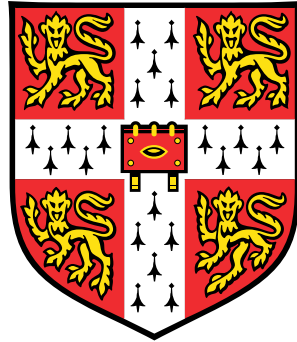


A Unified Framework for Simulating Impact-Induced Detonation of a Combustible Material in an Elasto-Plastic Confiner



Haran Jackson

The Centre for Scientific Computing

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy in Scientific Computing

Fitzwilliam College

Supervisor: Dr N. Nikiforakis

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text.

Word Count (including tables, figure legends, and appendices):

Haran Jackson
January 4th 2018

Acknowledgements

I would like to thank my supervisor, Dr Nikos Nikiforakis, for all the help and guidance he has given me. Additionally, I would like to thank Dr Louisa Michael, Geraint Harcombe, Tomé Gouveia, Knut Sverdrup, and Bruno Dog for the many useful discussions we had.

Research Output

The following materials were produced over the course of my PhD.

Publications

- H Jackson, G Harcombe, N Nikiforakis, *A solver based on eigendecomposition of the Cauchy tensor for Godunov-Romenski-type continuum models* (in preparation)
- H Jackson, N Nikiforakis, *A Riemann ghost fluid method for modeling multimaterial interfaces with the GPR model* (in preparation)
- H Jackson, *The Montecinos-Balsara ADER-FV polynomial basis: convergence properties & extension to non-conservative multidimensional systems* (Computers & Fluids, 2018)
- H Jackson, *A fast numerical scheme for the Godunov-Peshkov-Romenski model of continuum mechanics* (Journal of Computational Physics, 2017)
- H Jackson, *On the eigenvalues of the ADER-WENO Galerkin predictor* (Journal of Computational Physics, 2017)

Conference Presentations

- International Conference on Computational Science 2017 (Zürich, CH), *Paper: A Fast Numerical Scheme for the Godunov-Peshkov-Romenski Model of Continuum Mechanics*
- SIAM International Conference on Numerical Combustion 2017 (Orlando, FL), *Minisymposium: A New Approach for Cook-off Modeling*
- Scientific Computation in the University of Cambridge Seminar Day 2017 (Cambridge, UK), *Poster: A Numerical Method based on Operator Splitting for the GPR Model of Continuum Mechanics*
- Cavendish Graduate Student Conference 2016 (Cambridge, UK), *Poster: A New Framework for Simulating Multimaterial Systems and Gaseous Cookoff*

Open-Source Software

- ADER-WENO (github.com/haranjackson/ADER-WENO): A Python implementation of the ADER-WENO method for solving hyperbolic systems of PDEs
- Julia-WENO (github.com/haranjackson/Julia-WENO): An optimized Julia implementation of the WENO reconstruction algorithm, of any order of accuracy
- Euler1D (github.com/haranjackson/Euler1D): A few first- and second-order methods for solving the 1D Euler equations, implemented in C++
- ProjectionMethod (github.com/haranjackson/ProjectionMethod): A C++ implementation of Chorin's Project Method
- NewtonKrylov (github.com/haranjackson/NewtonKrylov): A C++ implementation of the Newton-Krylov algorithm, with Python bindings
- LGMRES (github.com/haranjackson/LGMRES): A C++ implementation of the LGMRES algorithm, with Python bindings
- LegendreGauss (github.com/haranjackson/LegendreGauss): C++ code to compute the Legendre-Gauss nodes and weights on $[-1,1]$, based on NumPy's `leggauss` function

Abstract

Contents

Contents	vi
List of Figures	viii
List of Tables	ix
Nomenclature	xi
0 Introduction	1
0.1 Background	1
0.2 Objectives of this Study	1
0.3 Mathematical Model	1
0.4 Numerical Methods	4
1 Objective 1: Extending the GPR Model	13
1.1 Equations of State	13
1.2 Multiple Materials	16
1.3 Reactive Materials	16
2 Objective 2: Improved Numerical Methods	17
2.1 Convergence Properties of the Standard ADER-WENO Method	17
2.2 Extension of the Montecinos-Balsara ADER-WENO Method	20
2.3 An Operator Splitting Scheme for the GPR Model	29
2.4 An Operator Splitting Scheme for GPR-type Continuum Models	51

3	Objective 3: Simulating Material Interfaces	53
3.1	Ghost Fluid Methods	53
3.2	A Riemann Ghost Fluid Method for the GPR Model	53
3.3	Results	53
4	Impact-Induced Detonation in an Elasto-Plastic Confiner	54
4.1	Results	54
5	Conclusions	55
	References	56
A	System Matrices	59
B	Primitive System	63
C	Matrices used in the Numerical Methods	71
C.1	WENO Matrices for $N = 2$	71
D	Jacobians	72
D.1	Jacobian of Distortion ODEs	72
D.2	Jacobian of Thermal Impulse ODEs	72

List of Figures

2.1	The (shaded) region to which x_i, x_j are confined in the evolution of the distortion ODEs	36
2.2	The components of the distortion tensor in the Strain Relaxation Test	41
2.3	The singular values of the distortion tensor and the energy in the Strain Relaxation Test	41
2.4	The components of the stress tensor in the Strain Relaxation Test	41
2.5	Results of solving Stokes' First Problem ($\mu = 10^{-2}, \mu = 10^{-3}, \mu = 10^{-4}$) with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)	43
2.6	Density, velocity, and pressure for the Viscous Shock problem, solved with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)	46
2.7	Viscous stress and heat flux for the Viscous Shock problem, solved with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)	47
2.8	Results of solving the problem of Heat Conduction in Gas with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)	47

List of Tables

1.1	e_{ref}, p_{ref}, Γ for different kinds of Mie-Gruneisen equations of state	15
2.1	Initial conditions for the slow opposing shear flow test	42
2.2	Initial conditions for the heat conduction test	45
2.3	Wall time for various tests (all with 200 cells) under the ADER-WENO method and the Split-WENO method	50
2.4	Time steps taken for various tests (all with 200 cells) under the ADER-WENO method and the Split-WENO method	50
2.5	Convergence rates for the Split-WENO method ($N = 2$)	50
2.6	Convergence rates for the Split-WENO method ($N = 3$)	50
2.7	Convergence rates for the ADER-DG PNPM method ($N, M = 2$)	52

Nomenclature

Roman Symbols

\boldsymbol{J}	Thermal impulse vector
\boldsymbol{q}	Heat flux vector
\boldsymbol{v}	Velocity
\boldsymbol{A}	Distortion tensor
c_0	Adiabatic speed of sound
c_h	Characteristic velocity of heat waves
c_p	Specific heat capacity at constant pressure
c_s	Characteristic velocity of transverse perturbations
c_v	Specific heat capacity at constant volume
E	Total specific energy
p	Pressure
p_∞	Pressure constant in stiffened gas equation of state
s	Entropy
T	Temperature
t	Time variable
x	Space variable

Greek Symbols

α	Constant related to characteristic velocity of heat waves
δ	Kronecker delta

γ	Ratio of specific heat capacities, equal to $\frac{c_p}{c_v}$
ρ	Density
ρ_0	Reference density
σ	Viscous shear stress tensor
τ_1	Strain dissipation time
τ_2	Thermal impulse dissipation time

Other Symbols

$\ \cdot\ $	Euclidean vector norm
$\ \cdot\ _F$	Frobenius matrix norm

Acronyms / Abbreviations

DG	Discontinuous Galerkin
EOS	Equation of State
FV	Finite Volume
WENO	Weighted Essentially Non-Oscillatory

Notes

Unless otherwise stated, repeated indices in vector, matrix and tensor quantities are to be summed over. If M is a matrix, then M_i is taken to be the i th column of M (note, not the i th row). MATLAB-style index notation is used, such that $M_{i:j}$ refers to the matrix consisting of the columns $i \dots j$ of M (including columns i and j). $M_{i:j,m:n}$ refers to the submatrix of M with corners at M_{im} and M_{jn} .

Chapter 0

Introduction

0.1 Background

0.2 Objectives of this Study

0.3 Mathematical Model

The GPR model, first introduced in Peshkov and Romenski [18] and expanded upon in Dumbser et al. [?], takes the following form:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_k)}{\partial x_k} = 0 \quad (1a)$$

$$\frac{\partial (\rho v_i)}{\partial t} + \frac{\partial (\rho v_i v_k + p \delta_{ik} - \sigma_{ik})}{\partial x_k} = 0 \quad (1b)$$

$$\frac{\partial A_{ij}}{\partial t} + \frac{\partial (A_{ik} v_k)}{\partial x_j} + v_k \left(\frac{\partial A_{ij}}{\partial x_k} - \frac{\partial A_{ik}}{\partial x_j} \right) = -\frac{\psi_{ij}}{\theta_1} \quad (1c)$$

$$\frac{\partial (\rho J_i)}{\partial t} + \frac{\partial (\rho J_i v_k + T \delta_{ik})}{\partial x_k} = -\frac{\rho H_i}{\theta_2} \quad (1d)$$

$$\frac{\partial (\rho s)}{\partial t} + \frac{\partial (\rho s v_k + H_k)}{\partial x_k} = \frac{\rho}{T} \left(\frac{\psi_{kl} \psi_{kl}}{\theta_1} + \frac{H_k H_k}{\theta_2} \right) \quad (1e)$$

where θ_1 and θ_2 are positive scalar functions, and $\psi = \frac{\partial E}{\partial A}$ and $\mathbf{H} = \frac{\partial E}{\partial J}$. Entropy does not decrease during the dissipative time evolution:

$$\frac{\partial (\rho s)}{\partial t} + \frac{\partial (\rho s v_k + H_k)}{\partial x_k} \geq 0 \quad (2)$$

(1e) can be replaced with the following equation, which will be used instead when solving the model in this study:

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho E v_k + (p \delta_{ik} - \sigma_{ik}) v_i + q_k)}{\partial x_k} = 0 \quad (3)$$

Note that (1a), (1b), (1c), (1d), (3) can be written in the following form:

$$\frac{\partial Q}{\partial t} + \nabla \cdot \mathbf{F} + \mathbf{B} \cdot \nabla Q = S \quad (4)$$

The following definitions are given:

$$p = \rho^2 \left. \frac{\partial E}{\partial \rho} \right|_{s,A} \quad (5a)$$

$$\sigma = -\rho A^T \left. \frac{\partial E}{\partial A} \right|_{\rho,s} \quad (5b)$$

$$T = \left. \frac{\partial E}{\partial s} \right|_{\rho,A} \quad (5c)$$

$$\mathbf{q} = T \frac{\partial E}{\partial \mathbf{J}} \quad (5d)$$

To close the system, the EOS must be specified, from which the above quantities and the sources can be derived. E is the sum of the contributions of the energies at the molecular scale (microscale), the material element¹ scale (mesoscale), and the flow scale (macroscale):

$$E = E_1(\rho, s) + E_2(\rho, s, A, \mathbf{J}) + E_3(\mathbf{v}) \quad (6)$$

In previous studies, E_1 has to been taken to be either the ideal gas EOS, the stiffened gas EOS, or the shock Mie-Gruneisen EOS. A more general set of choices for E_1 is given in Section 1.1.

E_2 has the following quadratic form:

$$E_2 = \frac{c_s(\rho, s)^2}{4} \|\text{dev}(G)\|_F^2 + \frac{\alpha(\rho, s)^2}{2} \|\mathbf{J}\|^2 \quad (7)$$

c_s is the characteristic velocity of transverse perturbations. In previous studies, c_s has always been constant. In this study, it will be extended to have a ρ dependence, as outlined in Section 1.1. α is a constant related to the characteristic velocity of propagation of heat waves:

$$c_h = \frac{\alpha}{\rho} \sqrt{\frac{T}{c_v}} \quad (8)$$

¹The concept of a *material element* corresponds to that of a fluid parcel from fluid dynamics, applied to both fluids and solids.

0.3 Mathematical Model

In previous studies, α has been taken to be constant, as it will in this study.

$G = A^T A$ is the Gramian matrix of the distortion tensor, and $\text{dev}(G)$ is the deviator (trace-free part) of G :

$$\text{dev}(G) = G - \frac{1}{3} \text{tr}(G) I \quad (9)$$

E_3 is the usual specific kinetic energy per unit mass:

$$E_3 = \frac{1}{2} \|\mathbf{v}\|^2 \quad (10)$$

The following forms are taken:

$$\theta_1 = \frac{\tau_1 c_s^2}{3 |A|^{\frac{5}{3}}} \quad (11a)$$

$$\theta_2 = \tau_2 \alpha^2 \frac{\rho T_0}{\rho_0 T} \quad (11b)$$

$$\tau_1 = \frac{6\mu}{\rho_0 c_s^2} \quad (12a)$$

$$\tau_2 = \frac{\rho_0 \kappa}{T_0 \alpha^2} \quad (12b)$$

The justification of these choices is that classical Navier–Stokes–Fourier theory is recovered in the stiff limit $\tau_1, \tau_2 \rightarrow 0$ (see [6]). This results in the following relations:

$$\sigma = -\rho c_s^2 G \text{dev}(G) \quad (13a)$$

$$\mathbf{q} = \alpha^2 T \mathbf{J} \quad (13b)$$

$$-\frac{\psi}{\theta_1(\tau_1)} = -\frac{3}{\tau_1} |A|^{\frac{5}{3}} A \text{dev}(G) \quad (13c)$$

$$-\frac{\rho \mathbf{H}}{\theta_2(\tau_2)} = -\frac{T \rho_0}{T_0 \tau_2} \mathbf{J} \quad (13d)$$

The following constraint also holds (see [18]):

$$\det(A) = \frac{\rho}{\rho_0} \quad (14)$$

The GPR model and Godunov and Romenski's 1970s model of elastoplastic deformation in fact rely upon the same equations. The realization of Peshkov and Romenski was that these are the equations of motion for an arbitrary continuum - not just a solid - and so the model can be applied to fluids too. Unlike in previous continuum models, material elements have not only finite size, but also internal structure, encoded in the distortion tensor.

The strain dissipation time τ_1 of the HPR model is a continuous analogue of Frenkel's "particle settled life time" [?]; the characteristic time taken for a particle to move by a distance of the same order of magnitude as the particle's size. Thus, τ_1 characterizes the time taken for a material element to rearrange with its neighbors. $\tau_1 = \infty$ for solids and $\tau_1 = 0$ for inviscid fluids. It is in this way that the HPR model seeks to describe all three major phases of matter, as long as a continuum description is appropriate for the material at hand.

The evolution equation for \mathbf{J} and its contribution to the energy of the system are derived from Romenski's model of hyperbolic heat transfer, originally proposed in [14, 21], and implemented in [19, 20]. In this model, \mathbf{J} is effectively defined as the variable conjugate to the entropy flux, in the sense that the latter is the derivative of the specific internal energy with respect to \mathbf{J} . Romenski remarks that it is more convenient to evolve \mathbf{J} and E than the heat flux or the entropy flux, and thus the equations take the form given here. τ_2 characterizes the speed of relaxation of the thermal impulse due to heat exchange between material elements.

0.4 Numerical Methods

The GPR model, being non-conservative, with stiff source terms, represents a particularly challenging set of PDEs. In this study they are solved by an ADER-WENO method. First, the cell-wise constant state variable data from the current time step is reconstructed using high-order spatial polynomials according to the WENO method. This reconstruction is then extended to a reconstruction in both space and time for each individual cell in the domain, using the Discontinuous Galerkin method. A finite volume solver is then used to couple neighboring cells and produce the cell-wise constant data at the next time step.

0.4.1 The WENO Reconstruction

First introduced by Liu et al. [?] and developed by Jiang and Shu [?], WENO methods are used to produce high order polynomial approximations to piece-wise constant data. Many variations exist. In this study, the method of [9] is used.

Consider the domain $[0, L]$. Take $K, N \in \mathbb{N}$. The order of accuracy of the resulting method will be $N + 1$. Take the set of grid points $x_i = \frac{iL}{K}$ for $i = 0, \dots, K$ and let $\Delta x = \frac{L}{K}$. Denote cell $[x_i, x_{i+1}]$ by C_i . Given cell-wise constant data u on $[0, L]$, an order N polynomial reconstruction of u in C_i will be performed. Define the scaled space variable:

$$\chi^i = \frac{1}{\Delta x} (x - x_i) \quad (15)$$

Denoting the Gauss-Legendre abscissae on $[0, 1]$ by $\{\chi_0, \dots, \chi_N\}$, define the nodal basis of order N : the Lagrange interpolating polynomials $\{\psi_0, \dots, \psi_N\}$ with the following property:

$$\psi_i(\chi_j) = \delta_{ij} \quad (16)$$

If N is even, take the stencils:

$$\begin{cases} S_1 &= \{C_{i-\frac{N}{2}}, \dots, C_{i+\frac{N}{2}}\} \\ S_2 &= \{C_{i-N}, \dots, C_i\} \\ S_3 &= \{C_i, \dots, C_{i+N}\} \end{cases} \quad (17)$$

If N is odd, take the stencils:

$$\begin{cases} S_1 &= \{C_{i-\lfloor \frac{N}{2} \rfloor}, \dots, C_{i+\lceil \frac{N}{2} \rceil}\} \\ S_2 &= \{C_{i-\lceil \frac{N}{2} \rceil}, \dots, C_{i+\lfloor \frac{N}{2} \rfloor}\} \\ S_3 &= \{C_{i-N}, \dots, C_i\} \\ S_4 &= \{C_i, \dots, C_{i+N}\} \end{cases} \quad (18)$$

The data is reconstructed on S_j as:

$$\sum_p \psi_p(\chi^i(x)) \hat{w}_p^{ij} \quad (19)$$

where the \hat{w}_p^{ij} are solutions to the following linear system:

$$\frac{1}{\Delta x} \int_{x_k}^{x_{k+1}} \sum_p \psi_p(\chi^k(x)) \hat{w}_p^{ij} dx = u_k \quad \forall C_k \in S_j \quad (20)$$

where u_k is the value of u in C_k . This can be written as $M_j \hat{\mathbf{w}}^{ij} = \mathbf{u}_{[j_0:j_N]}$ where $\{j_0, \dots, j_N\}$ indexes the cells in S_j . In this study reconstructions with $N = 2$ are used. The matrices of these linear systems are given in C.1, along with their inverses, which are precomputed to accelerate the solution of these systems.

Define the oscillation indicator matrix:

$$\Sigma_{mn} = \sum_{\alpha=1}^N \int_0^1 \psi_m^{(\alpha)} \psi_n^{(\alpha)} d\chi \quad (21)$$

and the oscillation indicator for each stencil:

$$o_j = \Sigma_{mn} \hat{w}_m^{ij} \hat{w}_n^{ij} \quad (22)$$

The full reconstruction in C_i is:

$$w_i(x) = \sum_p \psi_p(\chi^i(x)) \bar{w}_p^i \quad (23)$$

where $\bar{w}_p^i = \omega_j \hat{w}_p^{ij}$ is the weighted coefficient of the p th basis function, with weights:

$$\omega_j = \frac{\tilde{\omega}_j}{\sum_k \tilde{\omega}_k} \quad \tilde{\omega}_j = \frac{\zeta_j}{(o_j + \varepsilon)^r} \quad (24)$$

In this study, $r = 8$, $\varepsilon = 10^{-14}$, $\zeta_j = 10^5$ if S_j is a central stencil, and $\zeta_j = 1$ if S_j is a side stencil, as in [5].

The reconstruction can be extended to two dimensions by taking:

$$v^i = \frac{1}{\Delta y} (y - y_i) \quad (25)$$

and defining stencils in the y-axis in an analogous manner. The data in C_i is then reconstructed using stencil S_j as:

$$\sum_{p,q} \psi_p \left(\chi^i(x) \right) \psi_q \left(v^i(x) \right) \tilde{w}_{pq}^{ij} \quad (26)$$

where the coefficients of the weighted 1D reconstruction are used as cell averages:

$$M_j \tilde{w}_p^{ij} = \bar{w}_p^{[j_0:j_N]} \quad \forall p \in \{0, \dots, N\} \quad (27)$$

The oscillation indicator is calculated for each p in the same manner as the 1D case. The reconstruction method is easily further extensible to three dimensions, now using the coefficients \bar{w}_{pq} of the weighted 2D reconstruction as cell averages.

0.4.2 The Galerkin Predictor

Take a non-conservative, hyperbolic system of the form:

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} + \mathbf{B}(\mathbf{Q}) \cdot \frac{\partial \mathbf{Q}}{\partial x} = \mathbf{S}(\mathbf{Q}) \quad (28)$$

where \mathbf{Q} is the vector of conserved variables, \mathbf{F} is the conservative nonlinear flux, \mathbf{B} is the block matrix corresponding to the purely non-conservative component of the system, and $\mathbf{S}(\mathbf{Q})$ is the algebraic source vector.

Take the grid for the previous section, and time steps $t_0 < t_1 < \dots$ while defining $\Delta t_n = t_{n+1} - t_n$. Combining the techniques presented in [5?], the Discontinuous Galerkin method produces at each time step t_n a local polynomial approximation to \mathbf{Q} on each space-time cell $C_i \times [t_n, t_{n+1}]$.

Now define the scaled time variable:

$$\tau^n = \frac{1}{\Delta t_n} (t - t_n) \quad (29)$$

Thus, (28) becomes:

$$\frac{\partial \mathbf{Q}}{\partial \tau^n} + \frac{\partial \mathbf{F}^*(\mathbf{Q})}{\partial \chi^i} + \mathbf{B}^*(\mathbf{Q}) \cdot \frac{\partial \mathbf{Q}}{\partial \chi^i} = \mathbf{S}^*(\mathbf{Q}) \quad (30)$$

where

$$\mathbf{F}^* = \frac{\Delta t_n}{\Delta x} \mathbf{F} \quad \mathbf{B}^* = \frac{\Delta t_n}{\Delta x} \mathbf{B} \quad \mathbf{S}^* = \Delta t_n \mathbf{S} \quad (31)$$

The non-dimensionalization notation and spacetime cell indexing notation will be dropped for simplicity in what follows. Now define the set of spatio-temporal basis functions:

$$\{\theta_k(\chi, \tau)\} = \{\psi_p(\chi) \psi_s(\tau) : 0 \leq p, s \leq N\} \quad (32)$$

Denoting the Galerkin predictor by \mathbf{q} , take the following set of approximations:

$$\mathbf{Q} \approx \mathbf{q} = \theta_\beta \mathbf{q}_\beta \quad (33a)$$

$$\mathbf{F}(\mathbf{Q}) \approx \theta_\beta \mathbf{F}_\beta \quad (33b)$$

$$\mathbf{B}(\mathbf{Q}) \cdot \frac{\partial \mathbf{Q}}{\partial \chi} \approx \theta_\beta \mathbf{B}_\beta \quad (33c)$$

$$\mathbf{S}(\mathbf{Q}) \approx \theta_\beta \mathbf{S}_\beta \quad (33d)$$

for some coefficients $\mathbf{q}_\beta, \mathbf{F}_\beta, \mathbf{B}_\beta, \mathbf{S}_\beta$.

If $\{\psi_0, \dots, \psi_N\}$ is a nodal basis, the *nodal basis representation* may be used:

$$\mathbf{F}_\beta = \mathbf{F}(\mathbf{q}_\beta) \quad (34a)$$

$$\mathbf{B}_\beta = \mathbf{B}(\mathbf{q}_\beta) \cdot \left(\frac{\partial \theta_\gamma(\chi_\beta, \tau_\beta)}{\partial \chi} \mathbf{q}_\gamma \right) \quad (34b)$$

$$\mathbf{S}_\beta = \mathbf{S}(\mathbf{q}_\beta) \quad (34c)$$

where (χ_β, τ_β) are the coordinates of the node corresponding to basis function θ_β .

If a modal basis is used, $\mathbf{F}_\beta, \mathbf{B}_\beta, \mathbf{S}_\beta$ may be found from the previous values of \mathbf{q}_β in the iterative processes described below.

For functions $f(\chi, \tau) = f_\chi(\chi) f_\tau(\tau)$ and $g(\chi, \tau) = g_\chi(\chi) g_\tau(\tau)$, define the following integral operators:

$$[f, g]^t = f_\tau(t) g_\tau(t) \langle f_\chi, g_\chi \rangle \quad (35a)$$

$$\{f, g\} = \langle f_\tau, g_\tau \rangle \langle f_\chi, g_\chi \rangle \quad (35b)$$

Multiplying (30) by test function θ_α , using the polynomial approximations for Q, F, B, S , and integrating over space and time gives:

$$\left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\} \mathbf{q}_\beta = - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\} \mathbf{F}_\beta + \{ \theta_\alpha, \theta_\beta \} (\mathbf{S}_\beta - \mathbf{B}_\beta) \quad (36)$$

0.4.2.1 The Discontinuous Galerkin Method

This method of computing the Galerkin predictor allows solutions to be discontinuous at temporal cell boundaries, and is also suitable for stiff source terms.

Integrating (36) by parts in time gives:

$$\begin{aligned} \left([\theta_\alpha, \theta_\beta]^1 - \left\{ \frac{\partial \theta_\alpha}{\partial \tau}, \theta_\beta \right\} \right) \mathbf{q}_\beta &= [\theta_\alpha, \mathbf{w}]^0 - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\} \mathbf{F}_\beta \\ &\quad + \{ \theta_\alpha, \theta_\beta \} (\mathbf{S}_\beta - \mathbf{B}_\beta) \end{aligned} \quad (37)$$

where \mathbf{w} is the reconstruction obtained at the start of the time step with the WENO method. Define the following:

$$U_{\alpha\beta} = [\theta_\alpha, \theta_\beta]^1 - \left\{ \frac{\partial \theta_\alpha}{\partial \tau}, \theta_\beta \right\} \quad (38a)$$

$$V_{\alpha\beta} = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\} \quad (38b)$$

$$\mathbf{W}_\alpha = [\theta_\alpha, \psi_\gamma]^0 \mathbf{w}_\gamma \quad (38c)$$

$$Z_{\alpha\beta} = \{ \theta_\alpha, \theta_\beta \} \quad (38d)$$

Thus:

$$U_{\alpha\beta} \mathbf{q}_\beta = \mathbf{W}_\alpha - V_{\alpha\beta} \mathbf{F}_\beta + Z_{\alpha\beta} (\mathbf{S}_\beta - \mathbf{B}_\beta) \quad (39)$$

This nonlinear system in \mathbf{q}_β is solved by a Newton method. The source terms must be solved implicitly if they are stiff. Note that \mathbf{W} has no dependence on \mathbf{q} .

0.4.2.2 The Continuous Galerkin Method

This method of computing the Galerkin predictor is not suitable for stiff source terms, but it provides substantial savings on computational cost and ensures continuity across temporal cell boundaries.

$\{\psi_0, \dots, \psi_N\}$ must be chosen in such a way that the first $N + 1$ elements of $\{\theta_\beta\}$ have only a spatial dependence. The first $N + 1$ elements of \mathbf{q} are then fixed by demanding continuity at $\tau = 0$:

$$\mathbf{q}(\chi, 0) = \mathbf{w}(\chi) \quad (40)$$

where \mathbf{w} is spatial the reconstruction obtained at the start of the time step with the WENO method.

For a given vector $\mathbf{v} \in \mathbb{R}^{(N+1)^2}$ and matrix $X \in M_{(N+1)^2, (N+1)^2}(\mathbb{R})$, let $\mathbf{v} = (\mathbf{v}^0, \mathbf{v}^1)$ and $X = \begin{pmatrix} X^{00} & X^{01} \\ X^{10} & X^{11} \end{pmatrix}$ where \mathbf{v}^0, X^{00} are the components relating solely to the first $N + 1$ components of \mathbf{v} . We only need to find the latter components of \mathbf{q} , and thus, from (36), we have:

$$\begin{aligned} \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\}^{11} \mathbf{q}_\beta^1 &= \{\theta_\alpha, \theta_\beta\}^{11} (\mathbf{S}_\beta^1 - \mathbf{B}_\beta^1) - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\}^{11} \mathbf{F}_\beta^1 \\ &+ \{\theta_\alpha, \theta_\beta\}^{10} (\mathbf{S}_\beta^0 - \mathbf{B}_\beta^0) - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\}^{10} \mathbf{F}_\beta^0 \end{aligned} \quad (41)$$

Define the following:

$$U_{\alpha\beta} = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\}^{11} \quad (42a)$$

$$V_{\alpha\beta} = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\}^{11} \quad (42b)$$

$$\mathbf{W}_\alpha = \{\theta_\alpha, \theta_\beta\}^{10} (\mathbf{S}_\beta - \mathbf{B}_\beta)^0 - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi} \right\}^{10} \mathbf{F}_\beta^0 \quad (42c)$$

$$Z_{\alpha\beta} = \{\theta_\alpha, \theta_\beta\}^{11} \quad (42d)$$

Thus:

$$U_{\alpha\beta} \mathbf{q}_\beta^1 = \mathbf{W}_\alpha - V_{\alpha\beta} \mathbf{F}_\beta^1 + Z_{\alpha\beta} (\mathbf{S}_\beta^1 - \mathbf{B}_\beta^1) \quad (43)$$

Note that, as with the discontinuous Galerkin method, \mathbf{W} has no dependence on the degrees of freedom in \mathbf{q} .

0.4.3 The Finite Volume Scheme

Following the formulation of [5], integrating (28) over $[t_n, t_{n+1}] \times C_i$ gives:

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \Delta t_n (\mathbf{S}_i^n - \mathbf{P}_i^n) - \frac{\Delta t_n}{\Delta x} (\mathbf{D}_{i+1}^n - \mathbf{D}_i^n) \quad (44)$$

where

$$\mathbf{Q}_i^n = \frac{1}{\Delta x} \int_{x_i}^{x_{i+1}} \mathbf{Q}(x, t_n) dx \quad (45a)$$

$$\mathbf{S}_i^n = \frac{1}{\Delta t_n \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_i}^{x_{i+1}} \mathbf{S}(\mathbf{Q}) dx dt \quad (45b)$$

$$\mathbf{P}_i^n = \frac{1}{\Delta t_n \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_i}^{x_{i+1}} \mathbf{B}(\mathbf{Q}) \cdot \frac{\partial \mathbf{Q}}{\partial x} dx dt \quad (45c)$$

$$\mathbf{D}_i^n = \frac{1}{\Delta t_n} \int_{t_n}^{t_{n+1}} \mathcal{D}(\mathbf{Q}^-(x_i, t), \mathbf{Q}^+(x_i, t)) dt \quad (45d)$$

$\mathbf{Q}^-, \mathbf{Q}^+$ are the left and right extrapolated states at the x_i boundary. $\mathbf{S}_i^n, \mathbf{P}_i^n, \mathbf{D}_i^n$ are calculated using an $N + 1$ -point Gauss-Legendre quadrature, replacing \mathbf{Q} with \mathbf{q}_h .

M , as defined in Section 0.4.2, is a diagonalizable matrix with decomposition $M = R\Lambda R^{-1}$ where the columns of R are the right eigenvectors and Λ is the diagonal matrix of eigenvalues. Define the following matrix:

$$|M| = R |\Lambda| R^{-1} \quad (46)$$

Using these definitions, the interface terms arising in the FV formula have the following form:

$$\mathcal{D}(\mathbf{q}^-, \mathbf{q}^+) = \frac{1}{2} (\mathbf{F}(\mathbf{q}^-) + \mathbf{F}(\mathbf{q}^+) + \hat{\mathbf{B}} \cdot (\mathbf{q}^+ - \mathbf{q}^-) - \hat{M} \cdot (\mathbf{q}^+ - \mathbf{q}^-)) \quad (47)$$

\hat{M} is chosen to either correspond to a Rusanov/Lax-Friedrichs flux [22]:

$$\hat{M} = \max \left(\max \left| \Lambda \left(\mathbf{q}^- \right) \right|, \max \left| \Lambda \left(\mathbf{q}^+ \right) \right| \right) \quad (48)$$

or a simplified Osher–Solomon flux [7?]:

$$\hat{M} = \int_0^1 \left| M \left(\mathbf{q}^- + z \left(\mathbf{q}^+ - \mathbf{q}^- \right) \right) \right| dz \quad (49)$$

\hat{B} takes the following form:

$$\hat{B} = \int_0^1 B \left(\mathbf{q}^- + z \left(\mathbf{q}^+ - \mathbf{q}^- \right) \right) dz \quad (50)$$

It was found that the Osher-Solomon flux would often produce slightly less diffusive results, but that it was more computationally expensive, and also had a greater tendency to introduce numerical artifacts.

0.4.4 Time Step and Boundary Conditions

Let Λ_i^n be the set of eigenvalues of the HPR system evaluated at \mathbf{Q}_i^n . $C_{cfl} < 1$ is a constant (usually taken to be 0.9, unless the problem being simulated is particularly demanding, requiring a lower value). A semi-analytic form for Λ is given in Section ???. The eigenvalues determine the speed of propagation of information in the solution to the Riemann Problem at the cell interfaces, and the time step is chosen to ensure that the characteristics do not enter into other cells between t_n and t_{n+1} :

$$\Delta t_n = \frac{C_{cfl} \cdot \Delta x}{\max_i |\Lambda_i^n|} \quad (51)$$

Transmissive boundary conditions (allowing material and heat to pass through) are implemented by setting the state variables in the boundary cells to the same value as their non-boundary neighbors. Reflective boundary conditions are implemented in the same way, except that the directions of the velocity and thermal impulse vectors in the boundary cells are reversed.

Chapter 1

Objective 1: Extending the GPR Model

1.1 Equations of State

1.1.1 Mie-Gruneisen Models

It is required to specify the microscale energy E_1 appearing in (6). In this study, several different possible choices for E_1 are put into the same framework by expressing them in the following Mie-Gruneisen form:

$$E_1(\rho, p) = e(\rho, p) = e_{ref}(\rho) + \frac{p - p_{ref}(\rho)}{\rho\Gamma(\rho)} \quad (1.1a)$$

$$T = T_{ref} + \frac{e - e_{ref}}{c_v} = T_{ref} + \frac{p - p_{ref}(\rho)}{c_v \rho \Gamma(\rho)} \quad (1.1b)$$

The forms taken by e_{ref} , p_{ref} , Γ for various different instances of this class are given in Table 1.1 on page 15. The first five entries in the table are standard results. The sixth is derived thus. The Godunov-Romenski hyperelastic EOS is given by:

$$E(\rho, S, A) = \frac{c_0^2}{2\alpha^2} \left(|A^T A|^{\alpha/2} - 1 \right)^2 + c_v T_0 |A^T A|^{\gamma/2} (e^{S/c_v} - 1) + \frac{b_0^2}{4} |A^T A|^{\beta/2} \|\text{dev}(A^T A)\|^2 \quad (1.2)$$

Using the relation $\det(A) = \frac{\rho}{\rho_0}$, this can be thought of as taking the form:

$$E_1(\rho, s) + \frac{c_s(\rho)^2}{4} \|\text{dev}(G)\|_F^2 \quad (1.3)$$

where $c_s = b_0 \left(\frac{\rho}{\rho_0} \right)^\beta$. Considering only the microscale energy component, note that we have:

$$p = \rho^2 \frac{\partial E_1}{\partial \rho} = \rho \left(\frac{c_0^2}{\alpha} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right) \left(\frac{\rho}{\rho_0} \right)^\alpha + \gamma c_v T_0 \left(\frac{\rho}{\rho_0} \right)^\gamma (e^{S/c_v} - 1) \right) \quad (1.4)$$

Therefore:

$$E_1 - \frac{c_0^2}{2\alpha^2} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right)^2 = \frac{p}{\gamma \rho} - \frac{c_0^2}{\gamma \alpha} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right) \left(\frac{\rho}{\rho_0} \right)^\alpha \quad (1.5)$$

Thus E_1 can be put in Mie-Gruneisen form:

$$E_1 = \frac{p - \frac{c_0^2 \rho}{\alpha} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right) \left(\frac{\rho}{\rho_0} \right)^\alpha}{\gamma \rho} + \frac{c_0^2}{2\alpha^2} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right)^2 \quad (1.6)$$

Although this is a versatile class of equations of state - and it is fit for the purposes that the model is put to here - it should be noted that many other choices are available.

1.1.2 Variable Transverse Perturbation Speed

Taking (1.1a) and using the fact that $p = \rho^2 e_\rho$, we have:

$$e_\rho - \frac{\Gamma}{\rho} e = \frac{p_{ref}}{\rho^2} - \frac{\Gamma}{\rho} e_{ref} \quad (1.7)$$

The solutions to this equation for different forms of Γ, p_{ref}, e_{ref} take the form below, where f is an arbitrary function of s , g depends on the form of Γ , and \hat{e} is a particular solution of the equation (which will be equal to e_{ref} if $p_{ref} = \rho^2 \frac{\partial e_{ref}}{\partial \rho}$).

$$e = f(s) g(\rho) + \hat{e}(\rho) \quad (1.8)$$

We have:

$$p = \rho^2 e_\rho = \rho^2 \left(f(s) g'(\rho) + \hat{e}'(\rho) \right) \quad (1.9)$$

Thus:

$$\frac{\frac{p}{\rho^2} - \hat{e}'(\rho)}{g'(\rho)} = f(s) = \frac{e - \hat{e}(\rho)}{g(\rho)} \quad (1.10)$$

Therefore:

$$E_1(\rho, p) = e(\rho, p) = \hat{e}(\rho) + \frac{g(\rho)}{g'(\rho)} \left(\frac{p}{\rho^2} - \hat{e}'(\rho) \right) \quad (1.11)$$

Equation of State	$p_{ref}(\rho)$	$e_{ref}(\rho)$	$\Gamma(\rho)$	T_{ref}
Ideal Gas	0	0	$\Gamma_0 (= \gamma - 1)$	0
Stiffened Gas	$-p_\infty$	$\frac{p_\infty}{\rho}$	$\Gamma_0 (= \gamma - 1)$	0
Shock Mie-Gruneisen	$\frac{c_0^2 \left(\frac{1}{\rho_0} - \frac{1}{\rho} \right)^2}{\left(\frac{1}{\rho_0} - s \left(\frac{1}{\rho_0} - \frac{1}{\rho} \right) \right)^2}$	$\frac{p_{ref}}{2} \left(\frac{1}{\rho_0} - \frac{1}{\rho} \right)$	$\Gamma_0 \frac{\rho_0}{\rho}$	0
JWL	$A e^{-\frac{R_1 \rho_0}{\rho}} + B e^{-\frac{R_2 \rho_0}{\rho}}$	$\frac{A}{\rho_0 R_1} e^{-\frac{R_1 \rho_0}{\rho}} + \frac{B}{\rho_0 R_2} e^{-\frac{R_2 \rho_0}{\rho}}$	Γ_0	0
Cochran-Chan	$A \left(\frac{\rho}{\rho_0} \right)^{\epsilon_1} - B \left(\frac{\rho}{\rho_0} \right)^{\epsilon_2}$	$\frac{A}{\rho_0 (\epsilon_1 - 1)} \left(\left(\frac{\rho}{\rho_0} \right)^{\epsilon_1 - 1} - 1 \right) - \frac{B}{\rho_0 (\epsilon_1 - 1)} \left(\left(\frac{\rho}{\rho_0} \right)^{\epsilon_2 - 1} - 1 \right)$	Γ_0	0
Godunov-Romenski	$\frac{c_0^2 \rho}{\alpha} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right) \left(\frac{\rho}{\rho_0} \right)^\alpha$	$\frac{c_0^2}{2\alpha^2} \left(\left(\frac{\rho}{\rho_0} \right)^\alpha - 1 \right)^2$	$\Gamma_0 (= \gamma)$	T_0

Table 1.1: e_{ref} , p_{ref} , Γ for different kinds of Mie-Gruneisen equations of state

We now add another term to the energy, giving it the following form:

$$E(\rho, p) = e(\rho, p) = f(s)g(\rho) + \hat{e}(\rho) + B(\rho)h(A) \quad (1.12)$$

We then have:

$$p = \rho^2 \left(f(s)g'(\rho) + \hat{e}'(\rho) + B'(\rho)h(A) \right) \quad (1.13)$$

Thus:

$$\frac{\frac{p}{\rho^2} - \hat{e}'(\rho) - B'(\rho)h(A)}{g'(\rho)} = f(s) = \frac{e - \hat{e}(\rho) - B(\rho)h(A)}{g(\rho)} \quad (1.14)$$

Therefore:

$$e = \frac{g(\rho)}{g'(\rho)} \left(\frac{p}{\rho^2} - \hat{e}'(\rho) - B'(\rho)h(A) \right) + \hat{e}(\rho) + B(\rho)h(A) \quad (1.15)$$

Noting that $\frac{g(\rho)}{g'(\rho)} = \frac{\rho}{\Gamma(\rho)}$, this can be expressed as:

$$E(\rho, p) = e(\rho, p) = E_1(\rho, p) + \left(B(\rho) - \frac{\rho}{\Gamma(\rho)} B'(\rho) \right) h(A) \quad (1.16)$$

If the EOS comprises a microscale energy component of Mie-Gruneisen type, and a mesoscale energy component with speed of transverse vibrations dependent upon ρ , then this provides a method to recover the EOS in terms of ρ and p (by substituting $B(\rho) = \frac{c_s^2(\rho)}{4}$ and $h(A) = \left\| \text{dev}(A^T A) \right\|_F^2$).

1.2 Multiple Materials

1.3 Reactive Materials

Chapter 2

Objective 2: Improved Numerical Methods

2.1 Convergence Properties of the Standard ADER-WENO Method

Extending the Galerkin method described in Section 0.4.2 to three dimensions, the following system must be solved for \mathbf{q} :

$$U_{\alpha\beta}\mathbf{q}_\beta = \mathbf{W}_\alpha - V_{\alpha\beta}^1\mathbf{F}_\beta - V_{\alpha\beta}^2\mathbf{G}_\beta - V_{\alpha\beta}^3\mathbf{H}_\beta + Z_{\alpha\beta}(\mathbf{S}_\beta - \mathbf{B}_\beta) \quad (2.1)$$

where now we have the 3 scaled spatial variables χ_1, χ_2, χ_3 and \mathbf{G}, \mathbf{H} are the flux components in the second and third spatial directions, respectively. In the case of the continuous Galerkin method, it is assumed that (2.1) is to be solved for only the non-fixed degrees of freedom in \mathbf{q} . The matrices $V_{\alpha\beta}^i$ are defined thus:

$$V_{\alpha\beta}^i = \left\langle \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi_i} \right\rangle \quad (2.2)$$

For the discontinuous Galerkin method, \mathbf{W}_α now takes the form:

$$\mathbf{W}_\alpha = [\theta_\alpha, \Psi_\gamma]^0 \mathbf{w}_\gamma \quad (2.3)$$

where $\Psi_\gamma(\chi_1, \chi_2, \chi_3)$ is an element of the following set, enumerated by γ :

$$\{\psi_i(\chi_1) \psi_j(\chi_2) \psi_c(\chi_3) : 0 \leq i, j, k \leq N\} \quad (2.4)$$

For the continuous Galerkin method, \mathbf{W}_α now takes the form:

2.1 Convergence Properties of the Standard ADER-WENO Method

$$\begin{aligned} W_\alpha &= \{\theta_\alpha, \theta_\beta\}^{10} (\mathbf{S}_\beta - \mathbf{B}_\beta)^0 \\ &\quad - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^1} \right\}^{10} \mathbf{F}_\beta^0 - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^2} \right\}^{10} \mathbf{G}_\beta^0 - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^2} \right\}^{10} \mathbf{H}_\beta^0 \end{aligned} \quad (2.5)$$

[3] remark that for the continuous Galerkin case, the eigenvalues of $U^{-1}V^i$ are all 0 for $0 \leq N \leq 5$, for $i = 1, 2, 3$. [?] state the same result for the discontinuous Galerkin case. This implies that in the conservative, homogeneous case ($\mathbf{B} = \mathbf{S} = \mathbf{0}$), owing to the Banach Fixed Point Theorem, existence and uniqueness of a solution are established, and convergence to this solution is guaranteed. As noted in [?], in the linear case it is implied that the iterative procedure converges after at most $N + 1$ iterations.

In [3] it is conjectured that the result concerning the eigenvalues of $U^{-1}V^i$ holds for any N , and any number of spatial dimensions. A solution to this conjecture is provided here. For the theory in linear algebra required for this section, please consult a standard textbook on the subject, such as [?].

2.1.1 The Discontinuous Galerkin Case

First, given the basis polynomials $\{\psi_0, \dots, \psi_N\}$, define the following matrices:

$$\mathbb{N}_{ij} = \langle \psi_i, \psi_j \rangle \quad (2.6a)$$

$$\mathbb{Q}_{ij} = \langle \psi_i, \psi'_j \rangle \quad (2.6b)$$

Note that \mathbb{N} is the Gram matrix, which by linear independence of $\{\psi_0, \dots, \psi_N\}$ is invertible. Note also that if $p \in P_N$ then $p = a_j \psi_j$ for some unique coefficient vector \mathbf{a} . Thus, taking inner products with ψ_i , we have $\langle \psi_i, \psi_j \rangle a_j = \langle \psi_i, p \rangle$ for $i = 0, \dots, N$. This produces the following result:

$$p = a_j \psi_j \Leftrightarrow \mathbf{a} = \mathbb{N}^{-1} \mathbf{x}, \quad x_i = \langle \psi_i, p \rangle \quad (2.7)$$

Without loss of generality, take the ordering:

$$\alpha = \alpha_t (N + 1)^3 + \alpha_x (N + 1)^2 + \alpha_y (N + 1) + \alpha_z \quad (2.8)$$

2.1 Convergence Properties of the Standard ADER-WENO Method

where $0 \leq \alpha_t, \alpha_x, \alpha_y, \alpha_z \leq N$. Using the same ordering for β , we have:

$$U_{\alpha\beta} = (\psi_{\alpha_t}(1) \psi_{\beta_t}(1) - \mathfrak{I}_{\beta_t \alpha_t}) \cdot \aleph_{\alpha_x \beta_x} \cdot \aleph_{\alpha_y \beta_y} \cdot \aleph_{\alpha_z \beta_z} \quad (2.9a)$$

$$V_{\alpha\beta}^1 = \aleph_{\alpha_t \beta_t} \cdot \mathfrak{I}_{\alpha_x \beta_x} \cdot \aleph_{\alpha_y \beta_y} \cdot \aleph_{\alpha_z \beta_z} \quad (2.9b)$$

Thus:

$$U = C \otimes \aleph \otimes \aleph \otimes \aleph \quad (2.10a)$$

$$V^1 = \aleph \otimes \mathfrak{I} \otimes \aleph \otimes \aleph \quad (2.10b)$$

where $C_{ij} = \psi_i(1) \psi_j(1) - \mathfrak{I}_{ji}$. Thus:

$$U^{-1}V^1 = (C^{-1}\aleph) \otimes (\aleph^{-1}\mathfrak{I}) \otimes I \otimes I \quad (2.11)$$

Therefore:

$$(U^{-1}V^1)^k = (C^{-1}\aleph)^k \otimes (\aleph^{-1}\mathfrak{I})^k \otimes I \otimes I \quad (2.12)$$

A matrix X is nilpotent ($X^k = 0$ for some $k \in \mathbb{N}$) if and only if all its eigenvalues are 0. The conjecture will be proved if it is shown that $(\aleph^{-1}\mathfrak{I})^k = 0$ for some $k \in \mathbb{N}$, as this would imply that $(U^{-1}V^1)^k = 0$, and thus all eigenvalues of $U^{-1}V^1$ are 0.

Taking $\mathbf{a} \in \mathbb{R}^{N+1}$, define:

$$p = a_0\psi_0 + \dots + a_N\psi_N \in P_N \quad (2.13)$$

Note that:

$$(\mathfrak{I}\mathbf{a})_i = \langle \psi_i, \psi'_0 \rangle a_0 + \dots + \langle \psi_i, \psi'_N \rangle a_N = \langle \psi_i, p' \rangle \quad (2.14)$$

Thus, by (2.7):

$$(\aleph^{-1}\mathfrak{I}\mathbf{a})_i \psi_i = p' \quad (2.15)$$

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

By induction:

$$\left(\left(\aleph^{-1} \beth \right)^k \mathbf{a} \right)_i \psi_i = p^{(k)} \quad (2.16)$$

for any $k \in \mathbb{N}$. As $p \in P_N$, $(\aleph^{-1} \beth)^{N+1} \mathbf{a} = \mathbf{0}$. As \mathbf{a} was chosen arbitrarily, $(\aleph^{-1} \beth)^{N+1} = 0$. Thus, the conjecture is solved.

This proof is easily adapted to show that $U^{-1}V^2$ and $U^{-1}V^3$ are nilpotent, and clearly extends to any number of spatial dimensions. No specific choice has been made for $N \in \mathbb{N}$ and thus the result holds in general.

2.1.2 The Continuous Galerkin Case

In addition to \aleph, \beth , we now define \aleph', \beth' where each new matrix is equal to the original, with its first row and column removed (the row and column corresponding to the constant-term polynomial ψ_0). Take the following ordering:

$$\alpha = \alpha_t (N+1)^3 + \alpha_x (N+1)^2 + \alpha_y (N+1) + \alpha_z \quad (2.17)$$

where now $0 \leq \alpha_x, \alpha_y, \alpha_z \leq N$ and $0 \leq \alpha_t \leq N-1$. Using the same ordering for β , we now have:

$$U_{\alpha\beta} = \beth'_{\alpha_t\beta_t} \cdot \aleph_{\alpha_x\beta_x} \cdot \aleph_{\alpha_y\beta_y} \cdot \aleph_{\alpha_z\beta_z} \quad (2.18a)$$

$$V_{\alpha\beta}^1 = \aleph'_{\alpha_t\beta_t} \cdot \beth_{\alpha_x\beta_x} \cdot \aleph_{\alpha_y\beta_y} \cdot \aleph_{\alpha_z\beta_z} \quad (2.18b)$$

The proof for the continuous case follows in the same manner as the proof for the discontinuous case, with:

$$U^{-1}V^1 = \left((\beth')^{-1} \aleph' \right) \otimes \left(\aleph^{-1} \beth \right) \otimes I \otimes I \quad (2.19)$$

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

[16] have proposed a new, more efficient class of basis polynomials. While the method was

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

given for conservative, one-dimensional systems in the original paper, it is extended here to general non-conservative, multidimensional systems.

Take a non-homogeneous, non-conservative, hyperbolic system of the form:

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{Q}) + \vec{\mathbf{B}}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}) \quad (2.20)$$

where \mathbf{Q} is the vector of conserved variables, $\vec{\mathbf{F}} = (F_1, F_2, F_3)$ and $\vec{\mathbf{B}} = (B_1, B_2, B_3)$ are respectively the conservative nonlinear fluxes and matrices corresponding to the purely non-conservative components of the system, and $\mathbf{S}(\mathbf{Q})$ is the algebraic source vector.

Define spatial variables $x^{(1)}, x^{(2)}, x^{(3)}$. Take the space-time cell $C = [x_{i_1}^{(1)}, x_{i_1+1}^{(1)}] \times [x_{i_2}^{(2)}, x_{i_2+1}^{(2)}] \times [x_{i_3}^{(3)}, x_{i_3+1}^{(3)}] \times [t_n, t_{n+1}]$. Define the scaled spatial and temporal variables:

$$\chi^{(k)} = \frac{x^{(k)} - x_{i_k}^{(k)}}{x_{i_k+1}^{(k)} - x_{i_k}^{(k)}} \quad (2.21a)$$

$$\tau = \frac{t - t_n}{t_{n+1} - t_n} \quad (2.21b)$$

Thus, C becomes:

$$(\chi^{(1)}, \chi^{(2)}, \chi^{(3)}, \tau) \in [0, 1]^4 \quad (2.22)$$

By rescaling $\vec{\mathbf{F}}, \vec{\mathbf{B}}, \mathbf{S}$ by the appropriate constant factors, and defining $\tilde{\nabla} = (\partial_{\chi^{(1)}}, \partial_{\chi^{(2)}}, \partial_{\chi^{(3)}})$, within C equation (2.20) becomes:

$$\frac{\partial \mathbf{Q}}{\partial \tau} + \tilde{\nabla} \cdot \vec{\mathbf{F}}(\mathbf{Q}) + \vec{\mathbf{B}}(\mathbf{Q}) \cdot \tilde{\nabla} \mathbf{Q} = \mathbf{S}(\mathbf{Q}) \quad (2.23)$$

A basis $\{\psi_0, \dots, \psi_N\}$ of P_N and inner product $\langle \cdot, \cdot \rangle$ are now required to produce a polynomial reconstruction of \mathbf{Q} within C . Traditionally, this basis has been chosen to be either nodal ($\psi_i(\alpha_j) = \delta_{ij}$ where $\{\alpha_0, \dots, \alpha_N\}$ are a set of nodes, e.g. the Gauss-Legendre abscissae - see [3]), or modal (e.g. the Legendre polynomials - see [1]).

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

[16] take the following approach. $\langle \cdot, \cdot \rangle$ is taken to be the usual integral product on $[0, 1]$. Supposing that $N = 2n + 1$ for some $n \in \mathbb{N}$, Gauss-Legendre nodes $\{\alpha_0, \dots, \alpha_n\}$ are taken. The basis $\Psi = \{\psi_0, \dots, \psi_N\} \subset P_N$ is taken with the following properties for $i = 0, \dots, n$:

$$\begin{cases} \psi_i(\alpha_j) = \delta_{ij} & \psi'_i(\alpha_j) = 0 \\ \psi_{n+1+i}(\alpha_j) = 0 & \psi'_{n+1+i}(\alpha_j) = \delta_{ij} \end{cases} \quad (2.24)$$

Define the following subsets:

$$\Psi^0 = \{\psi_i : 0 \leq i \leq n\} \quad (2.25a)$$

$$\Psi^1 = \{\psi_i : n + 1 \leq i \leq 2n + 1\} \quad (2.25b)$$

The WENO method (as used in [9]) produces an order- N polynomial reconstruction $w(\chi^{(1)}, \chi^{(2)}, \chi^{(3)})$ of the data at time t_n in $[x_{i_1}^{(1)}, x_{i_1+1}^{(1)}] \times [x_{i_2}^{(2)}, x_{i_2+1}^{(2)}] \times [x_{i_3}^{(3)}, x_{i_3+1}^{(3)}]$. It is used as initial data in the problem of finding the Galerkin predictor. Taking representation $w = w_{abc}\psi_a(\chi^{(1)})\psi_b(\chi^{(2)})\psi_c(\chi^{(3)})$ we have for $0 \leq i, j, k \leq n$:

$$w_{ijk} = w(\alpha_i, \alpha_j, \alpha_k) \quad (2.26a)$$

$$w_{(n+i+1)jk} = \partial_{\chi^{(1)}} w(\alpha_i, \alpha_j, \alpha_k) \quad (2.26b)$$

$$w_{i(n+j+1)k} = \partial_{\chi^{(2)}} w(\alpha_i, \alpha_j, \alpha_k) \quad (2.26c)$$

$$w_{ij(n+k+1)} = \partial_{\chi^{(3)}} w(\alpha_i, \alpha_j, \alpha_k) \quad (2.26d)$$

Take the following temporal nodes, where τ_1, \dots, τ_N are the usual Legendre-Gauss nodes on $[0, 1]$ and $\tau_0 = 0$ or $\tau_0 = 1$ if we are performing a Continuous Galerkin / Discontinuous Galerkin reconstruction, respectively:

$$\{\tau_0, \dots, \tau_N\} \quad (2.27)$$

Define $\Phi = \{\phi_0, \dots, \phi_N\} \subset P_N$ to be the set of Lagrange interpolating polynomials on the temporal nodes. We now define the spatio-temporal polynomial basis $\Theta = \Phi \otimes \Psi \otimes \Psi \otimes \Psi = \{\theta_\beta\}$ for $0 \leq \beta \leq (N + 1)^4 - 1$. Define subsets $\Theta^{\iota\xi\kappa} = \Phi \otimes \Psi^\iota \otimes \Psi^\xi \otimes \Psi^\kappa = \{\theta_\mu^{\iota\xi\kappa}\}$ where $\iota, \xi, \kappa \in \{0, 1\}$ for $0 \leq \mu \leq (N + 1)(n + 1)^3 - 1$.

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

Denoting the Galerkin predictor by q , take the following set of approximations:

$$Q \approx \theta_\beta q_\beta = \theta_\mu^{\iota\xi\kappa} q_\mu^{\iota\xi\kappa} \quad (2.28a)$$

$$\vec{F}(Q) \approx \theta_\beta \vec{F}_\beta = \theta_\mu^{\iota\xi\kappa} \vec{F}_\mu^{\iota\xi\kappa} \quad (2.28b)$$

$$\vec{B}(Q) \cdot \vec{\nabla} Q \approx \theta_\beta B_\beta = \theta_\mu^{\iota\xi\kappa} B_\mu^{\iota\xi\kappa} \quad (2.28c)$$

$$S(Q) \approx \theta_\beta S_\beta = \theta_\mu^{\iota\xi\kappa} S_\mu^{\iota\xi\kappa} \quad (2.28d)$$

for some coefficients $q_\beta, \vec{F}_\beta, B_\beta, S_\beta$. The *nodal basis representation* is used for the coefficients of Θ^{000} :

$$\vec{F}_\mu^{000} = \vec{F}(q_\mu^{000}) \quad (2.29a)$$

$$B_\mu^{000} = B_1(q_\mu^{000}) q_\mu^{100} + B_2(q_\mu^{000}) q_\mu^{010} + B_3(q_\mu^{000}) q_\mu^{001} \quad (2.29b)$$

$$S_\mu^{000} = S(q_\mu^{000}) \quad (2.29c)$$

In general, we have:

$$\vec{F}_\mu^{\iota\xi\kappa} = \partial_\chi^\iota \partial_v^\xi \partial_\zeta^\kappa (\vec{F}(Q)) \quad (2.30a)$$

$$B_\mu^{\iota\xi\kappa} = \partial_\chi^\iota \partial_v^\xi \partial_\zeta^\kappa (\vec{B}(Q) \cdot \vec{\nabla} Q) \quad (2.30b)$$

$$S_\mu^{\iota\xi\kappa} = \partial_\chi^\iota \partial_v^\xi \partial_\zeta^\kappa (S(Q)) \quad (2.30c)$$

where the right-hand-side is evaluated at the nodal point corresponding to μ . The full expressions are omitted here for brevity's sake, but note that for a one-dimensional system:

$$F_{1\mu}^{100} = \frac{\partial F(q_\mu^{000})}{\partial Q} \cdot q_\mu^{100} \quad (2.31a)$$

$$B_\mu^{100} = \left(\frac{\partial B_1(q_\mu^{000})}{\partial Q} \cdot q_\mu^{100} \right) \cdot q_\mu^{100} + B_1(q_\mu^{000}) \cdot \left(\frac{\partial^2 \theta_\kappa^{000}(\chi_\mu, \tau_\mu)}{\partial \chi^2} q_\mu^{000} + \frac{\partial^2 \theta_\kappa^{100}(\chi_\mu, \tau_\mu)}{\partial \chi^2} q_\mu^{100} \right) \quad (2.31b)$$

$$S_\mu^{100} = \frac{\partial S(q_\mu^{000})}{\partial Q} \cdot q_\mu^{100} \quad (2.31c)$$

where χ_μ, τ_μ are the spatial and temporal coordinates where $\theta_\mu^{100} = 0$ and $\partial_\chi \theta_\mu^{100} = 1$. Note that $\frac{\partial B_1}{\partial Q}$ is a rank 3 tensor.

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

Consider functions f, g of the following form:

$$f(\tau, \chi^{(1)}, \chi^{(2)}, \chi^{(3)}) = f_\tau(\tau) f_1(\chi^{(1)}) f_2(\chi^{(2)}) f_3(\chi^{(3)}) \quad (2.32a)$$

$$g(\tau, \chi^{(1)}, \chi^{(2)}, \chi^{(3)}) = g_\tau(\tau) g_1(\chi^{(1)}) g_2(\chi^{(2)}) g_3(\chi^{(3)}) \quad (2.32b)$$

Define the following integral operators:

$$[f, g]^t = f_\tau(t) g_\tau(t) \langle f_1, g_1 \rangle \langle f_2, g_2 \rangle \langle f_3, g_3 \rangle \quad (2.33a)$$

$$\{f, g\} = \langle f_\tau, g_\tau \rangle \langle f_1, g_1 \rangle \langle f_2, g_2 \rangle \langle f_3, g_3 \rangle \quad (2.33b)$$

Multiplying (2.28b) by test function θ_α , using the polynomial approximations for $\vec{Q}, \vec{F}, \vec{B}, S$, and integrating over space and time gives:

$$\left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\} \mathbf{q}_\beta = \{ \theta_\alpha, \theta_\beta \} (\mathbf{S}_\beta - \mathbf{B}_\beta) - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\} \mathbf{F}_{k\beta} \quad (2.34)$$

2.2.1 The Discontinuous Galerkin Method

This method of computing the Galerkin predictor allows solutions to be discontinuous at temporal cell boundaries, and is also suitable for stiff source terms. Integrating (2.34) by parts in time gives:

$$\begin{aligned} \left([\theta_\alpha, \theta_\beta]^1 - \left\{ \frac{\partial \theta_\alpha}{\partial \tau}, \theta_\beta \right\} \right) \mathbf{q}_\beta &= [\theta_\alpha, \mathbf{w}]^0 + \{ \theta_\alpha, \theta_\beta \} (\mathbf{S}_\beta - \mathbf{B}_\beta) \\ &\quad - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\} \mathbf{F}_{k\beta} \end{aligned} \quad (2.35)$$

where \mathbf{w} is the reconstruction obtained at the start of the time step with the WENO method. Take the following ordering:

$$\theta_{(N+1)^3 h + (N+1)^2 i + (N+1) j + k}(\tau, \chi, v, \zeta) = \phi_h(\tau) \psi_i(\chi) \psi_j(v) \psi_k(\zeta) \quad (2.36)$$

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

where $0 \leq h, i, j, k \leq N$. Thus, define the following:

$$U_{\alpha\beta} = [\theta_\alpha, \theta_\beta]^1 - \left\{ \frac{\partial \theta_\alpha}{\partial \tau}, \theta_\beta \right\} = (R^1 - M^{\tau,1}) \otimes (M^\chi)^3 \quad (2.37a)$$

$$V_{\alpha\beta}^k = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\} = M^\tau \otimes (M^\chi)^{k-1} \otimes M^{\chi,1} \otimes (M^\chi)^{3-k} \quad (2.37b)$$

$$\mathbf{W}_\alpha = [\theta_\alpha, \Psi_\gamma]^0 \mathbf{w}_\gamma = R^0 \otimes (M^\chi)^3 \quad (2.37c)$$

$$Z_{\alpha\beta} = \{\theta_\alpha, \theta_\beta\} = M^\tau \otimes (M^\chi)^3 \quad (2.37d)$$

where $\{\Psi_\gamma\} = \Psi \otimes \Psi \otimes \Psi$ and:

$$\begin{cases} M_{ij}^\tau = \langle \phi_i, \phi_j \rangle & M_{ij}^{\tau,1} = \langle \phi'_i, \phi_j \rangle \\ M_{ij}^\chi = \langle \psi_i, \psi_j \rangle & M_{ij}^{\chi,1} = \langle \psi_i, \psi'_j \rangle \\ R_{ij}^1 = \phi_i(1) \phi_j(1) & \mathbf{R}_i^0 = \phi_i(0) \end{cases} \quad (2.38)$$

Thus:

$$U_{\alpha\beta} \mathbf{q}_\beta = \mathbf{W}_\alpha + Z_{\alpha\beta} (\mathbf{S}_\beta - \mathbf{B}_\beta) - V_{\alpha\beta}^{(k)} \mathbf{F}_{k\beta} \quad (2.39)$$

Take the definitions:

$$\begin{cases} D &= (M^\chi)^{-1} M^{\chi,1} \\ E &= (R^1 - M^{\tau,1}) \end{cases} \quad (2.40)$$

Noting that $E1 = \mathbf{R}^0$, we have, by inversion of U :

$$\begin{aligned} \mathbf{q} &= (\mathbf{1} \otimes I^3) \mathbf{w} + (E^{-1} M^\tau \otimes I^3) (\mathbf{S} - \mathbf{B}) \\ &\quad - (E^{-1} M^\tau \otimes I^{k-1} \otimes D \otimes I^{3-k}) \mathbf{F}_k \end{aligned} \quad (2.41)$$

Thus, we have:

$$\begin{aligned} \mathbf{q}_{hijk} &= \mathbf{w}_{ijk} + (E^{-1} M^\tau)_{hm} (\mathbf{S}_{mijk} - \mathbf{B}_{mijk}) \\ &\quad - (E^{-1} M^\tau)_{hm} (D_{in} (\mathbf{F}_1)_{mnjk} + D_{jn} (\mathbf{F}_2)_{mink} + D_{kn} (\mathbf{F}_3)_{mijn}) \end{aligned} \quad (2.42)$$

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

Note then that $\mathbf{q}^{\iota\xi\kappa}$ is a function of $\mathbf{S}^{\iota\xi\kappa}, \mathbf{B}^{\iota\xi\kappa}, \vec{\mathbf{F}}$:

$$\mathbf{q}^{\iota\xi\kappa} = \mathcal{F}(\mathbf{S}^{\iota\xi\kappa}) + \mathcal{F}(\mathbf{B}^{\iota\xi\kappa}) + \mathcal{G}_{\iota\xi\kappa}(\vec{\mathbf{F}}^{000}, \dots, \vec{\mathbf{F}}^{111}) \quad (2.43)$$

where $\mathcal{F}, \mathcal{G}_{\iota\xi\kappa}$ are linear functions. Note in turn that, by (2.30c):

$$\mathbf{S}^{\iota\xi\kappa} = \mathcal{H} \left(\bigcup_{(0,0,0) \leq (a,b,c) \leq (\iota,\xi,\kappa)} \mathbf{q}^{abc} \right) \quad (2.44)$$

where \mathcal{H} is a nonlinear function.

In the case of stiff source terms, the following Picard iteration procedure can be used to solve (2.42), as adapted from [16]:

$$\begin{aligned} (\mathbf{q}^{\iota\xi\kappa})_{m+1} = & \mathcal{F} \left(\mathcal{H} \left((\mathbf{q}^{\iota\xi\kappa})_{m+1} \cup \bigcup_{\substack{(0,0,0) \leq (a,b,c) \leq (\iota,\xi,\kappa) \\ (a,b,c) \neq (\iota,\xi,\kappa)}}} (\mathbf{q}^{abc})_m \right) \right) \\ & + \mathcal{F}((\mathbf{B}^{\iota\xi\kappa})_m) + \mathcal{G}_{\iota\xi\kappa}((\vec{\mathbf{F}}^{000})_m, \dots, (\vec{\mathbf{F}}^{111})_m) \end{aligned} \quad (2.45)$$

2.2.2 The Continuous Galerkin Method

This method of computing the Galerkin predictor is not suitable for stiff source terms, but is less computationally expensive and ensures continuity across temporal cell boundaries. The first $N + 1$ elements of \mathbf{q} are fixed by imposing the following condition:

$$\mathbf{q}(\chi, 0) = \mathbf{w}(\chi) \quad (2.46)$$

For $\mathbf{v} \in \mathbb{R}^{(N+1)^2}$ and $X \in M_{(N+1)^2, (N+1)^2}(\mathbb{R})$, let $\mathbf{v} = (\mathbf{v}^0, \mathbf{v}^1)$ and $X = \begin{pmatrix} X^{00} & X^{01} \\ X^{10} & X^{11} \end{pmatrix}$ where \mathbf{v}^0, X^{00} are the components relating solely to the first $N + 1$ components of \mathbf{v} . We only need to find the latter components of \mathbf{q} , and thus, from (2.34), we have:

$$\begin{aligned} \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\}^{11} \mathbf{q}_\beta^1 &= \{\theta_\alpha, \theta_\beta\}^{11} (\mathbf{S}_\beta^1 - \mathbf{B}_\beta^1) - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\}^{11} \mathbf{F}_{k\beta}^1 \\ &+ \{\theta_\alpha, \theta_\beta\}^{10} (\mathbf{S}_\beta^0 - \mathbf{B}_\beta^0) - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\}^{10} \mathbf{F}_{k\beta}^0 \end{aligned} \quad (2.47)$$

Define the following:

$$U_{\alpha\beta} = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \tau} \right\}^{11} \quad (2.48a)$$

$$V_{\alpha\beta}^k = \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\}^{11} \quad (2.48b)$$

$$\mathbf{W}_\alpha = \{\theta_\alpha, \theta_\beta\}^{10} (\mathbf{S}_\beta - \mathbf{B}_\beta)^0 - \left\{ \theta_\alpha, \frac{\partial \theta_\beta}{\partial \chi^{(k)}} \right\}^{10} \mathbf{F}_{k\beta}^0 \quad (2.48c)$$

$$Z_{\alpha\beta} = \{\theta_\alpha, \theta_\beta\}^{11} \quad (2.48d)$$

Thus:

$$U_{\alpha\beta} \mathbf{q}_\beta^1 = \mathbf{W}_\alpha + Z_{\alpha\beta} (\mathbf{S}_\beta^1 - \mathbf{B}_\beta^1) - V_{\alpha\beta}^k \mathbf{F}_{k\beta}^1 \quad (2.49)$$

Note that, as with the discontinuous Galerkin method, \mathbf{W} has no dependence on the degrees of freedom in \mathbf{q} . As the source terms are not stiff, the following iteration is used:

$$U_{\alpha\beta} (\mathbf{q}_\beta^1)_{m+1} = \mathbf{W}_\alpha + Z_{\alpha\beta} ((\mathbf{S}_\beta^1)_m - (\mathbf{B}_\beta^1)_m) - V_{\alpha\beta}^k (\mathbf{F}_{k\beta}^1)_m \quad (2.50)$$

2.2.3 Convergence Properties

In [12] it was proved that for traditional choices of polynomial bases, the eigenvalues of $U^{-1}V^i$ are all 0 for any $N \in \mathbb{N}$, for $i = 1, 2, 3$. This implies that in the conservative, homogeneous case ($\vec{\mathbf{B}} = \mathbf{S} = \mathbf{0}$), owing to the Banach Fixed Point Theorem, existence and uniqueness of a solution are established, and convergence to this solution is guaranteed. As noted in [?], in the linear case it is implied that the iterative procedure converges after at most $N + 1$ iterations. A proof of this result for the Montecinos-Balsara polynomial basis class is now provided here. For the theory in linear algebra required for this section, please consult a standard textbook on the subject, such as [?].

2.2 Extension of the Montecinos-Balsara ADER-WENO Method

Take the definitions (2.38), (2.40). Consider that:

$$U^{-1}V^k = E^{-1}M^\tau \otimes I^{k-1} \otimes D \otimes I^{3-k} \quad (2.51)$$

Therefore:

$$(U^{-1}V^k)^m = (E^{-1}M^\tau)^m \otimes (I^{k-1})^m \otimes D^m \otimes (I^{3-k})^m \quad (2.52)$$

A matrix X is nilpotent ($X^k = 0$ for some $k \in \mathbb{N}$) if and only if all its eigenvalues are 0. Note that $U^{-1}V^k$ is nilpotent if $D^m = 0$ for some $m \in \mathbb{N}$.

Note that if $p \in P_N$ then $p = a_j \psi_j$ for some unique coefficient vector \mathbf{a} . Thus, taking inner products with ψ_i , we have $\langle \psi_i, \psi_j \rangle a_j = \langle \psi_i, p \rangle$ for $i = 0, \dots, N$. This produces the following result:

$$p = a_j \psi_j \Leftrightarrow \mathbf{a} = (M^X)^{-1} \mathbf{x}, \quad x_i = \langle \psi_i, p \rangle \quad (2.53)$$

Taking $\mathbf{a} \in \mathbb{R}^{N+1}$, define:

$$p = a_0 \psi_0 + \dots + a_N \psi_N \in P_N \quad (2.54)$$

Note that:

$$(M^{X,1} \mathbf{a})_i = \langle \psi_i, \psi'_0 \rangle a_0 + \dots + \langle \psi_i, \psi'_N \rangle a_N = \langle \psi_i, p' \rangle \quad (2.55)$$

Thus, by (2.53):

$$((M^X)^{-1} M^{X,1} \mathbf{a})_i \psi_i = (D\mathbf{a})_i \psi_i = p' \quad (2.56)$$

By induction:

$$(D^m \mathbf{a})_i \psi_i = p^{(m)} \quad (2.57)$$

for any $m \in \mathbb{N}$. As $p \in P_N$, $D^{N+1} \mathbf{a} = \mathbf{0}$. As \mathbf{a} was chosen arbitrarily, $D^{N+1} = 0$. No specific choice has been made for $N \in \mathbb{N}$ and thus the result holds in general.

2.3 An Operator Splitting Scheme for the GPR Model

Thus, in the case that $\vec{B} = S = 0$, existence and uniqueness of a solution are established, and convergence to this solution is guaranteed for the iterative solution to (2.39) in the Discontinuous Galerkin case, and (2.49) in the Continuous Galerkin case.

2.3 An Operator Splitting Scheme for the GPR Model

Note that (1a), (1b), (1c), (1d), (3) can be written in the following form:

$$\frac{\partial Q}{\partial t} + \nabla \cdot F(Q) + B(Q) \cdot \nabla Q = S(Q) \quad (2.58)$$

As described in [22], a viable way to solve inhomogeneous systems of PDEs is to employ an operator splitting. That is, the following subsystems are solved:

$$\frac{\partial Q}{\partial t} + \nabla \cdot F(Q) + B(Q) \cdot \nabla Q = 0 \quad (2.59a)$$

$$\frac{dQ}{dt} = S(Q) \quad (2.59b)$$

The advantage of this approach is that specialized solvers can be employed to compute the results of the different subsystems. Let $H^{\delta t}, S^{\delta t}$ be the operators that take data $Q(x, t)$ to $Q(x, t + \delta t)$ under systems (2.59a) and (2.59b) respectively. A second-order scheme (in time) for solving the full set of PDEs over time step $[0, \Delta t]$ is obtained by calculating $Q_{\Delta t}$ using a Strang splitting:

$$Q_{\Delta t} = S^{\frac{\Delta t}{2}} H^{\Delta t} S^{\frac{\Delta t}{2}} Q_0 \quad (2.60)$$

In the scheme proposed here, the homogeneous subsystem will be solved using a WENO reconstruction of the data, followed by a finite volume update, and the temporal ODEs will be solved with appropriate ODE solvers. This new scheme will be referred to here as *the Split-WENO method*.

2.3.1 The Homogeneous System

A WENO reconstruction of the cell-averaged data is performed at the start of the time step (as described in ??). Focusing on a single cell C_i at time t_n , we have $w^n(x) =$

2.3 An Operator Splitting Scheme for the GPR Model

$w_p^n \Psi_p(\chi(x))$ in C_i where Ψ_p is a tensor product of basis functions in each of the spatial dimensions. The flux in C is approximated by $F(x) \approx F(w_p) \Psi_p(\chi(x))$. w_p are stepped forwards half a time step using the update formula:

$$\frac{w_p^{n+\frac{1}{2}} - w_p^n}{\Delta t/2} + F(w_k^n) \cdot \nabla \Psi_k(\chi_p) + B(w_p^n) \cdot (w_k^n \nabla \Psi_k(\chi_p)) = 0 \quad (2.61)$$

i.e.

$$w_p^{n+\frac{1}{2}} = w_p^n - \frac{\Delta t}{2\Delta x} \left(F(w_k^n) \cdot \nabla \Psi_k(\chi_p) + B(w_p^n) \cdot (w_k^n \nabla \Psi_k(\chi_p)) \right) \quad (2.62)$$

where χ_p is the node corresponding to Ψ_p . This evolution to the middle of the time step is similar to that used in the second-order MUSCL and SLIC schemes (see [22]) and, as with those schemes, it is integral to giving the method presented here its second-order accuracy.

Integrating (2.59a) over C gives:

$$Q_i^{n+1} = Q_i^n - \Delta t_n \left(P_i^{n+\frac{1}{2}} + D_i^{n+\frac{1}{2}} \right) \quad (2.63)$$

where

$$Q_i^n = \frac{1}{V} \int_C Q(x, t_n) dx \quad (2.64a)$$

$$P_i^{n+\frac{1}{2}} = \frac{1}{V} \int_C B(Q(x, t_{n+\frac{1}{2}})) \cdot \nabla Q(x, t_{n+\frac{1}{2}}) dx \quad (2.64b)$$

$$D_i^{n+\frac{1}{2}} = \frac{1}{V} \oint_{\partial C} \mathcal{D}(Q^-(s, t_{n+\frac{1}{2}}), Q^+(s, t_{n+\frac{1}{2}})) ds \quad (2.64c)$$

where V is the volume of C and Q^-, Q^+ are the interior and exterior extrapolated states at the boundary of C , respectively.

Note that (2.59a) can be rewritten as:

$$\frac{\partial Q}{\partial t} + M(Q) \cdot \nabla Q = 0 \quad (2.65)$$

where $M = \frac{\partial F}{\partial Q} + B$. Let n be the normal to the boundary at point $s \in \partial C$. For the GPR model, $\hat{M} = M(Q(s)) \cdot n$ is a diagonalizable matrix with decomposition $\hat{M} = \hat{R} \hat{\Lambda} \hat{R}^{-1}$

2.3 An Operator Splitting Scheme for the GPR Model

where the columns of \hat{R} are the right eigenvectors and $\hat{\Lambda}$ is the diagonal matrix of eigenvalues. Define also $\hat{F} = F \cdot n$ and $\hat{B} = B \cdot n$. Using these definitions, the interface terms arising in the FV formula have the following form:

$$\mathcal{D}(Q^-, Q^+) = \frac{1}{2} \left(\hat{F}(Q^+) + \hat{F}(Q^-) + \tilde{B}(Q^+ - Q^-) + \tilde{M}(Q^+ - Q^-) \right) \quad (2.66)$$

\tilde{M} is chosen to either correspond to a Rusanov/Lax-Friedrichs flux (see [22]):

$$\tilde{M} = \max \left(\max |\hat{\Lambda}(Q^+)|, \max |\hat{\Lambda}(Q^-)| \right) \quad (2.67)$$

or a simplified Osher–Solomon flux (see [7, 8]):

$$\tilde{M} = \int_0^1 |\hat{M}(Q^- + z(Q^+ - Q^-))| dz \quad (2.68)$$

where

$$|\hat{M}| = \hat{R} |\hat{\Lambda}| \hat{R}^{-1} \quad (2.69)$$

\tilde{B} takes the following form:

$$\tilde{B} = \int_0^1 \hat{B}(Q^- + z(Q^+ - Q^-)) dz \quad (2.70)$$

It was found that the Osher–Solomon flux would often produce slightly less diffusive results, but that it was more computationally expensive, and also had a greater tendency to introduce numerical artefacts.

$P_i^{n+\frac{1}{2}}, D_i^{n+\frac{1}{2}}$ are calculated using an $N + 1$ -point Gauss-Legendre quadrature, replacing $Q(x, t_{n+\frac{1}{2}})$ with $w^{n+\frac{1}{2}}(x)$.

2.3.2 The Temporal ODEs

Noting that $\frac{d\rho}{dt} = 0$ over the ODE time step, the operator S entails solving the following systems:

$$\frac{dA}{dt} = \frac{-3}{\tau_1} |A|^{\frac{5}{3}} A \operatorname{dev}(G) \quad (2.71a)$$

$$\frac{d\mathbf{J}}{dt} = -\frac{1}{\tau_2} \frac{T\rho_0}{T_0\rho} \mathbf{J} \quad (2.71b)$$

These systems can be solved concurrently with a stiff ODE solver. The Jacobians of these two systems to be used in an ODE solver are given in [D.1](#) and [D.2](#). However, these systems can also be solved separately, using the analytical results presented in [Section 2.3.3](#), under specific assumptions. The second-order Strang splitting is then:

$$\mathbf{Q}_{\Delta t} = D^{\frac{\Delta t}{2}} T^{\frac{\Delta t}{2}} H^{\Delta t} T^{\frac{\Delta t}{2}} D^{\frac{\Delta t}{2}} \mathbf{Q}_0 \quad (2.72)$$

where $D^{\delta t}, T^{\delta t}$ are the operators solving the distortion and thermal impulse ODEs respectively, over timestep δt . This allows us to bypass the relatively computationally costly process of solving these systems numerically.

2.3.3 GPR-Specific Performance Improvements

2.3.3.1 The Thermal Impulse ODEs

Taking the EOS for the GPR model (??) and denoting by $E_2^{(A)}, E_2^{(J)}$ the components of E_2 depending on A and \mathbf{J} respectively, we have:

$$\begin{aligned} T &= \frac{E_1}{c_v} \\ &= \frac{E - E_2^{(A)}(A) - E_3(\mathbf{v})}{c_v} - \frac{1}{c_v} E_2^{(J)}(\mathbf{J}) \\ &= c_1 - c_2 \|\mathbf{J}\|^2 \end{aligned} \quad (2.73)$$

2.3 An Operator Splitting Scheme for the GPR Model

where:

$$c_1 = \frac{E - E_2^{(A)}(A) - E_3(\mathbf{v})}{c_v} \quad (2.74a)$$

$$c_2 = \frac{\alpha^2}{2c_v} \quad (2.74b)$$

Over the time period of the ODE (2.71b), $c_1, c_2 > 0$ are constant. We have:

$$\frac{dJ_i}{dt} = - \left(\frac{1}{\tau_2} \frac{\rho_0}{T_0 \rho} \right) J_i (c_1 - c_2 \|\mathbf{J}\|^2) \quad (2.75)$$

Therefore:

$$\frac{d}{dt} (J_i^2) = J_i^2 (-a + b (J_1^2 + J_2^2 + J_3^2)) \quad (2.76)$$

where

$$a = \frac{2\rho_0}{\tau_2 T_0 \rho c_v} (E - E_2^{(A)}(A) - E_3(\mathbf{v})) \quad (2.77a)$$

$$b = \frac{\rho_0 \alpha^2}{\tau_2 T_0 \rho c_v} \quad (2.77b)$$

Note that this is a generalized Lotka-Volterra system in $\{J_1^2, J_2^2, J_3^2\}$. It has the following analytical solution:

$$\mathbf{J}(t) = \mathbf{J}(0) \sqrt{\frac{1}{e^{at} - \frac{b}{a}(e^{at} - 1) \|\mathbf{J}(0)\|^2}} \quad (2.78)$$

2.3.3.2 The Distortion ODEs

Reduced Distortion ODEs Let $k_0 = \frac{3}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{5}{3}} > 0$ and let A have singular value decomposition $U\Sigma V^T$. Then:

$$G = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^2 V^T \quad (2.79)$$

2.3 An Operator Splitting Scheme for the GPR Model

$$\text{tr}(G) = \text{tr}(V\Sigma^2V^T) = \text{tr}(\Sigma^2V^TV) = \text{tr}(\Sigma^2) \quad (2.80)$$

Therefore:

$$\begin{aligned} \frac{dA}{dt} &= -k_0 U \Sigma V^T \left(V \Sigma^2 V^T - \frac{\text{tr}(\Sigma^2)}{3} I \right) \\ &= -k_0 U \Sigma \left(\Sigma^2 - \frac{\text{tr}(\Sigma^2)}{3} I \right) V^T \\ &= -k_0 U \Sigma \text{dev}(\Sigma^2) V^T \end{aligned} \quad (2.81)$$

It is a common result (see [10]) that:

$$d\Sigma = U^T dA V \quad (2.82)$$

and thus:

$$\frac{d\Sigma}{dt} = -k_0 \Sigma \text{dev}(\Sigma^2) \quad (2.83)$$

Using a fast 3×3 SVD algorithm (such as in [15]), U, V, Σ can be obtained, after which the following procedure is applied to Σ , giving $A(t) = U \Sigma(t) V^T$.

Denote the singular values of A by a_1, a_2, a_3 . Then:

$$\Sigma \text{dev}(\Sigma^2) = \begin{pmatrix} a_1 \left(a_1^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3} \right) & 0 & 0 \\ 0 & a_2 \left(a_2^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3} \right) & 0 \\ 0 & 0 & a_3 \left(a_3^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3} \right) \end{pmatrix} \quad (2.84)$$

Letting $x_i = \frac{a_i^2}{\det(A)^{\frac{2}{3}}} = \frac{a_i^2}{\left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}}}$ we have:

$$\frac{dx_i}{d\tau} = -3x_i(x_i - \bar{x}) \quad (2.85)$$

where $\tau = \frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} t$ and \bar{x} is the arithmetic mean of x_1, x_2, x_3 . This ODE system travels along the surface $\Psi = \{x_1, x_2, x_3 > 0, x_1 x_2 x_3 = 1\}$ to the point $x_1, x_2, x_3 = 1$. This surface is symmetrical in the planes $x_1 = x_2$, $x_1 = x_3$, $x_2 = x_3$. As such, given that the system is

2.3 An Operator Splitting Scheme for the GPR Model

autonomous, the paths of evolution of the x_i cannot cross the intersections of these planes with Ψ . Thus, any non-strict inequality of the form $x_i \geq x_j \geq x_k$ is maintained for the whole history of the system. By considering (2.85) it is clear that in this case x_i is monotone decreasing, x_k is monotone increasing, and the time derivative of x_j may switch sign.

Note that we have:

$$\begin{cases} \frac{dx_i}{d\tau} = -x_i(2x_i - x_j - x_k) = -x_i\left(2x_i - x_j - \frac{1}{x_i x_j}\right) \\ \frac{dx_j}{d\tau} = -x_j(2x_j - x_k - x_i) = -x_j\left(2x_j - x_i - \frac{1}{x_i x_j}\right) \end{cases} \quad (2.86)$$

Thus, an ODE solver can be used on these two equations to effectively solve the ODEs for all 9 components of A . Note that:

$$\frac{dx_j}{dx_i} = \frac{x_j(2x_j - x_i - \frac{1}{x_i x_j})}{x_i(2x_i - x_j - \frac{1}{x_i x_j})} \quad (2.87)$$

This has solution:

$$x_j = \frac{c + \sqrt{c^2 + 4(1-c)x_i^3}}{2x_i^2} \quad (2.88)$$

where

$$c = -\frac{x_{i,0}(x_{i,0}x_{j,0}^2 - 1)}{x_{i,0} - x_{j,0}} \in (-\infty, 0] \quad (2.89)$$

In the case that $x_{i,0} = x_{j,0}$, we have $x_i = x_j$ for all time. Thus, the ODE system for A has been reduced to a single ODE, as $x_j(x_i)$ can be inserted into the RHS of the equation for $\frac{dx_i}{d\tau}$. However, it is less computationally expensive to evolve the system presented in (2.86).

Bounds on Reduced Distortion ODEs If any of the relations in $x_i \geq x_j \geq x_k$ are in fact equalities, equality is maintained throughout the history of the system. This can be seen by noting that the time derivatives of the equal variables are in this case equal. If $x_j = x_k$ then $x_i = \frac{1}{x_j^2}$. Combining these results, the path of the system in (x_i, x_j) coordinates is in fact confined to the curved triangular region:

$$\left\{ (x_i, x_j) : x_i \leq x_i^0 \cap x_i \geq x_j \cap x_i \geq \frac{1}{x_j^2} \right\} \quad (2.90)$$

2.3 An Operator Splitting Scheme for the GPR Model

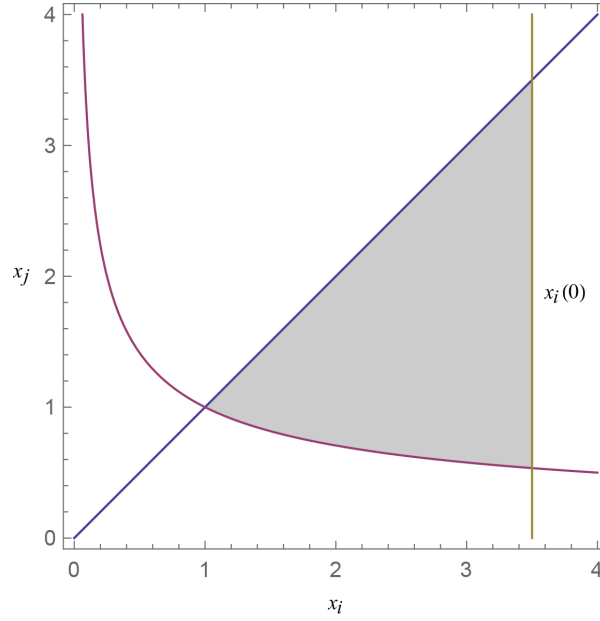


Figure 2.1: The (shaded) region to which x_i, x_j are confined in the evolution of the distortion ODEs

This is demonstrated in Figure 2.1 on page 36. By (2.86), the rate of change of x_i at a particular value $x_i = x_i^*$ is given by:

$$-x_i^* \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) \quad (2.91)$$

Note that:

$$\begin{aligned} \frac{d}{dx_j} \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) &= -1 + \frac{1}{x_i^* x_j^2} = 0 \\ \Rightarrow x_j &= \frac{1}{\sqrt{x_i^*}} \end{aligned} \quad (2.92)$$

$$\frac{d^2}{dx_j^2} \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) = \frac{-2}{x_i^* x_j^3} < 0 \quad (2.93)$$

Thus, x_i decreases fastest on the line $x_i = \frac{1}{x_j^2}$ (the bottom boundary of the region given in Figure 2.1 on page 36), and slowest on the line $x_i = x_j$. The rates of change of x_i along

2.3 An Operator Splitting Scheme for the GPR Model

these two lines are given respectively by:

$$\frac{dx_i}{d\tau} = -2x_i \left(x_i - \sqrt{\frac{1}{x_i}} \right) \quad (2.94a)$$

$$\frac{dx_i}{d\tau} = -x_i \left(x_i - \frac{1}{x_i^2} \right) \quad (2.94b)$$

These have implicit solutions:

$$\tau = (f(\sqrt{x_i}) + g(\sqrt{x_i})) - \left(f(\sqrt{x_i^0}) + g(\sqrt{x_i^0}) \right) \equiv F_1(x_i; x_i^0) \quad (2.95a)$$

$$\tau = (f(x_i) - g(x_i)) - \left(f(x_i^0) - g(x_i^0) \right) \equiv F_2(x_i; x_i^0) \quad (2.95b)$$

where

$$f(x_i) = \frac{1}{6} \log \left(\frac{x_i^2 + x_i + 1}{(x_i - 1)^2} \right) \quad (2.96a)$$

$$g(x_i) = \frac{1}{\sqrt{3}} \tan^{-1} \left(\frac{2x_i + 1}{\sqrt{3}} \right) \quad (2.96b)$$

As (2.85) is an autonomous system of ODEs, it has the property that its limit $x_1 = x_2 = x_3 = 1$ is never obtained in finite time, in precise arithmetic. In floating point arithmetic we may say that the system has converged when $x_i - 1 < \epsilon$ (machine epsilon) for each i . This happens when:

$$\tau > F_2(1 + \epsilon; x_i^0) \quad (2.97)$$

This provides a quick method to check whether it is necessary to run the ODE solver in a particular cell. If the following condition is satisfied then we know the system in that cell converges to the ground state over the time interval in which the ODE system is calculated:

$$\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t > F_2(1 + \epsilon; \max \{x_i^0\}) \quad (2.98)$$

If the fluid is very inviscid, resulting in a stiff ODE, the critical time is lower, and there is more chance that the ODE system in the cell reaches its limit in Δt . This check potentially saves a lot of computationally expensive stiff ODE solves. The same goes for if the flow is

2.3 An Operator Splitting Scheme for the GPR Model

slow-moving, as the system will be closer to its ground state at the start of the time step and is more likely to converge over Δt . Similarly, if the following condition is satisfied then we know for sure that an ODE solver is necessary, as the system certainly will not have converged over the timestep:

$$\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t < F_1 \left(1 + \epsilon; \max \{x_i^0\} \right) \quad (2.99)$$

Analytical Approximation We now explore cases when even the reduced ODE system (2.86) need not be solved numerically. Define the following variables:

$$m = \frac{x_1 + x_2 + x_3}{3} \quad (2.100a)$$

$$u = \frac{(x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_1)^2}{3} \quad (2.100b)$$

It is a standard result that $m \geq \sqrt[3]{x_1 x_2 x_3}$. Thus, $m \geq 1$. Note that u is proportional to the internal energy contribution from the distortion. From (2.85) we have:

$$\frac{du}{d\tau} = -18 \left(1 - m \left(m^2 - \frac{5}{6}u \right) \right) \quad (2.101a)$$

$$\frac{dm}{d\tau} = -u \quad (2.101b)$$

Combining these equations, we have:

$$\frac{d^2 m}{d\tau^2} = -\frac{du}{d\tau} = 18 \left(1 - m \left(m^2 - \frac{5}{6}u \right) \right) \quad (2.102)$$

Therefore:

$$\begin{cases} \frac{d^2 m}{d\tau^2} + 15m \frac{dm}{d\tau} + 18(m^3 - 1) = 0 \\ m(0) = m_0 \\ m'(0) = -u_0 \end{cases} \quad (2.103)$$

We make the following assumption, noting that it is true in all physical situations tested in this study:

$$m(t) = 1 + \eta(t), \quad \eta \ll 1 \quad \forall t \geq 0 \quad (2.104)$$

2.3 An Operator Splitting Scheme for the GPR Model

Thus, we have the linearized ODE:

$$\begin{cases} \frac{d^2\eta}{d\tau^2} + 15\frac{d\eta}{d\tau} + 54\eta = 0 \\ \eta(0) = m_0 - 1 \\ \eta'(0) = -u_0 \end{cases} \quad (2.105)$$

This is a Sturm-Liouville equation with solution:

$$\eta(\tau) = \frac{e^{-9\tau}}{3} \left((9m_0 - u_0 - 9)e^{3\tau} - (6m_0 - u_0 - 6) \right) \quad (2.106)$$

Thus, we also have:

$$u(\tau) = e^{-9\tau} \left(e^{3\tau} (18m_0 - 2u_0 - 18) - (18m_0 - 3u_0 - 18) \right) \quad (2.107)$$

Once $m_{\Delta t} = 1 + \eta \left(\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t \right)$ and $u_{\Delta t} = u \left(\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t \right)$ have been found, we have:

$$\frac{x_i + x_j + x_k}{3} = m_{\Delta t} \quad (2.108a)$$

$$\frac{(x_i - x_j)^2 + (x_j - x_k)^2 + (x_k - x_i)^2}{3} = u_{\Delta t} \quad (2.108b)$$

$$x_i x_j x_k = 1 \quad (2.108c)$$

This gives:

$$x_i = \frac{\sqrt[3]{6 \left(\sqrt{81\Delta^2 - 6u_{\Delta t}^3} + 9\Delta \right)}}{6} + \frac{u_{\Delta t}}{\sqrt[3]{6 \left(\sqrt{81\Delta^2 - 6u_{\Delta t}^3} + 9\Delta \right)}} + m_{\Delta t} \quad (2.109a)$$

$$x_j = \frac{1}{2} \left(\sqrt{\frac{x_i (3m_{\Delta t} - x_i)^2 - 4}{x_i}} + 3m_{\Delta t} - x_i \right) \quad (2.109b)$$

$$x_k = \frac{1}{x_i x_j} \quad (2.109c)$$

where

$$\Delta = -2m_{\Delta t}^3 + m_{\Delta t} u_{\Delta t} + 2 \quad (2.110)$$

2.3 An Operator Splitting Scheme for the GPR Model

Note that taking the real parts of the above expression for x_i gives:

$$x_i = \frac{\sqrt{6u_{\Delta t}}}{3} \cos\left(\frac{\theta}{3}\right) + m_{\Delta t} \quad (2.111a)$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{6u_{\Delta t}^3 - 81\Delta^2}}{9\Delta}\right) \quad (2.111b)$$

At this point it is not clear which values of $\{x_i, x_j, x_k\}$ are taken by x_1, x_2, x_3 . However, this can be inferred from the fact that any relation $x_i \geq x_j \geq x_k$ is maintained over the lifetime of the system. Thus, the stiff ODE solver has been obviated by a few arithmetic operations.

2.3.4 Numerical Results

2.3.4.1 Strain Relaxation

In this section, the approximate analytic solver for the distortion ODEs, presented in 2.3.3.2, is compared with a numerical ODE solver. Initial data was taken from [2]:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{pmatrix}^{-1} \quad (2.112)$$

Additionally, the following parameter values were used: $\rho_0 = 1, c_s = 1, \mu = 10^{-2}$, giving $\tau_1 = 0.06$. As can be seen in Figure 2.2 on page 41, Figure 2.3 on page 41, and Figure 2.4 on page 41, the approximate analytic solver compares well with the numerical solver in its results for the distortion tensor A , and thus also the internal energy and stress tensor. The numerical ODE solver was the odeint solver from SciPy 0.18.1, based on the LSODA solver from the FORTRAN library ODEPACK (see [17]).

2.3.4.2 Stokes' First Problem

This problem is one of the few test cases with an analytic solution for the Navier-Stokes equations. It consists of two ideal gases in an infinite domain, meeting at the plane $x = 0$,

2.3 An Operator Splitting Scheme for the GPR Model

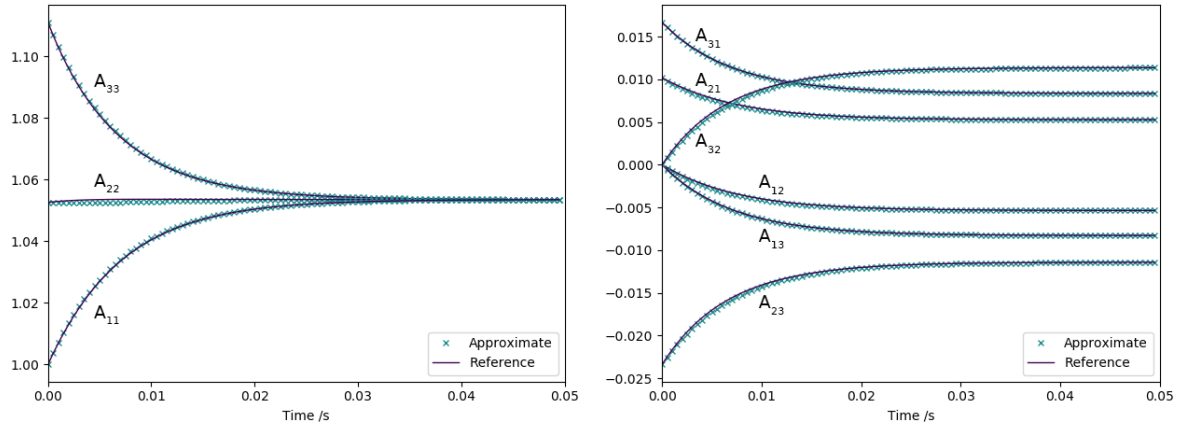


Figure 2.2: The components of the distortion tensor in the Strain Relaxation Test

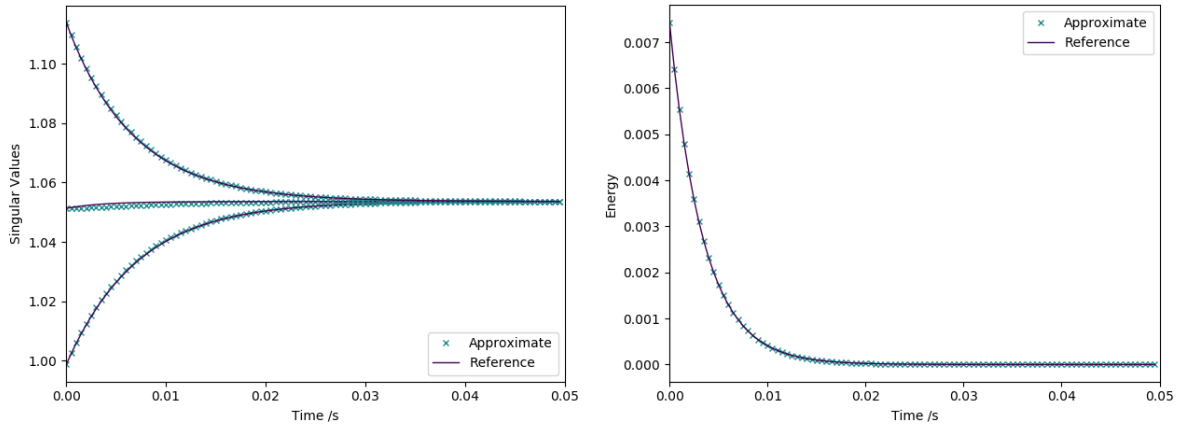


Figure 2.3: The singular values of the distortion tensor and the energy in the Strain Relaxation Test

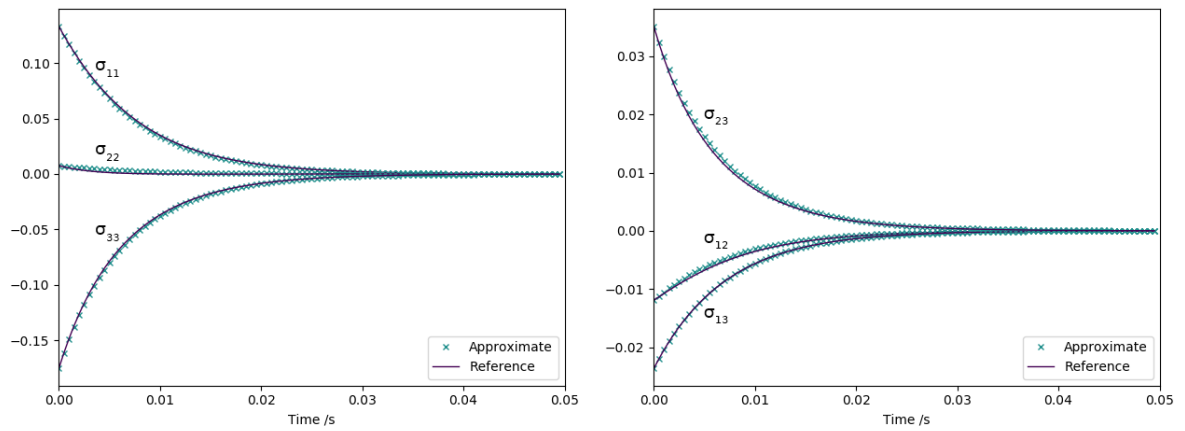


Figure 2.4: The components of the stress tensor in the Strain Relaxation Test

2.3 An Operator Splitting Scheme for the GPR Model

	ρ	p	\mathbf{v}	A	\mathbf{J}
$x < 0$	1	$1/\gamma$	$(0, -0.1, 0)$	I_3	$\mathbf{0}$
$x \geq 0$	1	$1/\gamma$	$(0, 0.1, 0)$	I_3	$\mathbf{0}$

Table 2.1: Initial conditions for the slow opposing shear flow test

initially flowing with equal and opposite velocity ± 0.1 in the y -axis. The initial conditions are given in Table 2.1 on page 42.

The flow has a low Mach number of 0.1, and this test case is designed to demonstrate the efficacy of the numerical methods in this flow regime. The exact solution to the Navier-Stokes equations is given by¹:

$$v = v_0 \operatorname{erf} \left(\frac{x}{2\sqrt{\mu t}} \right) \quad (2.113)$$

Heat conduction is neglected, and $\gamma = 1.4$, $c_v = 1$, $\rho_0 = 1$, $c_s = 1$. The viscosity is variously taken to be $\mu = 10^{-2}$, $\mu = 10^{-3}$, $\mu = 10^{-4}$ (resulting in $\tau_1 = 0.06$, $\tau_1 = 0.006$, $\tau_1 = 0.0006$, respectively). Due to the stiffness of the source terms in the equations governing A in the case that $\mu = 10^{-4}$, the step (2.62) in the WENO reconstruction under the Split-WENO method was not performed, and $\mathbf{w}_p^{n+\frac{1}{2}} \equiv \mathbf{w}_p^n$ was taken instead. This avoided the numerical diffusion that otherwise would have emerged at the interface at $x = 0$.

The results of simulations with 200 cells at time $t = 1$, using reconstruction polynomials of order $N = 2$, are presented in Figure 2.5 on page 43. The GPR model solved with both the ADER-WENO and Split-WENO methods closely matches the exact Navier-Stokes solution. Note that at $\mu = 10^{-2}$ and $\mu = 10^{-3}$, the ADER-WENO and Split-WENO methods are almost indistinguishable. At $\mu = 10^{-4}$ the Split-WENO method matches the curve of the velocity profile more closely, but overshoots slightly at the boundaries of the center region. This overshoot phenomenon is not visible in the ADER-WENO results.

¹In this problem, the Navier-Stokes equations reduce to $v_t = \mu v_{xx}$. Defining $\eta = \frac{x}{2\sqrt{\mu t}}$, and assuming $v = f(\eta)$, this becomes $f'' + 2\eta f' = 0$. The result follows by solving this equation with the boundary conditions $v(\pm\infty) = \pm v_0$.

2.3 An Operator Splitting Scheme for the GPR Model

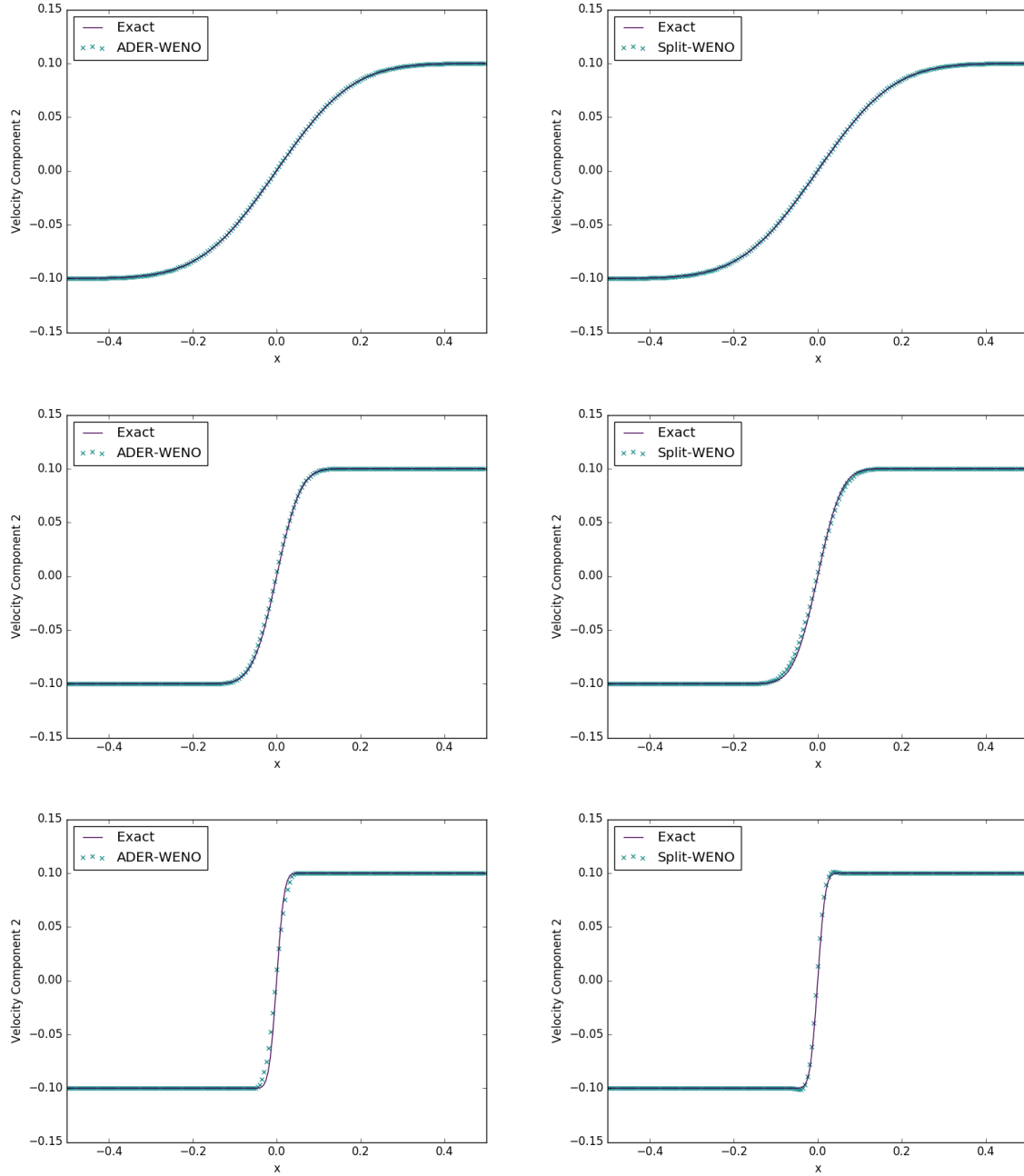


Figure 2.5: Results of solving Stokes' First Problem ($\mu = 10^{-2}, \mu = 10^{-3}, \mu = 10^{-4}$) with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

2.3.4.3 Viscous Shock

This test is designed to demonstrate that the numerical methods used are also able to cope with fast flows. First demonstrated by Becker [?], the Navier-Stokes equations have an analytic solution for $P_r = 0.75$ (see Johnson [?] for a full analysis). As noted by Dumbser [6], if the wave has nondimensionalised upstream velocity $\bar{v} = 1$ and Mach number M_c , then its nondimensionalised downstream velocity is:

$$a = \frac{1 + \frac{\gamma-1}{2}M_c^2}{\frac{\gamma+1}{2}M_c^2} \quad (2.114)$$

The wave's velocity profile $\bar{v}(x)$ is given by the roots of the following equation:

$$\frac{1 - \bar{v}}{(\bar{v} - a)^a} = c_1 \exp(-c_2 x) \quad (2.115a)$$

$$c_1 = \left(\frac{1 - a}{2}\right)^{1-a} \quad (2.115b)$$

$$c_2 = \frac{3}{4}R_e \frac{M_c^2 - 1}{\gamma M_c^2} \quad (2.115c)$$

c_1, c_2 are constants that affect the position of the center of the wave, and its stretch factor, respectively. Following the analysis of Morduchow and Libby [?], the nondimensional pressure and density profiles are given by:

$$\bar{p} = \frac{1}{\bar{v}} \left(1 + \frac{\gamma-1}{2}M_c^2 (1 - \bar{v}^2) \right) \quad (2.116)$$

$$\bar{\rho} = \frac{1}{\bar{v}} \quad (2.117)$$

To obtain an unsteady shock traveling into a region at rest, a constant velocity field $v = M_c c_0$ is imposed on the traveling wave solution presented here (where c_0 is the adiabatic sound speed). Thus, if p_0, ρ_0 are the downstream (reference) values for pressure and density:

$$v = M_c c_0 (1 - \bar{v}) \quad (2.118a)$$

$$p = p_0 \bar{p} \quad (2.118b)$$

$$\rho = \rho_0 \bar{\rho} \quad (2.118c)$$

2.3 An Operator Splitting Scheme for the GPR Model

	ρ	p	\mathbf{v}	A	\mathbf{J}
$x < 0$	2	1	$\mathbf{0}$	$\sqrt[3]{2} \cdot I_3$	$\mathbf{0}$
$x \geq 0$	0.5	1	$\mathbf{0}$	$\frac{1}{\sqrt[3]{2}} \cdot I_3$	$\mathbf{0}$

Table 2.2: Initial conditions for the heat conduction test

These functions are used as initial conditions, along with $A = \sqrt[3]{\rho}I$ and $\mathbf{J} = \mathbf{0}$. The downstream density and pressure are taken to be $\rho_0 = 1$ and $p_0 = \frac{1}{\gamma}$ (so that $c_0 = 1$). $M_c = 2$ and $R_e = 100$. The material parameters are taken to be: $\gamma = 1.4$, $p_\infty = 0$, $c_v = 2.5$, $c_s = 5$, $\alpha = 5$, $\mu = 2 \times 10^{-2}$, $\kappa = \frac{28}{3} \times 10^{-2}$ (resulting in $\tau_1 = 0.0048$, $\tau_2 = 0.005226$).

The results of a simulation with 200 cells at time $t = 0.2$, using reconstruction polynomials of order $N = 2$, are presented in Figure 2.6 on page 46 and Figure 2.7 on page 47. The shock was initially centered at $x = 0.25$, reaching $x = 0.65$ at the final time. Note that the density, velocity, and pressure results for both methods match the exact solution well, with the ADER-WENO method appearing to produce a slightly more accurate solution. The results for the two methods for the stress tensor and heat flux are close.

2.3.4.4 Heat Conduction in a Gas

This is a simple test case to ensure that the heat transfer terms in the implementation are working correctly. Two ideal gases at different temperatures are initially in contact at position $x = 0$. The initial conditions for this problem are given in Table 2.2 on page 45.

The material parameters are taken to be: $\gamma = 1.4$, $c_v = 2.5$, $\rho_0 = 1$, $p_0 = 1$, $c_s = 1$, $\alpha = 2$, $\mu = 10^{-2}$, $\kappa = 10^{-2}$ (resulting in $\tau_1 = 0.06$, $\tau_2 = 0.0025$). The results of a simulation with 200 cells at time $t = 1$, using reconstruction polynomials of order $N = 2$, are presented in Figure 2.8 on page 47. The ADER-WENO and Split-WENO methods are in perfect agreement for both the temperature and heat flux profiles. As demonstrated in [6], this means that they in turn agree very well with a reference Navier-Stokes-Fourier solution.

2.3.4.5 Speed

Both the ADER-WENO scheme and the Split-WENO scheme used in this study were implemented in Python3. All array functions were precompiled with Numba's JIT capabili-

2.3 An Operator Splitting Scheme for the GPR Model

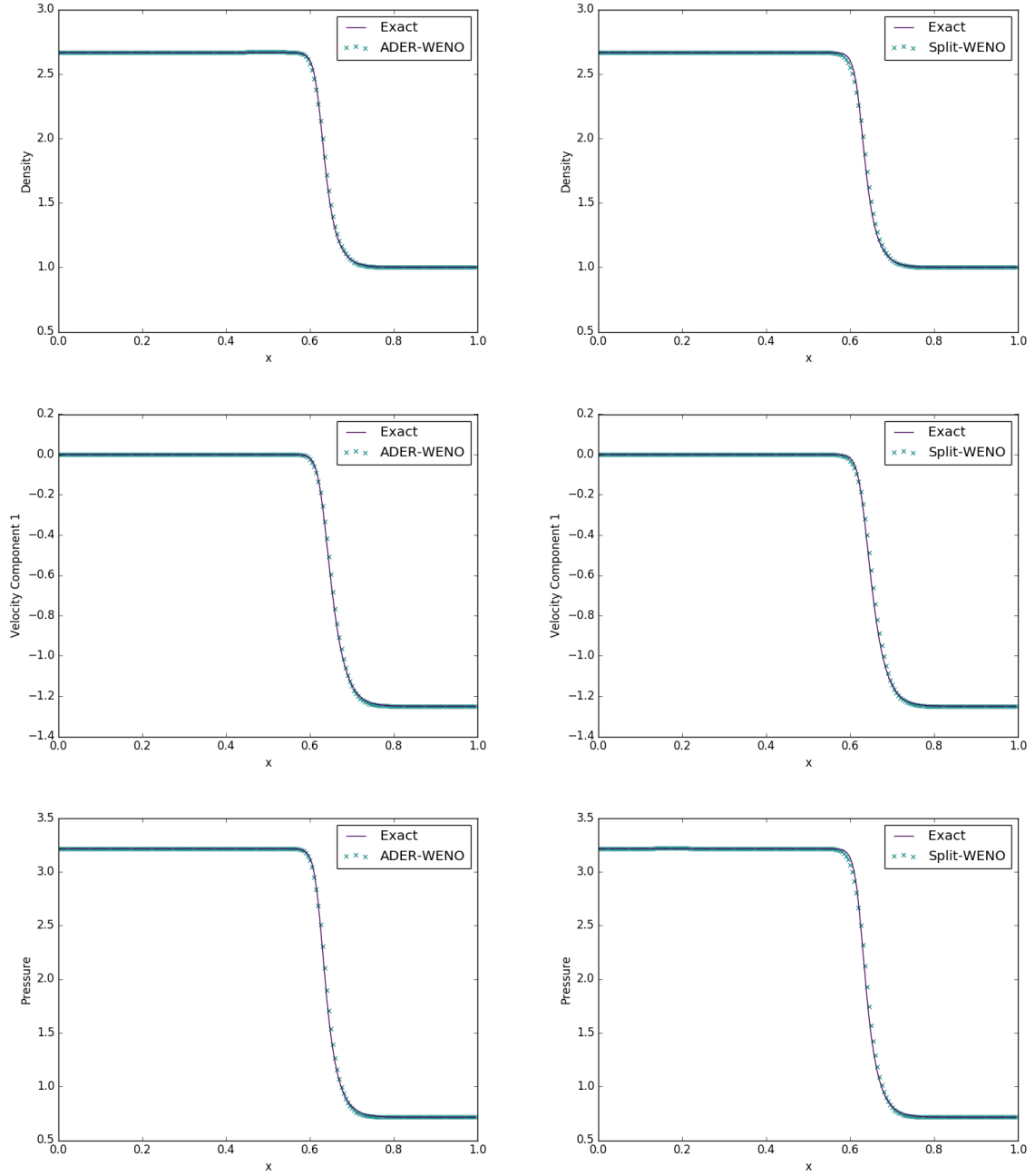


Figure 2.6: Density, velocity, and pressure for the Viscous Shock problem, solved with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

2.3 An Operator Splitting Scheme for the GPR Model

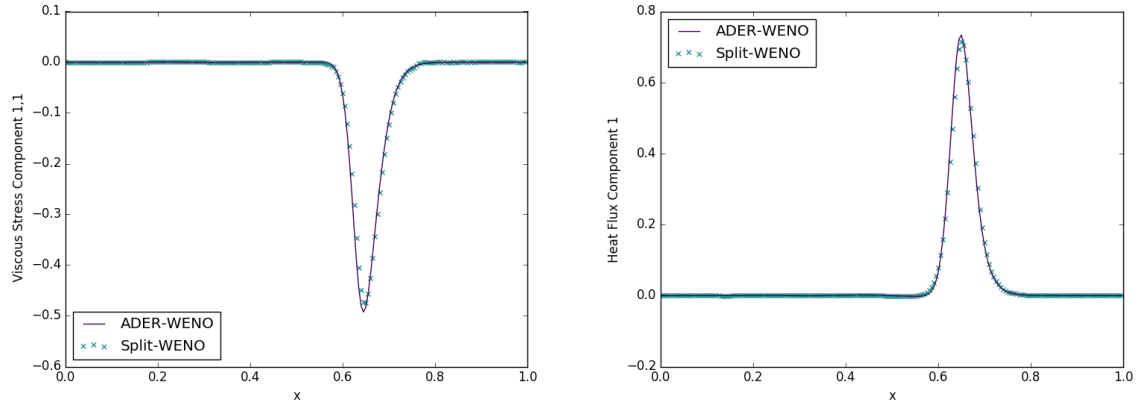


Figure 2.7: Viscous stress and heat flux for the Viscous Shock problem, solved with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

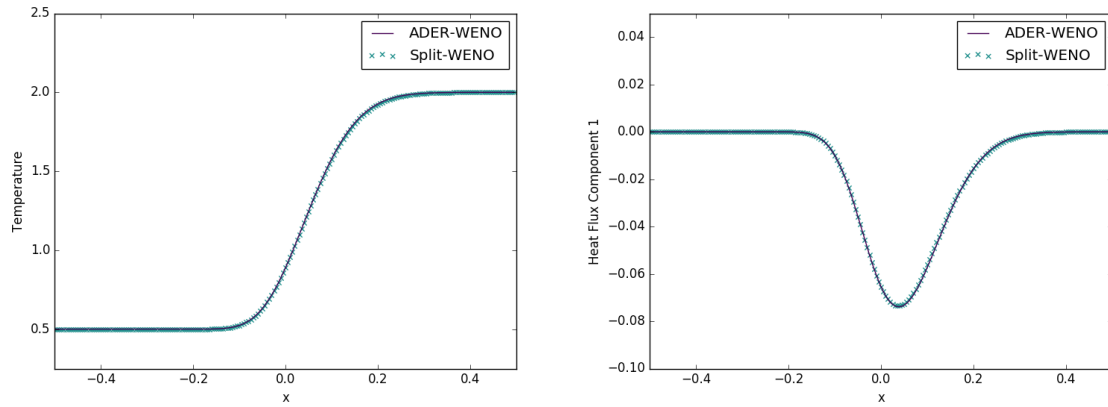


Figure 2.8: Results of solving the problem of Heat Conduction in Gas with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

2.3 An Operator Splitting Scheme for the GPR Model

ties and the root-finding procedure in the Galerkin predictor was performed using SciPy's Newton-Krylov solver, compiled against the Intel MKL. Clear differences in computational cost between the ADER-WENO and Split-WENO methods were apparent, as is to be expected, owing to the lack of Galerkin method in the Split-WENO scheme. The wall times for the various tests undertaken in this study are given in Table 2.3 on page 50, comparing the combined WENO and Galerkin methods of the ADER-WENO scheme to the combined WENO and ODE methods of the Split-WENO scheme. All computations were performed using an Intel Core i7-4910MQ, on a single core. The number of time steps taken are given in Table 2.4 on page 50. The differences between the methods in terms of the number of time steps taken in each test result from the fact that, for numerical stability, CFL numbers of 0.8 and 0.7 were required by the ADER-WENO method and the Split-WENO method, respectively.

Note that, unlike with the ADER-WENO scheme, the wall time for the Split-WENO scheme is unaffected by a decrease in the viscosity in Stokes' First Problem (and the corresponding increase in the stiffness of the source terms). This is because the analytic approximation to the distortion ODEs obviates the need for a stiff solver. The large difference in ADER-WENO solver times between the $\mu = 10^{-3}$ and $\mu = 10^{-4}$ cases is due to the fact that, in the latter case, a stiff solver must be employed for the initial guess to the root of the nonlinear system produced by the Discontinuous Galerkin method (as described in [11]).

2.3.4.6 Convergence

To assess the rate of convergence of the Split-WENO method, the convected isentropic vortex convergence study from [6] was performed. The initial conditions are given as $\rho = 1 + \delta\rho$, $p = 1 + \delta p$, $\mathbf{v} = (1, 1, 0) + \delta\mathbf{v}$, $A = \sqrt[3]{\rho}I$, $\mathbf{J} = \mathbf{0}$, where:

$$\delta T = -\frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2}e^{1-r^2} \quad (2.119a)$$

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1 \quad (2.119b)$$

$$\delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1 \quad (2.119c)$$

$$\delta\mathbf{v} = \frac{\epsilon}{2\pi}e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ x-5 \\ 0 \end{pmatrix} \quad (2.119d)$$

2.3 An Operator Splitting Scheme for the GPR Model

The 2D domain is taken to be $[0, 10]^2$. ϵ is taken to be 5. The material parameters are taken to be: $\gamma = 1.4$, $c_v = 2.5$, $\rho_0 = 1$, $p_0 = 1$, $c_s = 0.5$, $\alpha = 1$, $\mu = 10^{-6}$, $\kappa = 10^{-6}$ (resulting in $\tau_1 = 2.4 \times 10^{-5}$, $\tau_2 = 10^{-6}$). Thus, this can be considered to be a stiff test case.

The convergence rates in the L_1 , L_2 , L_∞ norms for the density variable are given in Table 2.5 on page 50 and Table 2.6 on page 50 for WENO reconstruction polynomial orders of $N = 2$ and $N = 3$, respectively. As expected, both sets of tests attain roughly second order convergence. For comparison, the corresponding results for this test from [6] - solved using a third-order P2P2 scheme - are given in Table 2.7 on page 52 for comparison.

2.3.5 Conclusions

In summary, a new numerical method based on an operator splitting, and including some analytical results, has been proposed for the GPR model of continuum mechanics. It has been demonstrated that this method is able to match current ADER-WENO methods in terms of accuracy on a range of test cases. It is significantly faster than the other currently available methods, and it is easier to implement. The author would recommend that if very high order-of-accuracy is required, and computational cost is not important, then ADER-WENO methods may present a better option, as by design the new method cannot achieve better than second-order accuracy. This new method clearly has applications in which it will prove useful, however.

In a similar manner to the operator splitting method presented in [13], the Split-WENO method is second-order accurate and stable even for very stiff problems (in particular, the reader is referred to the results of the $\mu = 10^{-4}$ variation of Stokes' First Problem in 2.3.4.2 and the convergence study in 2.3.4.6). However, it will inevitably suffer from the incorrect speed of propagation of discontinuities on regular, structured grids. This is due to a lack of spatial resolution in evaluating the source terms, as detailed in [13]. This issue can be rectified by the use of some form of shock tracking or mesh refinement, as noted in the cited paper. It is noted in [4] that operator splitting-based methods can result in schemes that are neither well-balanced nor asymptotically consistent. The extent to which these two conditions are violated by the Split-WENO method - and the severity in practice of any potential violation - is a topic of further research.

It should be noted that the assumption (2.104) used to derive the approximate analytical solver may break down for situations where the flow is compressed heavily in one direction

2.3 An Operator Splitting Scheme for the GPR Model

	ADER-WENO	Split-WENO	Speed-up
Stokes' First Problem ($\mu = 10^{-2}$)	265s	38s	7.0
Stokes' First Problem ($\mu = 10^{-3}$)	294s	38s	7.7
Stokes' First Problem ($\mu = 10^{-4}$)	536s	38s	14.1
Viscous Shock	297s	56s	5.3
Heat Conduction in a Gas	544s	94s	5.8

Table 2.3: Wall time for various tests (all with 200 cells) under the ADER-WENO method and the Split-WENO method

	Timesteps (ADER-WENO)	Timesteps (Split-WENO)
Stokes' First Problem ($\mu = 10^{-2}$)	385	442
Stokes' First Problem ($\mu = 10^{-3}$)	386	443
Stokes' First Problem ($\mu = 10^{-4}$)	385	442
Viscous Shock	562	645
Heat Conduction in a Gas	942	1077

Table 2.4: Time steps taken for various tests (all with 200 cells) under the ADER-WENO method and the Split-WENO method

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$\mathcal{O}(L_1)$	$\mathcal{O}(L_2)$	$\mathcal{O}(L_\infty)$
20	2.87×10^{-3}	7.15×10^{-3}	6.21×10^{-2}			
40	5.81×10^{-4}	1.62×10^{-3}	1.73×10^{-2}	2.30	2.14	1.85
60	1.98×10^{-4}	5.39×10^{-4}	5.94×10^{-3}	2.65	2.70	2.63
80	1.23×10^{-4}	3.47×10^{-4}	3.41×10^{-3}	1.67	1.52	1.92

Table 2.5: Convergence rates for the Split-WENO method ($N = 2$)

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$\mathcal{O}(L_1)$	$\mathcal{O}(L_2)$	$\mathcal{O}(L_\infty)$
10	1.01×10^{-2}	2.58×10^{-2}	1.27×10^{-1}			
20	1.68×10^{-3}	4.02×10^{-3}	2.93×10^{-2}	2.59	2.68	2.11
30	5.34×10^{-4}	1.57×10^{-3}	1.70×10^{-2}	2.83	2.32	1.34
40	3.32×10^{-4}	8.94×10^{-4}	7.55×10^{-3}	1.65	1.95	2.82

Table 2.6: Convergence rates for the Split-WENO method ($N = 3$)

2.4 An Operator Splitting Scheme for GPR-type Continuum Models

but not the others. The reason for this is that one of the singular values of the distortion tensor will be much larger than the others, and the mean of the squares of the singular values will not be close to its geometric mean, meaning that the subsequent linearization of the ODE governing the mean of the singular values fails. It should be noted that none of the situations covered in this study presented problems for the approximate analytical solver, and situations which may be problematic are in some sense unusual. In any case, a stiff ODE solver can be used to solve the system (2.86) if necessary, utilizing the Jacobians derived in the appendix, and so the Split-WENO method is still very much usable in these situations, albeit slightly slower.

It should be noted that both the ADER-WENO and Split-WENO methods, as described in this study, are trivially parallelizable on a cell-wise basis. Thus, given a large number of computational cores, deficiencies in the Split-WENO method in terms of its order of accuracy may be overcome by utilizing a larger number of computational cells and cores. The computational cost of each time step is significantly smaller than with the ADER-WENO method, and the number of grid cells that can be used scales roughly linearly with number of cores, at constant time per iteration.

2.4 An Operator Splitting Scheme for GPR-type Continuum Models

2.4 An Operator Splitting Scheme for GPR-type Continuum Models

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$\mathcal{O}(L_1)$	$\mathcal{O}(L_2)$	$\mathcal{O}(L_\infty)$
20	9.44×10^{-3}	2.20×10^{-3}	2.16×10^{-3}			
40	1.95×10^{-3}	4.50×10^{-4}	4.27×10^{-4}	2.27	2.29	2.34
60	7.52×10^{-4}	1.74×10^{-4}	1.48×10^{-4}	2.35	2.35	2.61
80	3.72×10^{-4}	8.66×10^{-5}	7.40×10^{-5}	2.45	2.42	2.41

Table 2.7: Convergence rates for the ADER-DG PNPM method ($N, M = 2$)

Chapter 3

Objective 3: Simulating Material Interfaces

3.1 Ghost Fluid Methods

Level Set Methods

The Original Ghost Fluid Method

The Riemann Ghost Fluid Method

3.2 A Riemann Ghost Fluid Method for the GPR Model

Eigenstructure of the GPR Model

Solving the Riemann Problem

3.3 Results

Chapter 4

Impact-Induced Detonation in an Elasto-Plastic Confiner

4.1 Results

Chapter 5

Conclusions

References

- [1] D. S. BALSARA, T. RUMPF, M. DUMBSER, AND C. D. MUNZ, *Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics*, Journal of Computational Physics, 228 (2009), pp. 2480–2516.
- [2] P. T. BARTON AND D. DRIKAKIS, *An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces*, Journal of Computational Physics, 229 (2010), pp. 5518–5540.
- [3] M. DUMBSER, D. S. BALSARA, E. F. TORO, AND C. D. MUNZ, *A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes*, Journal of Computational Physics, 227 (2008), pp. 8209–8253.
- [4] M. DUMBSER, C. ENAUX, AND E. F. TORO, *Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws*, Journal of Computational Physics, 227 (2008), pp. 3971–4001.
- [5] M. DUMBSER, A. HIDALGO, AND O. ZANOTTI, *High order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems*, Computer Methods in Applied Mechanics and Engineering, 268 (2014), pp. 359–387.
- [6] M. DUMBSER, I. PESHKOV, E. ROMENSKI, AND O. ZANOTTI, *High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat-conducting fluids and elastic solids*, Journal of Computational Physics, 314 (2016), pp. 824–862.
- [7] M. DUMBSER AND E. F. TORO, *A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems*, Journal of Scientific Computing, 48 (2011), pp. 70–88.
- [8] —, *On universal Osher-type schemes for general nonlinear hyperbolic conservation laws*, Communications in Computational Physics, 10 (2011), pp. 635–671.
- [9] M. DUMBSER, O. ZANOTTI, A. HIDALGO, AND D. S. BALSARA, *ADER-WENO finite volume schemes with space-time adaptive mesh refinement*, Journal of Computational Physics, 248 (2013), pp. 257–286.

-
- [10] M. B. GILES, *An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation*, tech. rep., University of Oxford, 2008.
 - [11] A. HIDALGO AND M. DUMBSER, *ADER schemes for nonlinear systems of stiff advection-diffusion-reaction equations*, Journal of Scientific Computing, 48 (2011), pp. 173–189.
 - [12] H. JACKSON, *On the Eigenvalues of the ADER-WENO Galerkin Predictor*, Journal of Computational Physics, 333 (2017), pp. 409–413.
 - [13] R. LEVEQUE AND H. YEE, *A Study of Numerical Methods for Hyperbolic Conservation Laws with Stiff Source Terms*, Journal of Computational Physics, 86 (1990), pp. 187–210.
 - [14] A. N. MALYSHEV AND E. I. ROMENSKII, *Hyperbolic equations for heat transfer. Global solvability of the Cauchy problem*, Siberian Mathematical Journal, 27 (1986), pp. 734–740.
 - [15] A. MCADAMS, A. SELLE, R. TAMSTORF, J. TERAN, AND E. SIFAKIS, *Computing the Singular Value Decomposition of 3×3 matrices with minimal branching and elementary floating point operations*, University of Wisconsin Madison, (2011).
 - [16] G. I. MONTECINOS AND D. S. BALSARA, *A cell-centered polynomial basis for efficient Galerkin predictors in the context of ADER finite volume schemes. The one-dimensional case*, Computers & Fluids, 156 (2017), pp. 220–238.
 - [17] T. E. OLIPHANT, *SciPy: Open source scientific tools for Python*, 2007.
 - [18] I. PESHKOV AND E. ROMENSKI, *A hyperbolic model for viscous Newtonian flows*, Continuum Mechanics and Thermodynamics, 28 (2016), pp. 85–104.
 - [19] E. ROMENSKI, D. DRIKAKIS, AND E. TORO, *Conservative models and numerical methods for compressible two-phase flow*, Journal of Scientific Computing, 42 (2010), pp. 68–95.
 - [20] E. ROMENSKI, A. D. RESNYANSKY, AND E. F. TORO, *Conservative hyperbolic model for compressible two-phase flow with different phase pressures and temperatures*, Quarterly of applied mathematics, 65(2) (2007), pp. 259–279.
 - [21] E. I. ROMENSKI, *Hyperbolic equations of Maxwell's nonlinear model of elastoplastic heat-conducting media*, Siberian Mathematical Journal, 30 (1989), pp. 606–625.

- [22] E. TORO, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, Springer, 2009.

Appendix A

System Matrices

The GPR model takes the form $\frac{\partial Q}{\partial t} + \frac{\partial F(Q)}{\partial x} + B(Q) \cdot \frac{\partial Q}{\partial x} = S(Q)$ where Q, F, B, S are given below.

$$\begin{aligned}
\mathbf{Q} &= \begin{pmatrix} \rho \\ \rho E \\ \rho v_1 \\ \rho v_2 \\ \rho v_3 \\ \rho v_3 \\ A_{11} \\ A_{12} \\ A_{13} \\ A_{21} \\ A_{22} \\ A_{23} \\ A_{31} \\ A_{32} \\ A_{33} \\ \rho J_1 \\ \rho J_2 \\ \rho J_3 \end{pmatrix} & \mathbf{F}_1 &= \begin{pmatrix} \rho v_1 \\ \rho v_1 E + v_1 p - v_m \sigma_{m1} + q_1 \\ \rho v_1^2 + p - \sigma_{11} \\ \rho v_1 v_2 - \sigma_{21} \\ \rho v_1 v_3 - \sigma_{31} \\ A_{1m} v_m \\ 0 \\ 0 \\ A_{2m} v_m \\ 0 \\ 0 \\ A_{3m} v_m \\ 0 \\ 0 \\ \rho J_1 v_1 + T \\ \rho J_2 v_1 \\ \rho J_3 v_1 \end{pmatrix} \\
& & \mathbf{F}_2 &= \begin{pmatrix} \rho v_2 \\ \rho v_2 E + v_2 p - v_m \sigma_{m2} + q_2 \\ \rho v_1 v_2 - \sigma_{12} \\ \rho v_2^2 + p - \sigma_{22} \\ \rho v_2 v_3 - \sigma_{32} \\ 0 \\ A_{1m} v_m \\ 0 \\ 0 \\ A_{2m} v_m \\ 0 \\ 0 \\ A_{3m} v_m \\ 0 \\ \rho J_1 v_2 \\ \rho J_2 v_2 + T \\ \rho J_3 v_2 \end{pmatrix} \\
& & \mathbf{F}_3 &= \begin{pmatrix} \rho v_3 \\ \rho v_3 E + v_3 p - v_m \sigma_{m3} + q_3 \\ \rho v_1 v_3 - \sigma_{13} \\ \rho v_2 v_3 - \sigma_{23} \\ \rho v_3^2 + p - \sigma_{33} \\ 0 \\ 0 \\ A_{1m} v_m \\ 0 \\ 0 \\ A_{2m} v_m \\ 0 \\ 0 \\ A_{3m} v_m \\ \rho J_1 v_3 \\ \rho J_2 v_3 \\ \rho J_3 v_3 + T \end{pmatrix}
\end{aligned}$$

Appendix B

Primitive System

Taking the ordering \mathbf{P} of primitive variables in (B.10), note that (3), (1b), (1c), (1d) can be stated as:

$$\frac{D\rho}{Dt} + \rho \frac{\partial v_k}{\partial x_k} = 0 \quad (\text{B.1a})$$

$$\frac{Dv_i}{Dt} + \frac{1}{\rho} \frac{\partial \Sigma_{ik}}{\partial x_k} = 0 \quad (\text{B.1b})$$

$$\frac{DA_{ij}}{Dt} + A_{ik} \frac{\partial v_k}{\partial x_j} = -\frac{\psi_{ij}}{\theta_1} \quad (\text{B.1c})$$

$$\frac{DJ_i}{Dt} + \frac{1}{\rho} \frac{\partial T \delta_{ik}}{\partial x_k} = -\frac{H_i}{\theta_2} \quad (\text{B.1d})$$

$$\frac{DE}{Dt} + \frac{1}{\rho} \frac{\partial (\Sigma_{ik} v_i + T H_k)}{\partial x_k} = 0 \quad (\text{B.1e})$$

where the stress tensor $\Sigma = pI + \rho A^T \psi$. Note that:

$$\frac{DE}{Dt} = \frac{\partial E}{\partial p} \frac{Dp}{Dt} + \frac{\partial E}{\partial \rho} \frac{D\rho}{Dt} + v_i \frac{Dv_i}{Dt} + \psi_{ij} \frac{DA_{ij}}{Dt} + H_i \frac{DJ_i}{Dt} \quad (\text{B.2})$$

Thus, the energy equation becomes:

$$\begin{aligned} \frac{\partial E}{\partial p} \frac{Dp}{Dt} - \frac{\partial E}{\partial \rho} \rho \frac{\partial v_k}{\partial x_k} - \frac{v_i}{\rho} \frac{\partial \Sigma_{ik}}{\partial x_k} - \psi_{ij} A_{ik} \frac{\partial v_k}{\partial x_j} - \frac{\psi_{ij} \psi_{ij}}{\theta_1} \\ - \frac{H_i}{\rho} \frac{\partial T \delta_{ik}}{\partial x_k} - \frac{H_i H_i}{\theta_2} + \frac{1}{\rho} \frac{\partial (\Sigma_{ik} v_i + T H_k)}{\partial x_k} = 0 \end{aligned} \quad (\text{B.3})$$

Simplifying:

$$\frac{\partial E}{\partial p} \frac{Dp}{Dt} + \left(\frac{p}{\rho} - \frac{\partial E}{\partial \rho} \rho \right) \frac{\partial v_k}{\partial x_k} + \frac{T}{\rho} \frac{\partial H_k}{\partial x_k} = \frac{\psi_{ij} \psi_{ij}}{\theta_1} + \frac{H_i H_i}{\theta_2} \quad (\text{B.4})$$

The full system then becomes:

$$\frac{D\rho}{Dt} + \rho \frac{\partial v_k}{\partial x_k} = 0 \quad (\text{B.5a})$$

$$\frac{Dv_i}{Dt} + \frac{1}{\rho} \frac{\partial \Sigma_{ik}}{\partial x_k} = 0 \quad (\text{B.5b})$$

$$\frac{DA_{ij}}{Dt} + A_{ik} \frac{\partial v_k}{\partial x_j} = -\frac{\psi_{ij}}{\theta_1} \quad (\text{B.5c})$$

$$\frac{DJ_i}{Dt} + \frac{1}{\rho} \frac{\partial T \delta_{ik}}{\partial x_k} = -\frac{H_i}{\theta_2} \quad (\text{B.5d})$$

$$\frac{Dp}{Dt} + \frac{p - \rho^2 E_\rho}{\rho E_p} \frac{\partial v_k}{\partial x_k} + \frac{T}{\rho E_p} \frac{\partial H_k}{\partial x_k} = \frac{\psi_{ij} \psi_{ij}}{\theta_1 E_p} + \frac{H_i H_i}{\theta_2 E_p} \quad (\text{B.5e})$$

Note that¹:

$$\frac{p - \rho^2 E_\rho}{\rho E_p} = \rho c_0^2 \quad (\text{B.7a})$$

$$\frac{\alpha^2 T}{\rho E_p} = \frac{\alpha^2 T}{\rho c_v T_p} = \frac{\rho c_h^2}{T_p} \quad (\text{B.7b})$$

Thus, expressing the system in terms of spatial derivatives of the primitive variables, we have:

$$\frac{D\rho}{Dt} + \rho \frac{\partial v_k}{\partial x_k} = 0 \quad (\text{B.8a})$$

$$\frac{Dv_i}{Dt} + \frac{\delta_{ik}}{\rho} \frac{\partial p}{\partial x_k} - \frac{(\beta + 1) \sigma_{ik}}{\rho^2} \frac{\partial \rho}{\partial x_k} - \frac{1}{\rho} \frac{\partial \sigma_{ik}}{\partial A_{mn}} \frac{\partial A_{mn}}{\partial x_k} = 0 \quad (\text{B.8b})$$

$$\frac{DA_{ij}}{Dt} + A_{ik} \frac{\partial v_k}{\partial x_j} = -\frac{\psi_{ij}}{\theta_1} \quad (\text{B.8c})$$

$$\frac{DJ_i}{Dt} + \frac{T_\rho \delta_{ik}}{\rho} \frac{\partial \rho}{\partial x_k} + \frac{T_p \delta_{ik}}{\rho} \frac{\partial p}{\partial x_k} = -\frac{H_i}{\theta_2} \quad (\text{B.8d})$$

$$\frac{Dp}{Dt} + \rho c_0^2 \frac{\partial v_k}{\partial x_k} + \frac{\rho c_h^2}{T_p} \frac{\partial J_k}{\partial x_k} = \frac{\|\psi\|_F^2}{\theta_1 E_p} + \frac{\|H\|^2}{\theta_2 E_p} \quad (\text{B.8e})$$

1

$$\begin{aligned} \frac{p - \rho^2 E_\rho}{\rho E_p} &= \frac{\rho^2 E_\rho|_s - \rho^2 E_\rho|_p}{\rho E_p|_\rho} = \rho \frac{E_\rho|_s - (E_\rho|_s + E_s|_\rho s_\rho|_p)}{E_s|_\rho s_p|_\rho} \\ &= \rho \frac{-s_\rho|_p}{s_p|_\rho} = \rho \frac{\partial p}{\partial \rho} \Big|_s \end{aligned} \quad (\text{B.6})$$

The result follows by the definition of c_0 .

Thus, the GPR system can be written in the following form:

$$\frac{\partial \mathbf{P}}{\partial t} + \mathbf{M} \cdot \nabla \mathbf{P} = \mathbf{S}_p \quad (\text{B.9})$$

$$\mathbf{P} = \begin{pmatrix} \rho \\ p \\ A_{11} \\ A_{21} \\ A_{31} \\ A_{12} \\ A_{22} \\ A_{32} \\ A_{13} \\ A_{23} \\ A_{33} \\ v_1 \\ v_2 \\ v_3 \\ J_1 \\ J_2 \\ J_3 \end{pmatrix} \quad (\text{B.10})$$

$$\mathbf{S}_p = \frac{1}{\theta_1 (\tau_1) E_p} \begin{pmatrix} 0 \\ \|\psi\|_F^2 \\ 0 \\ 0 \\ 0 \\ -\psi_{11} \\ -\psi_{21} \\ -\psi_{31} \\ -\psi_{12} \\ -\psi_{22} \\ -\psi_{32} \\ -\psi_{13} \\ -\psi_{23} \\ -\psi_{33} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{\theta_2 (\tau_2) E_p} \begin{pmatrix} 0 \\ \|\mathbf{H}\|^2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -H_1 \\ -H_2 \\ -H_3 \end{pmatrix} \quad (\text{B.11})$$

Appendix C

Matrices used in the Numerical Methods

C.1 WENO Matrices for $N = 2$

$$M_1 = \begin{pmatrix} 2\sqrt{\frac{5}{3}} + \frac{245}{18} & -\frac{236}{9} & \frac{245}{18} - 2\sqrt{\frac{5}{3}} \\ \sqrt{\frac{5}{3}} + \frac{65}{18} & -\frac{56}{9} & \frac{65}{18} - \sqrt{\frac{5}{3}} \\ \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{pmatrix} \quad (\text{C.1a})$$

$$M_2 = \begin{pmatrix} \sqrt{\frac{5}{3}} + \frac{65}{18} & -\frac{56}{9} & \frac{65}{18} - \sqrt{\frac{5}{3}} \\ \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \\ \frac{65}{18} - \sqrt{\frac{5}{3}} & -\frac{56}{9} & \sqrt{\frac{5}{3}} + \frac{65}{18} \end{pmatrix} \quad (\text{C.1b})$$

$$M_3 = \begin{pmatrix} \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \\ \frac{65}{18} - \sqrt{\frac{5}{3}} & -\frac{56}{9} & \sqrt{\frac{5}{3}} + \frac{65}{18} \\ \frac{245}{18} - 2\sqrt{\frac{5}{3}} & -\frac{236}{9} & 2\sqrt{\frac{5}{3}} + \frac{245}{18} \end{pmatrix} \quad (\text{C.1c})$$

$$M_1^{-1} = \begin{pmatrix} \frac{1}{60} (2 - 3\sqrt{15}) & \sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60} (62 - 9\sqrt{15}) \\ -\frac{1}{24} & \frac{1}{12} & \frac{23}{24} \\ \frac{1}{60} (3\sqrt{15} + 2) & -\sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60} (9\sqrt{15} + 62) \end{pmatrix} \quad (\text{C.2a})$$

$$M_2^{-1} = \begin{pmatrix} \frac{1}{60} (3\sqrt{15} + 2) & \frac{14}{15} & \frac{1}{60} (2 - 3\sqrt{15}) \\ -\frac{1}{24} & \frac{13}{12} & -\frac{1}{24} \\ \frac{1}{60} (2 - 3\sqrt{15}) & \frac{14}{15} & \frac{1}{60} (3\sqrt{15} + 2) \end{pmatrix} \quad (\text{C.2b})$$

$$M_3^{-1} = \begin{pmatrix} \frac{1}{60} (9\sqrt{15} + 62) & -\sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60} (3\sqrt{15} + 2) \\ \frac{23}{24} & \frac{1}{12} & -\frac{1}{24} \\ \frac{1}{60} (62 - 9\sqrt{15}) & \sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60} (2 - 3\sqrt{15}) \end{pmatrix} \quad (\text{C.2c})$$

Appendix D

Jacobians

D.1 Jacobian of Distortion ODEs

The Jacobian of the source function is used to speed up numerical integration of the ODE. It is derived thus:

$$\frac{\partial \text{dev}(G)_{ij}}{\partial A_{mn}} = \delta_{in} A_{mj} + \delta_{jn} A_{mi} - \frac{2}{3} \delta_{ij} A_{mn} \quad (\text{D.1})$$

Thus:

$$\begin{aligned} \frac{\partial (A \text{dev}(G))_{ij}}{\partial A_{mn}} &= \frac{\partial A_{it}}{\partial A_{mn}} \text{dev}(G)_{tj} + A_{it} \frac{\partial \text{dev}(G)_{tj}}{\partial A_{mn}} \\ &= \delta_{im} \delta_{tn} \left(A_{kt} A_{kj} - \frac{1}{3} A_{kl} A_{kl} \delta_{tj} \right) + A_{it} \left(\delta_{tn} A_{mj} + \delta_{jn} A_{mt} - \frac{2}{3} \delta_{tj} A_{mn} \right) \\ &= \delta_{im} A_{kn} A_{kj} - \frac{1}{3} \delta_{im} \delta_{jn} A_{kl} A_{kl} + A_{in} A_{mj} + \delta_{jn} A_{ik} A_{mk} - \frac{2}{3} A_{ij} A_{mn} \end{aligned} \quad (\text{D.2})$$

Thus:

$$\begin{aligned} J_A &\equiv \frac{-3}{\tau_1} \frac{\partial \left(\det(A)^{\frac{5}{3}} A \text{dev}(G) \right)_{ij}}{\partial A_{mn}} \\ &= \frac{-3}{\tau_1} \det(A)^{\frac{5}{3}} \left(\frac{5}{3} (A \text{dev}(G))_{ij} A_{mn}^{-T} + A_{in} A_{mj} + \delta_{jn} G'_{im} + \delta_{im} G_{jn} - \frac{1}{3} \delta_{im} \delta_{jn} A_{kl} A_{kl} - \frac{2}{3} A_{ij} A_{mn} \right) \\ &= \frac{1}{\tau_1} \det(A)^{\frac{5}{3}} \left(-5 (A \text{dev}(G)) \otimes A^{-T} + 2A \otimes A - 3(A \otimes A)^{1,3} + \|A\|_F^2 (I \otimes I)^{2,3} - 3(G' \otimes I + I \otimes G') \right) \end{aligned} \quad (\text{D.3})$$

where $G' = AA^T$ and $X^{a,b}$ refers to tensor X with indices a, b transposed.

D.2 Jacobian of Thermal Impulse ODEs

As demonstrated in [2.3.3.1](#), we have:

$$\frac{dJ_i}{dt} = \frac{J_i}{2} \left(-a + b (J_1^2 + J_2^2 + J_3^2) \right) \quad (\text{D.4})$$

D.2 Jacobian of Thermal Impulse ODEs

where

$$a = \frac{2\rho_0}{\tau_2 T_0 \rho c_v} (E - E_{2A}(A) - E_3(\mathbf{v})) \quad (\text{D.5a})$$

$$b = \frac{\rho_0 \alpha^2}{\tau_2 T_0 \rho c_v} \quad (\text{D.5b})$$

Thus, the Jacobian of the thermal impulse ODEs is:

$$\begin{pmatrix} \frac{b}{2}(3J_1^2 + J_2^2 + J_3^2) - \frac{a}{2} & bJ_1J_2 & bJ_1J_3 \\ bJ_1J_2 & \frac{b}{2}(J_1^2 + 3J_2^2 + J_3^2) - \frac{a}{2} & bJ_2J_3 \\ bJ_1J_3 & bJ_2J_3 & \frac{b}{2}(J_1^2 + J_2^2 + 3J_3^2) - \frac{a}{2} \end{pmatrix} \quad (\text{D.6})$$