1 Current work in 1-d

We are looking at the following 3 equations in 1 dimensions of you publications:

$$\frac{\partial}{\partial t}\rho + \frac{\partial}{\partial x}\rho u = 0 \tag{1.1}$$

$$\frac{\partial}{\partial t}\rho u + \frac{\partial}{\partial x}(\rho u^2 + \rho c(1-c)u_r^2 + P) = 0 \tag{1.2}$$

$$\frac{\partial}{\partial t}u_r + \frac{\partial}{\partial x}(uu_r + \frac{1-2c}{2}u_r^2 + \Psi(P)) = 0 \tag{1.3}$$

With the variables

$$U = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} \rho \\ \rho u \\ u_r \end{bmatrix}$$

the Jacobian is

$$J = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{u_2^2}{u_1^2} + c(1-c)u_3^2 + \partial_{u_1}P & \frac{2u_2}{u_1} & u_1c(1-c)2u_3 \\ -\frac{u_2}{u_1^2}u_3 + \partial_{u_1}\Psi(\rho) & \frac{u_3}{u_1} & \frac{u_2}{u_1} + (1-2c)u_3 \end{pmatrix} \tag{1.4}$$

Now the question we are still looking at is, what EOS to use and what to keeps constant. Currently we are looking at 3 cases

1. As in you publication and in the Fortran program:
   - Use the EOS
   $$P = K_2\rho_2^\gamma \tag{1.5}$$
   - Keep $c = \alpha\rho_2/\rho$ constant (In Fortran: `CCL`)
   - Do the initialization step as in the Fortran program, compute K_ρ (In Fortran: `CT = PLL/(RHOL**G)`) for a second EOS
   $$P = K_\rho\rho^\gamma \tag{1.6}$$
   and keep K_ρ constant
   - In the equation for ρu use
   $$P = K_\rho\rho^\gamma$$
   - In the equation for u_r use
   $$\Psi = \frac{\gamma}{\gamma-1}K_2^{1/\gamma}P^{(\gamma-1)/\gamma} - \frac{P}{\rho_1}$$
   - The corresponding Jacobian would be
   $$J = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{u_2^2}{u_1^2} + c(1-c)u_3^2 + \gamma K_\rho u_1^{\gamma-1} & \frac{2u_2}{u_1} & u_1c(1-c)2u_3 \\ -\frac{u_2}{u_1^2}u_3 + \left(\frac{K_2}{K_\rho}\right)^{1/\gamma}\gamma K_\rho u_1^{\gamma-2} - \frac{\gamma K_\rho u_1^{\gamma-1}}{\rho_1} & \frac{u_3}{u_1} & \frac{u_2}{u_1} + (1-2c)u_3 \end{pmatrix} \tag{1.7}$$
2. In the second version we don't use the second EOS $P = K_\rho\rho^\gamma$ at all. The set of equations is the same and we initialize all constants in the same way, but in each step
   - we are computing
   $$\rho_2 = \frac{c\rho\rho_1}{\rho(c-1)+\rho_1}$$

- and with this density value for ρ_2

$$P = K_2 \rho_2^\gamma.$$

- We are using these values for ρ_2 and P for the equations and the CLF condition.
- This makes the derivation of the Jacobian much more complicated and we get

$$J = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{u_2^2}{u_1^2} + c(1-c)u_3^2 + K_2\gamma\rho_2^{\gamma-1}\frac{c\rho_1^2}{(\rho_1+(c-1)\rho)^2} & \frac{2u_2}{u_1} & u_1 c(1-c)2u_3 \\ -\frac{u_2}{u_1^2}u_3 + K_2\left(\gamma\rho_2^{\gamma-2} - \frac{\gamma\rho_2^{\gamma-1}}{\rho_1}\right)\frac{c\rho_1^2}{(\rho_1+(c-1)\rho)^2} & \frac{u_3}{u_1} & \frac{u_2}{u_1} + (1-2c)u_3 \end{pmatrix} \tag{1.8}$$

  still wiht ρ_2 in it.

3. The last alternative is to look just at the EOS for the whole density $P = K_\rho \rho^\gamma$.

   - In this case we compute K_ρ as in the Fortran program to compare the results.
   - In the equation for u_r use again this equation for P, but we have to recompute Ψ.
   - Using $P = K_\rho \rho^\gamma$ I get for Ψ

$$\Psi = \frac{1}{c}\left(\frac{\gamma}{\gamma-1}\frac{P}{\rho} - \frac{P}{\rho_1}\right) \tag{1.9}$$

$$= \frac{K_g}{c}\left(\frac{\gamma}{\gamma-1}\rho^{\gamma-1} - \frac{\rho^\gamma}{\rho_1}\right) \tag{1.10}$$

   - With these expressions I get for the Jacobian

$$J = \begin{pmatrix} 0 & 1 & 0 \\ -\frac{u_2^2}{u_1^2} + c(1-c)u_3^2 + K_\rho\gamma u_1^{\gamma-1} & \frac{2u_2}{u_1} & u_1 c(1-c)2u_3 \\ -\frac{u_2}{u_1^2}u_3 + \frac{1}{c}\left(\frac{1}{u_1} - \frac{1}{\rho_1}\right)K_\rho\gamma u_1^{\gamma-1} & \frac{u_3}{u_1} & \frac{u_2}{u_1} + (1-2c)u_3 \end{pmatrix} \tag{1.11}$$

Currently we run all three cases with Lax-Friedrich and FORCE, but the code has to be checked again.

2 Current work in 2-d

We implemented the 5 equations as discussed before, but there are still errors in the code. Next week I want to have a look at the Jacobian and to help to find the bugs in the code.