A Fast Numerical Scheme for the Godunov-Peshkov-Romenski Model of Continuum Mechanics

Haran Jackson

Cavendish Laboratory, JJ Thomson Ave, Cambridge, UK, CB3 0HE

Abstract

A new second-order numerical scheme based on an operator splitting is proposed for the Godunov-Peshkov-Romenski model of continuum mechanics. The homogeneous part of the system is solved with a finite volume method based on a WENO reconstruction, and the temporal ODEs are solved using some analytic results presented here. Whilst it is not possible to attain arbitrary-order accuracy with this scheme (as with ADER-WENO schemes used previously), the attainable order of accuracy is often sufficient, and solutions are computationally cheap when compared with other available schemes. The new scheme is compared with an ADER-WENO scheme for various test cases, and a convergence study is undertaken to demonstrate its order of accuracy.

Keywords: Godunov-Peshkov-Romenski, GPR, Continuum Mechanics, Operator Splitting, ADER, WENO

Contents

1. Background

1.1. Motivation

The Godunov-Peshkov-Romenski model of continuum mechanics (as described in 1.2) presents an exciting possibility of being able to describe both fluids and solids within the same mathematical framework. This has the potential to streamline development of simulation software by reducing the number of different systems of equations that require solvers, and cutting down on the amount of theoretical work required, for example in the treatment of interfaces in multimaterial problems. In addition to this, the hyperbolic nature of the GPR model ensures that the nonphysical instantaneous transmission of information appearing in certain non-hyperbolic models (such as the Navier-Stokes equations) cannot occur. Parallelization also tends to be easier with hyperbolic models, allowing us to leverage the great advances that have been made in parallel computing architectures in recent years.

At the time of writing, the GPR model has been solved for a variety of fluid and solid problems using the ADER-WENO method (Dumbser et al. [8], Boscheri et al. [4]). ADER-WENO methods (described in 1.3) are extremely effective in producing arbitrarily-high order solutions to hyperbolic systems of PDEs, but in some situations their accompanying computational cost may prove burdensome. A new method is presented in this study that is simple to implement and computationally cheaper than a corresponding ADER-WENO method if only second order accuracy is required. This may prove useful in the design of simulation software addressing problems in which not just accuracy but also speed and usability are of paramount importance.

1.2. The GPR Model

The GPR model, first introduced in Peshkov and Romenski [23], has its roots in Godunov and Romenski's 1970s model of elastoplastic deformation (see Godunov and Romenski [14]). It was expanded upon in Dumbser et al. [8] to include thermal conduction. This expanded model takes the following form:

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

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

$$\frac{\partial A_{ij}}{\partial t} + \frac{\partial (A_{ik} v_k)}{\partial x_j} + v_k \left(\frac{\partial A_{ij}}{\partial x_k} - \frac{\partial A_{ik}}{\partial x_j} \right) = -\frac{\psi_{ij}}{\theta_1(\tau_1)} \tag{1c}$$

$$\frac{\partial (\rho J_i)}{\partial t} + \frac{\partial (\rho J_i v_k + T \delta_{ik})}{\partial x_k} = -\frac{\rho H_i}{\theta_2(\tau_2)} \tag{1d}$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho E v_k + (p \delta_{ik} - \sigma_{ik}) v_i + q_k)}{\partial x_k} = 0 \tag{1e}$$

$\rho, \mathbf{v}, p, \delta, \sigma, T, E, \boldsymbol{q}$ retain their usual meanings. θ_1 and θ_2 are positive scalar functions, chosen according to the properties of the material being modeled. A is the distortion tensor (containing information about the deformation and rotation of material elements), $\mathbf{J}$ is the thermal impulse vector (a thermal analogue of momentum), τ_1 is the strain dissipation time, and τ_2 is the thermal impulse relaxation time. $\psi = \frac{\partial E}{\partial A}$ and $\boldsymbol{H} = \frac{\partial E}{\partial \boldsymbol{J}}$.

The following definitions are given:

$$p = \rho^2 \frac{\partial E}{\partial \rho} \tag{2a}$$

$$\sigma = -\rho A^T \frac{\partial E}{\partial A} \tag{2b}$$

$$T = \frac{\partial E}{\partial s} \tag{2c}$$

$$\boldsymbol{q} = \frac{\partial E}{\partial s} \frac{\partial E}{\partial \boldsymbol{J}} \tag{2d}$$

To close the system, the equation of state (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 element[1] scale (mesoscale), and the flow scale (macroscale):

$$E = E_1 (\rho, p) + E_2 (A, \boldsymbol{J}) + E_3 (\boldsymbol{v}) \tag{3}$$

The EOS used in this study (and described in the following passages) is taken from Dumbser et al. [8]. It should be noted, however, that this is just one particular choice, and there are many others that may be used.

For an ideal or stiffened gas, E_1 is given by:

$$E_1 = \frac{p + \gamma p_\infty}{(\gamma - 1) \rho} \tag{4}$$

---

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

where $p_\infty = 0$ for an ideal gas.

E_2 is chosen to have the following quadratic form:

$$E_2 = \frac{c_s^2}{4} \|\text{dev}(G)\|_F^2 + \frac{\alpha^2}{2} \|\boldsymbol{J}\|^2 \tag{5}$$

c_s is the characteristic velocity of propagation of transverse perturbations. α is a constant related to the characteristic velocity of propagation of heat waves:

$$c_h = \frac{\alpha}{\rho} \sqrt{\frac{T}{c_v}} \tag{6}$$

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

$$\text{dev}(G) = G - \frac{1}{3} \text{tr}(G) I \tag{7}$$

E_3 is the usual specific kinetic energy per unit mass:

$$E_3 = \frac{1}{2} \|\boldsymbol{v}\|^2 \tag{8}$$

The following forms are chosen:

$$\theta_1(\tau_1) = \frac{\tau_1 c_s^2}{3 |A|^{\frac{5}{3}}} \tag{9a}$$

$$\theta_2(\tau_2) = \tau_2 \alpha^2 \frac{\rho T_0}{\rho_0 T} \tag{9b}$$

$$\tau_1 = \frac{6\mu}{\rho_0 c_s^2} \tag{10a}$$

$$\tau_2 = \frac{\rho_0 \kappa}{T_0 \alpha^2} \tag{10b}$$

The justification of these choices is that classical Navier–Stokes–Fourier theory is recovered in the stiff limit $\tau_1, \tau_2 \to 0$ (see Dumbser et al. [8]). This results in the following relations:

$$\sigma = -\rho c_s^2 G \, \text{dev}(G) \tag{11a}$$

$$\boldsymbol{q} = \alpha^2 T \boldsymbol{J} \tag{11b}$$

$$-\frac{\psi}{\theta_1(\tau_1)} = -\frac{3}{\tau_1} |A|^{\frac{5}{3}} A \, \text{dev}(G) \tag{11c}$$

$$-\frac{\rho \boldsymbol{H}}{\theta_2(\tau_2)} = -\frac{T\rho_0}{T_0 \tau_2} \boldsymbol{J} \tag{11d}$$

The following constraint also holds (see Peshkov and Romenski [23]):

$$\det(A) = \frac{\rho}{\rho_0} \tag{12}$$

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" Frenkel [12]; 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 J and its contribution to the energy of the system are derived from Romenski's model of hyperbolic heat transfer, originally proposed in Malyshev and Romenskii [19], Romenski [26], and implemented in Romenski et al. [25, 24]. In this model, 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 J. Romenski remarks that it is more convenient to evolve 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.

1.3. The ADER-WENO Method

The ADER-WENO method was used in Dumbser et al. [8], Boscheri et al. [4] to solve the GPR system. It produces arbitrarily high-order solutions to hyperbolic systems of PDEs and has been shown to be particularly effective for a wide range of systems (e.g. the classical Euler equations of gas dynamics, the special relativistic hydrodynamics and ideal magnetohydrodynamics equations, and the Baer-Nunziato model for compressible two-phase flow - see Balsara et al. [1], Zanotti and Dumbser [28]). The first step in the process - the WENO method - will be used later in this study and is therefore discussed in detail here. The remaining steps are described qualitatively, with references for further information given.

WENO (Weighted Essentially Non-Oscillatory) methods are used to produce high order polynomial approximations to piece-wise constant data. Many variations exist. In this study, the method of Dumbser et al. [11] 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 $x_i = \frac{i \cdot L}{K}$ for $i = 0, \ldots, K$ 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:

$$\chi^i = \frac{1}{\Delta x}(x - x_i) \tag{13}$$

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

$$\psi_i(\chi_j) = \delta_{ij} \tag{14}$$

If N is even, take the stencils:

$$\begin{cases} S_1 &= \left\{C_{i-\frac{N}{2}}, \ldots, C_{i+\frac{N}{2}}\right\} \\ S_2 &= \{C_{i-N}, \ldots, C_i\} \\ S_3 &= \{C_i, \ldots, C_{i+N}\} \end{cases} \tag{15}$$

If N is odd, take the stencils:

$$
\begin{cases}
S_1 & = \left\{ C_{i-\lfloor \frac{N}{2} \rfloor}, \ldots, C_{i+\lceil \frac{N}{2} \rceil} \right\} \\
S_2 & = \left\{ C_{i-\lceil \frac{N}{2} \rceil}, \ldots, C_{i+\lfloor \frac{N}{2} \rfloor} \right\} \\
S_3 & = \{ C_{i-N}, \ldots, C_i \} \\
S_4 & = \{ C_i, \ldots, C_{i+N} \}
\end{cases}
\tag{16}
$$

The data is reconstructed on S_j as:

$$
\sum_p \psi_p \left(\chi^i(x) \right) \hat{w}_p^{ij}
\tag{17}
$$

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

$$
\frac{1}{\Delta x} \int_{x_k}^{x_{k+1}} \sum_p \psi_p \left(\chi^k(x) \right) \hat{w}_p^{ij} dx = u_k \qquad \forall C_k \in S_j
\tag{18}
$$

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 $\{j_0, \ldots, j_N\}$ indexes the cells in S_j. In this study reconstructions with $N = 2$ are used. The matrices of these linear systems are given in 8.3, along with their inverses, which are precomputed to accelerate the solution of these systems.

Define the oscillation indicator matrix:

$$
\Sigma_{mn} = \sum_{\alpha=1}^{N} \int_0^1 \psi_m^{(\alpha)} \psi_n^{(\alpha)} d\chi
\tag{19}
$$

and the oscillation indicator for each stencil:

$$
o_j = \Sigma_{mn} \hat{w}_m^{ij} \hat{w}_n^{ij}
\tag{20}
$$

The full reconstruction in C_i is:

$$
w_i(x) = \sum_p \psi_p \left(\chi^i(x) \right) \bar{w}_p^i
\tag{21}
$$

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

$$
\omega_j = \frac{\tilde{\omega}_j}{\sum_k \tilde{\omega}_k} \qquad \tilde{\omega}_j = \frac{\zeta_j}{\left(o_j + \varepsilon \right)^r}
\tag{22}
$$

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 Dumbser et al. [7].

The reconstruction can be extended to two dimensions by taking:

$$
\upsilon^i = \frac{1}{\Delta y} (y - y_i)
\tag{23}
$$

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

$$\sum_{p,q} \psi_p \left(\chi^i (x) \right) \psi_q \left(v^i (x) \right) \tilde{w}_{pq}^{ij} \tag{24}$$

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

$$M_j \tilde{w}_p^{ij} = \bar{w}_p^{[j_0:j_N]} \qquad \forall p \in \{0, \dots, N\} \tag{25}$$

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.

The next process in the ADER-WENO method is to perform a Continuous Galerkin or Discontinuous Galerkin spatio-temporal polynomial reconstruction of the data in each cell, using the WENO reconstruction as initial data at the start of the time step (see Balsara et al. [1] and Dumbser et al. [5] respectively for implementations of these two variations). The order of this reconstruction in time is usually taken to be the same as the spatial order, and the same basis polynomials are used. The process involves finding the root of a non-linear system, and this process is guaranteed to converge in exact arithmetic for certain classes of PDEs (see Jackson [16]). This root finding can be computationally expensive relative to the WENO reconstruction, especially if the source terms of the PDE system are stiff.

The final step in the ADER-WENO method is to perform a finite volume update of the data in each cell, using the boundary-extrapolated values of the cell-local Galerkin reconstructions to calculate the flux terms, and the interior values of the Galerkin reconstructions to calculate the interior volume integrals. See Dumbser et al. [7] for more details.

2. An Alternative Numerical Scheme

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

$$\frac{\partial Q}{\partial t} + \nabla \cdot F(Q) + B(Q) \cdot \nabla Q = S(Q) \tag{26}$$

As described in Toro [27], a viable way to solve inhomogeneous systems of PDEs is to employ an operator splitting. That is, the following subsystems are solved:

$$\frac{\partial Q}{\partial t} + \nabla \cdot F(Q) + B(Q) \cdot \nabla Q = 0 \tag{27a}$$

$$\frac{dQ}{dt} = S(Q) \tag{27b}$$

The advantage of this approach is that specialized solvers can be employed to compute the results of the different sub-systems. Let $H^{\delta t}, S^{\delta t}$ be the operators that take data $Q(x, t)$ to $Q(x, t + \delta t)$ under systems (27a) and (27b) 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:

$$Q_{\Delta t} = S^{\frac{\Delta t}{2}} H^{\Delta t} S^{\frac{\Delta t}{2}} Q_0 \tag{28}$$

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*.

2.1. The Homogeneous System

A WENO reconstruction of the cell-averaged data is performed at the start of the time step (as described in 1.3). Focusing on a single cell C_i at time t_n, we have $\boldsymbol{w}^n(\boldsymbol{x}) = \boldsymbol{w}^n{}_p \Psi_p(\boldsymbol{\chi}(\boldsymbol{x}))$ 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 $\boldsymbol{F}(\boldsymbol{x}) \approx \boldsymbol{F}(\boldsymbol{w}_p) \Psi_p(\boldsymbol{\chi}(\boldsymbol{x}))$. $\boldsymbol{w}_p$ are stepped forwards half a time step using the update formula:

$$\frac{\boldsymbol{w}_p^{n+\frac{1}{2}} - \boldsymbol{w}_p^n}{\Delta t / 2} + \boldsymbol{F}\left(\boldsymbol{w}_k^n\right) \cdot \nabla \Psi_k\left(\boldsymbol{\chi}_p\right) + \boldsymbol{B}\left(\boldsymbol{w}_p^n\right) \cdot \left(\boldsymbol{w}_k^n \nabla \Psi_k\left(\boldsymbol{\chi}_p\right)\right) = \boldsymbol{0} \tag{29}$$

i.e.

$$\boldsymbol{w}_p^{n+\frac{1}{2}} = \boldsymbol{w}_p^n - \frac{\Delta t}{2\Delta x}\left(\boldsymbol{F}\left(\boldsymbol{w}_k^n\right) \cdot \nabla \Psi_k\left(\boldsymbol{\chi}_p\right) + \boldsymbol{B}\left(\boldsymbol{w}_p^n\right) \cdot \left(\boldsymbol{w}_k^n \nabla \Psi_k\left(\boldsymbol{\chi}_p\right)\right)\right) \tag{30}$$

where $\boldsymbol{\chi}_p$ 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 Toro [27]) and, as with those schemes, it is integral to giving the method presented here its second-order accuracy.

Integrating (27a) over C gives:

$$\boldsymbol{Q}_i^{n+1} = \boldsymbol{Q}_i^n - \Delta t_n \left(\boldsymbol{P}_i^{n+\frac{1}{2}} + \boldsymbol{D}_i^{n+\frac{1}{2}}\right) \tag{31}$$

where

$$\boldsymbol{Q}_i^n = \frac{1}{V} \int_C \boldsymbol{Q}(\boldsymbol{x}, t_n)\, d\boldsymbol{x} \tag{32a}$$

$$\boldsymbol{P}_i^{n+\frac{1}{2}} = \frac{1}{V} \int_C \boldsymbol{B}\left(\boldsymbol{Q}\left(\boldsymbol{x}, t_{n+\frac{1}{2}}\right)\right) \cdot \nabla \boldsymbol{Q}\left(\boldsymbol{x}, t_{n+\frac{1}{2}}\right)\, d\boldsymbol{x} \tag{32b}$$

$$\boldsymbol{D}_i^{n+\frac{1}{2}} = \frac{1}{V} \oint_{\partial C} \mathcal{D}\left(\boldsymbol{Q}^-\left(\boldsymbol{s}, t_{n+\frac{1}{2}}\right), \boldsymbol{Q}^+\left(\boldsymbol{s}, t_{n+\frac{1}{2}}\right)\right)\, d\boldsymbol{s} \tag{32c}$$

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

Note that (27a) can be rewritten as:

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \boldsymbol{M}(\boldsymbol{Q}) \cdot \nabla \boldsymbol{Q} = \boldsymbol{0} \tag{33}$$

where $\boldsymbol{M} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{Q}} + \boldsymbol{B}$. Let $\boldsymbol{n}$ be the normal to the boundary at point $\boldsymbol{s} \in \partial C$. For the GPR model, $\hat{M} = \boldsymbol{M}(\boldsymbol{Q}(\boldsymbol{s})) \cdot \boldsymbol{n}$ is a diagonalizable matrix with decomposition $\hat{M} = \hat{R}\hat{\Lambda}\hat{R}^{-1}$ where the columns of $\hat{R}$ are the right eigenvectors and $\hat{\Lambda}$ is the diagonal matrix of eigenvalues. Define also $\hat{\boldsymbol{F}} = \boldsymbol{F} \cdot \boldsymbol{n}$ and $\hat{\boldsymbol{B}} = \boldsymbol{B} \cdot \boldsymbol{n}$. Using these definitions, the interface terms arising in the FV formula have the following form:

$$\mathcal{D}(\boldsymbol{Q}^-, \boldsymbol{Q}^+) = \frac{1}{2}\left(\hat{\boldsymbol{F}}(\boldsymbol{Q}^+) + \hat{\boldsymbol{F}}(\boldsymbol{Q}^-) + \tilde{B}(\boldsymbol{Q}^+ - \boldsymbol{Q}^-) + \tilde{M}(\boldsymbol{Q}^+ - \boldsymbol{Q}^-)\right) \tag{34}$$

$\tilde{M}$ is chosen to either correspond to a Rusanov/Lax-Friedrichs flux (see Toro [27]):

$$\tilde{M} = \max\left(\max\left|\hat{\Lambda}\left(\boldsymbol{Q}^+\right)\right|, \max\left|\hat{\Lambda}\left(\boldsymbol{Q}^-\right)\right|\right) \tag{35}$$

or a simplified Osher–Solomon flux (see Dumbser and Toro [9, 10]):

$$\tilde{M} = \int_0^1 \left|\hat{M}\left(\boldsymbol{Q}^- + z\left(\boldsymbol{Q}^+ - \boldsymbol{Q}^-\right)\right)\right| dz \tag{36}$$

where

$$\left|\hat{M}\right| = \hat{R}\left|\hat{\Lambda}\right|\hat{R}^{-1} \tag{37}$$

$\tilde{B}$ takes the following form:

$$\tilde{B} = \int_0^1 \hat{B}\left(\boldsymbol{Q}^- + z\left(\boldsymbol{Q}^+ - \boldsymbol{Q}^-\right)\right) dz \tag{38}$$

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.

$\boldsymbol{P}_i^{n+\frac{1}{2}}, \boldsymbol{D}_i^{n+\frac{1}{2}}$ are calculated using an $N+1$-point Gauss-Legendre quadrature, replacing $\boldsymbol{Q}\left(\boldsymbol{x}, t_{n+\frac{1}{2}}\right)$ with $\boldsymbol{w}^{n+\frac{1}{2}}(\boldsymbol{x})$.

2.2. The Temporal ODEs

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

$$\frac{dA}{dt} = \frac{-3}{\tau_1}|A|^{\frac{5}{3}} A\,\mathrm{dev}\,(G) \tag{39a}$$

$$\frac{d\boldsymbol{J}}{dt} = -\frac{1}{\tau_2}\frac{T\rho_0}{T_0\rho}\boldsymbol{J} \tag{39b}$$

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 8.1 and 8.2. However, these systems can also be solved separately, using the analytical results presented in Section 3, under specific assumptions. The second-order Strang splitting is then:

$$\boldsymbol{Q}_{\Delta t} = D^{\frac{\Delta t}{2}} T^{\frac{\Delta t}{2}} H^{\Delta t} T^{\frac{\Delta t}{2}} D^{\frac{\Delta t}{2}} \boldsymbol{Q}_0 \tag{40}$$

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.

3. GPR-Specific Performance Improvements

3.1. The Thermal Impulse ODEs

Taking the EOS for the GPR model (3) and denoting by $E_2^{(A)}, E_2^{(J)}$ the components of E_2 depending on A and $\boldsymbol{J}$ respectively, we have:

$$T = \frac{E_1}{c_v} \tag{41}$$

$$= \frac{E - E_2^{(A)}(A) - E_3(v)}{c_v} - \frac{1}{c_v}E_2^{(J)}(J)$$

$$= c_1 - c_2 \|J\|^2$$

where:

$$c_1 = \frac{E - E_2^{(A)}(A) - E_3(v)}{c_v} \tag{42a}$$

$$c_2 = \frac{\alpha^2}{2c_v} \tag{42b}$$

Over the time period of the ODE (39b), $c_1, c_2 > 0$ are constant. We have:

$$\frac{dJ_i}{dt} = -\left(\frac{1}{\tau_2}\frac{\rho_0}{T_0\rho}\right)J_i\left(c_1 - c_2\|J\|^2\right) \tag{43}$$

Therefore:

$$\frac{d}{dt}\left(J_i^2\right) = J_i^2\left(-a + b\left(J_1^2 + J_2^2 + J_3^2\right)\right) \tag{44}$$

where

$$a = \frac{2\rho_0}{\tau_2 T_0 \rho c_v}\left(E - E_2^{(A)}(A) - E_3(v)\right) \tag{45a}$$

$$b = \frac{\rho_0\alpha^2}{\tau_2 T_0 \rho c_v} \tag{45b}$$

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

$$J(t) = J(0)\sqrt{\frac{1}{e^{at} - \frac{b}{a}(e^{at} - 1)\|J(0)\|^2}} \tag{46}$$

3.2. The Distortion ODEs

3.2.1. 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:

$$G = \left(U\Sigma V^T\right)^T U\Sigma V^T = V\Sigma^2 V^T \tag{47}$$

$$\text{tr}(G) = \text{tr}\left(V\Sigma^2 V^T\right) = \text{tr}\left(\Sigma^2 V^T V\right) = \text{tr}\left(\Sigma^2\right) \tag{48}$$

Therefore:

$$\frac{dA}{dt} = -k_0 U\Sigma V^T \left(V\Sigma^2 V^T - \frac{\text{tr}\left(\Sigma^2\right)}{3} I \right) \tag{49}$$

$$= -k_0 U\Sigma \left(\Sigma^2 - \frac{\text{tr}\left(\Sigma^2\right)}{3} \right) V^T$$

$$= -k_0 U\Sigma \, \text{dev}\left(\Sigma^2\right) V^T$$

It is a common result (see Giles [13]) that:

$$d\Sigma = U^T dA V \tag{50}$$

and thus:

$$\frac{d\Sigma}{dt} = -k_0 \Sigma \, \text{dev}\left(\Sigma^2\right) \tag{51}$$

Using a fast 3×3 SVD algorithm (such as in McAdams et al. [20]), 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:

$$\Sigma \, \text{dev}\left(\Sigma^2\right) = \begin{pmatrix} a_1 \left(a_1^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3}\right) & 0 & 0 \\ 0 & a_1 \left(a_1^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3}\right) & 0 \\ 0 & 0 & a_1 \left(a_1^2 - \frac{a_1^2 + a_2^2 + a_3^2}{3}\right) \end{pmatrix} \tag{52}$$

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:

$$\frac{dx_i}{d\tau} = -3x_i \left(x_i - \bar{x}\right) \tag{53}$$

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 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 (53) 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:

$$\begin{cases} \frac{dx_i}{d\tau} = -x_i \left(2x_i - x_j - x_k\right) = -x_i \left(2x_i - x_j - \frac{1}{x_i x_j}\right) \\ \frac{dx_j}{d\tau} = -x_j \left(2x_j - x_k - x_i\right) = -x_j \left(2x_j - x_i - \frac{1}{x_i x_j}\right) \end{cases} \tag{54}$$

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

$$\frac{dx_j}{dx_i} = \frac{x_j}{x_i} \frac{2x_j - x_i - \frac{1}{x_i x_j}}{2x_i - x_j - \frac{1}{x_i x_j}} \tag{55}$$

This has solution:

$$x_j = \frac{c + \sqrt{c^2 + 4(1-c)x_i^3}}{2x_i^2} \tag{56}$$

where

$$c = -\frac{x_{i,0}\left(x_{i,0}x_{j,0}^2 - 1\right)}{x_{i,0} - x_{j,0}} \in (-\infty, 0] \tag{57}$$

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 (54).

3.2.2. 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:

$$\left\{ (x_i, x_j) : x_i \leq x_i^0 \ \cap \ x_i \geq x_j \ \cap \ x_i \geq \frac{1}{x_j^2} \right\} \tag{58}$$

This is demonstrated in Figure 1 on page 13. By (54), the rate of change of x_i at a particular value $x_i = x_i^*$ is given by:
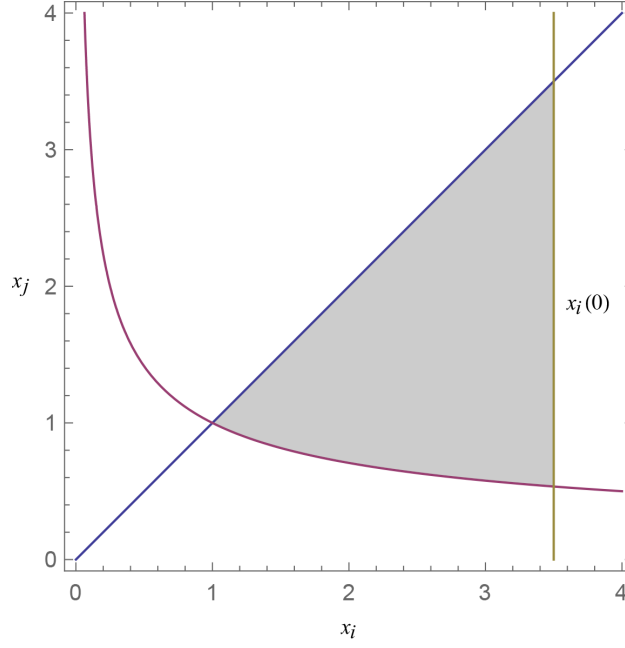
$$-x_i^* \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) \tag{59}$$

Note that:

$$\frac{d}{dx_j} \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) = -1 + \frac{1}{x_i^* x_j^2} = 0 \tag{60}$$

$$\Rightarrow x_j = \frac{1}{\sqrt{x_i^*}}$$

$$\frac{d^2}{dx_j^2} \left(2x_i^* - x_j - \frac{1}{x_i^* x_j} \right) = \frac{-2}{x_i^* x_j^3} < 0 \tag{61}$$

Thus, x_i decreases fastest on the line $x_i = \frac{1}{x_j^2}$ (the bottom boundary of the region given in Figure 1 on page 13), and slowest on the line $x_i = x_j$. The rates of change of x_i along these two lines are given respectively by:

Figure 1: The (shaded) region to which x_i, x_j are confined in the evolution of the distortion ODEs

$$\frac{dx_i}{d\tau} = -2x_i\left(x_i - \sqrt{\frac{1}{x_i}}\right) \tag{62a}$$

$$\frac{dx_i}{d\tau} = -x_i\left(x_i - \frac{1}{x_i^2}\right) \tag{62b}$$

These have implicit solutions:

$$\tau = \left(f\left(\sqrt{x_i}\right) + g\left(\sqrt{x_i}\right)\right) - \left(f\left(\sqrt{x_i^0}\right) + g\left(\sqrt{x_i^0}\right)\right) \equiv F_1\left(x_i; x_i^0\right) \tag{63a}$$

$$\tau = \left(f\left(x_i\right) - g\left(x_i\right)\right) - \left(f\left(x_i^0\right) - g\left(x_i^0\right)\right) \equiv F_2\left(x_i; x_i^0\right) \tag{63b}$$

where

$$f\left(x_i\right) = \frac{1}{6}\log\left(\frac{x_i^2 + x_i + 1}{(x_i - 1)^2}\right) \tag{64a}$$

$$g\left(x_i\right) = \frac{1}{\sqrt{3}}\tan^{-1}\left(\frac{2x_i + 1}{\sqrt{3}}\right) \tag{64b}$$

As (53) 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:

$$\tau > F_2\left(1 + \epsilon; x_i^0\right) \tag{65}$$

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:

$$\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t > F_2 \left(1 + \epsilon; \max \left\{ x_i^0 \right\} \right) \tag{66}$$

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 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:

$$\frac{2}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}} \Delta t < F_1 \left(1 + \epsilon; \max \left\{ x_i^0 \right\} \right) \tag{67}$$

3.2.3. Analytical Approximation

We now explore cases when even the reduced ODE system (54) need not be solved numerically. Define the following variables:

$$m = \frac{x_1 + x_2 + x_3}{3} \tag{68a}$$

$$u = \frac{(x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_1)^2}{3} \tag{68b}$$

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 (53) we have:

$$\frac{du}{d\tau} = -18 \left(1 - m \left(m^2 - \frac{5}{6} u \right) \right) \tag{69a}$$

$$\frac{dm}{d\tau} = -u \tag{69b}$$

Combining these equations, we have:

$$\frac{d^2 m}{d\tau^2} = -\frac{du}{d\tau} = 18 \left(1 - m \left(m^2 - \frac{5}{6} u \right) \right) \tag{70}$$

Therefore:

$$\begin{cases} \frac{d^2 m}{d\tau^2} + 15 m \frac{dm}{d\tau} + 18 \left(m^3 - 1 \right) = 0 \\ m(0) = m_0 \\ m'(0) = -u_0 \end{cases} \tag{71}$$

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

$$m(t) = 1 + \eta(t), \quad \eta \ll 1 \ \forall t \geq 0 \tag{72}$$

Thus, we have the linearized ODE:

$$\begin{cases} \frac{d^2\eta}{d\tau^2} + 15\frac{d\eta}{d\tau} + 54\eta = 0 \\ \eta(0) = m_0 - 1 \\ \eta'(0) = -u_0 \end{cases} \tag{73}$$

This is a Sturm-Liouville equation with solution:

$$\eta(\tau) = \frac{e^{-9\tau}}{3}\left((9m_0 - u_0 - 9)e^{3\tau} - (6m_0 - u_0 - 6)\right) \tag{74}$$

Thus, we also have:

$$u(\tau) = e^{-9\tau}\left(e^{3\tau}(18m_0 - 2u_0 - 18) - (18m_0 - 3u_0 - 18)\right) \tag{75}$$

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:

$$\frac{x_i + x_j + x_k}{3} = m_{\Delta t} \tag{76a}$$

$$\frac{\left(x_i - x_j\right)^2 + \left(x_j - x_k\right)^2 + (x_k - x_i)^2}{3} = u_{\Delta t} \tag{76b}$$

$$x_i x_j x_k = 1 \tag{76c}$$

This gives:

$$x_i = \frac{\sqrt[3]{6\left(\sqrt{81\Delta^2 - 6u_{\Delta t}^3} + 9\Delta\right)}}{6} + \frac{u_{\Delta t}}{\sqrt[3]{6\left(\sqrt{81\Delta^2 - 6u_{\Delta t}^3} + 9\Delta\right)}} + m_{\Delta t} \tag{77a}$$

$$x_j = \frac{1}{2}\left(\sqrt{\frac{x_i(3m_{\Delta t} - x_i)^2 - 4}{x_i}} + 3m_{\Delta t} - x_i\right) \tag{77b}$$

$$x_k = \frac{1}{x_i x_j} \tag{77c}$$

where

$$\Delta = -2m_{\Delta t}^3 + m_{\Delta t}u_{\Delta t} + 2 \tag{78}$$

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

$$x_i = \frac{\sqrt{6u_{\Delta t}}}{3}\cos\left(\frac{\theta}{3}\right) + m_{\Delta t} \tag{79a}$$

$$\theta = \tan^{-1}\left(\frac{\sqrt{6u_{\Delta t}^3 - 81\Delta^2}}{9\Delta}\right) \tag{79b}$$

	ρ	p	v	A	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 1: Initial conditions for the slow opposing shear flow test

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.

4. Numerical Results

4.1. Strain Relaxation

In this section, the approximate analytic solver for the distortion ODEs, presented in 3.2.3, is compared with a numerical ODE solver. Initial data was taken from Barton and Drikakis [2]:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ -0.01 & 0.95 & 0.02 \\ -0.015 & 0 & 0.9 \end{pmatrix}^{-1} \tag{80}$$

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 on page 17, Figure 3 on page 17, and Figure 4 on page 17, 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 Oliphant [22]).

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 $x = 0$, initially flowing with equal and opposite velocity ± 0.1 in the y-axis. The initial conditions are given in Table 1 on page 16.

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 by[2]:

$$v = v_0 \operatorname{erf}\left(\frac{x}{2\sqrt{\mu t}}\right) \tag{81}$$

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 (30) in the WENO reconstruction under the Split-WENO method was not performed, and $w_p^{n+\frac{1}{2}} \equiv w_p^n$ was taken instead. This avoided the numerical diffusion that otherwise would have emerged at the interface at $x = 0$.

---

[2]In 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 $v(\pm\infty) = \pm v_0$.
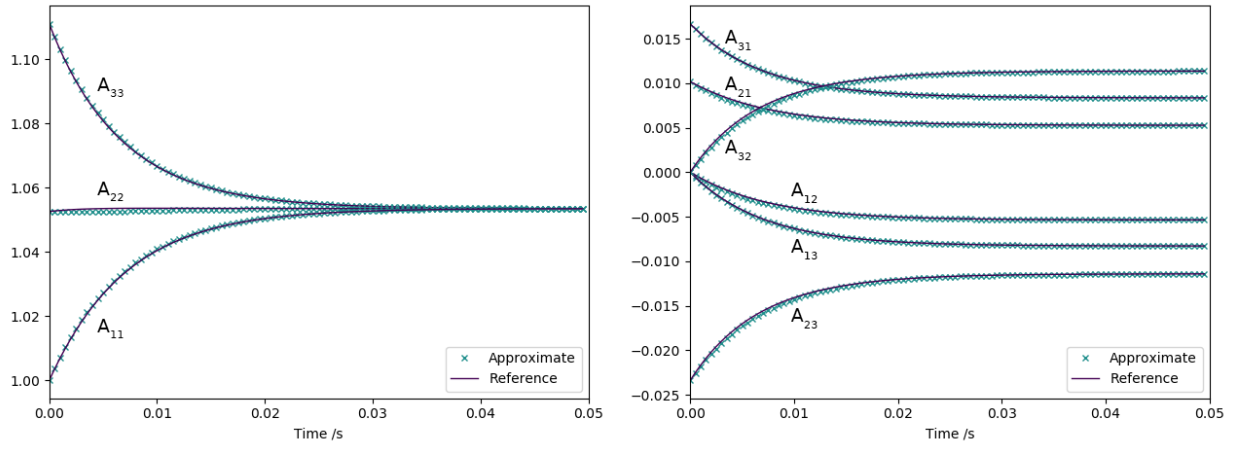
Figure 2: The components of the distortion tensor in the Strain Relaxation Test
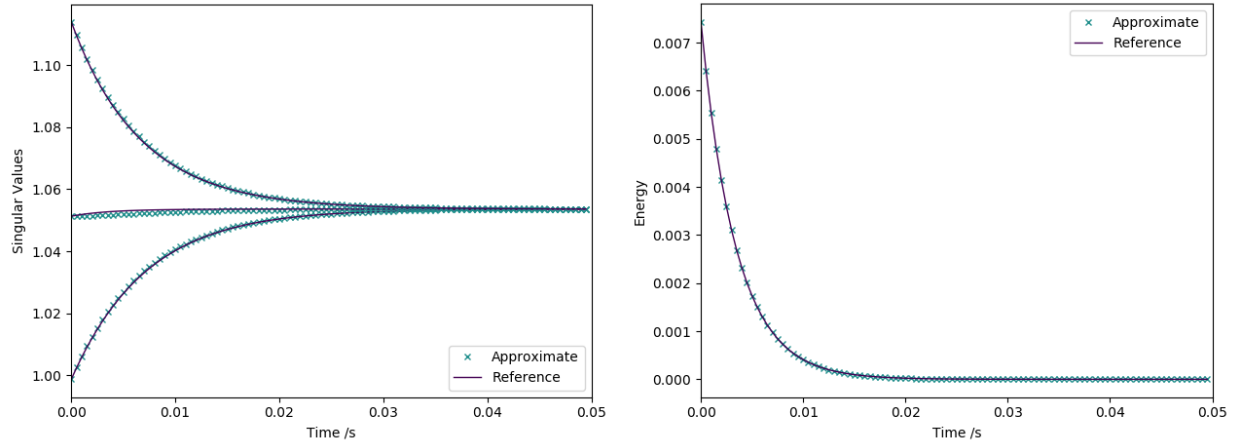

Figure 3: The singular values of the distortion tensor and the energy in the Strain Relaxation Test
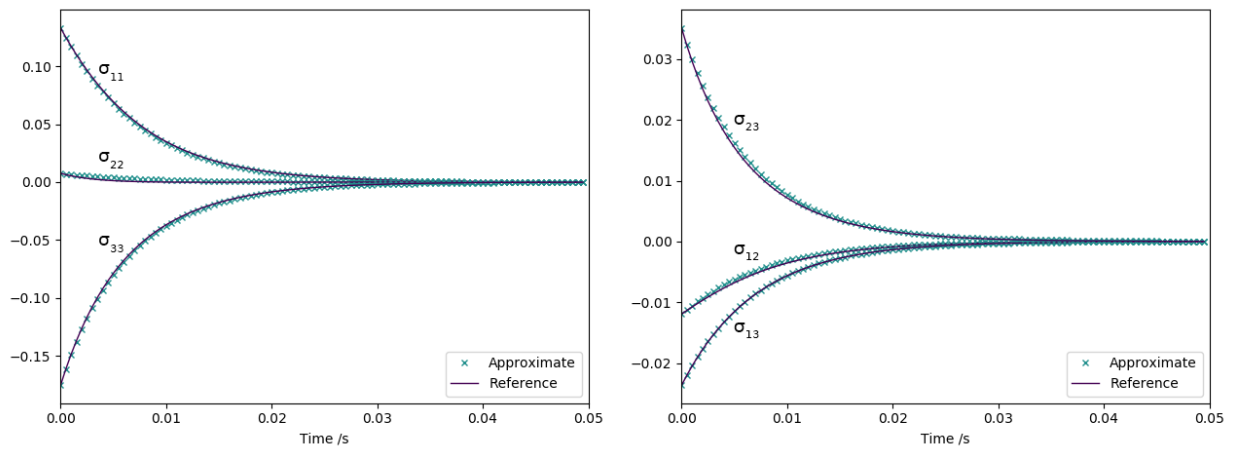

Figure 4: The components of the stress tensor in the Strain Relaxation Test

The results of simulations with 200 cells at time $t = 1$, using reconstruction polynomials of order $N = 2$, are presented in Figure 5 on page 19. 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.

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 Becker [3], the Navier-Stokes equations have an analytic solution for $P_r = 0.75$ (see Johnson Johnson [17] for a full analysis). As noted by Dumbser Dumbser et al. [8], if the wave has nondimensionalised upstream velocity $\bar{v} = 1$ and Mach number M_c, then its nondimensionalised downstream velocity is:

$$a = \frac{1 + \frac{\gamma-1}{2}M_c^2}{\frac{\gamma+1}{2}M_c^2} \tag{82}$$

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

$$\frac{1 - \bar{v}}{(\bar{v} - a)^a} = c_1 \exp(-c_2 x) \tag{83a}$$

$$c_1 = \left(\frac{1 - a}{2}\right)^{1-a} \tag{83b}$$

$$c_2 = \frac{3}{4} R_e \frac{M_c^2 - 1}{\gamma M_c^2} \tag{83c}$$

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 Morduchow and Libby [21], the nondimensional pressure and density profiles are given by:

$$\bar{p} = \frac{1}{\bar{v}}\left(1 + \frac{\gamma - 1}{2}M_c^2\left(1 - \bar{v}^2\right)\right) \tag{84}$$

$$\bar{\rho} = \frac{1}{\bar{v}} \tag{85}$$

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:

$$v = M c_0 (1 - \bar{v}) \tag{86a}$$

$$p = p_0 \bar{p} \tag{86b}$$

$$\rho = \rho_0 \bar{\rho} \tag{86c}$$

These functions are used as initial conditions, along with $A = \sqrt[3]{\rho}I$ and $\boldsymbol{J} = \boldsymbol{0}$. 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 $\tau_1 = 0.0048$, $\tau_2 = 0.005226$).
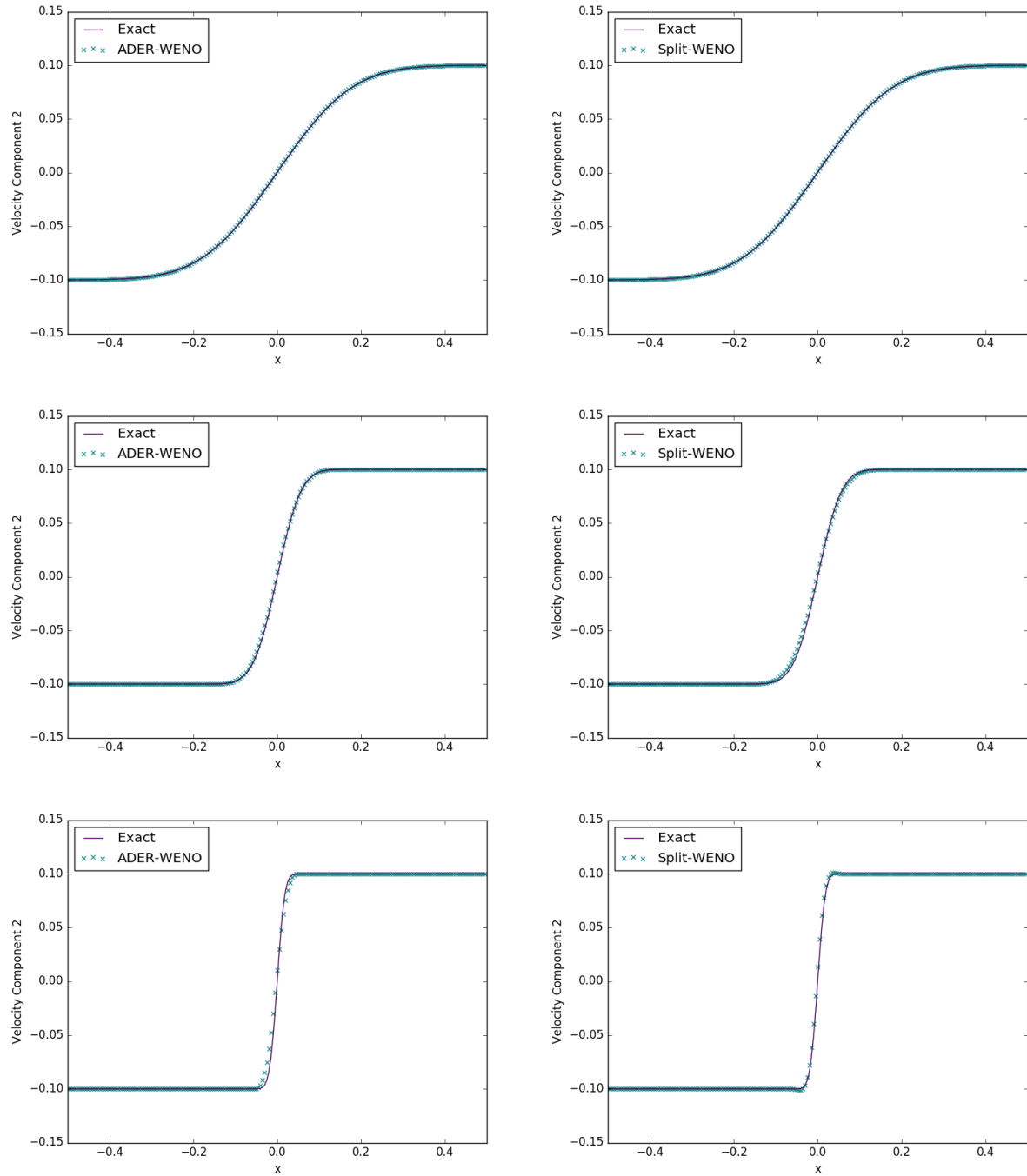
Figure 5: Results of solving Stokes' First Problem ($\mu = 10^{-2}, \mu = 10^{-3}, \mu = 10^{-4}$) with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

	ρ	p	v	A	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: Initial conditions for the heat conduction test

The results of a simulation with 200 cells at time $t = 0.2$, using reconstruction polynomials of order $N = 2$, are presented in Figure 6 on page 21 and Figure 7 on page 22. 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.

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 on page 20.

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 8 on page 22. The ADER-WENO and Split-WENO methods are in perfect agreement for both the temperature and heat flux profiles. As demonstrated in Dumbser et al. [8], this means that they in turn agree very well with a reference Navier-Stokes-Fourier solution.

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 capabilities 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 3 on page 22, 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 4 on page 23. 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 Hidalgo and Dumbser [15]).

4.6. Convergence

To assess the rate of convergence of the Split-WENO method, the convected isentropic vortex convergence study from Dumbser et al. [8] was performed. The initial conditions are given as $\rho = 1 + \delta\rho$, $p = 1 + \delta p$, $v = (1, 1, 0) + \delta v$, $A = \sqrt[3]{\rho} I$, $J = 0$, where:
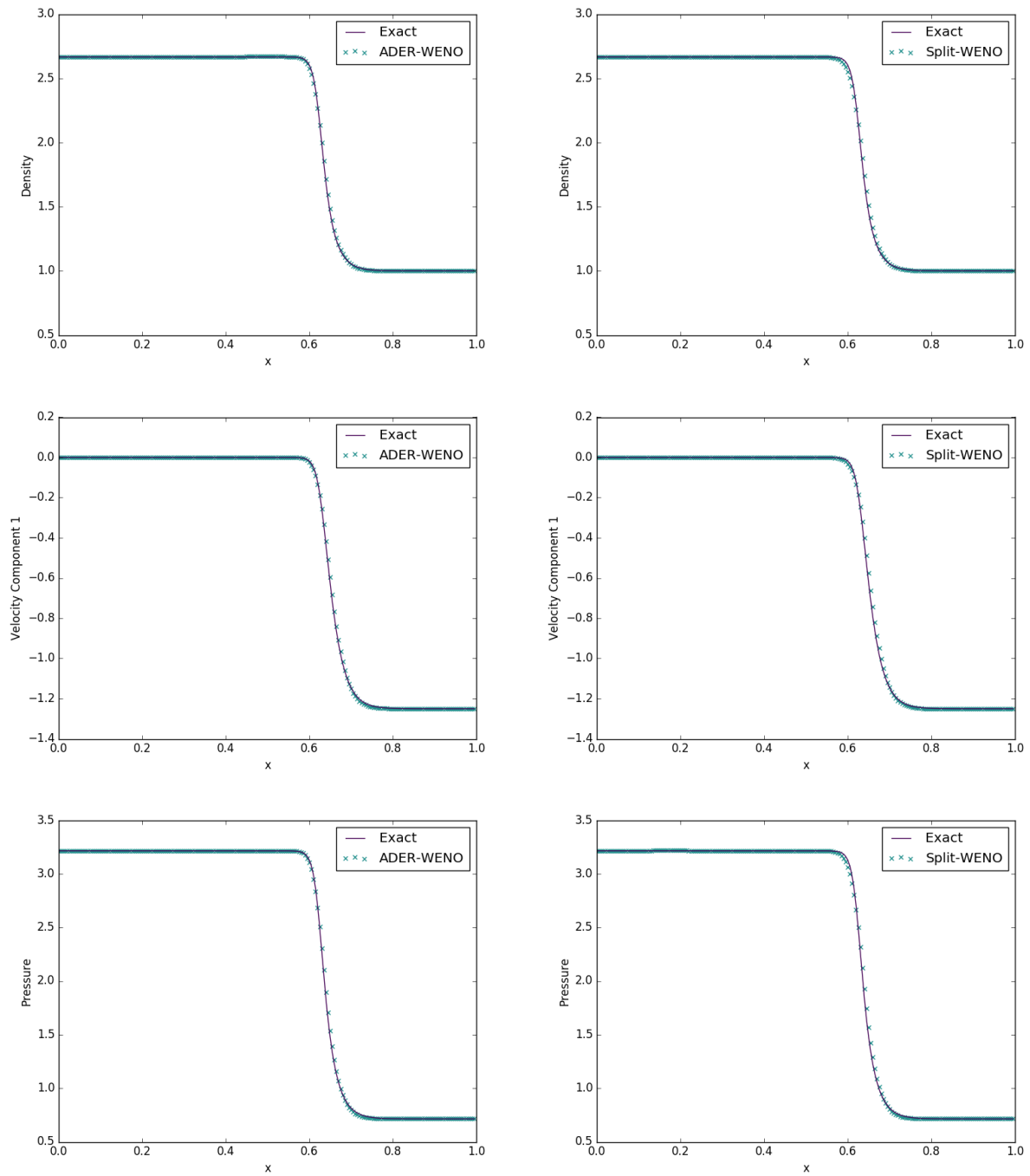
Figure 6: Density, velocity, and pressure for the Viscous Shock problem, solved with an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)
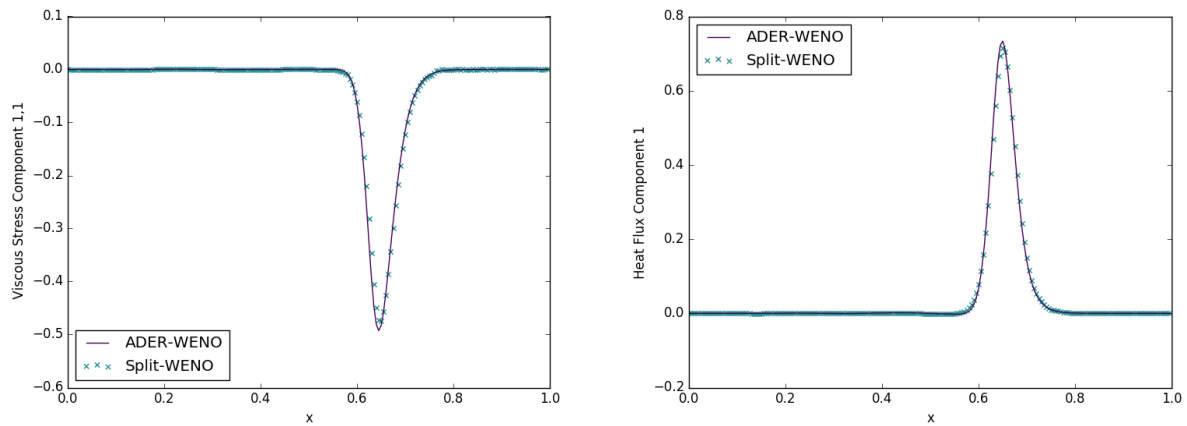
Figure 7: Viscous stress and heat flux for the Viscous Shock problem, solved with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)
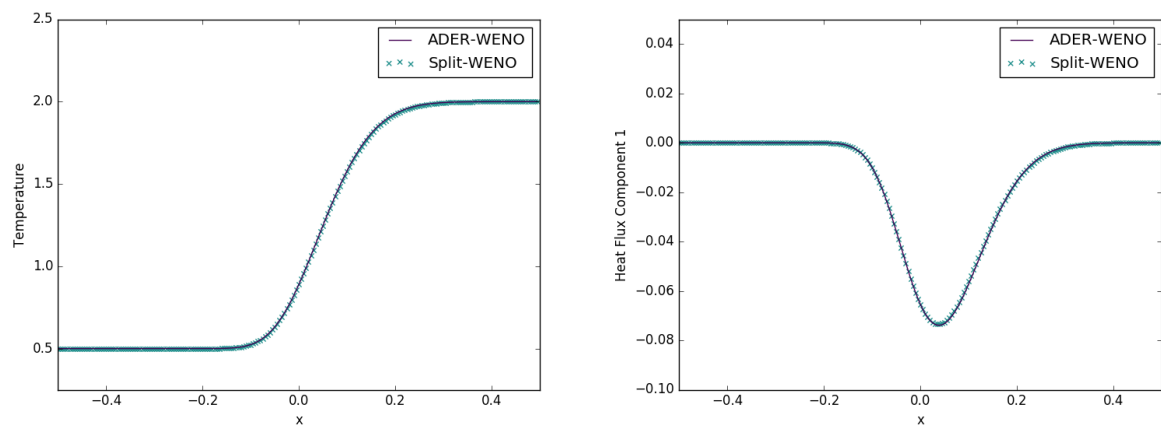
Figure 8: Results of solving the problem of Heat Conduction in Gas with both an ADER-WENO scheme and a Split-WENO scheme ($N = 2$)

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

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$O(L_1)$	$O(L_2)$	$O(L_\infty)$
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 5: Convergence rates for the Split-WENO method ($N = 2$)

$$\delta T = -\frac{(\gamma - 1)\,\epsilon^2}{8\gamma\pi^2} e^{1-r^2} \tag{87a}$$

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1 \tag{87b}$$

$$\delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1 \tag{87c}$$

$$\delta v = \frac{\epsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ x-5 \\ 0 \end{pmatrix} \tag{87d}$$

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 5 on page 23 and Table 6 on page 23 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 Dumbser et al. [8] - solved using a third-order P2P2 scheme - are given in Table 7 on page 24 for comparison.

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$O(L_1)$	$O(L_2)$	$O(L_\infty)$
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 6: Convergence rates for the Split-WENO method ($N = 3$)

Grid Size	$\epsilon(L_1)$	$\epsilon(L_2)$	$\epsilon(L_\infty)$	$O(L_1)$	$O(L_2)$	$O(L_\infty)$
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 7: Convergence rates for the ADER-DG PNPM method ($N, M = 2$)

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 Leveque and Yee [18], 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 4.2 and the convergence study in 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 Leveque and Yee [18]. 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 Dumbser et al. [6] 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 practise of any potential violation - is a topic of further research.

It should be noted that the assumption (72) used to derive the approximate analytical solver may break down for situations where the flow is compressed heavily in one direction 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 (54) 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.

6. References

[1] Balsara, D. S., Rumpf, T., Dumbser, M., Munz, C. D., 2009. Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics. Journal of Computational Physics 228 (7), 2480–2516.
URL http://dx.doi.org/10.1016/j.jcp.2008.12.003
[2] Barton, P. T., Drikakis, D., 2010. An Eulerian method for multi-component problems in non-linear elasticity with sliding interfaces. Journal of Computational Physics 229 (15), 5518–5540.
URL http://dx.doi.org/10.1016/j.jcp.2010.04.012

[3] Becker, R., 1929. Impact Waves and Detonation. Zeitschrift für Physik 8, 381.

[4] Boscheri, W., Dumbser, M., Loubère, R., 2016. Cell centered direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume schemes for nonlinear hyperelasticity. Computers and Fluids 134-135, 111–129.

[5] Dumbser, M., Balsara, D. S., Toro, E. F., Munz, C. D., 2008. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. Journal of Computational Physics 227 (18), 8209–8253.

[6] Dumbser, M., Enaux, C., Toro, E. F., 2008. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. Journal of Computational Physics 227 (8), 3971–4001.

[7] Dumbser, M., Hidalgo, A., Zanotti, O., 2014. High order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems. Computer Methods in Applied Mechanics and Engineering 268, 359–387.

[8] Dumbser, M., Peshkov, I., Romenski, E., Zanotti, O., 2016. 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, 824–862.
URL http://arxiv.org/abs/1511.08995

[9] Dumbser, M., Toro, E. F., 2011. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. Journal of Scientific Computing 48 (1-3), 70–88.

[10] Dumbser, M., Toro, E. F., 2011. On universal Osher-type schemes for general nonlinear hyperbolic conservation laws. Communications in Computational Physics 10 (3), 635–671.

[11] Dumbser, M., Zanotti, O., Hidalgo, A., Balsara, D. S., 2013. ADER-WENO finite volume schemes with space-time adaptive mesh refinement. Journal of Computational Physics 248, 257–286.

[12] Frenkel, J., 1947. Kinetic Theory of Liquids. Oxford University Press.

[13] Giles, M. B., 2008. An extended collection of matrix derivative results for forward and reverse mode algorithmic differentiation. Tech. rep., University of Oxford.
URL http://eprints.maths.ox.ac.uk/1079/

[14] Godunov, S. K., Romenski, E., 2003. Elements of continuum mechanics and conservation laws.

[15] Hidalgo, A., Dumbser, M., 2011. ADER schemes for nonlinear systems of stiff advection-diffusion-reaction equations. Journal of Scientific Computing 48 (1-3), 173–189.

[16] Jackson, H., 2017. On the Eigenvalues of the ADER-WENO Galerkin Predictor. Journal of Computational Physics 333 (March 2017), 409–413.
URL http://dx.doi.org/10.1016/j.jcp.2016.12.058

[17] Johnson, B. M., 2013. Analytical shock solutions at large and small Prandtl number. Journal of Fluid Mechanics 726, 1–12.
URL http://adsabs.harvard.edu/abs/2013arXiv1305.7132J

[18] Leveque, R., Yee, H., 1990. A Study of Numerical Methods for Hyperbolic Conservation Laws with Stiff Source Terms. Journal of Computational Physics 86, 187–210.

[19] Malyshev, A. N., Romenskii, E. I., 1986. Hyperbolic equations for heat transfer. Global solvability of the Cauchy problem. Siberian Mathematical Journal 27 (5), 734–740.

[20] McAdams, A., Selle, A., Tamstorf, R., Teran, J., Sifakis, E., 2011. Computing the Singular Value Decomposition of 3 x 3 matrices with minimal branching and elementary floating point operations. University of Wisconsin Madison.

[21] Morduchow, M., Libby, P. A., 1949. On a Complete Solution of the One-Dimensional Flow Equations of a Viscous, Heat-Conducting, Compressible Gas. Tech. rep., Polytechnic Institute of Brooklyn.

[22] Oliphant, T. E., 2007. SciPy: Open source scientific tools for Python.
URL http://www.scipy.org/

[23] Peshkov, I., Romenski, E., 2016. A hyperbolic model for viscous Newtonian flows. Continuum Mechanics and Thermodynamics 28 (1-2), 85–104.

[24] Romenski, E., Drikakis, D., Toro, E., 2010. Conservative models and numerical methods for compressible two-phase flow. Journal of Scientific Computing 42 (1), 68–95.

[25] Romenski, E., Resnyansky, A. D., Toro, E. F., 2007. Conservative hyperbolic model for compressible two-phase flow with different phase pressures and temperatures. Quarterly of applied mathematics 65(2) (2), 259–279.

[26] Romenski, E. I., 1989. Hyperbolic equations of Maxwell's nonlinear model of elastoplastic heat-conducting media. Siberian Mathematical Journal 30 (4), 606–625.

[27] Toro, E., 2009. Riemann solvers and numerical methods for fluid dynamics: a practical introduction. Springer.

[28] Zanotti, O., Dumbser, M., 2016. Efficient conservative ADER schemes based on WENO reconstruction and space-time predictor in primitive variables. Computational Astrophysics and Cosmology 3 (1), 1.
URL http://www.comp-astrophys-cosmol.com/content/3/1/1

7. Acknowledgments

8. Appendix

8.1. Jacobian of Distortion ODEs

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

$$\frac{\partial \operatorname{dev}(G)_{ij}}{\partial A_{mn}} = \delta_{in} A_{mj} + \delta_{jn} A_{mi} - \frac{2}{3} \delta_{ij} A_{mn} \tag{88}$$

Thus:

$$\begin{aligned}
\frac{\partial (A \operatorname{dev}(G))_{ij}}{\partial A_{mn}} &= \frac{\partial A_{it}}{\partial A_{mn}} \operatorname{dev}(G)_{tj} + A_{it} \frac{\partial \operatorname{dev}(G)_{tj}}{\partial A_{mn}} \\
&= \delta_{im} \delta_{tn} \left(A_{kt} A_{kj} - \frac{1}{3} A_{kl} A_{kl} \delta_{tj} \right) + A_{it} \left(\delta_{tn} A_{mj} + \delta_{jn} A_{mt} - \frac{2}{3} \delta_{tj} A_{mn} \right) \\
&= \delta_{im} A_{kn} A_{kj} - \frac{1}{3} \delta_{im} \delta_{jn} A_{kl} A_{kl} + A_{in} A_{mj} + \delta_{jn} A_{ik} A_{mk} - \frac{2}{3} A_{ij} A_{mn}
\end{aligned} \tag{89}$$

Thus:

$$\begin{aligned}
J_A &\equiv \frac{-3}{\tau_1} \frac{\partial \left(\det(A)^{\frac{5}{3}} A \operatorname{dev}(G) \right)_{ij}}{\partial A_{mn}} \\
&= \frac{-3}{\tau_1} \det(A)^{\frac{5}{3}} \left(\frac{5}{3} (A \operatorname{dev}(G))_{ij} A_{mn}^{-T} + A_{in} A_{mj} + \delta_{jn} G'_{im} + \delta_{im} G_{jn} - \frac{1}{3} \delta_{im} \delta_{jn} A_{kl} A_{kl} - \frac{2}{3} A_{ij} A_{mn} \right) \\
&= \frac{1}{\tau_1} \det(A)^{\frac{5}{3}} \left(-5 (A \operatorname{dev}(G)) \otimes A^{-T} + 2A \otimes A - 3 (A \otimes A)^{1,3} + \|A\|_F^2 (I \otimes I)^{2,3} - 3 \left(G' \otimes I + I \otimes G \right)^{2,3} \right)
\end{aligned} \tag{90}$$

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

8.2. Jacobian of Thermal Impulse ODEs

As demonstrated in 3.1, we have:

$$\frac{dJ_i}{dt} = \frac{J_i}{2} \left(-a + b \left(J_1^2 + J_2^2 + J_3^2 \right) \right) \tag{91}$$

where

$$a = \frac{2\rho_0}{\tau_2 T_0 \rho c_v} \left(E - E_{2A}(A) - E_3(\mathbf{v}) \right) \tag{92a}$$

$$b = \frac{\rho_0 \alpha^2}{\tau_2 T_0 \rho c_v} \tag{92b}$$

Thus, the Jacobian of the thermal impulse ODEs is:

$$\begin{pmatrix}
\frac{b}{2} \left(3J_1^2 + J_2^2 + J_3^2 \right) - \frac{a}{2} & b J_1 J_2 & b J_1 J_3 \\
b J_1 J_2 & \frac{b}{2} \left(J_1^2 + 3J_2^2 + J_3^2 \right) - \frac{a}{2} & b J_2 J_3 \\
b J_1 J_3 & b J_2 J_3 & \frac{b}{2} \left(J_1^2 + J_2^2 + 3J_3^2 \right) - \frac{a}{2}
\end{pmatrix} \tag{93}$$

8.3. WENO Matrices for N = 2

$$M_1 = \begin{pmatrix} 2\sqrt{\frac{5}{3}} + \frac{245}{18} & -\frac{236}{9} & \frac{245}{18} - 2\sqrt{\frac{5}{3}} \\ \sqrt{\frac{5}{3}} + \frac{65}{18} & -\frac{56}{9} & \frac{65}{18} - \sqrt{\frac{5}{3}} \\ \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \end{pmatrix} \tag{94a}$$

$$M_2 = \begin{pmatrix} \sqrt{\frac{5}{3}} + \frac{65}{18} & -\frac{56}{9} & \frac{65}{18} - \sqrt{\frac{5}{3}} \\ \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \\ \frac{65}{18} - \sqrt{\frac{5}{3}} & -\frac{56}{9} & \sqrt{\frac{5}{3}} + \frac{65}{18} \end{pmatrix} \tag{94b}$$

$$M_3 = \begin{pmatrix} \frac{5}{18} & \frac{4}{9} & \frac{5}{18} \\ \frac{65}{18} - \sqrt{\frac{5}{3}} & -\frac{56}{9} & \sqrt{\frac{5}{3}} + \frac{65}{18} \\ \frac{245}{18} - 2\sqrt{\frac{5}{3}} & -\frac{236}{9} & 2\sqrt{\frac{5}{3}} + \frac{245}{18} \end{pmatrix} \tag{94c}$$

$$M_1^{-1} = \begin{pmatrix} \frac{1}{60}\left(2 - 3\sqrt{15}\right) & \sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60}\left(62 - 9\sqrt{15}\right) \\ -\frac{1}{24} & \frac{1}{12} & \frac{23}{24} \\ \frac{1}{60}\left(3\sqrt{15} + 2\right) & -\sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60}\left(9\sqrt{15} + 62\right) \end{pmatrix} \tag{95a}$$

$$M_2^{-1} = \begin{pmatrix} \frac{1}{60}\left(3\sqrt{15} + 2\right) & \frac{14}{15} & \frac{1}{60}\left(2 - 3\sqrt{15}\right) \\ -\frac{1}{24} & \frac{13}{12} & -\frac{1}{24} \\ \frac{1}{60}\left(2 - 3\sqrt{15}\right) & \frac{14}{15} & \frac{1}{60}\left(3\sqrt{15} + 2\right) \end{pmatrix} \tag{95b}$$

$$M_3^{-1} = \begin{pmatrix} \frac{1}{60}\left(9\sqrt{15} + 62\right) & -\sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60}\left(3\sqrt{15} + 2\right) \\ \frac{23}{24} & \frac{1}{12} & -\frac{1}{24} \\ \frac{1}{60}\left(62 - 9\sqrt{15}\right) & \sqrt{\frac{3}{5}} - \frac{1}{15} & \frac{1}{60}\left(2 - 3\sqrt{15}\right) \end{pmatrix} \tag{95c}$$