water as a proxy for bones or kidney stones, with instrumentation such as pressure gauges or photographs used to explore the interaction of the shock wave with the object. In some cases high-speed photographs of the shock wave have been obtained. Creating phantoms from clear birefringent materials and using polarized light it is even possible to photograph the shock wave propagating through the object [\[48\]](#). We have used some of these experiments to help validate our numerical approach [\[19\]](#).

Other researchers have also developed computational models for shock wave therapy and lithotripsy. In prior work the pressure field has been modeled using linear and nonlinear acoustics as well as the Euler equations with the Tait equation of state. Hamilton [\[24\]](#) used linear geometrical acoustics, which holds under the assumption of weak shock strength, to calculate the reflection of the spherical wave. The diffraction of the wave at the corner of the reflector was calculated using the Kirchoff integral method. Christopher's model [\[12\]](#) of the HM3 lithotripter used Hamilton's result as a starting point and considered nonplanar sources. Coleman

et al. [17], Averkiou and Cleveland [3] used models based on the KZK equation. Tanguay [51] solved the full Euler equations and incorporated cavitation effects as well as the edge wave.

Our approach differs from these in that we consider the wave propagation in both the fluid and solid by solving a single set of equations that can model both materials. This approach allows us to investigate not only compression and tension effects of ESWT, but also the propagation of shear waves in the solid. Sapozhnikov and Cleveland [16] have investigated the effect of shear waves on spherical and cylindrical stones using linear elasticity with a plane wave initial condition. This initial condition is an unfocused wave, which yields good results for small objects, but would fail to capture the full ESWT pressure wave interaction with three-dimensional bone geometries.

2. Model equations

To accurately model shock wave formation and propagation it is generally necessary to use nonlinear equations of compressible flow. In this work we use nonlinear equations for compressible liquids in the fluid domain (water or soft tissue) and linear elasticity in the solid domain (bone). The nonlinear compressible equations are written in a Lagrangian framework in terms of a reference configuration, as is done for the linear elasticity equations. This allows both sets of equations to be written in the same form. We apply finite volume methods to this form of the equations so that a single computational grid (or set of nested grids with AMR) can be used over the entire domain. Interfaces between fluid and solid are represented by choosing averaged material parameters in each grid cell, as discussed further in Section 4.4.

The system of equations we solve has the general form of a hyperbolic system of 9 equations

$$q_t + f(q, x, y, z)_x + g(q, x, y, z)_y + h(q, x, y, z)_z = 0, \quad (1)$$

where the vector q consists of the 6 components of the symmetric strain tensor followed by the momenta, and the fluxes in general may be spatially varying based on material properties:

$$\begin{aligned} q &= [\epsilon^{11} \quad \epsilon^{22} \quad \epsilon^{33} \quad \epsilon^{12} \quad \epsilon^{23} \quad \epsilon^{13} \quad \rho u \quad \rho v \quad \rho w]^T, \\ f(q, x, y, z) &= [u \quad 0 \quad 0 \quad v/2 \quad 0 \quad w/2 \quad \sigma^{11} \quad \sigma^{12} \quad \sigma^{13}]^T, \\ g(q, x, y, z) &= [0 \quad v \quad 0 \quad u/2 \quad w/2 \quad 0 \quad \sigma^{12} \quad \sigma^{22} \quad \sigma^{23}]^T, \\ h(q, x, y, z) &= [0 \quad 0 \quad w \quad 0 \quad v/2 \quad u/2 \quad \sigma^{13} \quad \sigma^{23} \quad \sigma^{33}]^T. \end{aligned} \quad (2)$$

In these expressions, T denotes transposition, $\rho = \rho(x, y, z)$ is the density of the

material (the “background density” independent of the wave propagating through the material) and the stress tensor $\sigma = \sigma(q, x, y, z)$ is in general a spatially varying function of q , linear in the solid and nonlinear in the fluid.

Within the fluid domain $\sigma = -pI$, where p is the scalar pressure and I is the identity matrix. The pressure is a nonlinear function of the strain as discussed further below. In the solid domain, σ is a linear function of ϵ and is nondiagonal, allowing us to model the propagation of shear waves as well as compressional waves.

In [Section 2.1](#) below we present the compressible fluid equations in their standard Eulerian form (the Euler equations) and discuss two possible equations of state, the Tammann EOS and the simpler Tait EOS in which the pressure is a function of density (or strain) alone, allowing us to drop the energy equation from the Euler equations. Then in [Section 2.2](#) we rewrite these equations in the Lagrangian form given above. This can be done when modeling ESWT because the deformations are sufficiently small that the geometric nonlinearity of the equations can be ignored, adopting a Lagrangian frame and only considering the nonlinearity of the stress-strain relation as given by the equation of state.

In [Section 2.3](#) we discuss the linear elasticity model used to model bone.

2.1. Compressible fluids in Eulerian form. Much of the previous work on ESWT has been centered around the use of the Euler equations with the Tait or Tammann equations of state. These equations of state are typically used for modeling underwater explosions like the spark plug source of the lithotripter device [24; 28]. In this section we discuss the full Euler equations and proceed to show why the Tait equation of state is sufficient for modeling ESWT. Since this equation of state is a function only of the density, and can be rewritten as a function of strain, we show in [Section 2.2](#) how it can be modeled within the framework of elasticity, which enables us to model both the fluid and solid with the single system of equations given above.

In three space dimensions the Euler equations take the form

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E + p) \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho u^2 + p \\ \rho vv \\ v(E + p) \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} \rho v \\ \rho uv \\ \rho vv \\ \rho w^2 + p \\ w(E + p) \end{bmatrix} = 0. \quad (3)$$

The total energy is $E = \rho e + \frac{1}{2}(u^2 + v^2 + w^2)$.

Several of the problems we investigated are axially symmetric and this enabled us to reduce the three-dimensional equations to a two-dimensional form. If we first rewrite the equations in cylindrical coordinates (r, θ, z) and assume no variation and

zero velocity in the θ direction, the system we obtain is reduced to two variables, r and z . The equations are

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u_r \\ \rho w_z \\ E \end{bmatrix} + \frac{\partial}{\partial r} \begin{bmatrix} \rho u_r \\ \rho u_r^2 + p \\ \rho u_r w_z \\ u_r(E + p) \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} \rho w_z \\ \rho u_r w_z \\ \rho w_z^2 + p \\ w_z(E + p) \end{bmatrix} = \begin{bmatrix} -(\rho u_r)/r \\ -(\rho u_r^2)/r \\ -(\rho u_r w_z) \\ u_r(E + p)/r \end{bmatrix}, \quad (4)$$

where u_r and w_z denote the velocities in the r and z directions. These equations are of the same form as the two-dimensional Euler equations, with the addition of geometric source terms that are a result of the variable transformation. The source terms are never evaluate at $r = 0$ since we are using a finite volume method where quantities are evaluated at cell-centers, that is, the smallest value of r in a calculation is $\Delta x/2$. We prefer to keep the equations in conservation form, so they can be efficiently solved using finite volume methods.

In order to solve the system (3) or (4), we need to close the system with a relation between the pressure and conserved variables. The Tamman EOS [28] is applicable to a wide range of liquids, even with very strong shock waves. This equation of state has the form

$$p = p(\rho, e) = (\gamma - 1)\rho e - \gamma p_\infty, \quad (5)$$

where p , ρ and e are the pressure, density and specific internal energy, respectively, while γ and p_∞ are constants depending on the fluid. If $p_\infty = 0$ this is the standard EOS for an ideal gas, with γ generally satisfying $1 < \gamma < 5/3$, while for water $\gamma \approx 7.15$ and $p_\infty \approx 300$ MPa. For sufficiently weak shocks, this can be approximated by the Tait equation of state,

$$p = p(\rho) = B \left[\left(\frac{\rho}{\rho_0} \right)^n - 1 \right], \quad (6)$$

where B is a pressure term that is a weak function of entropy, but is typically treated as a constant, and corresponds to p_∞ from (5) while n corresponds to γ . Here ρ_0 is the background density measured at one atmospheric pressure. In our work we take $B = 300$ MPa and $n = 7.15$.

It has been common practice to use the Tait EOS in shock wave therapy and lithotripsy models [47; 41]. This has been justified by noting studies that show that entropy changes across the shock are very small even up to pressure jumps of 200 MPa [41], which is beyond the range used in ESWT. To verify this assumption, we performed computational experiments to compare the Tamman and Tait equations of state for typical ESWT shock waves. Since we have used the f-wave approach in our computational model, we can solve (4) with a spatially

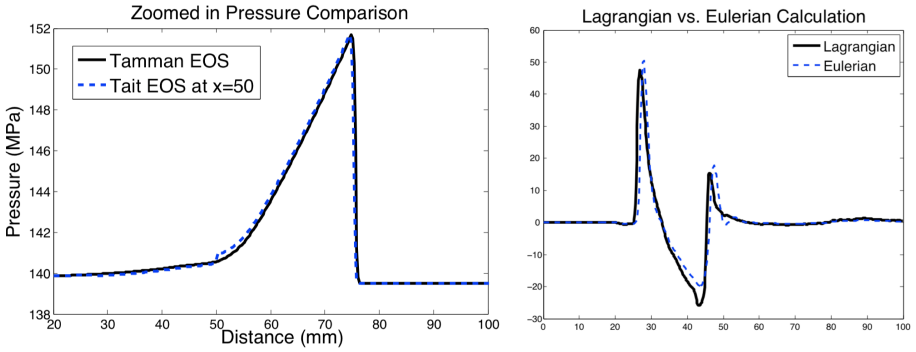


Figure 2. Left: comparison of a pressure wave calculation performed using both the Tait (blue dashed curve) and Tammann (black curve) equations of state. The results are nearly identical. Right: Comparison of the pressure pulse at F2 obtained in the Euler calculation (blue dashed curve) and the Lagrangian calculation (black curve). It is clear that the two sets of equations give good agreement. The wave in the Lagrangian case is slightly attenuated, but this may be due to error in initializing the calculation. In these calculations $\Delta x = 0.5$ mm.

varying equation of state. We set up an experiment where the resulting shockwave (generated using the Tammann equation of state), was over 150 MPa. Figure 2, left, shows the results from this experiment. The black solid curve is the result from solving with the Tammann EOS in the entire domain. The blue dashed curve shows the result gotten by switching to the Tait EOS at $x = 50$. This enabled us to compare the two equations of state with the exact same initial condition. There is a small disagreement at $x = 50$ caused by a slight reflection at the interface due to the change in the equation of state. Otherwise, the pressure profiles are nearly identical, giving confidence that the calculations we are interested in can be done by solving the Euler equations with the Tait equation of state. This allows us to drop the equation for energy and obtain the simplified system

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \end{bmatrix} + \frac{\partial}{\partial z} \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \quad (7)$$

2.2. Compressible fluids in Lagrangian form. In the case of a fluid where the shear modulus is zero, the stress tensor can be written as $\sigma(\epsilon) = -pI$, where p is the pressure in the fluid and I is the identity matrix. In the case of ESWT, the pressure only depends on changes in the density, and we can write $p(\epsilon)$ as a function of the strain tensor ϵ . Consider the movement of a material with respect to a reference configuration and let $\delta = (\delta^x, \delta^y, \delta^z)$ be the infinitesimal displacement.

In three space dimensions, the full strain tensor is

$$\epsilon = \begin{pmatrix} \delta_x^x & \frac{1}{2}(\delta_y^x + \delta_x^y) & \frac{1}{2}(\delta_z^x + \delta_x^z) \\ \frac{1}{2}(\delta_y^x + \delta_x^y) & \delta_y^y & \frac{1}{2}(\delta_z^y + \delta_y^z) \\ \frac{1}{2}(\delta_z^x + \delta_x^z) & \frac{1}{2}(\delta_z^y + \delta_y^z) & \delta_z^z \end{pmatrix}, \quad (8)$$

where subscripts denote partial derivatives.

In the case of small deformations, we have from conservation of mass that

$$\rho = \frac{\rho_0}{1 + \text{tr}(\epsilon)} \quad (9)$$

where ρ_0 is the equilibrium density.

If we insert this into the Tait equation of state (6) we get

$$p(\epsilon) = B \left[\left(\frac{1}{1 + \text{tr}(\epsilon)} \right)^n - 1 \right]. \quad (10)$$

Using the Lagrangian form is only valid in the case where the displacements are small, so we calculated the maximum value of the displacements in a two-dimensional axisymmetric calculation with the Euler equations. We found that for a maximum peak pressure of 50 MPa, the corresponding maximum velocity was 10^{-3} m/s. We then calculated the maximum displacement by integrating the velocity over the time of the calculation and found this to be on the order of 10^{-5} mm. The size of the grid cell is on the order of 10^{-1} mm, so the displacements are 4 orders of magnitude smaller than the width of the grid cells. It is therefore reasonable to assume that the density in each grid cell is essentially constant and that the Lagrangian framework of the elasticity equations will be valid for the fluid.

To test this, we took the same initial condition for the two-dimensional axisymmetric Euler equations with the Tammann equation of state and the corresponding two-dimensional axisymmetric Lagrangian form of the equations with the Tait equation of state and measured the pressure at the focus, F2. The results in Figure 2, right, demonstrate reasonably good agreement between the two cases, but the Lagrangian form is slightly attenuated. This may be due to conversion of the initial condition from the conserved variables in the Euler equations (4) to those in the elasticity equations (1).

Since the displacements are small, we also considered the possibility that nonlinearity in the fluid could be ignored, so we could instead use a linearized version of the Tait equation of state. Then we would be able to simply use the linear elasticity equations throughout the domain, in both the fluid and solid materials. If we assume a small perturbation to the strain, $\epsilon + \delta\epsilon$, we can expand the Tait EOS (6) as a Taylor series about ϵ ,

$$p(\epsilon + \delta\epsilon) = p_0 + p'(\delta\epsilon)\epsilon + \frac{p''(\delta\epsilon)}{2}\epsilon^2 + \dots \quad (11)$$

If we keep the first two terms of the expansion, the EOS has been linearized and we will call this the *linear Tait EOS*. Similarly, we will refer to the equation obtained by keeping the first three terms of the expansion as the *quadratic Tait EOS*. One-dimensional tests of both possibilities are shown in [Figure 3](#), for three different wave amplitudes. For a wave with maximum amplitude less than 3 MPa there is fairly

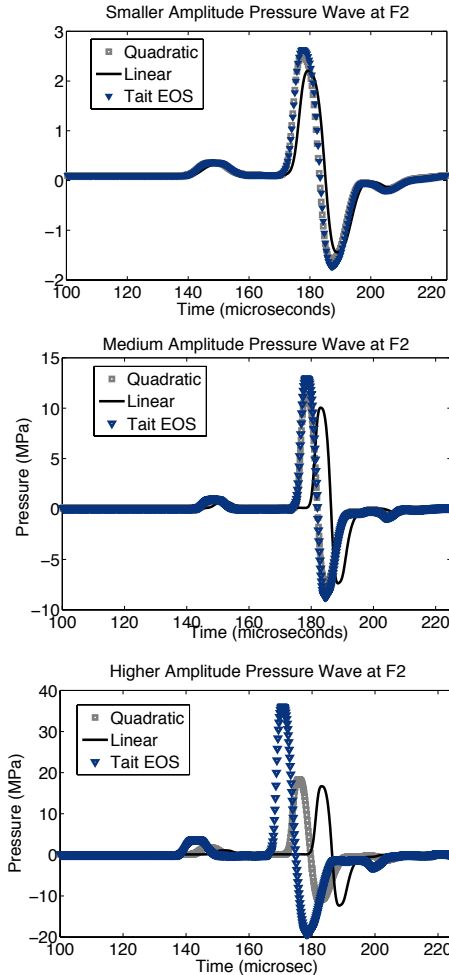


Figure 3. Pressure gauge measurement at F2 of different versions of the Tait EOS at different amplitudes. The triangular markers indicate the full nonlinear Tait EOS, the solid line is a linearized version and the square markers are a quadratic version. The linearized versions of the EOS work reasonably well at small amplitudes, but it’s clear from the bottom figure that as the pressures increase to those observed in ESWT, the full nonlinear equation of state must be used.

good agreement, however, as the amplitude is increased, as is required for ESWT, the linear and quadratic equations of state do not capture the correct behavior. Thus we used the full Tait EOS in the fluid domain.

2.3. Elasticity equations. In the current work we model bone as a linear isotropic solid. We use the equations (1) together with Hooke's law

$$\sigma^{11} = C_{11}\epsilon^{11} + C_{12}\epsilon^{22} + C_{13}\epsilon^{33}, \quad (12)$$

$$\sigma^{22} = C_{21}\epsilon^{11} + C_{22}\epsilon^{22} + C_{23}\epsilon^{33}, \quad (13)$$

$$\sigma^{33} = C_{31}\epsilon^{11} + C_{32}\epsilon^{22} + C_{33}\epsilon^{33}, \quad (14)$$

$$\sigma^{12} = C_{44}\epsilon^{12}, \quad (15)$$

$$\sigma^{13} = C_{55}\epsilon^{13}, \quad (16)$$

$$\sigma^{23} = C_{66}\epsilon^{23}, \quad (17)$$

where the spatially varying scalar coefficients $C_{ij}(x, y, z)$ are determined by the properties of the material being modeled. The parameters used for the bone model were found in [37].

For an isotropic material we can relate the C_{ij} above to the two Lamé parameters, λ and μ , that are used to model different elastic materials. $C_{ii} = \lambda + 2\mu$ for $i = 1, \dots, 3$, $C_{ii} = 2\mu$ for $i = 4, \dots, 6$, and $C_{ij} = \lambda$ for $i \neq j$. Here μ is the shear modulus and $\lambda + 2\mu$ is the bulk modulus of the material. Note that the λ here is different from the λ^i used to denote the eigenvalues elsewhere in the paper.

Linear elasticity has been used extensively in the literature to model both trabecular and cortical bone [29; 21]. Linear viscoelastic models have also been used for ultrasound wave propagation in bone [22]. Our model could be extended to orthotropic models, requiring 9 material parameters, as has also been used for bone modeling; see, for example, [52].

3. Eigenstructure of the hyperbolic system

The full three-dimensional system of equations (1) models both the nonlinear fluid and the linear elastic bone as described in the preceding sections. This system can be written in quasilinear form:

$$q_t + A(q, x, y, z)q_x + B(q, x, y, z)q_y + C(q, x, y, z)q_z = 0, \quad (18)$$

where A , B and C are the Jacobians of the flux functions in the x , y and z directions respectively. For the multidimensional methods implemented in Clawpack, we need the solution to the Riemann problem along slices in each coordinate direction. Here we provide the details for the solution in the x direction, but the solution in the y and z directions are similar with appropriate permutations to the B and C matrices.

The corresponding Jacobian for this system in the x direction is:

$$A(q, x, y, z) = \frac{\partial f(q, x, y, z)}{\partial x} = - \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{\rho_0} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2\rho_0} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2\rho_0} \\ \sigma_{\epsilon^{11}}^{11} & \sigma_{\epsilon^{22}}^{11} & \sigma_{\epsilon^{33}}^{11} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\epsilon^{12}}^{12} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{\epsilon^{13}}^{13} & 0 & 0 & 0 \end{pmatrix}, \quad (19)$$

where $\sigma_{\epsilon^{33}}^{11}$, for example, denotes the partial derivative of σ^{11} with respect to ϵ^{33} . In the linear elastic case this is simply the coefficient C_{13} , but the above form also applies to the nonlinear compressible equations. The spatial variation in $f(q, x, y, z)$ and the Jacobian A result from allowing the material parameters such as density and elastic moduli to vary in space. The Jacobians in the y and z directions are similar with the entries permuted appropriately.

The eigenvalues for system (19) are

$$\lambda^{1,2} = \pm \sqrt{\frac{\sigma_{\epsilon^{11}}^{11}}{\rho_0}}; \quad \lambda^{3,4} = \pm \sqrt{\frac{\sigma_{\epsilon^{12}}^{12}}{2\rho_0}}; \quad \lambda^{5,6} = \pm \sqrt{\frac{\sigma_{\epsilon^{13}}^{13}}{2\rho_0}}; \quad \lambda^{7,8,9} = 0. \quad (20)$$

When modeling a fluid where the shear stress is zero, there are seven zero-speed eigenvalues since $\sigma_{\epsilon^{12}}^{12} = \sigma_{\epsilon^{13}}^{13} = 0$. Only the compressional waves corresponding to $\lambda^{1,2}$ propagate with nonzero speed. Note that the Tait equation of state (10) gives

$$\sigma_{\epsilon^{11}}^{11} = \frac{\partial \sigma^{11}}{\partial \epsilon^{11}} = Bn \left(\frac{1}{1 + \epsilon^{11} + \epsilon^{22} + \epsilon^{33}} \right)^{n+1} = \frac{n(p+B)}{1 + \text{tr } \epsilon}. \quad (21)$$

In the small amplitude acoustic limit $\epsilon \rightarrow 0$, from (20) we obtain the wave speeds

$$\pm \sqrt{\frac{n(p+B)}{\rho_0}}, \quad (22)$$

which are the expected waves speeds for compressional waves in the Lagrangian form with this equation of state.

For the elastic solid, on the other hand, waves 1 and 2 correspond to P-waves while waves 4–6 correspond to S-waves, and the expected wave speeds are recovered

based on the elastic coefficients given in [Section 2.3](#). For example, in the x direction the P-wave speeds are

$$\pm \sqrt{\frac{C_{11}}{\rho_0}}, \quad (23)$$

and the S-wave speeds are

$$\pm \sqrt{\frac{C_{44}}{2\rho_0}} \quad \text{and} \quad \pm \sqrt{\frac{C_{55}}{2\rho_0}}. \quad (24)$$

The corresponding eigenvectors for system (19) are

$$\begin{aligned} r^{1,2} &= [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \pm \sqrt{\rho_0 \sigma_{\epsilon_{11}}^{11}} \ 0 \ 0]^T, \\ r^{3,4} &= [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ \pm \sqrt{2\rho_0 \sigma_{\epsilon_{12}}^{12}} \ 0]^T, \\ r^{5,6} &= [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ \pm \sqrt{2\rho_0 \sigma_{\epsilon_{13}}^{13}}]^T, \end{aligned} \quad (25)$$

for the P-waves and S-waves, and

$$\begin{aligned} r^7 &= [-\sigma_{\epsilon_{22}}^{11} \ \sigma_{\epsilon_{11}}^{11} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T, \\ r^8 &= [-\sigma_{\epsilon_{33}}^{11} \ 0 \ \sigma_{\epsilon_{11}}^{11} \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T, \\ r^9 &= [0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^T, \end{aligned} \quad (26)$$

for the stationary waves.

3.1. Axisymmetric form of the equations. We used the two-dimensional axisymmetric form of the equations to generate an initial condition for our three-dimensional calculations, as well as for validation of our model.

The three-dimensional equations in cylindrical coordinates are:

$$\begin{aligned} \epsilon_t^{rr} &= \frac{\partial u}{\partial r}, & \epsilon_t^{\theta\theta} &= \frac{u}{r} + \frac{1}{r} \frac{\partial v}{\partial \theta}, & \epsilon_t^{zz} &= \frac{\partial w}{\partial z}, \\ \epsilon_t^{rz} &= \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \right), & \epsilon_t^{r\theta} &= \frac{1}{2} \left(\frac{\partial v}{\partial r} + \frac{1}{r} \frac{\partial u}{\partial \theta} - \frac{v}{r} \right), & \epsilon_t^{\theta z} &= \frac{1}{2r} \left(\frac{\partial w}{\partial \theta} + \frac{\partial v}{\partial z} \right), \\ \rho u_t &= \frac{1}{r} \frac{\partial \sigma^{r\theta}}{\partial \theta} + \frac{\partial \sigma^{rr}}{\partial r} + \frac{\sigma^{rr} - \sigma^{\theta\theta}}{r} + \frac{\partial \sigma^{rz}}{\partial z}, \\ \rho v_t &= \frac{1}{r} \frac{\partial \sigma^{\theta\theta}}{\partial \theta} + \frac{\partial \sigma^{r\theta}}{\partial r} + \frac{2\sigma^{r\theta}}{r} + \frac{\partial \sigma^{z\theta}}{\partial z}, \\ \rho w_t &= \frac{1}{r} \frac{\partial \sigma^{z\theta}}{\partial \theta} + \frac{\partial \sigma^{zz}}{\partial z} + \frac{\partial \sigma^{rz}}{\partial r} + \frac{\sigma^{rz}}{r}. \end{aligned} \quad (27)$$

If we assume that $v = \epsilon_{\theta z} = \epsilon_{r\theta} = 0$ and there is no variation in the θ direction, then the system (27) simplifies to

$$\begin{aligned} \epsilon_t^{rr} &= \frac{\partial u}{\partial r}, & \epsilon_t^{\theta\theta} &= \frac{u}{r}, & \epsilon_t^{zz} &= \frac{\partial w}{\partial z}, & \epsilon_t^{rz} &= \frac{1}{2} \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial r} \right), \\ \rho u_t &= \frac{\partial \sigma^{rr}}{\partial r} + \frac{\sigma^{rr} - \sigma^{\theta\theta}}{r} + \frac{\partial \sigma^{rz}}{\partial z}, \\ \rho w_t &= \frac{\partial \sigma^{zz}}{\partial z} + \frac{\partial \sigma^{rz}}{\partial r} + \frac{\sigma^{rz}}{r}. \end{aligned} \quad (28)$$

It is interesting to note here that the strain in the $\theta\theta$ direction is nonzero and in this case is called the hoop strain. A uniform radial displacement is not a rigid body motion, as it would be in the two-dimensional plane strain case, but instead produces a circumferential strain. This is because the original circumference of the cylinder is $2\pi r$, but when there is a strain in the radial direction the circumference grows to $2\pi(r + u_r)$, inducing a strain $2\pi u_r / 2\pi r = u_r / r$.

The Jacobian for system (28) in the z direction is

$$f'(q) = - \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{\rho_0} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2\rho_0} \\ \sigma_{\epsilon_{rr}}^{rr} & \sigma_{\epsilon_{zz}}^{rr} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{\epsilon_{rz}}^{rz} & 0 & 0 \end{pmatrix}, \quad (29)$$

and has an eigenstructure that is equivalent to the two-dimensional elasticity equations, with the addition of a second zero-speed eigenvalue.

These equations have the structure

$$q_t + f(q)_r + g(q)_z = S(q, r), \quad (30)$$

with source terms

$$\epsilon_t^{\theta\theta} = \frac{u}{r}, \quad \rho u_t = \frac{\sigma_{rr} - \sigma_{\theta\theta}}{r}, \quad \rho w_t = \frac{\sigma_{rz}}{r}. \quad (31)$$

In Clawpack, we solve these equations with a fractional-step method. The full problem is split into two subproblems that are solved independently. We first solve the homogeneous system obtained by setting $S \equiv 0$ in (30) using the wave propagation algorithm described in Section 4, and then solve

$$q_t = S(q, r), \quad (32)$$

with an appropriate ODE solver. For (31), we use forward Euler.

4. Numerical methodology

We used the wave-propagation algorithms described in [35] and implemented in Clawpack [15] to solve the hyperbolic systems of PDEs derived in the preceding sections. In this section we provide the basic details of the numerical methodology and the approximate solution to the Riemann problem with a spatially varying flux function, similar to what was done in [36]. We also discuss computational issues that require the use of adaptive mesh refinement.

4.1. Riemann solvers and wave-propagation algorithms. Recall that the ‘‘Riemann problem’’ is the initial value problem for a one-dimensional hyperbolic system of the form

$$q_t + f(q, x)_x = 0, \quad (33)$$

with special initial data consisting of two constant states separated by a discontinuity

$$q_0(x) = \begin{cases} Q_l & \text{if } x < 0, \\ Q_r & \text{if } x > 0. \end{cases} \quad (34)$$

If the flux function is spatially varying then we also use a piecewise-defined flux function with

$$f(q, x) = \begin{cases} f_l(q) & \text{if } x < 0, \\ f_r(q) & \text{if } x > 0. \end{cases} \quad (35)$$

The Riemann problem plays a fundamental role in the theory and computation of hyperbolic problems, since the Riemann solution consists of waves propagating at constant speeds and can generally be computed. For nonlinear systems of equations this is often replaced by an approximate Riemann solver as will be discussed below.

For a linear system of equations $q_t + A(x)q_x = 0$ the Riemann solution is easily computed in terms of the eigenvectors and eigenvalues of the matrices A_l to the left of the interface and A_r to the right of the interface. We begin by discussing the linear case with a constant matrix A and turn to the variable-coefficient (heterogeneous media) case in Section 4.3. We assume the matrix A is diagonalizable,

$$A = R\Lambda R^{-1}, \quad (36)$$

where R is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. The Riemann solution is computed by decomposing $\Delta Q = Q_r - Q_l$ as a linear combination of eigenvectors of A ,

$$\Delta Q = \sum_{p=1}^m \alpha^p r^p, \quad \text{where } \alpha = R^{-1} \Delta Q. \quad (37)$$

We denote the p -th wave by ${}^pW_p = \alpha^p r^p$, where $p = 1, 2, \dots, m$ and the number of waves m is equal to the number of equations in the system.

We use finite volume methods in which Q_i^n represents a cell average of the vector q in cell i at time t_n (still in one space dimension). In Godunov's method the cell average is updated by the waves entering the cell from the interfaces to the left and the right, and each wave updates the cell average by ${}^{\circ}W^P$, the jump in q across the wave, multiplied by the distance the wave propagates over the time step and divided by the length of the cell, that is, the cell average is updated by $(\lambda^P \Delta t / \Delta x) {}^{\circ}W^P$. To express the total update to a cell, it is convenient to define matrices A^+ and A^- via

$$A^{\pm} = R \Lambda^{\pm} R^{-1}, \quad \text{where } \Lambda^{\pm} = \text{diag}(\lambda_p^{\pm}), \quad (38)$$

with $\lambda^+ = \max(\lambda, 0)$ and $\lambda^- = \min(\lambda, 0)$. Then the cell average is updated by

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2}). \quad (39)$$

Here $\Delta Q_{i-1/2} = Q_i - Q_{i-1}$ is the jump across the interface at $i - 1/2$, for example. For a linear system this is a generalization of the upwind method and is first order accurate.

Second order accuracy is achieved by adding in correction fluxes:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (A^+ \Delta Q + A^- \Delta Q) - \frac{\Delta t}{\Delta x} (\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}), \quad (40)$$

where

$$\tilde{F}_{i-1/2} = \frac{1}{2} \left(1 - \left| \frac{\lambda^P \Delta t}{\Delta x} \right| \right) |\lambda^P| {}^{\circ}W_{i-1/2}^P. \quad (41)$$

These terms convert the upwind method into a method of Lax–Wendroff type, matching terms through $\Delta t^2 A^2 q_{xx}$ in the Taylor series expansion of the solution at the end of the time step. This method generates dispersive errors, however, that can create large nonphysical oscillations near steep gradients or discontinuities in a solution, such as shock waves. To turn this into a “high-resolution” method, we use a wave limiter, replacing ${}^{\circ}W_{i-1/2}^P$ in (41) by $\tilde{W}_{i-1/2}^P$, a limited version of the wave. The wave ${}^{\circ}W_{i-1/2}^P$ is compared to the corresponding wave from the neighboring Riemann problem, either ${}^{\circ}W_{i-3/2}^P$ if $\lambda^P > 0$ or ${}^{\circ}W_{i+1/2}^P$ if $\lambda^P < 0$. If the waves are of comparable magnitude the full correction term is used for accuracy, but if there is a large discrepancy then the solution is not smooth at this point and a limited version is applied. See [34] or [35, Chapter 6] for more complete details.

In two or three space dimensions the idea is the same, but now a one-dimensional Riemann problem must be solved normal to each edge or face of the cell. The resulting waves update the cell averages and correction fluxes analogous to (41) are used along with limiters in each direction.

In addition, to achieve second-order accuracy and good stability properties, it is also necessary to use “transverse Riemann solvers” that further modify the correction fluxes \tilde{F} at each cell edge. The method described above is based on

propagating waves normal to each interface. In reality, the waves will propagate in a multidimensional manner and affect cell averages in cells above and below those that are directly adjacent to the interface.

In two dimensions, each “fluctuation” such as $A^- \Delta Q_{i-1/2,j}$ and $A^+ \Delta Q_{i-1/2,j}$ that results from solving a Riemann problem in the x direction is split into two pieces using the eigenstructure of the coefficient matrix B in the y direction, for example:

$$A^+ \Delta Q_{i-1/2,j} = B^- A^+ \Delta Q_{i-1/2,j} + B^+ A^+ \Delta Q_{i-1/2,j}. \quad (42)$$

These two pieces will modify the correction flux at the edges $(i, j - 1/2)$ and $(i, j + 1/2)$ respectively to capture the transverse motion of the right-going wave. Similarly, after solving a normal Riemann problem in the y direction using the B matrix, transverse problems are solved based on the eigenstructure of A . The net effect of all these corrections is to incorporate terms modeling the cross-derivative terms BAq_{xy} and ABq_{yx} of the Taylor series expansion in a properly upwinded manner. More details can be found in [34] or [35, Chapter 21]. The transverse correction terms are needed for accuracy, but also have the effect of improving the stability limit, allowing a Courant number near 1 to be used, relative to the maximum wave speed in any direction.

In three space dimensions there are two transverse directions for each normal Riemann solve, and terms modeling CAq_{xz} , etc. must also be included. Moreover, “double transverse” terms must be included, splitting the result of a transverse solve into eigenvectors of the remaining coefficient matrix, and modeling terms such as $BCAq_{xzy}$. The details are presented in [31] and fully implemented in Clawpack.

4.2. The nonlinear fluid Riemann solver. The compressible fluid equations in Lagrangian form discussed in Section 2.2 can be reduced to the quasilinear form (18) in which the Jacobian matrices depend only on q (for a spatially uniform fluid). To apply the wave-propagation algorithm we need to solve the Riemann problem orthogonal to each cell interface. For nonlinear problems this is usually done using an approximate Riemann solver, for example, by replacing $f(q)_x$ by $\hat{A}q_x$, where the matrix \hat{A} at each cell interface is chosen based on the data Q_l and Q_r to the left and right. We use the f-wave formulation of the wave-propagation algorithm [4], in which the jump in flux $f(Q_r) - f(Q_l)$ is split into eigenvectors of an approximate Jacobian matrix, rather than the jump in Q . This leads to an algorithm that is conservative for any choice of approximate Jacobian and also extends naturally to the case of spatially varying fluxes, as required near the fluid-solid boundary and discussed further below.

Rather than choose an approximate Jacobian \hat{A} and then determining its eigenvectors and eigenvalues, we simply choose the set of eigenvectors and associated

wave speeds based on the data and wave forms expected to result from this data. These vectors form a matrix \hat{R} and we then solve $\hat{R}\beta = f(Q_r) - f(Q_l)$ for the vector of wave strengths β . The choice of vectors in \hat{R} and associated wave speeds $\hat{\lambda}$ implicitly defines the Jacobian approximation $\hat{A} = \hat{R}\hat{\Lambda}\hat{R}^{-1}$, but this matrix is never needed.

The eigenvectors are taken to be the vectors displayed in (25) and (26). Recall that in the fluid case there are only two nonzero eigenvalues corresponding to the first two eigenvectors. For the eigenvector corresponding to $\lambda^1 < 0$ we use $\lambda^1 = -\sigma_{\epsilon_{11}}^{11}$ evaluated in the left state Q_l , while the eigenvector corresponding to $\lambda^2 > 0$ is determined using $\lambda^2 = \sigma_{\epsilon_{11}}^{11}$ evaluated in the right state Q_r . These vectors have nonzero components only in positions 1 and 7 and so the values of β^1 and β^2 can be determined by solving a 2×2 system:

$$\begin{bmatrix} 1 & 1 \\ \rho_l \lambda_l^1 & \rho_r \lambda_r^2 \end{bmatrix} \begin{bmatrix} \beta^1 \\ \beta^2 \end{bmatrix} = \begin{bmatrix} \Delta f^1 \\ \Delta f^7 \end{bmatrix}. \quad (43)$$

The solution is

$$\begin{aligned} \beta^1 &= \frac{\rho_r \lambda_r^2 \Delta f^1 - \Delta f^7}{\rho_r \lambda_r^2 - \rho_l \lambda_l^1}, \\ \beta^2 &= \frac{\Delta f^7 - \rho_l \lambda_l^1 \Delta f^1}{\rho_r \lambda_r^2 - \rho_l \lambda_l^1}. \end{aligned} \quad (44)$$

The remaining waves do not propagate and do not come into the wave-propagation algorithm.

4.3. The linear elastic Riemann solver. In the linear elastic material modeling bone, we take a similar approach and again use the f-wave formulation of the algorithm. In this case there are six waves with nonzero wave speeds given by the eigenvectors in (25). The eigenvectors are independent of q in the linear case, but can be spatially varying to represent varying bone structure, so the coefficients C_{ij} in (12) can vary from one grid cell to the next. Similar to the nonlinear case described above, to compute the decomposition of the flux difference into propagating waves we define the three left-going eigenvectors $r^{1,3,5}$ (with the minus sign in (25)) based on the coefficients in the left state, while the right-going eigenvectors $r^{2,4,6}$ are defined using the coefficients in the right state. Note that the flux vector $f(q)$ from (2), and hence any jump in flux, has zeros in three components which are easily seen to lead to $\beta^7 = \beta^8 = \beta^9$ when the flux difference is written as a linear combination of the eigenvectors, and the six remaining components of the flux difference uniquely define the coefficients β^1 through β^6 for the six propagating waves. The weights β^1 and β^2 are the same as is (44), and the others are

$$\beta^3 = \frac{\rho_r \lambda_r^4 \Delta f^4 - \Delta f^8}{\rho_r \lambda_r^4 - \rho_l \lambda_l^3}, \quad \beta^4 = \frac{\Delta f^8 - \rho_l \lambda_l^3 \Delta f^4}{\rho_r \lambda_r^4 - \rho_l \lambda_l^4}, \quad (45)$$

$$\beta^5 = \frac{\rho_r \lambda_r^6 \Delta f^6 - \Delta f^9}{\rho_r \lambda_r^6 - \rho_l \lambda_l^5}, \quad \beta^6 = \frac{\Delta f^9 - \rho_l \lambda_l^5 \Delta f^6}{\rho_r \lambda_r^6 - \rho_l \lambda_l^5}.$$

4.4. Interfaces and the Cartesian grid. In ESWT the pressure wave must propagate through a variety of materials, and in general the interfaces between different materials do not align with the grid. In our calculations we use a Cartesian grid. To handle grid cells that contain two materials, we perform a weighted average of the material properties. The stress-strain relationship in the averaged grid cells is taken to be that from linear elasticity, even if one of the materials is fluid. This approach is feasible because we use AMR to refine around the interfaces between the two materials. By using a fine enough grid, we are able to reduce the error introduced by the weighted average approximation. Figure 4, left, illustrates the interface between the fluid and the brass reflector from an axisymmetric calculation. Three grid resolutions are shown in this figure: a coarse grid on the right, a level 2 grid that is refined by a factor of 4 in each direction in the middle, and the finest grid on the left, where the grid lines are not drawn.

Figure 4, right, shows a comparison between the pressure wave measured at F2 for an AMR calculation versus a single grid calculation. The single grid calculation took 269 minutes to complete, just over 6 times as long as the run using AMR which finished in 44 minutes. These calculations were performed serially on a 2.8 GHz dual core AMD Opteron machine with 32 GB of memory. It's clear that the two calculations yield comparable pressure waves. The biggest difference is in the direct wave arriving around $t = 150$, which is not being resolved in the AMR

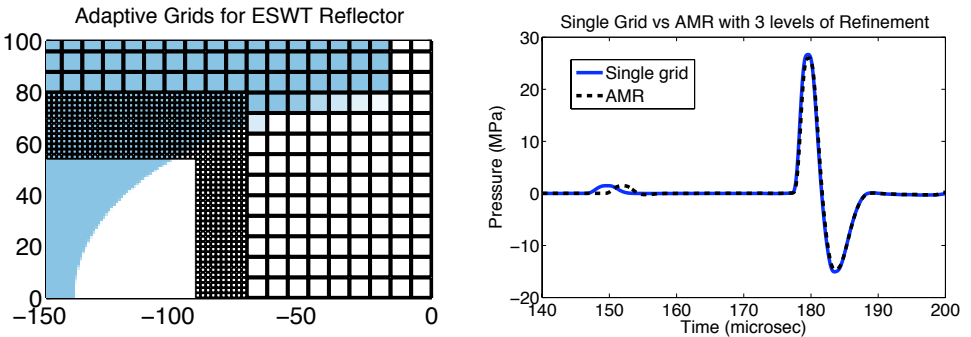


Figure 4. Left: resolution of the ellipsoid reflector with different levels of AMR. Right: two-dimensional axisymmetric calculation. The second is a comparison of the waveform obtained using AMR and a uniform grid. The finest grid resolution in the AMR calculation is the same as the resolution on the uniform grid.

calculation because we have refined only in the vicinity of the reflected wave of primary interest.

4.5. Adaptive mesh refinement. The pressure waveform found in ESWT contains a very thin region of high pressure that can not be resolved without a highly refined mesh. In Figure 5 we investigated the effect of grid refinement on the shock wave profile and found that with grid resolution greater than 0.25 mm, the wave form at F2 was not a shock. Note that near the shock we only expect our method to be first order, but the solution does converge to a shock as the grid is refined. Our calculations are done with the adaptive mesh refinement (AMR) in the style of Colella, Berger and Olinger [5; 8]. The AMR algorithms used in Clawpack are more fully described in [7]. For the three-dimensional calculations, a similar AMR algorithm is used, as implemented in Chombo. Here we only briefly review the main ideas.

The computational domain is covered by a rectangular level-1 grid, typically at a coarse resolution. Rectangular patches of the grid may be covered by level-2 grids, refined by some specified refinement ratio in each direction. Since we use explicit methods, the Courant–Friedrichs–Lewy condition generally requires that the time step be refined by the same factor on the level-2 grids, so several time steps must be taken on each level-2 grid for each time step on the level-1 grid. The level 1 grid is advanced first, and for each time step on the level-2 grid, ghost cell values around the boundary are filled in either by copying from adjacent grids at the same level, or using space-time interpolation from the level-1 grid for ghost cells that do not lie in an adjacent grid. This entire procedure is repeated recursively to obtain

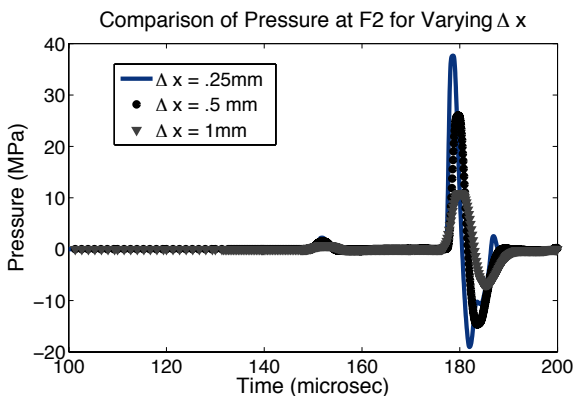


Figure 5. Effect of grid size on shock wave profile. As the grid is refined for the same initial condition, the shock wave profile steepens. The solution eventually converges to a profile with the same magnitude, though the convergence rate is only first order near a discontinuity.

higher levels of refinement; e.g., some portion of the collection of level-2 grids may be covered by level-3 grids and so on.

In order to adaptively refine the grid, it is important to specify appropriate refinement criteria. The perturbations to the strain are small, so gradients in the strain are too small to use as reliable refinement criteria. However, the small strains result in large changes in the pressure, so we refine in the area near the pressure wave. In order to handle the interfaces between two materials, we also use large gradients in background density as a secondary refinement criterion. Cells that are flagged as needing refinement are clustered into rectangular patches using the algorithm of Berger and Rigoutsis [6]. Regridding is done every few steps on each grid level in order to track propagating waves. Regions are automatically de-refined once the wave passes by, since cells in these regions are no longer flagged as needing refinement.

Figure 5 illustrates the behavior of the ESWT waveform as the grid is refined. What is evident from these experiments is that a coarse grid will not effectively capture the development of the shock, so around the propagating wave, we need at least $\Delta x = 0.25$ mm resolution. As the wave steepens into a shock, we no longer expect second order convergence, because in the region around a discontinuity, our methods are first order. However, since the discontinuities occur in a small region of the domain, the overall methodology is still second order.

In order to efficiently calculate a reasonable ESWT waveform in three dimensions, we utilized ChomboClaw [10], which uses the adaptive mesh refinement routines of CHOMBO with the wave propagation solvers of Clawpack. This code can be run in parallel using MPI on an NSF TeraGrid computer at TACC.

5. Results

We have used the approach described above to model ESWT pressure waves interacting with three-dimensional bone geometries comprised of idealized materials. We have modeled both simple objects that have been used in laboratory experiments as well as complex three-dimensional geometries extracted from CT scans of patient data [18]. Here we present results that demonstrate the efficacy of the Lagrangian formulation, as well as examples of calculations performed using real three-dimensional geometries.

The calculations were initialized using pressure data obtained from a two-dimensional axisymmetric calculation where we modeled the full geometry of the ellipsoidal reflector. The reflector was modeled using linear elasticity with material properties that can be found in [18]. We assumed the fluid was water with the corresponding parameters for the Tait equation of state found in Section 2.1. We saved the data at $t = 116 \mu\text{s}$ and used this to restart future calculations. For the

three-dimensional initial condition, we rotated the two-dimensional data about the x -axis. The material properties of averaged bone were obtained from [37] and used in the heterotopic ossification, cylinder and sphere calculations.

We have found in our experiments that interfaces between materials with large impedance differences have the most significant effect on maximum stress and energy deposition.

5.1. Reflection and focusing. In Figure 6, we show an axisymmetric calculation of the ESWT wave propagation and focusing in water alone, in a domain bounded by the ellipsoidal reflector of the Dornier HM3. Figure 6, top left, shows the initial spherical propagation of the pressure wave, as well as the grids where the calculation is being refined. The grid must be refined around the pressure wave as well as the reflector. Figure 6, top right and bottom left, shows the propagation of the wave and evolution of the adaptive grid structures. At later times the grid is only being refined near the pressure wave. The sharp results and absence of spurious oscillations in the pressure measurement at F2 indicate that AMR together with our Cartesian grid approach enables us to capture the reflection at the interface.

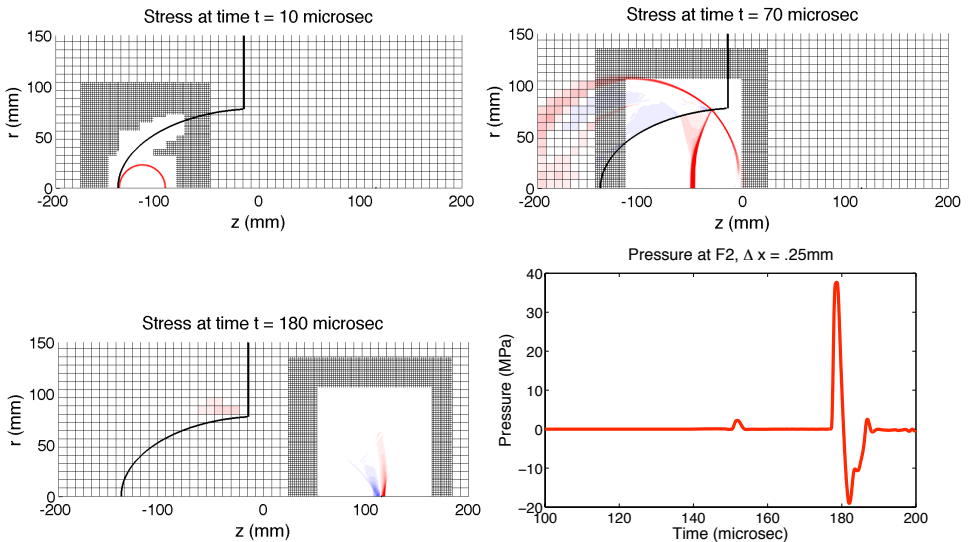


Figure 6. Axisymmetric calculation of the pressure pulse generated by a spherical high-pressure bubble centered $z = -115$ (the focus F1 of the ellipsoidal reflector). Three levels of AMR are used and grid lines are shown only on levels 1 and 2. The level-3 grid has a resolution of $\Delta z = \Delta r = 0.25$ mm. Top left: at $t = 10$ the pulse has nearly reached the reflector. Top right: at $t = 70$ the incident, transmitted, and reflected pulses are visible. Bottom left: at $t = 180$ the reflected pulse has focused near $z = 115$ (the focus F2). Bottom right: the time history of the pressure at F2. The direct (unreflected) wave passes F2 at $t \approx 150$ and the focused pulse arrives at $t \approx 180$.

5.2. Axisymmetric sphere. We used an axisymmetric test problem in order to compare the solutions obtained with the two-dimensional and three-dimensional codes. The initial condition for this experiment was an analytic form for an ESWT pressure wave used in [49]. In the two-dimensional case, we specified the pressure as a function of the radial distance from F1(-115,0). In the three-dimensional case, we rotated the same two-dimensional initial condition about the x -axis. The grid resolution was $\Delta x = 0.25$ mm.

Results with contour lines are shown in Figure 7. The maximum values in each of the three cases are nearly the same, but there are slight discrepancies in the contour

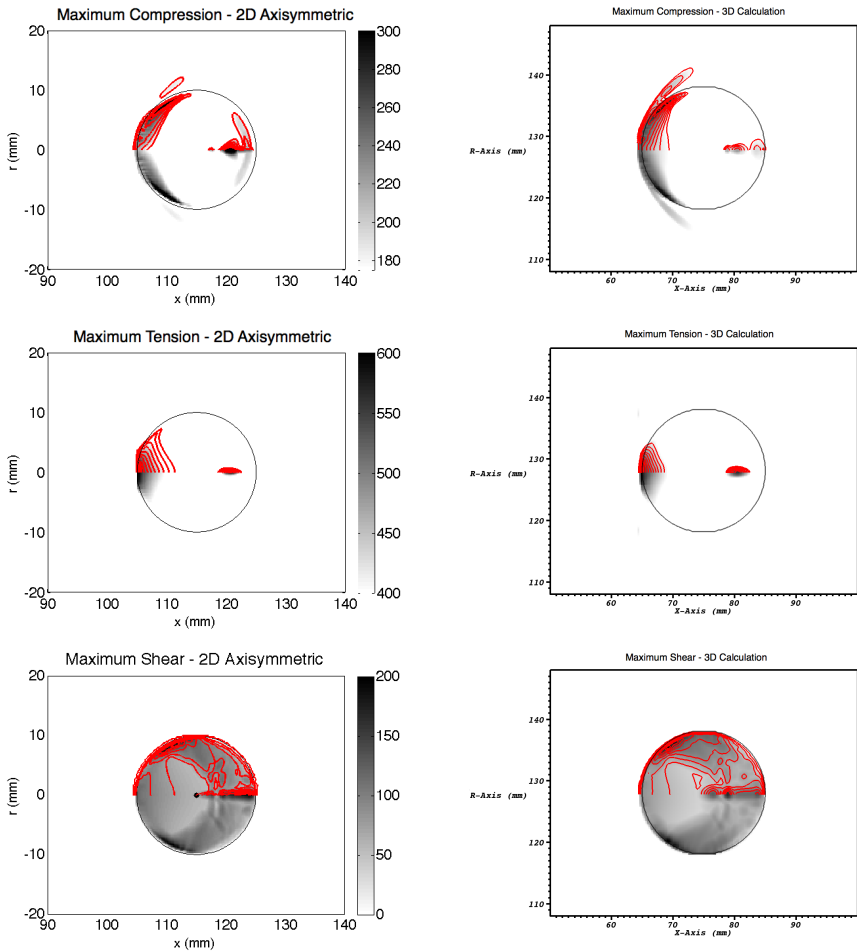


Figure 7. Results from calculation of a shockwave interacting with an acrylic sphere. The left column shows two-dimensional axisymmetric results and the right column shows a corresponding cross section of full three-dimensional calculation. Top: maximum compression; middle: maximum tension; bottom: maximum shear.

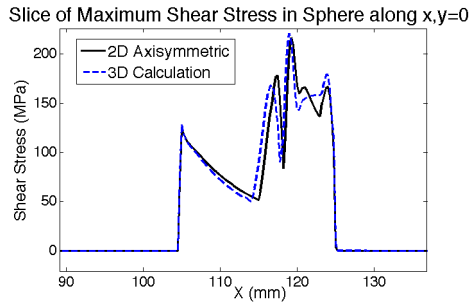


Figure 8. Comparison of maximum shear stress from two-dimensional and three-dimensional calculations as a function of x along $y = z = 0$. The difference in the results is likely caused by averaging of the initial condition onto the three-dimensional domain and the boundary conditions on the axisymmetric calculation at $r = 0$, but the two calculations predict comparable location and magnitude of maximal shear stress deposition.

lines. **Figure 8** shows a one-dimensional slice of the maximum shear calculation in the two-dimensional and three-dimensional codes, which makes it clear that the peak of maximal shear stress is in the same location and has the same value. The general shape of the maximum stress deposition pattern are similar in both cases. The difference in the two solutions is likely caused by the solid wall boundary condition that is used at $r = 0$. Only waves that are propagating normal to that boundary are perfectly reflected, otherwise some error is generated.

5.3. Nonunions. ESWT has recently been used for the treatment of nonunions or bone fractures that fail to heal [9]. One question that is of interest to clinicians is whether or not the angle of treatment has an effect on healing. We assume that healing is related to the magnitude of stress applied near the treatment area, although the connection between the applied force and biological response is not yet understood. In the fluid there is no shear stress. However, at the liquid-solid interfaces, shear stresses are generated by the shockwave and stimulate motion both at the surface and within the material. The motion of the biological materials (e.g., the periosteum, interstitial fluid, mechanotransduction) is likely to be important in the healing process [25; 44; 56; 13; 27; 43; 53], and modeling the magnitude and location of the stress deposition is a good first step toward understanding the shear and tensile displacements caused by ESWT. We should stress, however, that the healing mechanisms are not well understood and we are not claiming that magnitude of the applied stress is the most important or only biological mechanism involved in the healing process. As mentioned in **Section 1**, several studies have indicated that cyclic application of mechanical loading leads to the generation of new bone. The work of Isaksson et al. [27], indicates that the most accurate predictors for bone healing are those based on shear strain and fluid flow, however, there is no

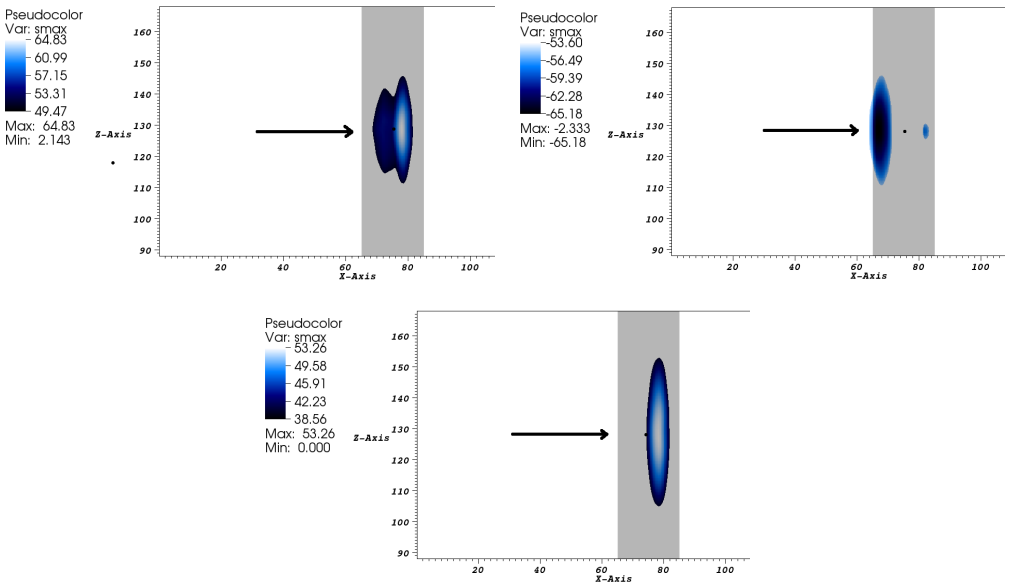


Figure 9. Three-dimensional results for the direct treatment of a complete cylinder. This figure shows two-dimensional slices of maximum compression, tension and shear along $y = 0$ for treatment where the ESWT wave propagates along the x -axis, as indicated by the arrow. The dot illustrates the location of F2.

single model that can predict all features of the healing process, so more work is necessary [39].

In an actual treatment, the clinician generally sets up the device so that the focus is aligned with the ailment. For example, in the case of a broken bone, the clinician will set up the device so that F2 is in the center of the break. However, given the heterogeneous media, it is not clear that the maximal stresses will actually be observed at F2, as would be expected in pure water. We used our model to investigate the location of maximal stress deposition relative to F2. In these calculations we considered two different geometries, a complete cylinder, representing the long shaft of a healthy bone, and a broken cylinder, representing a nonunion. The results from calculations where the idealized bone was perpendicular to the direction of the pressure wave front are shown in Figures 9 and 10. We found that the break has a significant impact on the location of stress deposition.

We used these geometries to perform a variety of experiments. We rotated the direction of treatment by 45 and 60 degrees relative to the x -axis and calculated both the magnitude of the maximum compressive, tensile and shear stresses, as well as the distance from the focus F2 of the device.

In the case of the broken cylinder, the maximum stress deposition in the direct experiment is similar to that of the unbroken cylinder, except that there are two

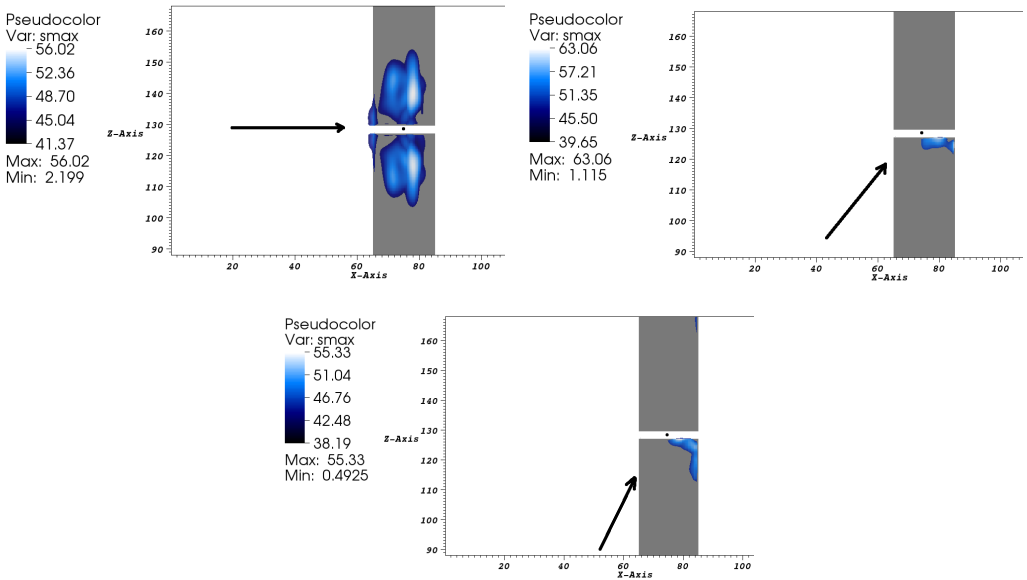


Figure 10. Three-dimensional results for the direct treatment of a broken cylinder. This figure shows two-dimensional slices of maximum compression along $y = 0$ for treatment along the x -axis, 45-degree rotation and 60-degree rotation about the y -axis. The arrows indicate the angle of treatment in each case and the dot illustrates the location of F2.

locations of maximal stress deposition on either side of the break. The pressures in the bone are larger than in the fluid due to reflection at the fluid-solid interface, so the contours of maximum stress are concentrated on either side of the gap. The location along the x -axis is nearly the same as in the unbroken cylinder, and the distances from the ideal focal point, F2, are also similar.

As the angle of treatment is varied, there is less of a shift in the z direction for the shear and compressive stresses. This is caused by the impedance difference between the fluid and solid material at the gap, which is located close to F2. If the gap were shifted along the z -axis from the focal point, there would be a corresponding shift in the location of maximum shear and compression. Geometrically, the shape of the regions of compressive and shear stress are quite different from the direct case. Instead of being an ellipsoidal shape, the regions are compressed into the corner of the lower-half of the cylinder. Again, this is caused by the impedance jump at the fluid-solid interface. The region of maximum tension deposition is similar to that of the unbroken cylinder case, though it is also affected by the gap and the tension is concentrated on the upper half of the cylinder.

It is clear from the literature [43; 44; 30; 13; 27; 11; 23], that mechanical loading is important in bone healing. The implication of our computational experiments is that the angle of treatment will affect stress deposition and therefore may be

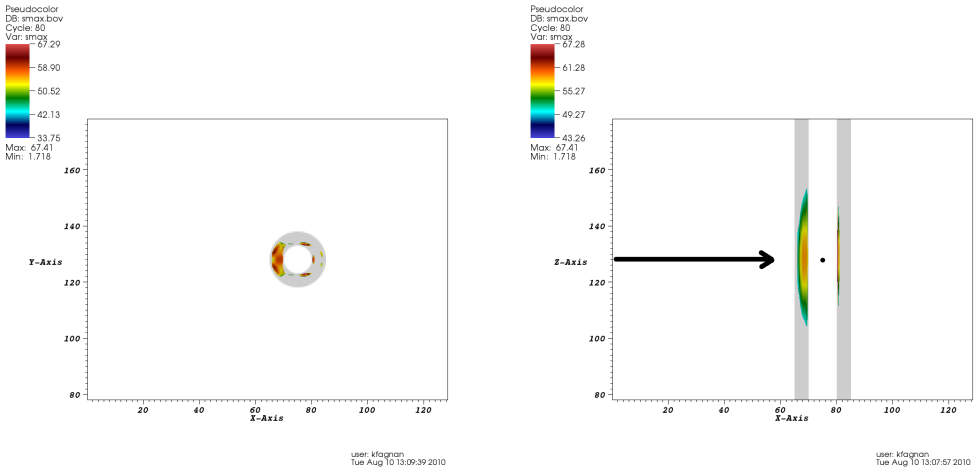


Figure 11. An additional interface calculation showing the more realistic treatment of a cylinder with a fluid-filled cavity. Left: a slice along $z = 0$ showing the concentration of stress in the front of the idealized bone, with additional smaller pockets of maximum stress due to reflection in the back half of the bone. Right: a slice along $y = 0$ which further demonstrates the stress concentration in the first half of the bone. The arrow in the figure at right indicates the direction of ESWT wave propagation and the dot indicates the location of F2.

important in treatment optimization. For example, in order to maximize shear stress at the tissue-bone interface, our preliminary computations indicate that it might be best to treat the patient at an oblique angle. However, if the goal is to maximize shear stress in the gap of the broken bone, then treating the patient at a 90-degree angle may be better than treating at either the 45- or 60-degree angle. We stress however that the biological mechanisms must be better understood and more experiments must be done in conjunction with laboratory and clinical treatments before these calculations could be used to make specific clinical recommendations.

In Figure 11 we show two-dimensional slices of a calculation with a more realistic, but still idealized, long bone geometry. In a the shaft of a long bone, there is a marrow-filled canal running through the center. Marrow is typically modeled as a viscoelastic material [37], but for this first approximation we just used a fluid-filled canal. The impedance difference in the two materials is similar and therefore illustrates the behavior that we are interested in, the change in maximal stress deposition. In contrast to the solid cylinder above, the contours of maximal stress are concentrated in the front side of the hollow cylinder. Figure 11, right, shows that there are also two regions of additional stress concentration in the backside of the hollow cylinder. This example highlights the importance in understanding where these impedance jumps occur in order to optimally treat the patient.

5.4. Heterotopic ossification. A heterotopic ossification (HO) is a growth of bone-like material in soft tissue. HOs often grow spontaneously in tissue that has been traumatized due to injury or amputation. An example of an HO is shown in [Figure 12](#), left, which shows the pelvis and an HO, using data extracted from a patient's CT scan. In this case, the HO has grown around the right hip joint and is inhibiting the patient's range of motion. The goal of the HO treatment is not to pulverize the ossification, but to break up the adhesion between the HO and the joint, in order to restore the patient's range of motion. There is no clear division between the HO and bone in the CT scan because both are composed of materials that have similar densities. However, the HO does not have the same woven structure that is present in bone, so the two will likely have different material properties, even though the densities are similar. This similarity means that we are uncertain as to how strong the connection or adhesion is between the HO and the bone, which will directly impact the number of shocks needed to restore the patient's range of motion.

We are able to use our model to investigate the effect of the angle of treatment on the observed stresses in the region near the HO. Since the composition and material properties of the ossification are not well understood, we can also use the model to vary the material properties of the ossification and investigate the sensitivity of the results to these parameters. We found that both the strength of the connection between the HO and bone, as well as the composition of the HO, had a significant effect on the location of maximum stress in the object [18].

It is challenging to infer anything meaningful from the images in the full three-dimensional calculation in [Figure 12](#), left, so we have also included a two-dimensional slice of the maximum shear in [Figure 12](#), right. Here the gray regions

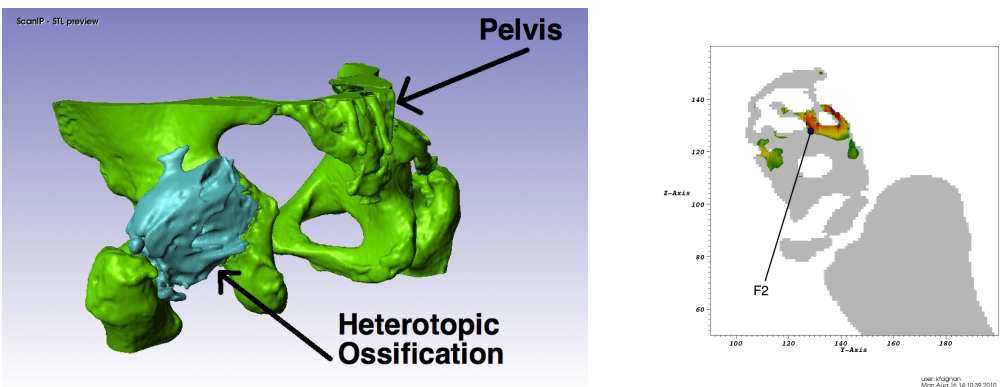


Figure 12. Left: the three-dimensional CT patient data illustrating the heterotopic ossification (blue) attached to the right hip joint (green). Right: a slice at $x = 115$ of the two-dimensional calculation shows how the pockets of fluid lead to stress concentration in the substructure of the ossification, the dot indicates the location of F2 and the direction of treatment is into the page.

represent the bone-like HO material and we assume any gaps are filled with fluid. It is clear that the interior of the ossification is complex and contains many fluid-filled pockets that affect, in this case, the location of the maximum shear stress.

Given the complex nature of the HO and subsequent difficulty interpreting the three-dimensional results, we have also used an idealized ossification to investigate some facets of the shockwave interaction with the varying material properties. One example of this is shown in [Figure 13](#), where we have simulated a case where the ossification (the crescent in the two-dimensional images) is not strongly attached to the bone (the cylinder) and calculated the maximal shear stress as a result of two different treatment angles. [Figure 13](#), left, illustrates the result when the ESWT device is aimed orthogonal to the gap between the HO and the cylinder. [Figure 13](#), right, is the result when the device is aimed so the shockwaves propagate parallel to the gap between the HO and bone. It has been indicated that maximum shear stress is important in causing the HO to break, so it is desirable to deposit the maximum amount of shear as close to the HO/bone interface as possible. In this case, it is better to treat the HO in the direction indicated in [Figure 13](#), right, since the shear stress is concentrated along the gap.

According to our computational results, the pockets of fluid within an HO and strength of adhesion to the bone surface will affect the stress deposition and therefore the location of the eventual break in the ossification. Further investigation is required to be conclusive, but our results indicate that if the fluid pockets are in the propagation path of the shock wave, they may cause the maximum stresses to occur away from the adhesion site, making the treatment less effective. In reality, the composition of the HO is unknown and we do not have a good characterization for the material properties of the ossifications. However, the strong impedance mismatch between fluid and bone, as well as the inability of the fluid to support shear stress, indicate that the presence of fluid-filled pockets will have an effect on the stress deposition. We should note here that our modeling work does not take into account the propagation of successive shocks or failure models within the material, which should ultimately be incorporated in order to determine the optimal treatment. This is an area for future work.

6. Conclusion

In this paper we have proposed a new model for ESWT. We have demonstrated that the Tait equation of state is sufficient for the pressures that arise in ESWT, that enables us to drop the energy equation from our model. We have shown that the fluid and solid can be modeled with the same set of Lagrangian equations since the particle displacements are small. This approach allowed us to utilize existing numerical methodology, consisting of high-resolution shock-capturing methods

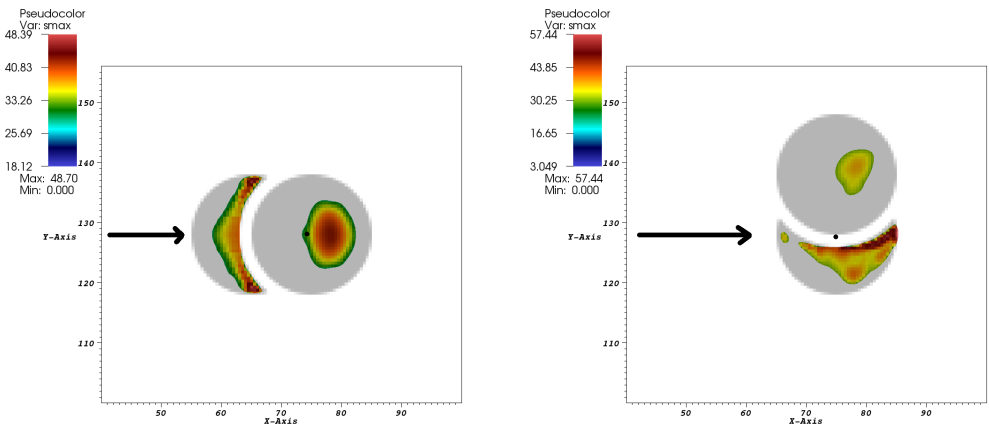


Figure 13. Calculations for an idealized ossification that demonstrates the difference in shear stress deposition when treating the HO from different directions. Since the goal is to disrupt the HO at the interface between the bone and the HO, the figure on the right indicates that it would be optimal send shock waves parallel to the break, instead of perpendicular to it. The arrows indicate the direction of ESWT propagation and the dots indicate the location of F2.

together with adaptive mesh refinement, to efficiently calculate solutions to these equations for a variety of idealized biological problems. We have also demonstrated that we can effectively handle interfaces between different materials on Cartesian grids. Using this methodology we were able to explore, even in geometrically complicated structures, how the interfaces between the fluid and solid materials affect the distribution of maximal stress in several problems of clinical interest. We should note that the models for the biological materials are idealized, so it is difficult to extrapolate from these experiments to reality without conducting further experiments.

Maximizing stress in specific regions seems important in both the healing and destruction of biological tissues. Shear stress is thought to play a role in the stimulation of biological tissues [54; 25; 20; 23; 13; 30; 43]. Mechanical loading is thought to play a role in the formation of bone tissue, and as discussed in Section 1, shear and compressive displacements generated by loading influence bone healing [43; 44; 30; 13; 27; 11; 23]. Shear stress is also important in predicting the break up of kidney stones [49].

The model we have developed has been used to investigate idealized nonunions and heterotopic ossifications, and we have shown a few examples to illustrate this. A broader range of calculations are available in [18].

The focus of this paper has been the effect that material interfaces between tissue and bone have on the transmission, reflection, and focusing of the shock wave. Very simple models have been used for the material on each side of the interface: compressible fluid with a Tait equation of state in the tissue and linear isotropic elasticity in the bone. We believe that this level of macroscopic modeling can already reveal interesting features of the stress that may be clinically important. In particular, focusing may occur in regions displaced from where it would be observed in pure water, and mode conversion at an interface can generate shear waves in the bone that are not present in the focusing shock wave in fluid.

To consider the effect of stress on individual osteocytes, a much more detailed model would be necessary that is beyond the scope of this work. In particular, this would require modeling the microscale fluid-filled canaliculi within the bone through which the osteocyte processes extend. Work is currently underway in this direction, and also on intermediate levels of modeling in which the bone is modeled as an orthotropic poroelastic material. These equations can be solved with essentially the same high resolution finite volume methods used here, after implementing a more complicated Riemann solver [32; 33], and with the same software for adaptive mesh refinement. Another possible extension is to investigate viscoelastic tissue models that may be superior to the Tait equation for water that is currently used.

Acknowledgements

This work was supported in part by NIH Grant 5R01AR53652-2, NSF Grants DMS-0609661 and DMS-0914942, and the Founders Term Professorship in Applied Mathematics at the University of Washington. The authors would like to thank Donna Calhoun and the ANAG group at Lawrence Berkeley Laboratory for their assistance with the ChomboClaw calculations. This research was supported in part by the National Science Foundation through TeraGrid resources provided by TACC under grant number TG-DMS090036T.

References

- [1] *Chombo: software for adaptive solutions of partial differential equations*, webpage, Applied Numerical Algorithms Group (ANAG), Lawrence Berkeley National Laboratory, 2009.
- [2] P. Augat, J. Merk, S. Wolf, and L. E. Claes, *Mechanical stimulation by external application of cyclic tensile strains does not effectively enhance bone healing*, *Journal of Orthopaedic Trauma* **15** (2001), 54–60.
- [3] M. Averkiou and R. Cleveland, *Modeling of an electrohydraulic lithotripter with the KZK equation*, *Journal for the Acoustical Society of America* **106** (1999), no. 1, 102–112.

- [4] D. S. Bale, R. J. Leveque, S. Mitran, and J. A. Rossmannith, *A wave propagation method for conservation laws and balance laws with spatially varying flux functions*, SIAM J. Sci. Comput. **24** (2002), no. 3, 955–978. MR 2004a:65098
- [5] M. J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, J. Comput. Phys. **82** (1989), 64–84.
- [6] M. J. Berger and I. Rigoutsos, *An algorithm for point clustering and grid generation*, IEEE Trans. Sys. Man & Cyber. **21** (1991), 1278–1286.
- [7] M. J. Berger and R. J. Leveque, *Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems*, SIAM J. Numer. Anal. **35** (1998), no. 6, 2298–2316. MR 2000c:65082
- [8] M. J. Berger and J. Olinger, *Adaptive mesh refinement for hyperbolic partial differential equations*, J. Comput. Phys. **53** (1984), no. 3, 484–512. MR 85h:65211
- [9] R. Biedermann, A. Martin, G. Handle, T. Auckenthaler, C. Bach, and M. Krismer, *Extracorporeal shock waves in the treatment of nonunions*, Journal of Trauma **54** (2003), no. 5, 936–42.
- [10] D. A. Calhoun, P. Colella, and R. J. LeVeque, *CHOMBO-CLAW software*, web site.
- [11] D. R. Carter, G. S. Beaupre, N. J. Giori, and J. A. Helms, *Mechanobiology of skeletal regeneration*, Clinical Orthopaedics and Related Research **355** (Suppl) (1998), S41–55.
- [12] T. Christopher, *Modeling the Dornier HM3 lithotripter*, Journal of the Acoustical Society of America **96** (1994), no. 5, 3088–3095.
- [13] L. E. Claes and C. A. Heigele, *Magnitudes of local stress and strain along osseous surfaces predict the course and type of fracture-healing*, The Journal of Biomechanics **32** (1999), 255–266.
- [14] L. E. Claes, H. J. Wilke, P. Augat, S. Rubenacker, and K. J. Margevicius, *Effect of dynamization on gap healing of diaphyseal fractures under external fixation*, Clinical Biomechanics **10** (1995), 227–234.
- [15] Clawpack Team, *CLAWPACK software*, web page, 2013.
- [16] R. O. Cleveland and O. Sapozhnikov, *Modeling elastic wave propagation in kidney stones with application to shock wave lithotripsy*, Journal of the Acoustical Society of America **118** (2005), no. 4, 2667–2676.
- [17] A. J. Coleman, J. Saunders, R. Preston, and D. Bacon, *Pressure waveforms generated by a Dornier extra-corporeal shock-wave lithotripter*, Ultrasound in Medicine and Biology **13** (1987), 651–657.
- [18] K. M. Fagnan, *High-resolution finite volume methods for extracorporeal shock wave therapy*, Ph.D. thesis, University of Washington, 2010. MR 2941302
- [19] K. Fagnan, R. J. LeVeque, T. J. Matula, and B. MacConaghy, *High-resolution finite volume methods for extracorporeal shock wave therapy*, Hyperbolic problems: theory, numerics, applications (S. Benzoni-Gavage and D. Serre, eds.), Springer, Berlin, 2008, pp. 503–510. MR 2549183
- [20] J. Freund, T. Colonius, and A. Evan, *A cumulative shear mechanism for tissue damage initiation in shock-wave lithotripsy*, Ultrasound in Medicine and Biology **33** (2007), 1495–1503.
- [21] Y. C. Fung, *Biomechanics: mechanical properties of living tissues*, Springer, 1993.
- [22] E. Garner, R. Lakes, T. Lee, C. Swan, and R. Brand, *Viscoelastic dissipation in compact bone: implications for stress-induced fluid flow in bone*, Journal of Biomechanical Engineering **122** (2000), 166–73.

- [23] A. E. Goodship and J. Kenwright, *The influence of induced micromovement on the healing of experimental tibial fractures*, *Journal of Bone and Joint Surgery British Volume* **67** (1985), 650–655.
- [24] M. Hamilton, *Transient axial solution for the reflection of a spherical wave from a concave ellipsoidal mirror*, *Journal of the Acoustical Society of America* **93** (1993), no. 3, 1256–1266.
- [25] M. V. Hillsley and J. A. Frangos, *Review of “Bone tissue engineering: the role of interstitial fluid flow”*, *Biotechnology and Bioengineering* **43** (1994), 573–581.
- [26] C. Huang and R. Ogawa, *Mechanotransduction in bone repair and regeneration*, *The FASEB Journal* **23** (2010), 3625–3632.
- [27] H. Isaksson, W. Wilson, C. C. van Donkelaar, R. Huiskes, and K. Ito, *Comparison of biophysical stimuli for mechano-regulation of tissue differentiation during fracture-healing*, *The Journal of Biomechanics* **39** (2006), 1507–1516.
- [28] M. J. Ivings, D. M. Causon, and E. F. Toro, *On Riemann solvers for compressible liquids*, *Internat. J. Numer. Methods Fluids* **28** (1998), no. 3, 395–418. [MR 99e:76088](#)
- [29] T. Keaveny, X. E. Guo, E. F. Wachtel, T. A. McMahon, and W. C. Hayes, *Trabecular bone exhibits fully linear elastic behavior and yields at low strains*, *Journal of Biomechanics* **27** (1994), 1127–1129, 1131–1136.
- [30] D. Lacroix and P. J. Prendergast, *A mechano-regulation model for tissue differentiation during fracture-healing: analysis of gap size and loading*, *The Journal of Biomechanics* **35** (2002), 1163–1171.
- [31] J. O. Langseth and R. J. LeVeque, *A wave propagation method for three-dimensional hyperbolic conservation laws*, *J. Comput. Phys.* **165** (2000), no. 1, 126–166. [MR 2001i:65110](#)
- [32] G. I. Lemoine, *Numerical modeling of poroelastic-fluid systems using high-resolution finite volume methods*, Ph.D. thesis, University of Washington, 2013.
- [33] G. I. Lemoine, M. Y. Ou, and R. J. LeVeque, *High-resolution finite volume modeling of wave propagation in orthotropic poroelastic media*, *SIAM J. Sci. Comput.* **35** (2013), no. 1, B176–B206. [MR 3033065](#)
- [34] R. J. LeVeque, *Wave propagation algorithms for multi-dimensional hyperbolic systems*, *J. Comput. Phys.* **131** (1997), 327–353.
- [35] ———, *Finite volume methods for hyperbolic problems*, *Cambridge Texts in Applied Mathematics*, no. 31, Cambridge University Press, 2002. [MR 2003h:65001](#)
- [36] ———, *Finite-volume methods for non-linear elasticity in heterogeneous media*, *Internat. J. Numer. Methods Fluids* **40** (2002), no. 1-2, 93–104. [MR 2003h:74078](#)
- [37] R. B. Martin, D. B. Burr, and N. A. Sharkey, *Skeletal tissue mechanics*, Springer, New York, 1998.
- [38] T. J. Matula, P. R. Hilmo, and M. R. Bailey, *A suppressor to prevent direct wave-induced cavitation in shock wave therapy devices*, *Journal of the Acoustical Society of America* **118** (2005), no. 1, 178–185.
- [39] E. F. Morgan, R. E. Gleason, L. N. M. Hayward, P. L. Leong, and K. T. Salisbury-Paolomares, *Mechanotransduction and fracture repair*, *Journal of Bone and Joint Surgery American Volume* **90** (Suppl 1) (2008), 25–30.
- [40] G. Mouzopoulos, M. Stamatakos, D. Mouzopoulos, and M. Tzurbakis, *Extracorporeal shock wave treatment for shoulder calcific tendonitis: a systematic review*, *Skeletal Radiology* **36** (2008), no. 9, 803–811.

- [41] M. Nakahara, K. Nagayama, and Y. Mori, *Shockwave dynamics of high pressure pulse in water and other biological materials based on Hugoniot data*, Japanese Journal of Applied Physics **47** (2008), 3510.
- [42] J. A. Ogden, A. Toth-Kischkat, and R. Schultheiss, *Principles of shock wave therapy*, Clinical Orthopaedics and Related Research **387** (2001), 8–17.
- [43] S. H. Park, K. O'Connor, H. McKellop, and A. Sarmiento, *The influence of active shear or compressive motion on fracture-healing*, The Journal of Bone and Joint Surgery American Volume **80** (1998), 868–878.
- [44] P. J. Prendergast, R. Huiskes, and K. Soballe, *Biophysical stimuli on cells during tissue differentiation at implant interfaces*, The Journal of Biomechanics **30** (1997), 539–548.
- [45] A. G. Robling, F. M. Hinant, D. B. Burr, and C. H. Turner, *Improved bone structure and strength after long-term mechanical loading is greatest if loading is separated into short bouts*, Journal of Bone Mineral Research **17** (2002), 1545–1554.
- [46] ———, *Shorter, more frequent mechanical loading sessions enhance bone mass*, Medicine and Science in Sports and Exercise **34** (2002), 196–202.
- [47] T. Saito, M. Marumoto, H. Yamashita, S. H. R. Hosseini, A. Nakagawa, T. Hirano, and K. Takayama, *Experimental and numerical studies of underwater shock wave attenuation*, Shock Waves **13** (2003), 139–148.
- [48] O. Sapozhnikov, M. Bailey, and R. O. Cleveland, *The role of shear and longitudinal waves in the kidney stone comminution by a lithotripter shock pulse*, Journal of the Acoustical Society of America **115** (2004), 2562.
- [49] O. Sapozhnikov, A. D. Maxwell, B. MacConaghy, and M. Bailey, *A mechanistic analysis of stone fracture in lithotripsy*, Journal of the Acoustical Society of America **121** (2007), no. 2, 1190–1202.
- [50] L. Saxon, A. Robling, I. Alam, and C. Turner, *Mechanosensitivity of the rat skeleton decreases after a long period of loading, but is improved with time off*, Bone **36** (2005), 454–464.
- [51] M. Tanguay, *Computation of bubbly cavitating flow in shock wave lithotripsy*, Ph.D. thesis, California Institute of Technology, 2004.
- [52] W. R. Taylor, E. Roland, H. Ploeg, D. Hertig, R. Klabunde, M. D. Warner, M. C. Hobatho, L. Rakotomanana, and S. E. Clift, *Determination of orthotropic bone elastic constants using FEA and modal analysis*, Journal of Biomechanics **35** (2002), 767–773.
- [53] C. H. Turner and F. M. Pavalko, *Mechanotransduction and functional response of the skeleton to physical stress: the mechanisms and mechanics of bone adaptation*, Journal of Orthopaedic Science **3** (1998), 346–355.
- [54] C.-J. Wang, K.-E. Huang, Y.-C. Sun, Y.-J. Yang, J.-Y. Ko, L.-H. Weng, and F.-S. Wang, *VEGF modulates angiogenesis and osteogenesis in shockwave-promoted fracture healing in rabbits*, Journal of Surgical Research **171** (2011), no. 1, 114–119.
- [55] C.-J. Wang, F.-S. Wang, J.-Y. Ko, H.-Y. Huang, C.-J. Chen, Y.-C. Sun, and Y.-J. Yang, *Extracorporeal shockwave therapy shows regeneration in hip necrosis*, Rheumatology **47** (2008), no. 4, 542–546.
- [56] S. Weinbaum, S. Cowin, and Y. Zeng, *A model for the excitation of osteocytes by mechanical loading-induced bone fluid shear stress*, Journal of Biomechanics **27** (1994), 339–360.

KIRSTEN FAGNAN: kmfagnan@lbl.gov

*NERSC/JGI, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, MS 943R0256,
Berkeley, CA 94720, United States*

RANDALL J. LEVEQUE: rjl@amath.washington.edu

*Department of Applied Mathematics, University of Washington, Box 352420,
Seattle, WA 98195-2420, United States*

THOMAS J. MATULA: matula@apl.washington.edu

*Applied Physics Laboratory, University of Washington, 1013 NE 40th Street, Box 355640,
Seattle, WA 98105-6698, United States*

Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.berkeley.edu/camcos.

Originality. Submission of a manuscript acknowledges that the manuscript is original and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

Language. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

Required items. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

Format. Authors are encouraged to use L^AT_EX but submissions in other varieties of T_EX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

References. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibT_EX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

Figures. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

White space. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

Proofs. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

Communications in Applied Mathematics and Computational Science

vol. 8

no. 1

2013

- On the origin of divergence errors in MHD simulations and consequences for numerical schemes 1
FRIEDEMANN KEMM
- Renormalized reduced models for singular PDEs 39
PANOS STINIS
- Legendre spectral-collocation method for Volterra integral differential equations with nonvanishing delay 67
YANPING CHEN and ZHENDONG GU
- A cartesian grid embedded boundary method for the compressible Navier–Stokes equations 99
DANIEL T. GRAVES, PHILLIP COLELLA, DAVID MODIANO, JEFFREY JOHNSON, BJORN SJOGREEN and XINFENG GAO
- Second-order accuracy of volume-of-fluid interface reconstruction algorithms II: An improved constraint on the cell size 123
ELBRIDGE GERRY PUCKETT
- Computational models of material interfaces for the study of extracorporeal shock wave therapy 159
KIRSTEN FAGNAN, RANDALL J. LEVEQUE and THOMAS J. MATULA