

# Case C1.2: Transonic Ringleb flow

Andrea Ferrero\* and Francesco Larocca<sup>†</sup>

*Department of Mechanical and Aerospace Engineering  
Politecnico di Torino, Italy*

## 1 Code description

Numerical simulations were performed with an high-order discontinuous Galerkin code written in Fortran 90 which can solve Euler equations. Several approximate Riemann problem solvers are available for the computation of convective fluxes. In particular all simulations were performed with the Osher solver. The numerical solution inside the element is represented with a modal basis obtained by a tensor product of Legendre polynomials. Integrals are approximated by Gauss quadrature formulas. Discontinuities can be stabilized with both limiters or adaptive filters. Curvilinear boundaries can be represented with quadrilateral elements transformed with high-order Serendipity mapping (linear, parabolic, cubic and quartic curvilinear elements are available). Time integration is performed with explicit Runge Kutta algorithms up to 4th order. Parallelization is supported on shared memory machines with OpenMP directives.

## 2 Case summary

### 2.1 Boundary conditions and mesh

Subsonic characteristic inflow and outflow boundary conditions are imposed at the inlet and at the outlet. Lateral boundaries are represented as inviscid solid walls. Tangency is imposed by the solution of a particular Riemann problem in which the wall state is forced to have zero normal velocity.

Simulations were performed on three structured meshes with 18x12, 36x24 and 72x48 quadrilateral elements. The mesh generator used for this problem [2] provides elements with straight edges. Then the edges of the elements on the boundaries (both solid walls and inlet/outlet) are curved by the introduction of additional points. Integrals are computed by the transformation of the physical element into the reference square on which Gauss quadrature formulas are easily applied. In particular we use fourth order Serendipity mapping for the elements

---

\*PhD candidate, email: andrea.ferrero@polito.it

<sup>†</sup>Professor, email: francesco.larocca@polito.it

on the boundary and classical bilinear mapping for the elements inside the domain.

## 2.2 Error computation

The  $L_2$  norm of the entropy error is evaluated according to the following expression

$$L_{2s} = \sqrt{\frac{\sum_e \int_{\Omega_e} (\Delta s)^2 dV}{\int_{\Omega} dV}} = \sqrt{\frac{\sum_e \sum_i \sum_j (\Delta s)^2 |J| w_i w_j}{\sum_e \sum_j \sum_j |J| W_i W_j}} \quad (1)$$

$$\Delta s = \frac{\frac{p}{\rho^\gamma} - \frac{p_\infty}{\rho_\infty^\gamma}}{\frac{p_\infty}{\rho_\infty}} \quad (2)$$

in which  $\Delta s$ ,  $|J|$ ,  $w_i$  and  $w_j$  represent the entropy error, the Jacobian determinant and the Gauss quadrature weights. In particular we use for this computation the same tensor products of Gauss quadrature formulas used for mass matrix evaluation: 2x2 points for  $p=1$ , 3x3 points for  $p=2$ , 4x4 points for  $p=3$ .

## 2.3 Time discretization and time step

The code is still under development and acceleration techniques for steady problems have not been implemented yet. We integrate the solution in time using explicit Runge-Kutta algorithms. For  $p=1$  and  $p=2$  we use TVD-RK2 and TVD-RK3 algorithm. For  $p=3$  we use SSP-RK4 algorithm. The time step is chosen according to the following stability limit ([1]):

$$\Delta t = \frac{\sigma \Delta x}{\lambda(2k+1)} \quad (3)$$

in which  $\lambda$  is the maximum propagation speed for the signals inside the cell,  $\Delta x$  is a representative cell dimension,  $p$  is the order of the polynomial reconstruction and  $\sigma$  is a stability coefficient. Numerical solution is initialized with the exact solution and we stop the computation when the L2-norm of the density residual is dropped by 8 orders of magnitude. In particular we monitor the residual of the zero order density modal coefficient, which represents the average density in the element.

## 2.4 Hardware specification

All computations were performed on a Linux machine with two Intel64 E5504 processors with 8 cores in total. The machine produces a Taubench time of 13.82 s on a single core. The number of employed cores is from 1 to 8, depending on the mesh size and scheme order.

p	Work units
1	3.9
2	5.5
3	10.7

Table 1: Work Units for 100 residual evaluations with 250000 DOFs

### 3 Results

In Fig.1 and 2 we report the L2 entropy error as a function of the length scale and work units for different meshes. There is only one simulation with the  $p=1$  case because we achieved a shockless solution only with a very fine mesh with  $144 \times 96$  elements. Fig. 2 shows clearly the need of acceleration techniques or implicit integration methods in order to make high order schemes advantageous respect to low order schemes in steady state problems.

### References

- [1] V. Wheatley, H. Kumar, P. Huguenot, On the role of Riemann solvers in Discontinuous Galerkin methods for magnetohydrodynamics, *J. Compu. Phys.* 229 (2010) 660-680.
- [2] K. Masatsuka, March 2009. <http://www.cfdbooks.com>

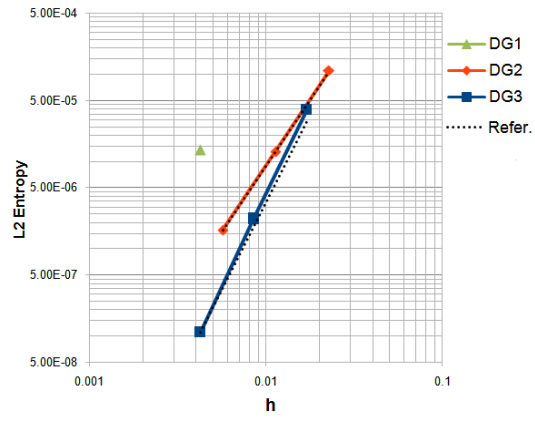


Figure 1: Entropy L2 error vs length scale



Figure 2: Entropy L2 error vs work units