Comparative study of density-based vs. pressure-based solvers for supersonic flow

Bachelor Thesis

Sean Bone

Supervisor:
Prof. Dr. P. Jenny
Insitute for Fluid Dynamics
ETH Zürich

November 21, 2020

Goal of the project

- Project EULER: build a sounding rocket to reach 30k ft ($\approx$ 9km)
- Supersonic!
- What solver to use for CFD?
- Opportunity for real-world validation

Solvers compared

- **rhoPimpleFoam:** *pressure*-based compressible and transient solver
- **rhoCentralFoam:** *density*-based compressible and transient solver

Conservation of mass (continuity), momentum, and energy:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho U) + \nabla \cdot (\rho U \otimes U) + \nabla p - \nabla \cdot \sigma = S, \tag{2}$$

$$\frac{\partial}{\partial t}E + \nabla \cdot (UE + Up) - \nabla \cdot \dot{q} - \nabla \cdot (U\sigma) = Q. \tag{3}$$

- **Pressure-based solvers:**
  - Solve the momentum equation for U
  - Derive *pressure equation* from continuity and momentum equations
  - Solve pressure equation for p
  - *Correct* the velocity field with the pressure field

- **Pressure-based solvers:**
  - Solve the momentum equation for U
  - Derive *pressure equation* from continuity and momentum equations
  - Solve pressure equation for p
  - *Correct* the velocity field with the pressure field
- **In general, we expect:**
  - Diffuse shocks
  - Shocks are diffused over time

- **Pressure-based solvers:**
  - Solve the momentum equation for U
  - Derive *pressure equation* from continuity and momentum equations
  - Solve pressure equation for p
  - *Correct* the velocity field with the pressure field
- **In general, we expect:**
  - Diffuse shocks
  - Shocks are diffused over time
- **Density-based solvers:**
  - Solve the momentum equation for U
  - Solve continuity equation for ρ
  - Use equations of state to obtain p
  - Shocks are explicitly considered when solving equations
  - Make assumptions about shock speeds

- **Pressure-based solvers:**
  - Solve the momentum equation for U
  - Derive *pressure equation* from continuity and momentum equations
  - Solve pressure equation for p
  - *Correct* the velocity field with the pressure field
- **In general, we expect:**
  - Diffuse shocks
  - Shocks are diffused over time

- **Density-based solvers:**
  - Solve the momentum equation for U
  - Solve continuity equation for ρ
  - Use equations of state to obtain p
  - Shocks are explicitly considered when solving equations
  - Make assumptions about shock speeds
- **In general, we expect:**
  - Sharp shocks
  - Shocks are conserved over time

Comparison 1: supersonic flow past wedge

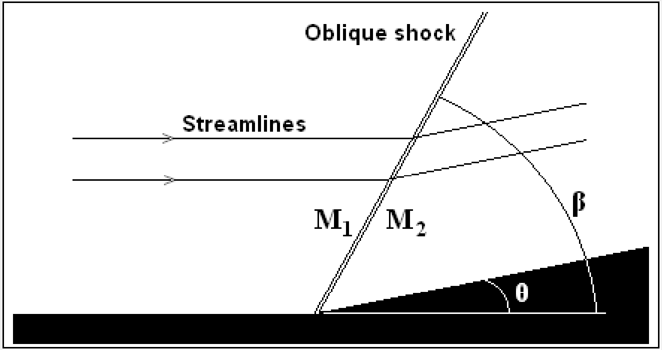


Figure 1: An illustration of the wedge problem (Source: Wikimedia Commons)

Comparison 1: supersonic flow past wedge

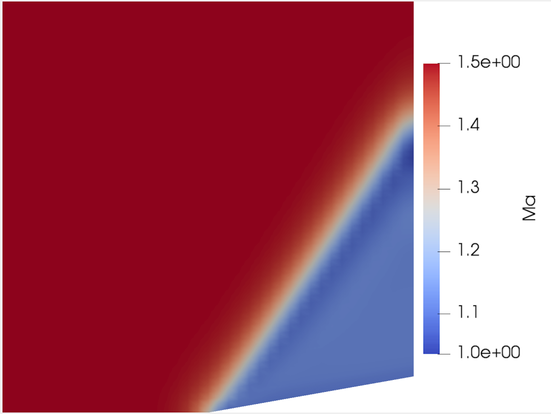


Figure 2: rhoPimpleFoam's solution

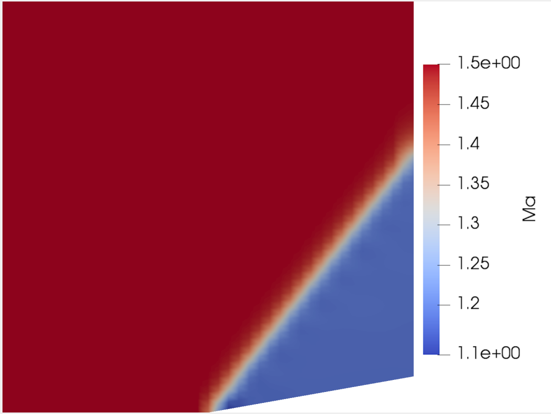


Figure 3: rhoCentralFoam's solution

Table 1: Results from the rhoPimpleFoam solver

θ	M_1	β	Rel. error	M_2	Rel. error	p_2/p_1	Rel. error
10	**1.5**	54	-4.73%	1.115618327	0.11%	1.666173008	0.00%
	2	39.5	0.47%	1.641311905	0.05%	1.707707792	0.07%
20	**2**	52	-2.66%	1.212560159	0.19%	2.842354736	-0.02%
	3	39	3.27%	1.988981188	-0.26%	3.798757552	0.73%

Table 2: Results from the rhoCentralFoam solver

θ	M_1	β	Rel. error	M_2	Rel. error	p_2/p_1	Rel. error
10	**1.5**	52	-8.25%	1.180810638	5.96%	1.576529862	-5.38%
	2	38.5	-2.07%	1.690511905	3.05%	1.658893471	-2.79%
20	**2**	49.5	-7.34%	1.331521595	10.02%	2.641398152	-7.09%
	3	37.5	-0.70%	2.127863366	6.71%	3.58902854	-4.83%

Comparison 1: conclusions

- **rhoPimpleFoam** (pressure-based):
  - Diffuses shock waves
  - Diffusion gets worse further downstream
  - Has overshooting
  - Predicts slope fairly well despite diffusion
  - Predicts values before/after shocks extremely well ($< 1\%$ error)
- **rhoCentralFoam** (density-based):
  - Resolves very sharp shocks
  - Preserves shocks effectively
  - No overshooting
  - Not always as accurate at predicting shock slope
  - Presents much larger errors when predicting values before/after shock (up to 10%! error)

Comparison 2: Sod's shock tube

- 1-D geometry
- Setup is a standard Riemann problem:

$$\begin{pmatrix} p_L \\ U_L \\ \rho_L \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \\ 1.0 \end{pmatrix} ; \quad \begin{pmatrix} p_R \\ U_R \\ \rho_R \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.0 \\ 0.125 \end{pmatrix} \tag{4}$$

Comparison 2: Sod's shock tube

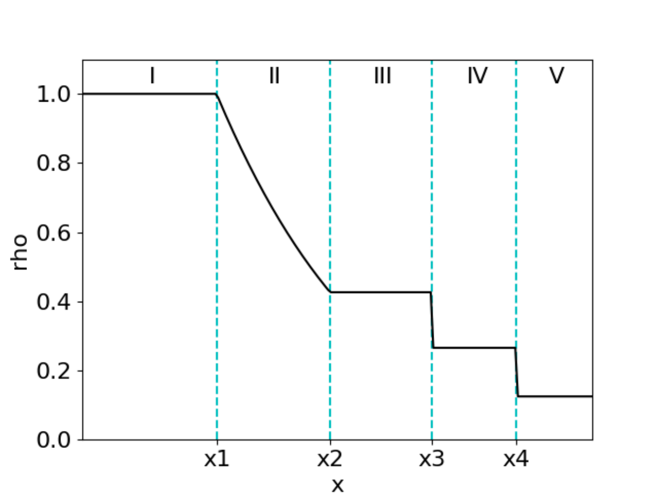


Figure 4: The exact solution of density at $t = 0.2$, with the ensuing fluid regions.

- BUT: OpenFOAM has variables p, U, $\mathbf{T}$
- Use ideal gas law to calculate value of T:

$$pV = nRT, \quad \rho = \frac{nM}{V} \quad \Rightarrow T = \frac{pM}{\rho R} \tag{5}$$

$$\begin{pmatrix} p_L \\ U_L \\ T_L \end{pmatrix} = \begin{pmatrix} 1.0 \\ 0.0 \\ 3.484290 \times 10^{-3} \end{pmatrix}; \quad \begin{pmatrix} p_R \\ U_R \\ T_R \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.0 \\ 2.787432 \times 10^{-3} \end{pmatrix} \tag{6}$$

Comparison 2: Sod's shock tube

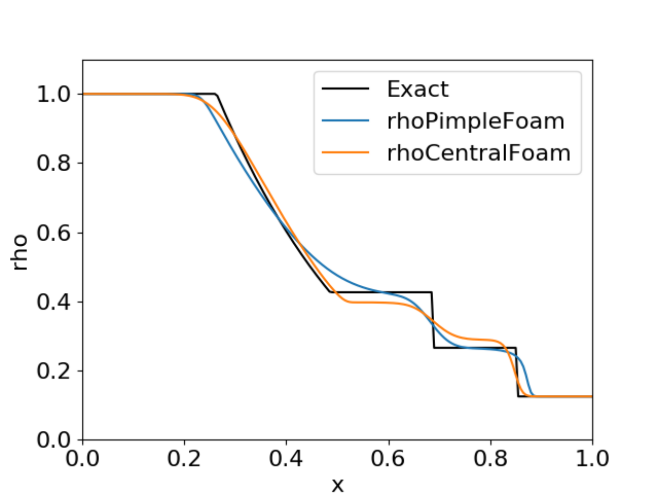


Figure 5: Comparison to exact solution of density at $t = 0.2$.

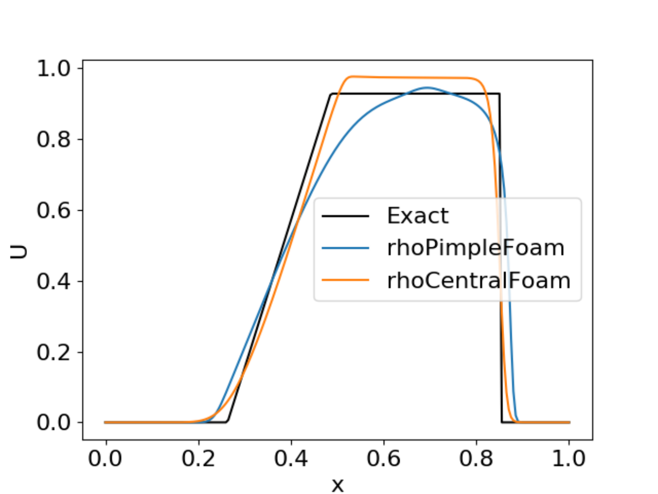
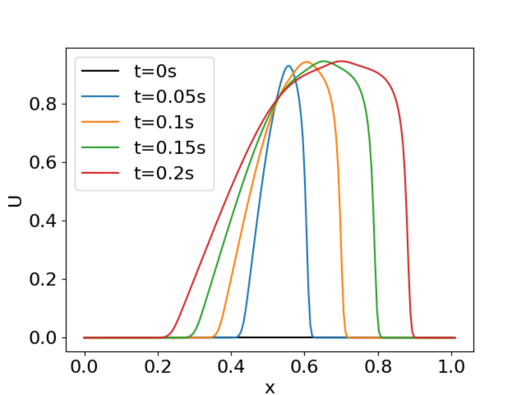
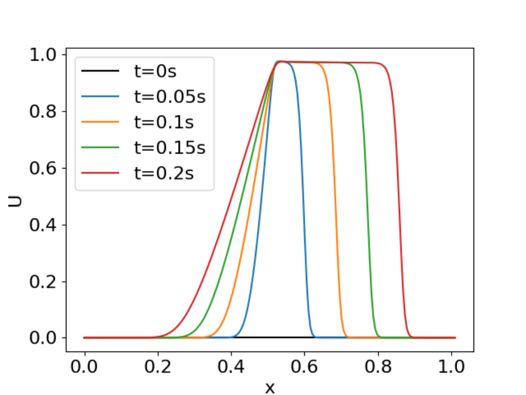


Figure 6: Comparison to exact solution of velocity at $t = 0.2$.



(a) rhoPimpleFoam



(b) rhoCentralFoam

Figure 7: Time evolution of the velocity solution from $t = 0$ to $t = 0.2$.

A more quantitative comparison metric, the L_1 norm:

$$L_1(Q^n, Q^*) = \frac{1}{Q_{ref}} \sum_{i=0}^{N=200} |Q^*(i \cdot \frac{1}{N}, n \cdot \Delta t) - Q_i^n| \tag{7}$$

Comparison 2: Sod's shock tube

Table 3: Normalized L_1 error norms of both numerical solutions

rhoPimpleFoam	**t = 0.05**	**t = 0.1**	**t = 0.15**	**t = 0.2**
p	1.9051	2.6689	3.3577	3.8180
ρ	1.7071	2.5219	3.2185	3.7374
U	4.7684	6.5525	8.3462	9.2785
T	2.7284	3.7757	4.8423	5.4644
rhoCentralFoam	**t = 0.05**	**t = 0.1**	**t = 0.15**	**t = 0.2**
p	1.7272	2.3440	2.8524	3.3806
ρ	1.6435	2.4459	3.1280	3.7782
U	3.7053	4.9110	6.0614	7.1656
T	3.4495	5.6248	7.6288	9.5964

- **rhoPimpleFoam** (pressure-based):
  - Diffuses shock waves
  - Diffusion gets worse further downstream
  - May show overshooting
  - Predicts shock speed fairly well despite diffusion
  - Predicts values before/after shocks quite well
- **rhoCentralFoam** (density-based):
  - Resolves very sharp shocks
  - Preserves shocks effectively
  - No overshooting
  - Not always as accurate at predicting shock speed
  - Presents larger errors when predicting values before/after shock

Overall conclusions

- **rhoPimpleFoam** (pressure-based):
  - Diffuses shock waves
  - Diffusion gets worse further downstream
  - May show overshooting
  - Predicts shock speed fairly well despite diffusion
  - Predicts values before/after shocks quite well
  - Overall errors are larger
- **rhoCentralFoam** (density-based):
  - Resolves very sharp shocks
  - Preserves shocks effectively
  - No overshooting
  - Not always as accurate at predicting shock speed
  - Presents larger errors when predicting values before/after shock
  - Overall errors are smaller

Table 4: Solver runtime

Runtime [s]	Oblique shock				Shock tube
rhoPimpleFoam	645.49	634.95	636.11	631.61	32.07
rhoCentralFoam	353.67	351.44	346.42	344.58	8.74
Speedup	183%	181%	184%	183%	367%

Future work and investigations

- Rocket simulations!
- Impact on drag coefficients
- Detailed study of time evolution of errors
- Detailed study of run-time performance

Thank you for your attention!