Validation of Tools to Accelerate High-Speed CFD Simulations Using OpenFOAM

Daniel E. R. Espinoza*, Thomas J. Scanlon†
and Richard E. Brown‡
Centre for Future Air-Space Transportation Technology,
University of Strathclyde, Glasgow, G1 1XJ, United Kingdom

Local time stepping (LTS) and adaptive mesh refinement (AMR) have been implemented into *rhoCentralFoam*, a compressible solver within the open source computational fluid dynamics (CFD) code OpenFOAM. The LTS solver has been validated using a compressible Couette channel with heat transfer and a supersonic flat plate. An excellent concurrence is found for these cases in comparison with the solutions with the obtained results being, respectively, 2.56 and 8.96 times faster compared to the unmodified solver. The AMR solver was employed to simulate hypersonic flow over a 30°wedge, and the Sod shock tube test case, and has also achieved satisfactory agreement with the analytical solutions. These results highlight the potentially significant computational cost savings that may be achieved when solving high-speed, compressible flows using this approach.

Nomenclature

F	Flux array
S	Source terms array
W	Conserved variables array
$\vec{v}$	Velocity vector, m/s
a	Acceleration factor
c	Courant number
H	Height of the channel, m
k	Thermal conductivity, W/(m·K)
L	Length of the shock tube
M	Mach number
p	Pressure, Pa
Re	Reynolds number
T	Temperature, K
t	Time, s
U	Flow speed in the x direction, m/s
x	Horizontal coordinate, m
y	Vertical coordinate, m
AMR	Adaptive Mesh Refinement
CFD	Computational Fluid Dynamics
DPLR	Data-Parallel Line Relaxation
LAURA	Langley Aerothermodynamic Upwind Relaxation Algorithm
LTS	Local time stepping
OpenFOAM	Open Source Field Operation And Manipulation
TVD	Total Variation Diminishing

Subscripts

∞	Upstream conditions
BW	Bottom wall
l	Initial conditions at left half of shock tube
max	Maximum
r	Initial conditions at right half of shock tube
TW	Top wall

Symbols

Δt	Time-step, s
μ	Dynamic viscosity, Pa·s
∇	Gradient operator
ρ	Density, kg/m^3

*Ph.D. Student, Department of Mechanical and Aerospace Engineering, AIAA student member
†Senior Lecturer, Department of Mechanical and Aerospace Engineering
‡Professor, Department of Mechanical and Aerospace Engineering, AIAA Senior member

I. Introduction

To be able to satisfactorily capture the essential flow features in the simulation of hypersonic flows, good practice dictates that a specific set of procedures should be followed. These include the categorisation of the correct physics (e.g. non-equilibrium effects,[1] modelling of turbulence[2]) as well as the selection of the most appropriate solution methodologies to capture the essential physics (e.g. shock-capturing and shock-fitting approaches,[3] choice of TVD schemes[4,5]). Whereas a significant amount of research has been dedicated to these areas, this paper focuses on how the spatial and temporal resolution of calculations may be improved in order to achieve satisfactory numerical results with an accompanying reduction in computational expense.

With regard to spatial resolution, a computational mesh that attempts to follow the shock structure is an essential feature of the solution methodology. In terms of the cell shape, a tetrahedral mesh can be created more quickly than a structured one and delivers greater flexibility, in particular for complex geometries. Poor predictions of heat transfer can result[6] however, and, for this reason, standard practice for unstructured meshes is to attempt to have prismatic cells aligned with solid surfaces and the shock wave itself.[7] This alignment of the grid with the shock wave is particularly important in order to avoid significant fluctuations occurring in the prediction of the stagnation point heat flux.[8] In addition, a sufficient degree of cell refinement is required near the shock wave simply in order to properly capture the spatial variation in the flow properties there.[9] In this paper we present a mesh refinement strategy which significantly improves the capturing of spatial gradients in the flow field.

The rate of temporal advance of the computation must also be such that the numerical stability and accuracy of the calculation are preserved. We present a time-acceleration technique which shows potential to greatly accelerate the computation of the initial transient within the computation and hence the rate of convergence of the calculation towards its steady state.

Some well-known hypersonic codes (e.g. LAURA[10] and DPLR[11]) contain features to deal with the spatial and temporal issues described above. Although the open source computational fluid dynamics code used in this paper, OpenFOAM,[12] does have such capabilities, the libraries that include these features have yet to be included in its high-speed compressible solvers. The objective of this paper is to incorporate local time stepping (LTS) and adaptive mesh refinement (AMR) into the OpenFOAM high-speed compressible solver *rhoCentralFoam*, and to validate the results of computations using this new approach against appropriate analytical solutions or experimental data.

II. Methodology

OpenFOAM is an open source computational fluid dynamics (CFD) software package consisting of a set of flexible C++ modules to resolve complex fluid flows. The package includes tools for meshing and pre- and post-processing, as well as solvers for both incompressible and compressible flows.[12]

Amongst the compressible solvers that are available in OpenFOAM, *rhoCentralFoam*[13] has been chosen for the simulations presented in this work, largely because of its satisfactory performance in previously-conducted high-speed compressible flow simulations.[13,14] *rhoCentralFoam* is an unsteady, compressible solver, that uses semi-discrete, non-staggered, Godunov-type central[15] and upwind-central[16] schemes. These schemes both avoid the explicit need for a Riemann solver, resulting in a numerical approach that is both simple and efficient.[16]

II.A. Local time stepping

Local time stepping (LTS) is a technique used to accelerate the convergence of computations towards their steady-state solutions. As *rhoCentralFoam* is an unsteady solver, the solution goes through an initial transient before reaching steady state. In the present form of *rhoCentralFoam*, the rate of temporal advance is the same for all the cells in the computational domain, and is thus limited by the smallest time-step

allowed for any cell. LTS allows each cell to advance in time at their own time-step.

While the Navier-Stokes equations have the form

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S} \tag{1}$$

where $\mathbf{W}$ represents the conserved variables, $\mathbf{F}$ the fluxes, $\mathbf{S}$ the source terms, t the time, and ∇ the gradient operator, the LTS technique solves the equation

$$\frac{1}{a}\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{S} \tag{2}$$

where a is an acceleration factor that depends on $\mathbf{W}$.
If the solution is steady, then the time derivative is exactly zero, and so a solution to Equation (2) is also a solution of Equation (1), independent of the value of a.

In the LTS procedure that has been implemented into OpenFOAM, the time derivative is modelled using an Euler scheme. Based on similar implementations in the OpenFOAM solver *LTSInterFoam* and those found in Ref. 17, the time-step Δt for each cell is selected individually as the minimum between the global, user-defined maximum time-step Δt_{max}, and the time-step that is determined by the maximum allowable Courant number c_{max} within the flow. In other words

$$\Delta t = \min\left(\Delta t_{max}, \Delta t(c_{max})\right) \tag{3}$$

In practice, the field Δt calculated using this approach is first smoothed spatially before being applied to the calculation of the LTS solution in each computational cell.

II.B. Adaptive Mesh Refinement

Adaptive mesh refinement (AMR) is a technique that is used to modify an existing mesh so as to improve its resolution of the flow properties in regions of interest, and thus to capture the flow with enhanced accuracy without an excessive increase in computational effort. The advantage of implementing this technique within OpenFOAM is that it allows the details of the flow near shocks to be captured without the need for an excessively fine mesh throughout the entire computational domain. Based on the implementations within the OpenFOAM solvers *interDyMFoam* and *rhoCentralDyMFoam*, the computational mesh is refined or coarsened locally during the computation in order to adapt to the spatial gradients within the flow properties. The approach uses a tree structure, and the maximum coarsening that can be obtained is equivalent to the cell density of the base mesh. When refining the mesh, the algorithm splits each hexahedral cell in two along each coordinate direction to generate eight new cells from the original parent cell. Figure 1 shows the refinement algorithm that is used by the AMR procedure and Figure 2 illustrates an example of a parent-child tree that could be obtained when a cell is split using this procedure.

II.C. Test cases

The following test cases have been used to validate the LTS and AMR algorithms as implemented within *rhoCentralFoam*.

II.C.1. LTS

For the purposes of validating the LTS procedure, a Couette flow and a supersonic flat plate have been simulated.

For the Couette flow, the separation between the upper and lower walls was 1m, and the lateral boundaries of the computation, modelled using cyclic boundary conditions, were placed 40 cm apart. The computational

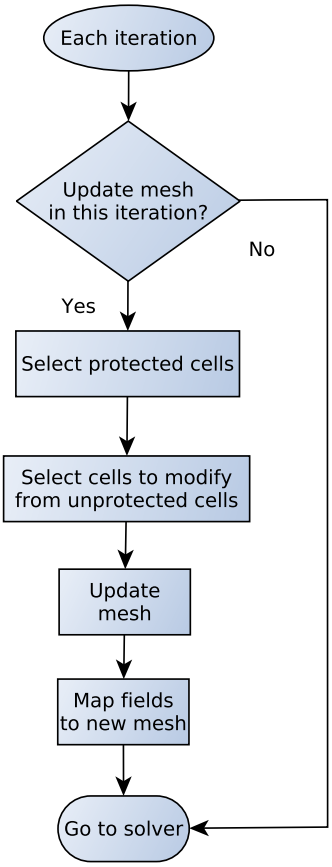


Figure 1. Flow chart for the refinement algorithm

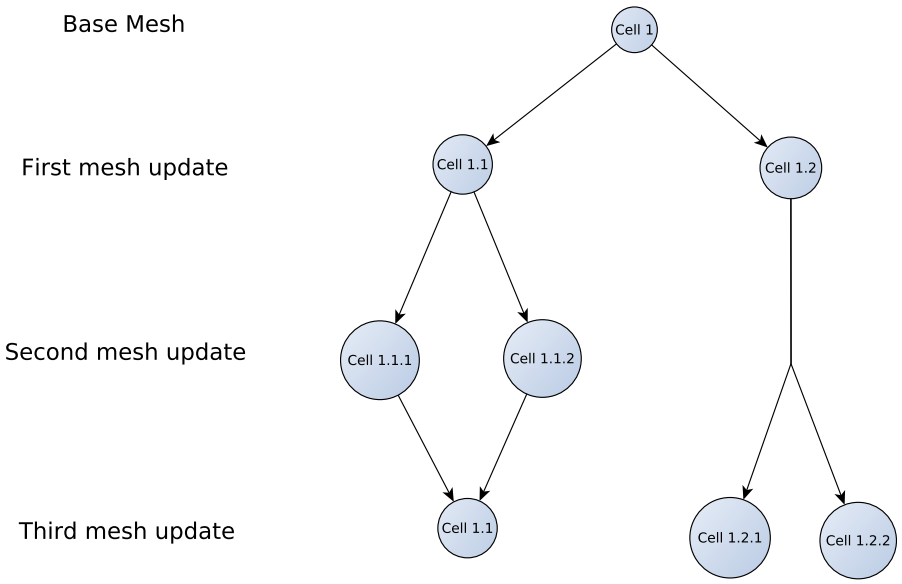


Figure 2. Parent-child tree for cell refinement and coarsening

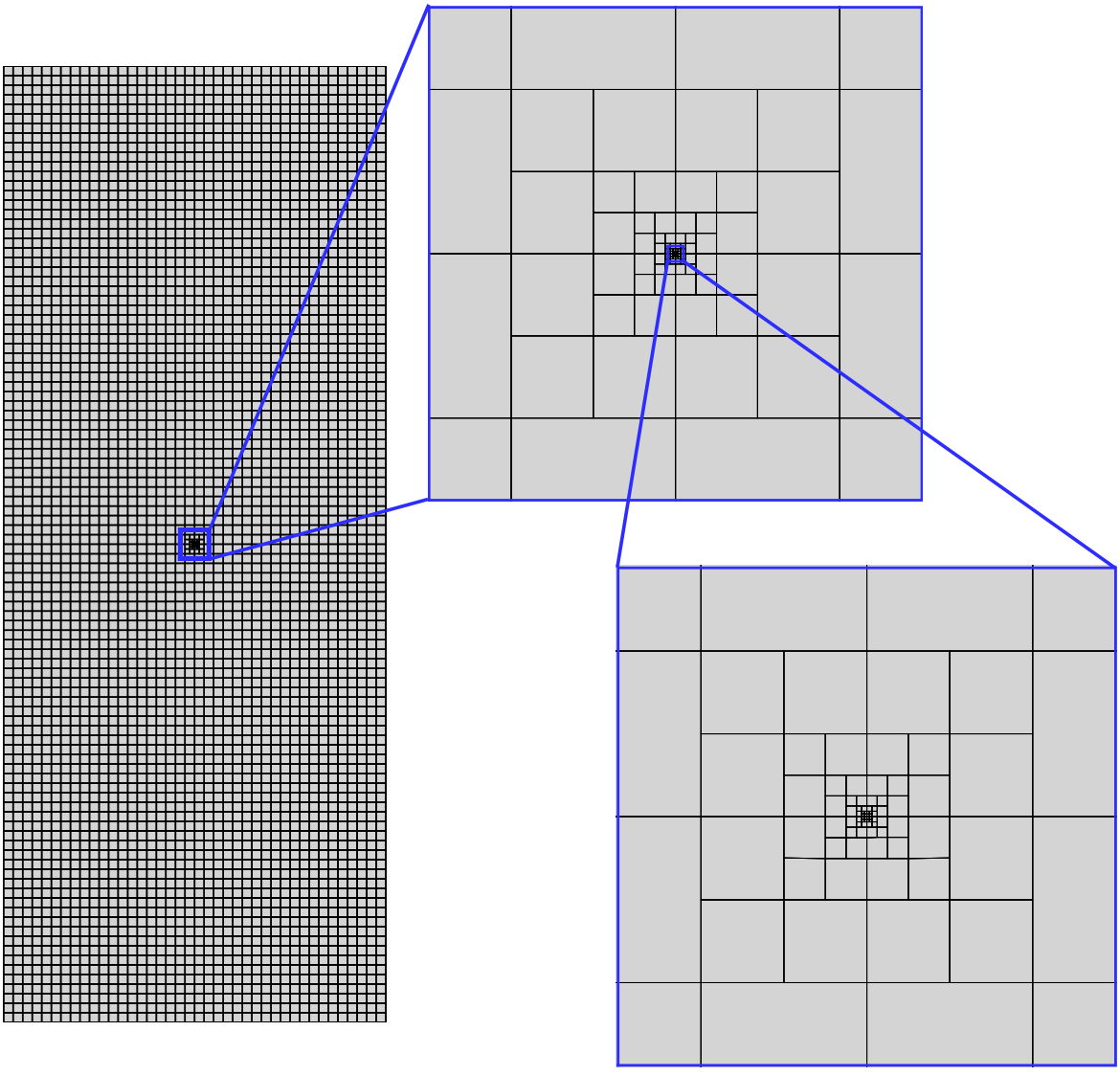


Figure 3. Mesh for the Couette flow test case

domain is illustrated in Figure 3. The base cell size was set to 0.01 m×0.01 m. Four cells were refined at the centre of the channel, then the four cells at the centre of the group of refined cells, and so on until 11 levels of refinement were achieved. The finest cell has a size of ≈4.88 μm×4.88 μm, as can be seen in Figure 3.

The fluid within the channel is air, and the initial and boundary conditions are set to

$$\begin{aligned} \vec{v}(t=0) &= 0m/s & T(t=0) &= 200K & p(t=0) &= 100Pa \\ U_{TW} &= 200m/s & T_{TW} &= 400K & & \\ U_{BW} &= 0m/s & T_{BW} &= 200K & & \end{aligned}$$

where $\vec{v}$ represents the velocity vector, T the temperature, p the pressure, U the flow speed in the x-direction, and the subscripts TW and BW refer to the top and bottom walls of the channel respectively.

The maximum allowable courant number in the flow was set to 0.2 and the maximum allowable time-step to 1×10^{-3} s.

For the supersonic flat plate,[18] the length of the plate within the domain was 0.1 m, with the domain extending 5×10^{-3} m behind the plate. Figure 4 shows the 2-Dimensional computational domain, with the flate plate in red, and consecutive zoom to be able to appreciate the mesh in detail. The smallest cells, at the leading edge of the plate, have a size of 1 μm×1 μm, whereas the coarsest cell, at the top right corner, has a size of ≈2.8 mm×2.8 mm.

The fluid within the domain is air, and the initial and boundary conditions are set so to match the Reynolds (Re) and Mach (M) numbers to those is Ref. 18, as well as the flat plate/upstream temperature ratio (that is, $Re = 500$, $M_\infty = 2$ and $\frac{T_W}{T_\infty} = 2$). Setting the upstream temperature to 300K, and using 0.01m as the reference length to calculate the Reynolds number, the initial and boundary conditions result in:

$$\begin{aligned} U(t=0) &= 694.55m/s & T(t=0) &= 300K & p(t=0) &= 114.46Pa \\ U_\infty &= 694.55m/s & T_\infty &= 300K & p_\infty &= 114.46Pa \end{aligned}$$

where the subscript ∞ refers to upstream conditions. The flat plate temperature is set to 600K and no-slip wall boundaries are assumed. The maximum allowable courant number in the flow was set to 0.1 and the maximum allowable time-step to 1×10^{-5} s.

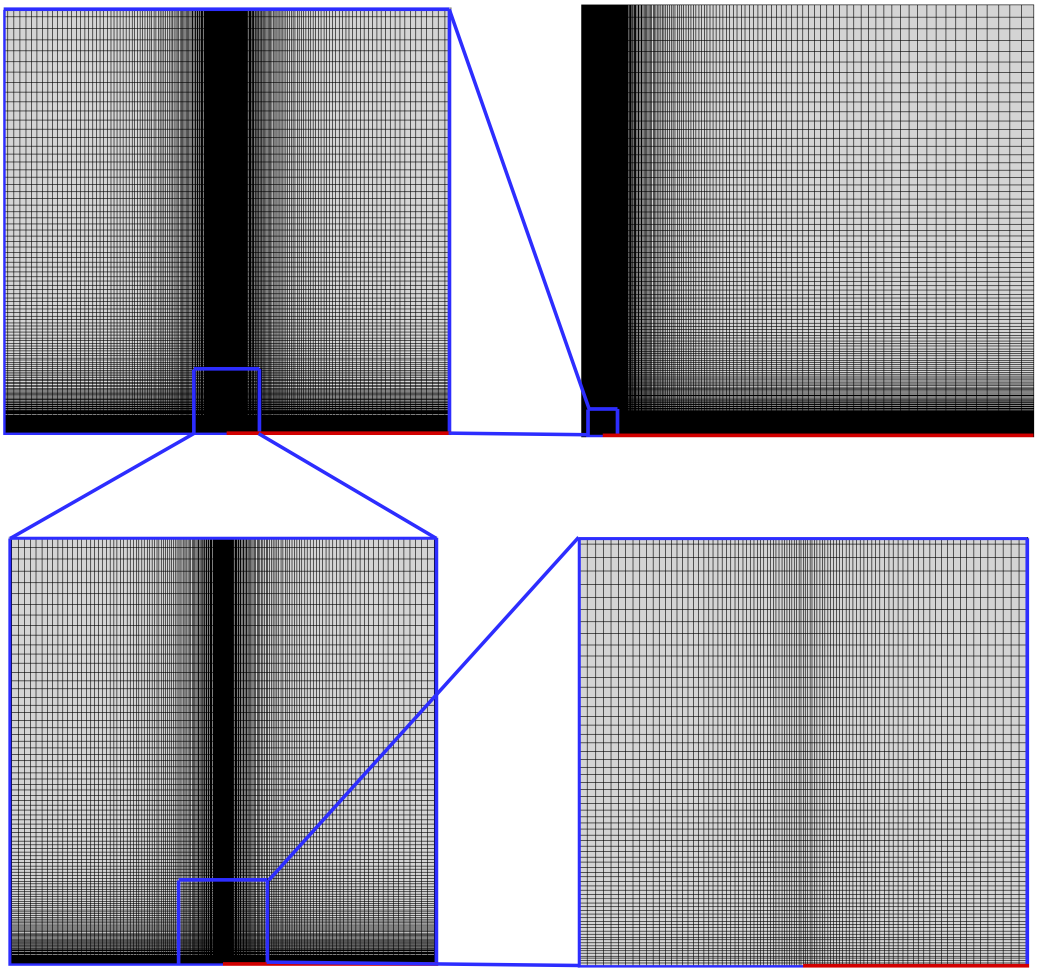


Figure 4. Mesh for the flat plate test case

II.C.2. AMR

A first verification of the implementation of the AMR technique within *rhoCentralFoam* was conducted by modelling the inviscid flow at Mach 5 over a two-dimensional 30° wedge. The initial mesh can be seen in Figure 5.

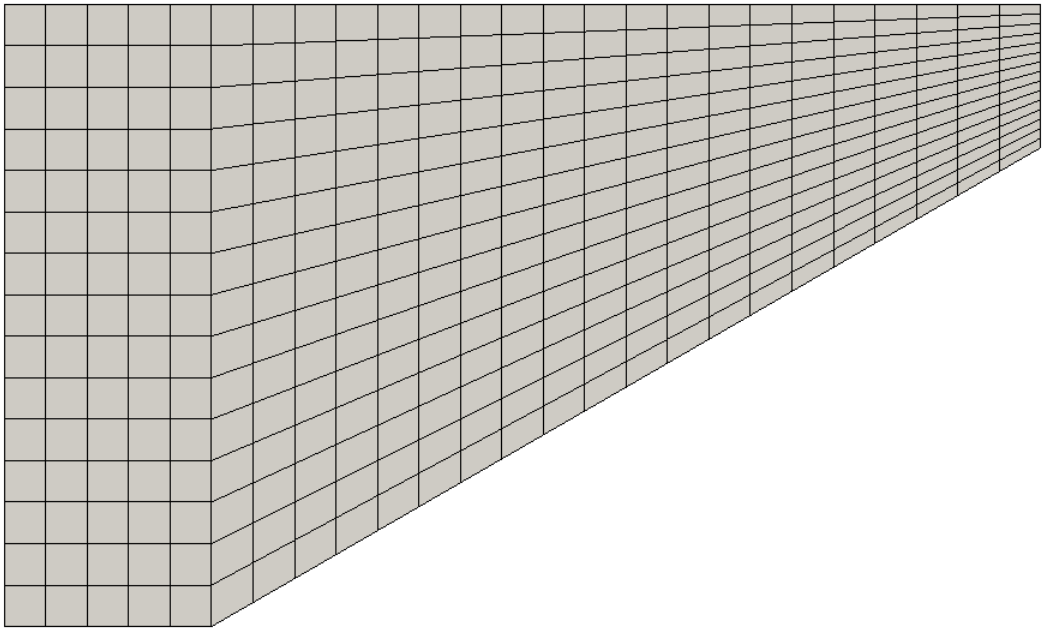


Figure 5. Initial mesh for the supersonic wedge test case

The fluid used for this simulation has been set up so to resolve the non-dimensional problem. As such, the initial and boundary conditions are

$$\begin{aligned} \vec{v}(t=0) &= 0 & M_\infty &= 5 \\ T(t=0) &= 1 & T_\infty &= 1 \\ p(t=0) &= 1 & p_\infty &= 1 \end{aligned}$$

No-slip, adiabatic wall boundaries are assumed.

Mesh refinement was instigated when $\|\nabla M\| > 1$, and coarsening when $\|\nabla M\| < 1$. A million time-steps were executed between each mesh update.

In addition, the Sod shock tube test case[19] was used to study the suitability of the AMR technique for transient simulations. The initial mesh can be seen in Figure 6.

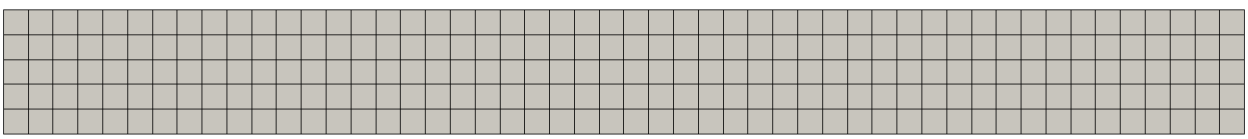


Figure 6. Initial mesh for the shock tube test case

The fluid used for this simulation has been set up so to resolve the non-dimensional problem. As such,

the initial conditions are

$$u(x/L<0,t=0)=u_l=0 \qquad u(x/L>0,t=0)=u_r=0$$
$$p(x/L<0,t=0)=p_l=1 \qquad p(x/L>0,t=0)=p_r=0.1$$
$$\rho(x/L<0,t=0)=\rho_l=1 \qquad \rho(x/L>0,t=0)=\rho_r=0.125$$

where L represents the length of the shock tube, ρ the density of the fluid, and the subscripts l and r refer to the left and right half of the shock tube respectively.

Two different refinement/coarsening criteria were compared. For the first criterion, mesh refinement was instigated when $\left\|\frac{\nabla p}{p}\right\| > 1$, and coarsening when $\left\|\frac{\nabla p}{p}\right\| < 0.5$. For the second, the mesh was refined when $\left\|\frac{\nabla \rho}{\rho}\right\| > 1$ and coarsened when $\left\|\frac{\nabla \rho}{\rho}\right\| < 0.5$. For both cases, five time-steps were executed between each mesh update.

III. Results and discussion

III.A. LTS

III.A.1. Couette flow

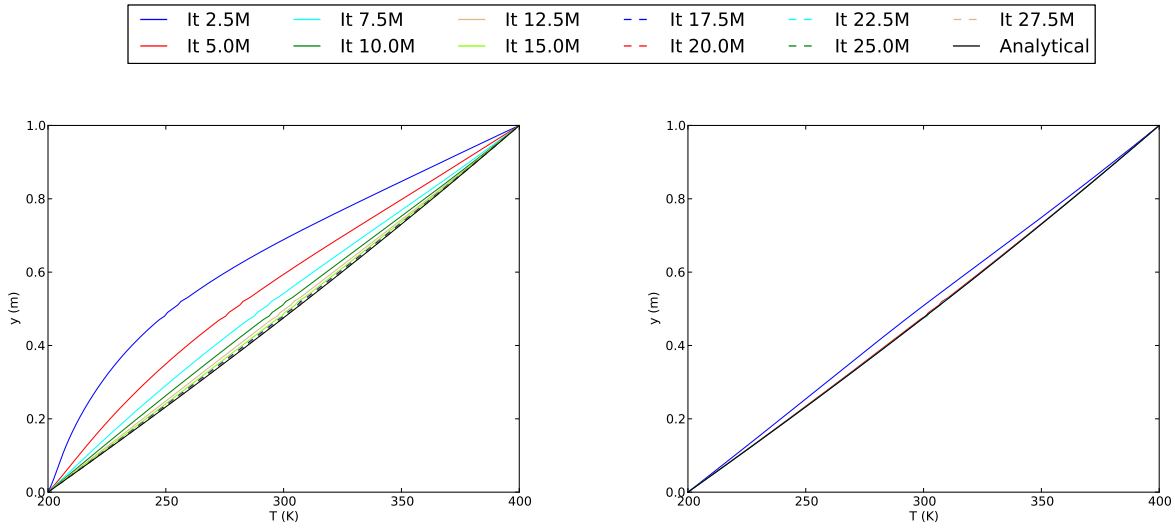
The figures presented below show the flow properties along the vertical centreline through the Couette channel and are compared with the analytical solution for the flow, namely:

$$T(y) = T_{BW} + (T_{TW} - T_{BW})\frac{y}{H} + \frac{\mu U_{TW}^2}{2k}\frac{y}{H}\left(1-\frac{y}{H}\right) \tag{4}$$
$$U(y) = U_{TW}\frac{y}{H} \tag{5}$$
$$V(y) = 0m/s \tag{6}$$
$$p(y) = constant \tag{7}$$

where H is the height of the channel, y the vertical coordinate, μ the dynamic viscosity of the flow, and k its thermal conductivity.

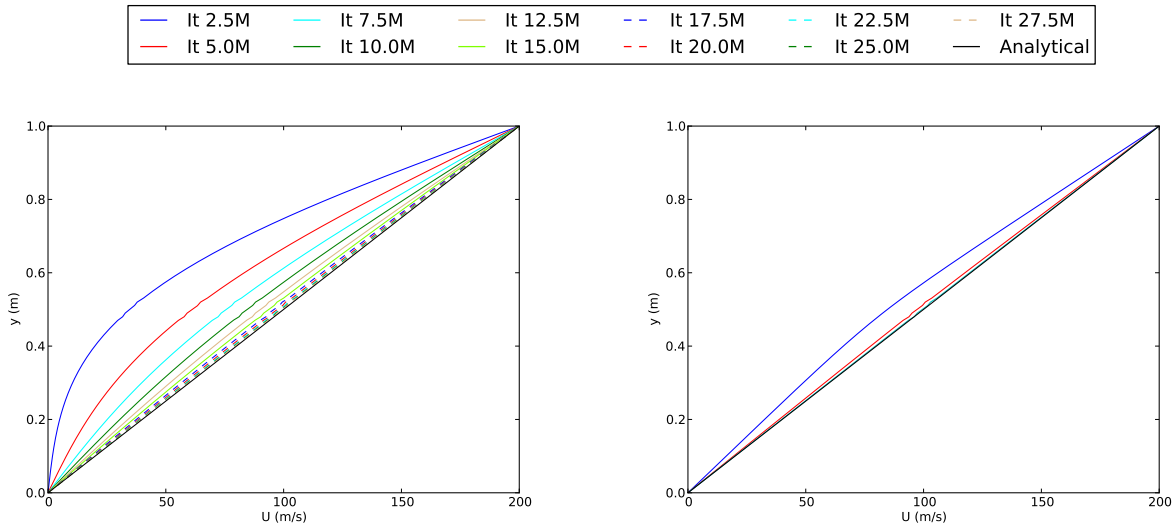
Figures 7 and 8 show the progress of the solution with respect to the number of iterations. Similar results were found for pressure and y-velocity. It is evident that the LTS solver converges to the analytical solution using much fewer iterations than the unmodified solver. Table 1 shows the individual cell time-step. Whereas the time-step for the unmodified solver is the same for all cells, for the LTS solver the minimum time-step is two orders of magnitude smaller than the maximum time-step. This difference allows the bigger cells, which have larger time-steps, to reach a steady state solution faster, accelerating the global convergence of the LTS solution.

Using the LTS solver, after 7.5 million iterations the maximum relative error is approximately 1%. However, for the unmodified *rhoCentralFoam* even after 27.5 million iterations the maximum relative error is still greater than that achieved by the LTS method in 7.5 million iterations. The computational time required to reach these numbers of iterations can be seen in Table 2. The LTS solver employed 2.56 times less computational time than the unmodified solver to reach a relative error of approximately 1%.



(a) Unmodified solver: T (b) LTS solver: T

Figure 7. Results from the unmodified and the LTS solvers for temperature



(a) Unmodified solver: U (b) LTS solver: U

Figure 8. Results from the unmodified and the LTS solvers for x-velocity

	Unmodified	LTS	
		Min	Max
It 2.5M	5.515 ns	20.148 ns	2.150 μs
It 5M	4.922 ns	1.237 ns	0.343 μs
It 7.5M	4.667 ns	1.233 ns	0.341 μs
It 10M	4.531 ns	-	-
It 12.5M	4.522 ns	-	-
It 15M	4.406 ns	-	-
It 17.5M	4.378 ns	-	-
It 20M	4.361 ns	-	-
It 22.5M	4.351 ns	-	-
It 25M	4.344 ns	-	-
It 27.5M	4.340 ns	-	-

Table 1. Simulation time-step in each iteration level, for the unmodified and LTS solvers

	Unmodified	LTS
It 2.5M	7h 26min 50s	9h 21min 17s
It 5M	14h 55min 57s	18h 52min 40s
It 7.5M	22h 37min 11s	28h 0min 44s
It 10M	29h 23min 21s	-
It 12.5M	35h 5min 21s	-
It 15M	41h 13min 13s	-
It 17.5M	47h 12min 38s	-
It 20M	52h 58min 3s	-
It 22.5M	59h 21min 11s	-
It 25M	65h 30min 11s	-
It 27.5M	71h 41min 41s	-

Table 2. Computational time required to reach each iteration level, for the unmodified and LTS solvers

III.A.2. Supersonic flat plate

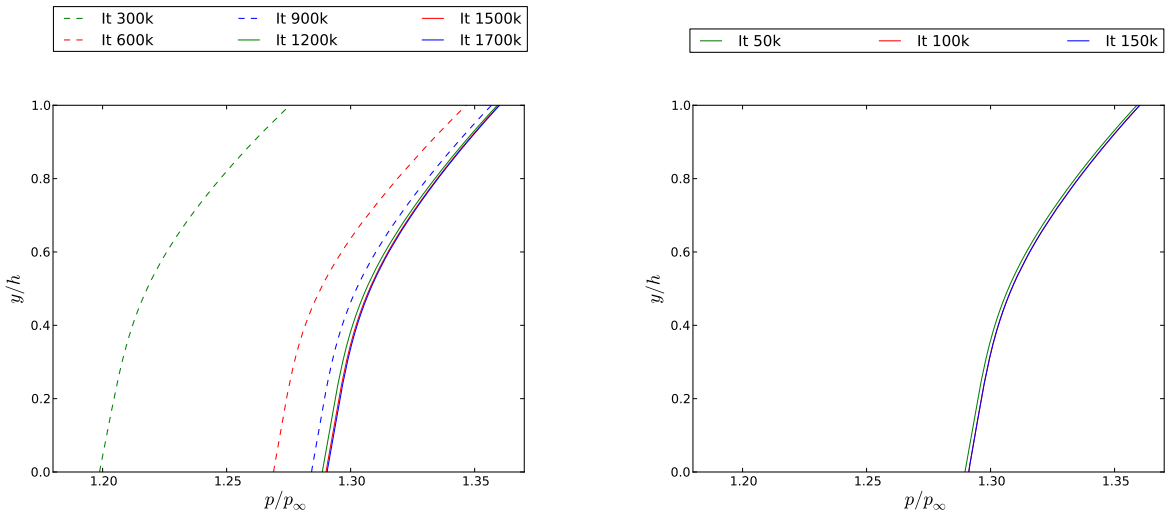
The following data show the flow properties along the vertical line located one reference length (0.01m) behind the leading edge of the flat plate.

The simulation has been considered converged when the difference between two solutions 5×10^4 iterations apart was smaller than 0.01% of the upstream values. The unmodified solver reached 1.7×10^6 iterations before this condition was fulfilled, whereas for the LTS solver only 1.5×10^5 iterations were required. The computational times required to reach this convergence criterion were 86 h 48 min 56 s for the unmodified solver and 9 h 41 min 21 s for LTS solver, thus achieving convergence 8.96 faster than the unmodified solver.

Figures 9 to 11 show the progress of the solution with the number of iterations. Similar results were found for temperature. In Figure 11 the x-velocity is also compared with the velocity profile from Ref. 18.

Iteration (×1000)	Unmodified	LTS
50	2h 45min 46s	3h 19min 48s
100	5h 25min 5s	6h 30min 32s
150	8h, 2min 17s	9h 41min 21s
300	15h 49min 2s	-
600	31h 12min 9s	-
900	46h 23min 12s	-
1200	61h 31min 8s	-
1500	76h 42min 21s	-
1700	86h 48min 56s	-

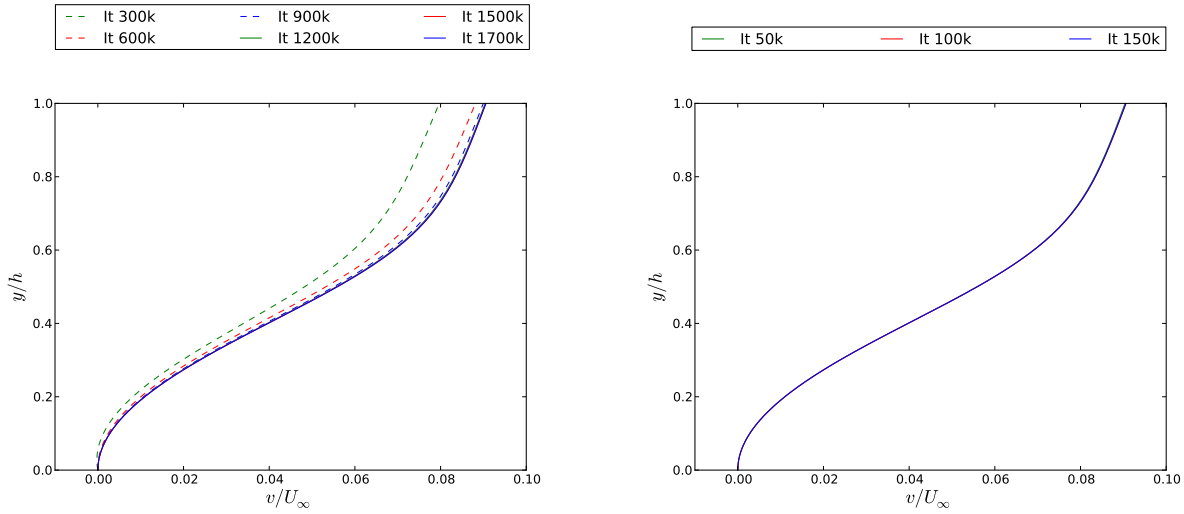
Table 3. Computational time required to reach each iteration level, for the unmodified and LTS solvers



(a) Unmodified solver: p

(b) LTS solver: p

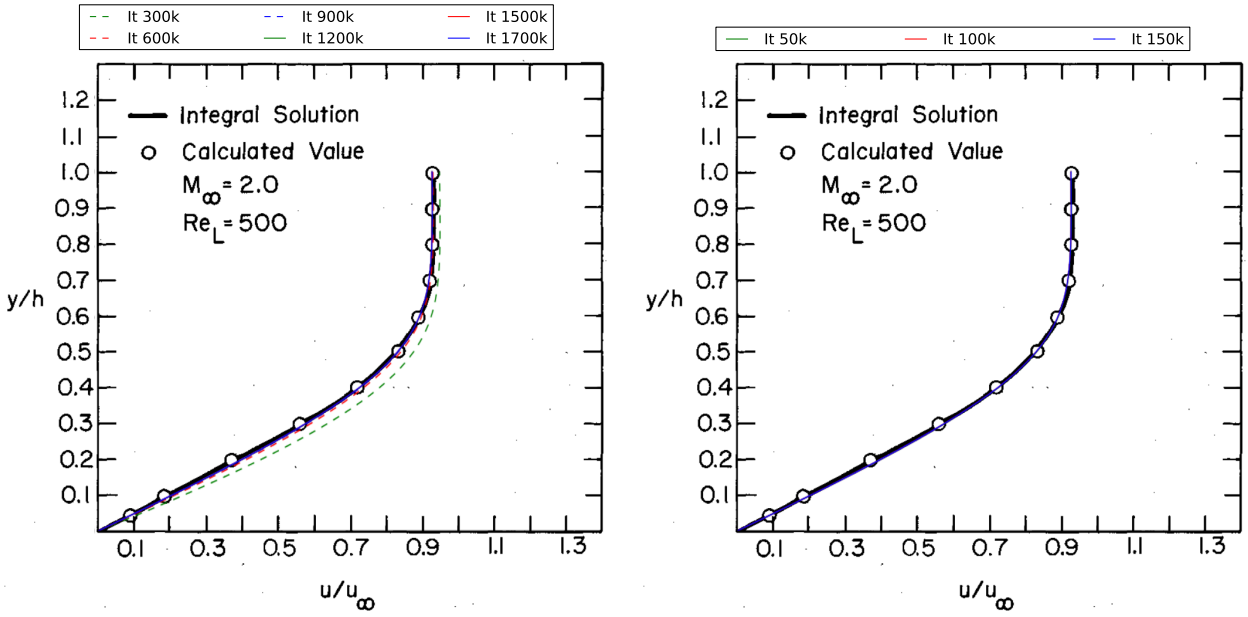
Figure 9. Results from the unmodified and the LTS solvers for pressure



(a) Unmodified solver: v

(b) LTS solver: v

Figure 10. Results from the unmodified and the LTS solvers for y-velocity



(a) Unmodified solver: u

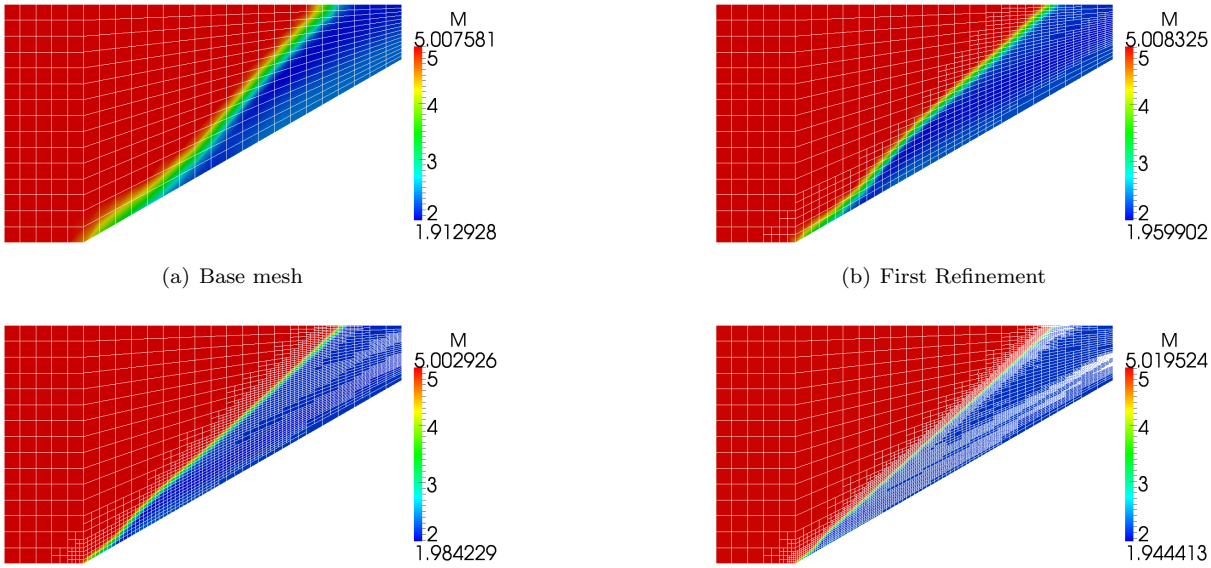
(b) LTS solver: u

Figure 11. Numerical laminar boundary layer profiles (x-velocity) from Ref. 18 and present work, for the unmodified and LTS solvers

III.B. AMR

III.B.1. Supersonic wedge

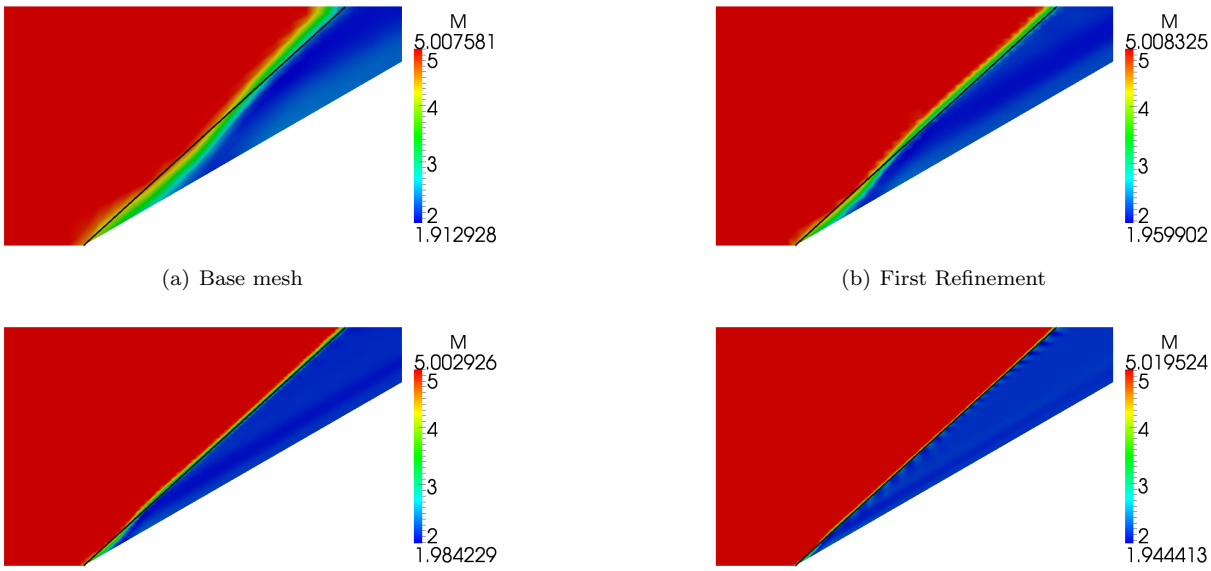
Figure 12 shows Mach number contour plots for the initial and refined meshes, as well as the meshes themselves. The shock wave is resolved with higher accuracy as the cell resolution is increased. This increase in accuracy can be also observed in Figure 13, where the analytical shock wave (black line) is superimposed on the previous Mach contour plots.
In Figure 14, the results for pressure, temperature, and Mach number on the top boundary are compared with the analytical solution, showing again the importance of cell refinement in the proximities of the shock wave.



(a) Base mesh (b) First Refinement

(c) Second Refinement (d) Third Refinement

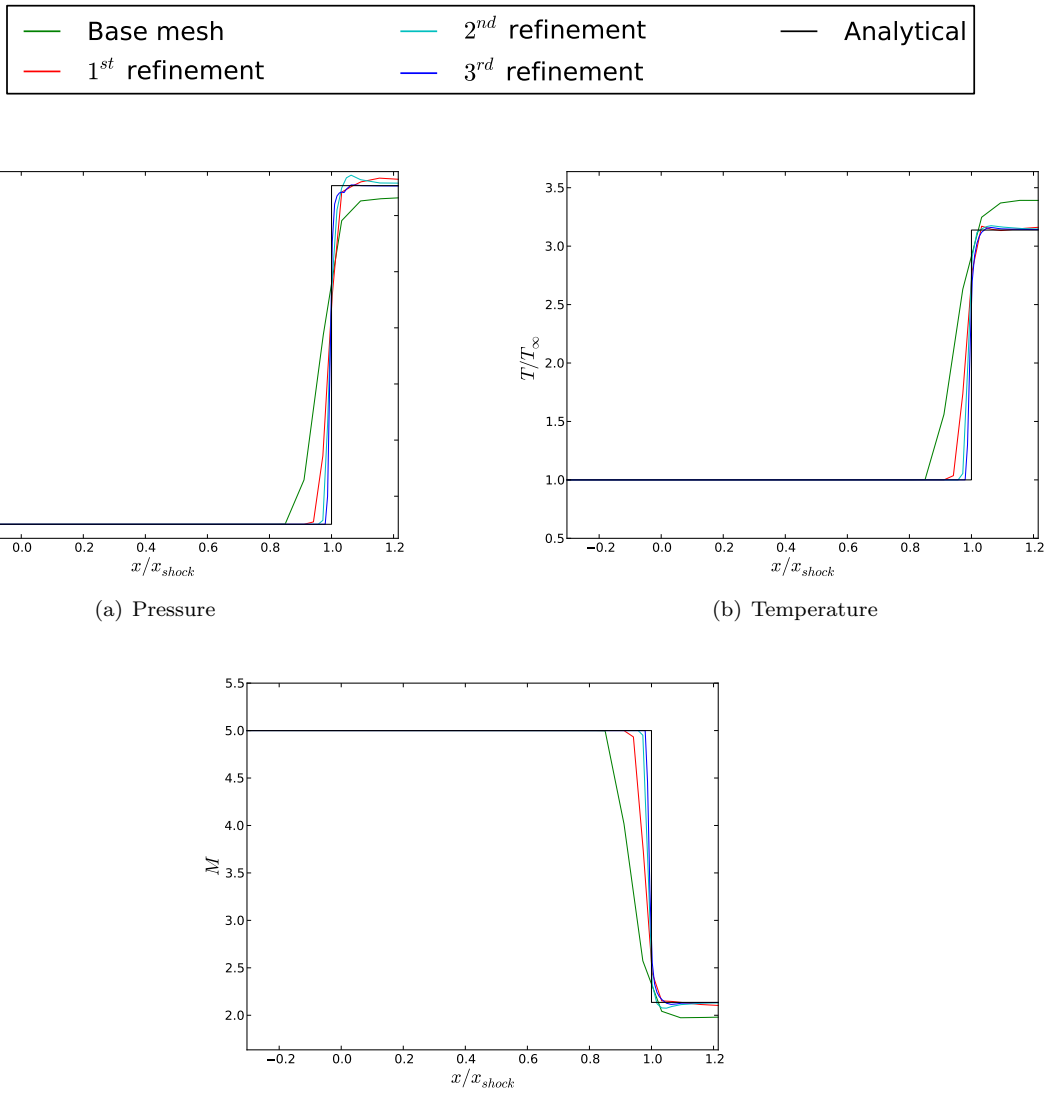
Figure 12. Mach number contour plots for different refinement levels



(a) Base mesh (b) First Refinement

(c) Second Refinement (d) Third Refinement

Figure 13. Comparison between Mach contour plots and analytical position of the shock wave for different refinement levels



(a) Pressure

(b) Temperature

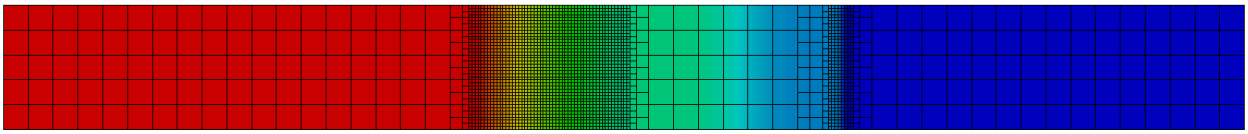
(c) Mach number

Figure 14. Solution comparison for the initial and refined meshes

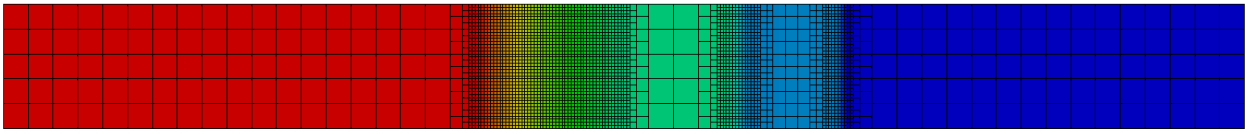
III.B.2. Sod shock tube

Figure 15 shows density contour plots for both refinement criteria, as well as the meshes themselves, for t=0.1s and t=0.2s. The two refinement criteria were $\frac{\nabla p}{p}$ and $\frac{\nabla \rho}{\rho}$. Both the shock wave and the rarefaction wave are resolved with high accuracy, as the cell resolution is increased. However, it is evident that the $\frac{\nabla p}{p}$ criterion does not provide enough resolution to accurately capture the contact discontinuity. These features can also be observed in Figure 16, where the analytical solution for the shock tube is compared with the computational results.

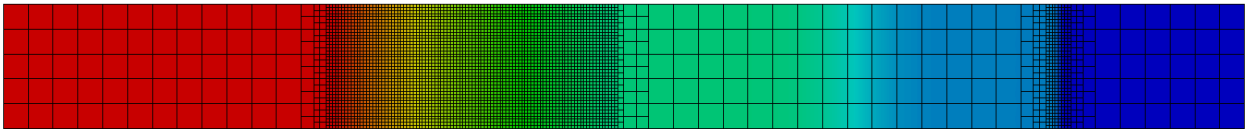
The explanation for this lack of refinement near the contact discontinuity is that whereas density changes through this type of discontinuity, pressure does not. This makes the contact discontinuity invisible to the $\frac{\nabla p}{p}$ criterion, and reflects the importance of selecting an appropriate refinement criterion.



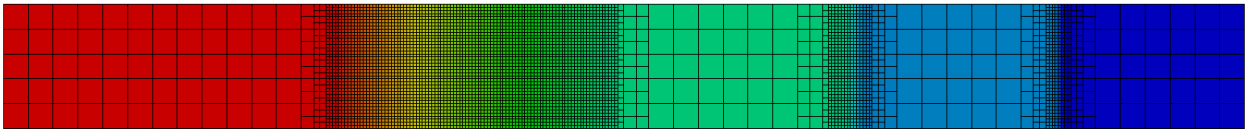
(a) $\nabla p/p$ criterion, t=0.1s



(b) $\nabla \rho/\rho$ criterion, t=0.1s

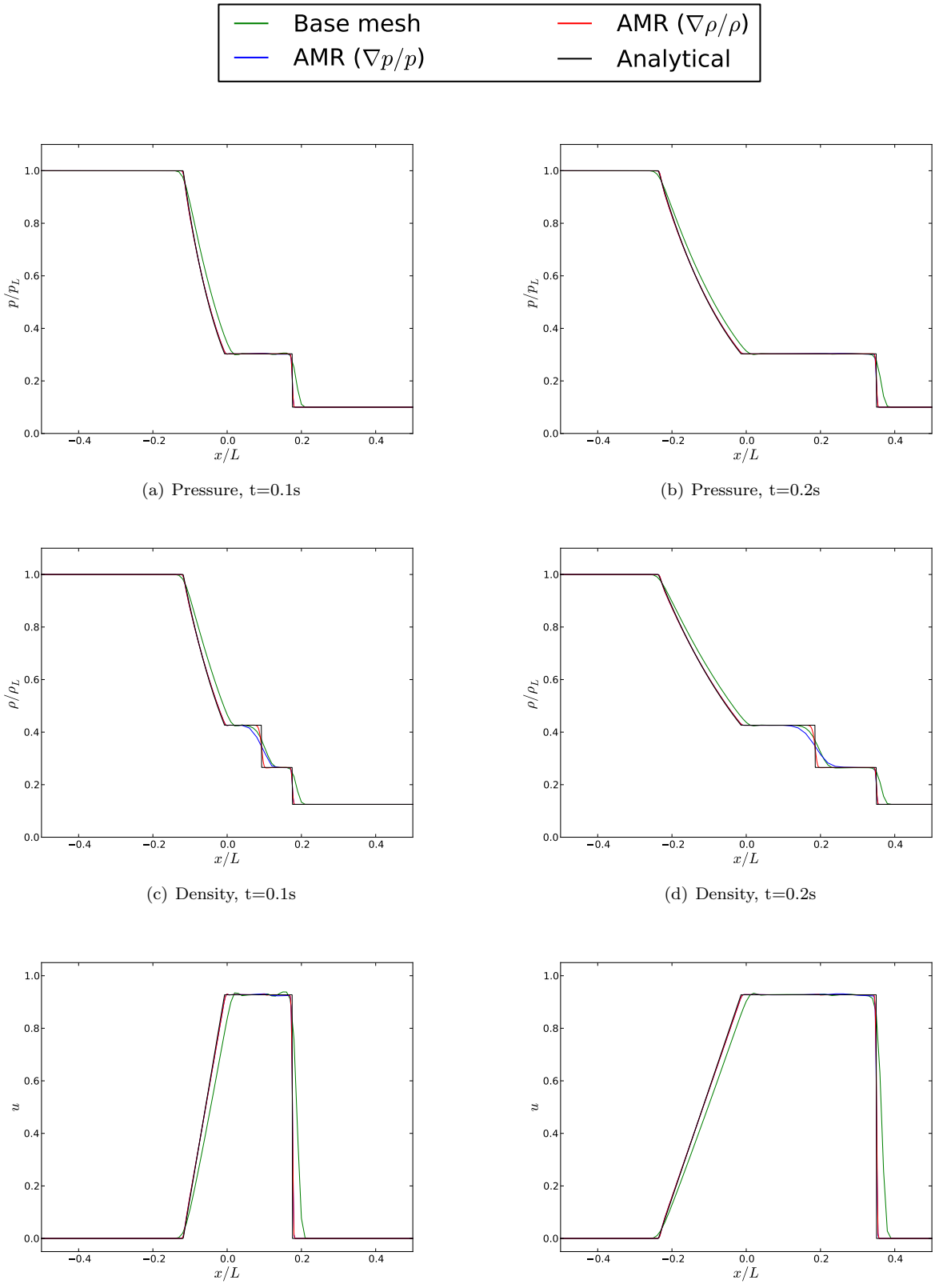


(c) $\nabla p/p$ criterion, t=0.2s



(d) $\nabla \rho/\rho$ criterion, t=0.2s

Figure 15. ρ contour plots for the different refinement criteria, for t=0.1s and t=0.2s



(a) Pressure, t=0.1s

(b) Pressure, t=0.2s

(c) Density, t=0.1s

(d) Density, t=0.2s

(e) Velocity, t=0.1s

(f) Velocity, t=0.2s

Figure 16. Solution comparison for initial and refined meshes, for t=0.1s and t=0.2s

IV. Conclusions

The local time stepping (LTS) and adaptive mesh refinement (AMR) techniques have been implemented into one of the OpenFOAM compressible flow solvers, *rhoCentralFoam*. The implementation of the LTS technique has resulted in a significant reduction of the computational time for the simulation of the Couette flow, matching the analytical solution to within 1.03% error 2.56 times faster than the unmodified solver. A more significant improvement was obtained for the flat plate, as the solution converged 8.96 times faster using local time stepping. The AMR technique has also been successfully implemented in *rhoCentralFoam*, producing an improved solution using significantly fewer computational cells compared with meshing procedures using global refinement. Future studies will apply the AMR and LTS techniques to additional high-speed compressible flow situations in order to fully categorise the extent of the computational time saving when applying these features.

References

[1]Maciel, E. and Pimenta, A., "Reentry Flows in Chemical Non-Equilibrium in Three-Dimensions," *WSEAS Transactions on Mathematics*, Vol. 11, No. 3, March 2012, pp. 262–282.

[2]Georgiadis, N., Yoder, D., Vyas, M., and Engblom, W., "Status of Turbulence Modeling for Hypersonic Propulsion Flowpaths," *Theoretical and Computational Fluid Dynamics*, Vol. 28, No. 3, June 2014, pp. 295–318.

[3]Bonfiglioli, A. and Paciorri, R., "Hypersonic Flow Computations on Unstructured Grids: Shock-Capturing vs. Shock-Fitting Approach," *40th Fluid Dynamics Conference and Exhibit*, June 2010.

[4]Fico, V., Emerson, D., and Reese, J., "A Parallel Compact-TVD Method for Compressible Fluid Dynamics Employing Shared and Distributed-Memory Paradigms," *Computers & Fluids*, Vol. 45, 2011, pp. 172–176.

[5]Wang, C., Wu, S. P., and Cao, N., "Application of Improved TVD Scheme in Hypersonic Heat-Flux Simulation," *Advanced Materials Research*, Vol. 588, 2012, pp. 1822–1826.

[6]Gnoffo, P. and White, J., "Computational Aerothermodynamic Simulation Issues on Unstructured Grids," *37th AIAA Thermophysics Conference*, June 2004.

[7]Gnoffo, P. A., "Simulation of Stagnation Region Heating in Hypersonic Flow on TetrahedralGrids," *18th AIAA CFD Conference*, June 2007.

[8]Saunders, D. A., Yoon, S., and Michael, J. W., "An Approach to Shock Envelope Grid Tailoring and Its Effect on Reentry Vehicle Solutions," *45th AIAA Aerospace Sciences Meeting and Exhibit*, Jan. 2007.

[9]Sha, P., Dinghua, P., Guohao, D., Zhengyu, T., Yueming, Y., and L., H., "Grid Dependency and Convergence of Hypersonic Aerothermal Simulation," *Acta Aeronautica et Astronautica Sinica*, Vol. 31, No. 3, March 2010.

[10]Gnoffo, P., "An Upwind-Biased, Point-Implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows," Tech. rep., NASA Technical Report 2953, 1990.

[11]Wright, M., Candler, G., and Bose, D., "Data-Parallel Line Relaxation Method of the Navier-Stokes Equations," *AIAA Journal*, Vol. 36, No. 9, Sept. 1998.

[12]www.openfoam.com, Accessed: 2014-10-23.

[13]Greenshields, C., Weller, H., Gasparini, L., and Reese, J., "Implementation of Semi-Discrete, Non-Staggered Central Schemes in a Colocated, Polyhedral, Finite Volume Framework, for High-Speed Viscous Flows," *Journal for Numerical Methods in Fluids*, Vol. 63, 2010, pp. 1–21.

[14]Gutiérrez, L. F., Tamagno, J. P., and Elaskar, S. A., "High Speed Flow Simulation Using OpenFOAM," *Mecánica Computacional*, Vol. XXXI, 2012, pp. 2939–2959.

[15]Kurganov, A. and Tadmor, E., "New High-Resolution Central Schemes for Nonlinear Conservation Laws and Convection-Diffusion Equations," *Journal of Computational Physics*, Vol. 160, No. 1, 2000, pp. 241–282.

[16]Kurganov, A., Noelle, S., and Petrova, G., "Semi-Discrete Central-Upwind Schemes for Hyperbolic Conservation Laws and Hamilton-Jacobi Equations," *SIAM J. Sci. Comput.*, Vol. 23, 2000, pp. 707–740.

[17]www.cfd-online.com, Accessed: 2014-07-15.

[18]Berman, H. A., Anderson, Jr., J. D., and Drummond, J. P., "Supersonic Flow over a Rearward Facing Step with Transverse Nonreacting Hydrogen Injection," *AIAA Journal*, Vol. 21, No. 12, 1983, pp. 1707–1713.

[19]Sod, G., "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 27, 1978, pp. 1–31.