

Lab Notes

Haran Jackson

January 24, 2017

Contents

1	To Do	2
1.1	Coding	2
1.2	Theoretical	2
1.3	Papers	2
2	Faster Solvers	3
2.1	Fast WENO Oscillation Indicator Calculation	3
2.2	Approximating Interface Terms in FV	4
2.3	Analytical Results for Basis Vectors	4
2.4	Operator Splitting	5
2.5	Distortion ODEs	7
2.5.1	Linearized Distortion ODEs Solver	7
2.5.2	Linearized Reduced Distortion ODE Solver	8
2.6	Primitive WENO and DG Reconstruction	9
2.7	Change to Row-Major Ordering	9
3	Slow Flow	21
3.1	Studying numerical smearing with slow flow past a barrier	21
4	RGFM	22

Chapter 1

To Do

1.1 Coding

- Test split-WENO in 2D
- Implement in CUDA
- Solid EOSs
- Implement Barton's damage model
- Test compressible Euler vs GPR - use narrowing domain from Toro's book
- Try conservative formulation [Peshkov, Grmela, Romenski]

1.2 Theoretical

- Investigate approximate Riemann solvers (e.g. Dumbser's HLLEM)
- Convergence of conservation of mass in RGFM

1.3 Papers

- Isobaric cookoff (make cookoff the focus)
- Split solver vs ADER-WENO
- Application of ADER-WENO to Barton's solid model (with Tome)
- HPR-RGFM paper (solve stationarity of interface solver under some conditions)
- Application of ADER-WENO to equations in other areas of Physics, Biology, & Economics

Chapter 2

Faster Solvers

2.1 Fast WENO Oscillation Indicator Calculation

The WENO oscillation indicator is defined as:

$$o = \Sigma_{mn} w_m w_n \quad (2.1)$$

where w_i are the WENO coefficients calculated for a particular stencil, and:

$$\Sigma_{mn} = \sum_{\alpha=1}^N \int_0^1 \psi_m^{(\alpha)} \psi_n^{(\alpha)} \quad (2.2)$$

By considering that:

$$o = \sum_{\alpha=1}^N \int_0^1 \left(\frac{d^\alpha w}{dx^\alpha} \right)^2 > 0 \quad (2.3)$$

we have that Σ is symmetric positive definite. Thus, it has a Cholesky decomposition $\Sigma = LL^T$. Thus:

$$o = \|w^T L\|^2 \quad (2.4)$$

L can be precalculated, and o calculated quickly as:

```

1: o = 0
2: for j = 1...n do
3:   tmp = 0
4:   for i = j...n do
5:     tmp = tmp + w(i) * L(i,j)
6:   end for
7:   o = o + tmp * tmp
8: end for

```

2.2 Approximating Interface Terms in FV

Instead of calculating the following:

$$\int D(q^-(x_0, t), q^+(x_0, t)) dt \quad (2.5)$$

I propose calculating the following:

$$D\left(\frac{1}{\Delta t} \int q^-(x_0, t) dt, \frac{1}{\Delta t} \int q^+(x_0, t) dt\right) \quad (2.6)$$

This obtains a large speedup with no discernable difference in the results of Stokes' First Problem.

2.3 Analytical Results for Basis Vectors

For $N = 1$, the Gauss-Legendre nodes on $[0, 1]$ are $\left\{\frac{1}{2}\left(1 - \frac{1}{\sqrt{3}}\right), \frac{1}{2}\left(1 + \frac{1}{\sqrt{3}}\right)\right\}$. Thus:

$$\psi_1(x) = -\sqrt{3}x + \frac{1 + \sqrt{3}}{2} \quad (2.7a)$$

$$\psi_2(x) = \sqrt{3}x + \frac{1 - \sqrt{3}}{2} \quad (2.7b)$$

$$\psi_1(1) = \frac{1 - \sqrt{3}}{2} \quad (2.8a)$$

$$\psi_2(1) = \frac{1 + \sqrt{3}}{2} \quad (2.8b)$$

$$\psi_1(1)\psi_1(1) = 1 - \frac{\sqrt{3}}{2} \quad (2.9a)$$

$$\psi_1(1)\psi_2(1) = -\frac{1}{2} \quad (2.9b)$$

$$\psi_2(1)\psi_2(1) = 1 + \frac{\sqrt{3}}{2} \quad (2.9c)$$

$$\int_m^{m+1} \psi_1(x) dx = \frac{-\sqrt{3}}{2} (2m+1) + \frac{1+\sqrt{3}}{2} = \frac{1}{2} - m\sqrt{3} \quad (2.10a)$$

$$\int_m^{m+1} \psi_2(x) dx = \frac{\sqrt{3}}{2} (2m+1) + \frac{1-\sqrt{3}}{2} = \frac{1}{2} + m\sqrt{3} \quad (2.10b)$$

The WENO matrices are:

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} - \sqrt{3} & \frac{1}{2} + \sqrt{3} \end{pmatrix}, \begin{pmatrix} \frac{1}{2} + \sqrt{3} & \frac{1}{2} - \sqrt{3} \\ \frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad (2.11)$$

The inverses are:

$$\frac{1}{\sqrt{3}} \begin{pmatrix} \sqrt{3} + \frac{1}{2} & -\frac{1}{2} \\ \sqrt{3} - \frac{1}{2} & \frac{1}{2} \end{pmatrix}, \frac{1}{\sqrt{3}} \begin{pmatrix} \frac{1}{2} & \sqrt{3} - \frac{1}{2} \\ -\frac{1}{2} & \sqrt{3} + \frac{1}{2} \end{pmatrix} \quad (2.12)$$

The weights for both nodes are 0.5 so $\int_0^1 \psi_i \psi_j dx = \frac{\delta_{ij}}{2}$ and $\int_0^1 \psi_i \psi_j' dx = (-1)^j \frac{\sqrt{3}}{2}$.

$$\begin{aligned} I_{11} - I_2^T &= \frac{1}{2} \begin{pmatrix} 2 - \sqrt{3} & -1 \\ -1 & 2 + \sqrt{3} \end{pmatrix} - \frac{1}{2} \begin{pmatrix} -\sqrt{3} & -\sqrt{3} \\ \sqrt{3} & \sqrt{3} \end{pmatrix} \\ &= \frac{1}{2} \begin{pmatrix} 2 & -(1 - \sqrt{3}) \\ -(1 + \sqrt{3}) & 2 \end{pmatrix} \end{aligned} \quad (2.13)$$

$$(I_{11} - I_2^T)^{-1} = \frac{1}{3} \begin{pmatrix} 2 & 1 - \sqrt{3} \\ 1 + \sqrt{3} & 2 \end{pmatrix} \quad (2.14)$$

Using a precalculated, analytical form of U in the DG predictor for $N = 1$ obtains a ~30% speedup on Stokes' First Problem.

2.4 Operator Splitting

Noting that $\frac{d\rho}{dt} = 0$ over the ODE time step, we must solve the following system:

$$\frac{dA_{ij}}{dt} = \frac{-\psi_{ij}}{\theta_1(\tau_1)} = \frac{-3}{\tau_1} |A|^{\frac{5}{3}} A \operatorname{dev}(G) \quad (2.15a)$$

$$\frac{dJ_i}{dt} = \frac{-H_i}{\theta_2(\tau_2)} = -\frac{1}{\tau_2} \frac{T\rho_0}{T_0\rho} J_i \quad (2.15b)$$

Many different solvers can be used to solve the homogeneous part of the system. So far, this has been tested with SLIC, WENO, and DG. A split-WENO scheme seems to be the fastest and most accurate method available. The results using split-WENO to solve Stokes' First Problem with $N = 1$ are shown in 2.1. These results are comparable to the corresponding results using ADER-WENO, as seen in 2.2.

The results of using a 3rd-order split-WENO scheme to solve Stokes' First Problem are shown in 2.3. Note the close agreement with the Navier-Stokes solution, closely matching the result using ADER-WENO. The split-WENO scheme took 15 times less CPU time than the ADER-WENO scheme.

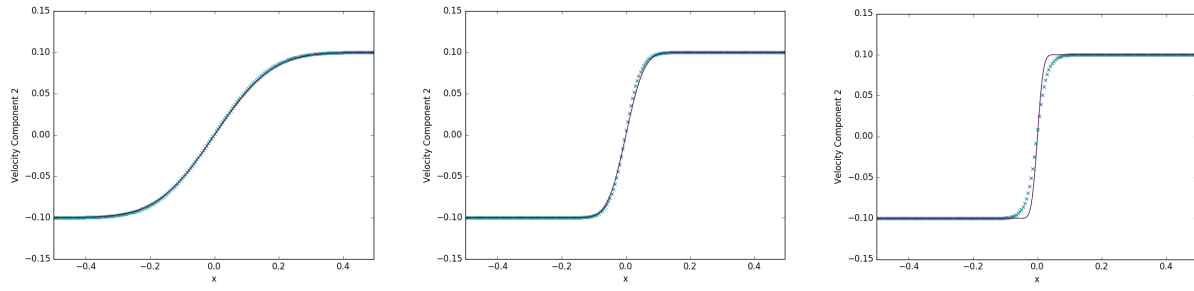


Figure 2.1: Results of solving Stokes' First Problem ($\mu = 10^{-2}, \mu = 10^{-3}, \mu = 10^{-4}$) with a split-WENO scheme ($N = 1$)

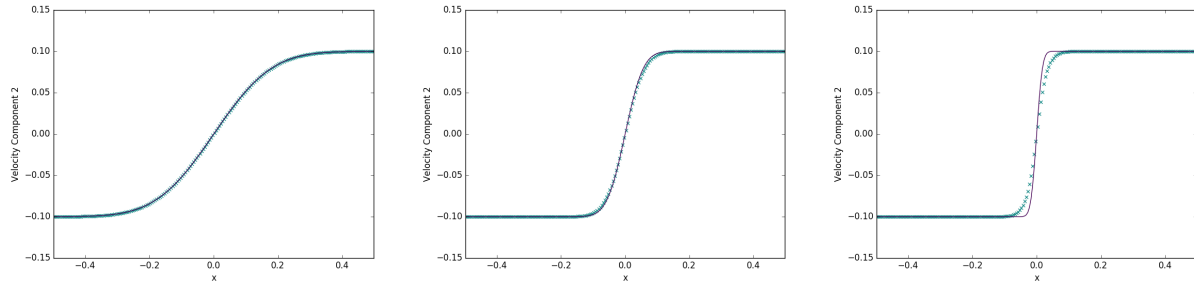


Figure 2.2: Results of solving Stokes' First Problem ($\mu = 10^{-2}, \mu = 10^{-3}, \mu = 10^{-4}$) with an ADER-WENO scheme ($N = 1$)

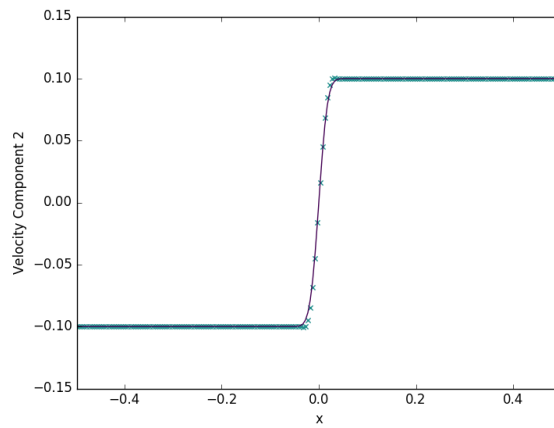


Figure 2.3: Results of solving Stokes' First Problem ($\mu = 10^{-4}$) with a split-WENO scheme ($N = 2$)

2.5 Distortion ODEs

2.5.1 Linearized Distortion ODEs Solver

Note that $A^* = \left(\frac{\rho}{\rho_0}\right)^{\frac{1}{3}} I$ is a stationary point of the ODE for A . Linearizing the ODE around A^* gives:

$$\begin{aligned}
\frac{dA}{dt} &\approx J_A(A^*)(A - A^*) \\
&= \frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{5}{3}} \left(\left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}} \delta_{in} \delta_{mj} + \left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}} \delta_{jn} \delta_{im} + \left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}} \delta_{im} \delta_{jn} - \frac{1}{3} \left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}} \delta_{im} \delta_{jn} \delta_{kl} \delta_{kl} - \frac{2}{3} \left(\frac{\rho}{\rho_0}\right)^{\frac{2}{3}} \delta_{ij} \delta_{mn} \right) \\
&\quad \times \left(A_{mn} - \left(\frac{\rho}{\rho_0}\right)^{\frac{1}{3}} \delta_{mn} \right) \\
&= \frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{7}{3}} \left(\delta_{in} \delta_{mj} + \delta_{im} \delta_{jn} - \frac{2}{3} \delta_{ij} \delta_{mn} \right) \left(A_{mn} - \left(\frac{\rho}{\rho_0}\right)^{\frac{1}{3}} \delta_{mn} \right) \\
&= \frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{7}{3}} \left(A_{mn} \left(\delta_{in} \delta_{mj} + \delta_{im} \delta_{jn} - \frac{2}{3} \delta_{ij} \delta_{mn} \right) - \left(\frac{\rho}{\rho_0}\right)^{\frac{1}{3}} \delta_{mn} \left(\delta_{in} \delta_{mj} + \delta_{im} \delta_{jn} - \frac{2}{3} \delta_{ij} \delta_{mn} \right) \right) \\
&= \frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{7}{3}} \left(\left(A_{ji} + A_{ij} - \frac{2}{3} \text{tr}(A) \delta_{ij} \right) - \left(\frac{\rho}{\rho_0}\right)^{\frac{1}{3}} (\delta_{ij} + \delta_{ij} - 2\delta_{ij}) \right) \\
&= \frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{7}{3}} \left(A + A^T - \frac{2}{3} \text{tr}(A) I \right)
\end{aligned} \tag{2.16}$$

The matrix for this system, in row-major form, is:

$$\frac{-3}{\tau_1} \left(\frac{\rho}{\rho_0}\right)^{\frac{7}{3}} \begin{pmatrix} \frac{4}{3} & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 & -\frac{2}{3} \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{2}{3} & 0 & 0 & 0 & \frac{4}{3} & 0 & 0 & 0 & -\frac{2}{3} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ -\frac{2}{3} & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 & \frac{4}{3} \end{pmatrix} \tag{2.17}$$

The eigenvalues and eigenvectors are:

$$\{0, 0, 0, 0, -2k, -2k, -2k, -2k, -2k\} \tag{2.18}$$

$$\begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{2.19}$$

where $k = \frac{3}{\tau_1} \left(\frac{\rho}{\rho_0} \right)^{\frac{7}{3}}$. Thus, the solution is:

$$\begin{aligned}
& \frac{A_{12} - A_{21}}{2} \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{A_{13} - A_{31}}{2} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} + \frac{A_{23} - A_{32}}{2} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix} \\
& + \frac{A_{11} + A_{22} + A_{33}}{3} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
& + \frac{2A_{22} - A_{11} - A_{33}}{3} e^{-2kt} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{2A_{33} - A_{11} - A_{22}}{3} e^{-2kt} \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
& + \frac{A_{12} + A_{21}}{2} e^{-2kt} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{A_{13} + A_{31}}{2} e^{-2kt} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} + \frac{A_{23} + A_{32}}{2} e^{-2kt} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}
\end{aligned} \tag{2.20}$$

This is equal to:

$$\frac{1}{2} (A - A^T) + \frac{\text{tr}(A)}{3} I + e^{-2kt} \left(\frac{1}{2} (A + A^T) - \frac{\text{tr}(A)}{3} I \right) \tag{2.21}$$

Results with Stokes' First Problem look good with this linearisation. The ODE step takes a negligible amount of time, meaning that if accuracy is maintained to second order, the solver is now fast enough.

2.5.2 Linearized Reduced Distortion ODE Solver

Taking system (??), note that the Jacobian of the system is given by:

$$J = -k \begin{pmatrix} 4x_1 - x_2 - x_3 & -x_1 & -x_1 \\ -x_2 & 4x_2 - x_3 - x_1 & -x_3 \\ -x_3 & -x_3 & 4x_3 - x_1 - x_2 \end{pmatrix} \tag{2.22}$$

Evaluated at stationary point $x_i = \sqrt[3]{c}$ we have:

$$J(\mathbf{x}_0) = -k \sqrt[3]{c} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \tag{2.23}$$

Thus, the system is linearized to:

$$\begin{aligned}
\frac{d\mathbf{x}}{dt} & \approx -k \sqrt[3]{c} \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \left(\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} - \sqrt[3]{c} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right) \\
& = k \sqrt[3]{c} \begin{pmatrix} -2 & 1 & 1 \\ 1 & -2 & 1 \\ 1 & 1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}
\end{aligned} \tag{2.24}$$

The eigenvalues of this system matrix are $\{-3k\sqrt[3]{c}, -3k\sqrt[3]{c}, 0\}$ and the eigenvectors are:

$$\begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (2.25)$$

Thus, the linearized solution is:

$$\mathbf{x}(t) = \frac{-2x_1 + x_2 + x_3}{3} e^{-3k\sqrt[3]{c}t} \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix} + \frac{x_1 - 2x_2 + x_3}{3} e^{-3k\sqrt[3]{c}t} \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix} + \frac{x_1 + x_2 + x_3}{3} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \quad (2.26)$$

This may represent a faster way to calculate the evolution of the stretch terms of A . Note that some kind of normalization will probably be necessary, as:

$$\frac{x_1 + x_2 + x_3}{3} \geq (x_1 x_2 x_3)^{\frac{1}{3}} \quad (2.27)$$

with equality if and only if $x_1 = x_2 = x_3$.

2.6 Primitive WENO and DG Reconstruction

As suggested in [?], the WENO and DG can be performed in primitive variables, which is less computationally expensive than evaluating fluxes using conserved variables. Achieves around 20% speedup in DG step, at double cost in WENO step. Minimal speedup in FV step, as both primitive and conserved variables must be calculated for the flux updates. Not enough.

2.7 Change to Row-Major Ordering

The original GPR papers state the equations for A in column-major order, probably because the authors use Fortran. For C++ and Python implementations it is faster to work in row-major order. ~10% speedup was achieved by implementing this.

The GPR equations are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho v_k)}{\partial x_k} = 0 \quad (2.28a)$$

$$\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho E v_k + (p\delta_{ik} - \sigma_{ik}) v_i + q_k)}{\partial x_k} = 0 \quad (2.28b)$$

$$\frac{\partial (\rho v_i)}{\partial t} + \frac{\partial (\rho v_i v_k + p\delta_{ik} - \sigma_{ik})}{\partial x_k} = 0 \quad (2.28c)$$

$$\frac{\partial A_{ij}}{\partial t} + \frac{\partial (A_{ik} v_k)}{\partial x_j} + v_k \left(\frac{\partial A_{ij}}{\partial x_k} - \frac{\partial A_{ik}}{\partial x_j} \right) = -\frac{\psi_{ij}}{\theta_1(\tau_1)} \quad (2.28d)$$

$$\frac{\partial (\rho J_i)}{\partial t} + \frac{\partial (\rho J_i v_k + T\delta_{ik})}{\partial x_k} = -\frac{\rho H_i}{\theta_2(\tau_2)} \quad (2.28e)$$

$$\Psi_{ij} = \rho v_i v_j - \sigma_{ij} \quad (2.35a)$$

$$\Phi_{ij}^k = \rho v_k \psi_{ij} - v_m \frac{\partial \sigma_{mk}}{\partial A_{ij}} \quad (2.35b)$$

$$\Omega_i = v_i (E + \rho E_\rho) - \frac{\sigma_{im} v_m}{\rho} + T_\rho H_i \quad (2.35c)$$

$$\Upsilon = \frac{\|\mathbf{v}\|^2 + \mathbf{H} \cdot \mathbf{J} - E - \rho E_\rho}{\rho E_p} \quad (2.35d)$$

$$\tilde{\mathbf{H}} = E_{JJ} \quad (2.35e)$$

$$\mathbf{S}_p = \frac{1}{\theta_1(\tau_1)} \begin{pmatrix} 0 \\ (\gamma - 1) \rho \|\psi\|_F^2 \\ 0 \\ 0 \\ 0 \\ -\psi_{11} \\ -\psi_{12} \\ -\psi_{13} \\ -\psi_{21} \\ -\psi_{22} \\ -\psi_{23} \\ -\psi_{31} \\ -\psi_{32} \\ -\psi_{33} \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{1}{\theta_2(\tau_2)} \begin{pmatrix} 0 \\ (\gamma - 1) \rho \|\mathbf{H}\|^2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -H_1 \\ -H_2 \\ -H_3 \end{pmatrix} \quad (2.44)$$

[illegible]

[illegible]

[illegible]

Chapter 3

Slow Flow

3.1 Studying numerical smearing with slow flow past a barrier

A checkerboard pattern appears around the corner of the barrier, leading to a crash, using reflective boundary conditions (in velocity) for the barrier. Do we need a staggered grid?

Chapter 4

RGFM

The RGFM does nothing without a temperature fix when applied to the heat conduction test. The linearisation upon which it is based results in a stationary solution when $q_L = q_R$ and $\sigma_L = \sigma_R$ initially. Barton's RGFM is similar. Should q be fixed? Maybe use analytical solution to heat equation at $t = \Delta t$?

Bibliography