2nd International Workshop on High-Order CFD Methods

May 27-28, 2013, Cologne, Germany

Abstract available at:
http://www.dlr.de/as/desktopdefault.aspx/tabid-8170/13999_read-39370/

Case C1.6: Vortex transport

Andrea Ferrero*and Francesco Larocca†
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 Flow conditions

"*Slow vortex*" : $M_\infty = 0.05 \qquad \beta = \frac{1}{50} \qquad R = 0.005[m]$
Only the slow vortex case is simulated. A vortex transported by a $M_\infty = 0.05$ flow moves in a square periodic domain. The vortex crosses the domain 50 times. The description of this problem makes use of SI units. Kinematic and thermodynamics variables are nondimensionalized inside the code according to the following reference variables:

$$P_{ref} = P_\infty^0 \qquad T_{ref} = T_\infty^0 \qquad U_{ref} = \sqrt{RT_{ref}} \qquad L_{ref} = L_x \tag{1}$$

where the superscript "0" identifies total quantities. After the computation the output is converted in order to evaluate the error in SI units.

*PhD candidate, email: andrea_ferrero@polito.it
†Professor, email: francesco.larocca@polito.it

2.2 Error computation

The L_2 norm of the cartesian components of velocity (u and v) and velocity magnitude (q) are evaluated after 50 time periods. For example the L_2u error for the component u is:

$$L_2u = \sqrt{\frac{\sum_e \int_{\Omega_e} (u - u_{ex})^2 dV}{\int_\Omega dV}} = \sqrt{\frac{\sum_e \sum_i \sum_j (u - u_{ex})^2 |J| w_i w_j}{\sum_e \sum_j \sum_j |J| W_i W_j}} \qquad (2)$$

in which u_{ex}, $|J|$, w_i and w_j represent the exact x-component (obtained by translation of initial condition), 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

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]):

$$\sigma \lambda \frac{\Delta t}{\Delta x} \leq \frac{1}{2p+1} \qquad (3)$$

in which λ is the maximum propagation speed for the signals inside the cell, Δx is a representative cell dimension and p is the order of the polynomial reconstruction. The parameter σ is set equal to 0.7 in all the simulations.

2.4 Meshes

We use two sets of meshes with 24x24, 48x48 and 96x96 elements. The first set is obtained by regular cartesian discretization of the domain. The second set is obtained by the introduction of random perturbations on the regular meshes. The internal nodes are perturbed both in x and y while the nodes on the boundary are perturbed only in the direction tangential to the boundary, keeping the mesh periodic. In Fig.1 an example of regular and perturbed meshes is represented.

2.5 Hardware specification

All computations were performed on a Linux machine with an Intel i7-3930x processor and 16 Gbytes of RAM. The machine produces a Taubench time of 6.50 s on a single core. Simulations were performed with a number of cores between 1 and 6, depending on the mesh size.

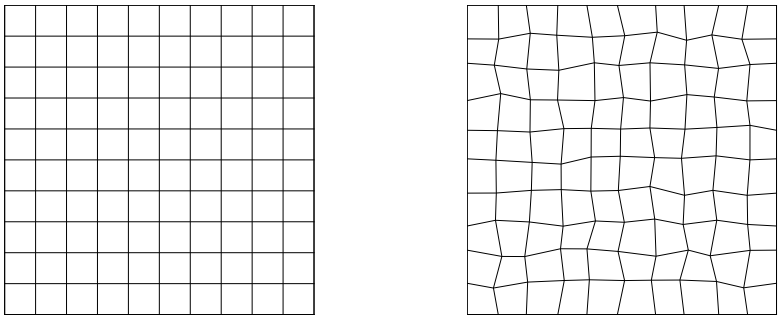

Figure 1: Example of regular and perturbed meshes (10x10 elements).

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.2 and 3 we report the velocity magnitude L2 error expressed in [m/s] as a function of the lenght scale and work units for the regular meshes. It is clear that high order methods give a significative improvement in performances for this test case. In Fig. 4 and 5 we compare the result obtained with regular and perturbed meshes. There is not an evident deterioration in the error for a given mesh size but there is an increase in the computational time due to the reduction of the mesh size in the perturbed case.

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.

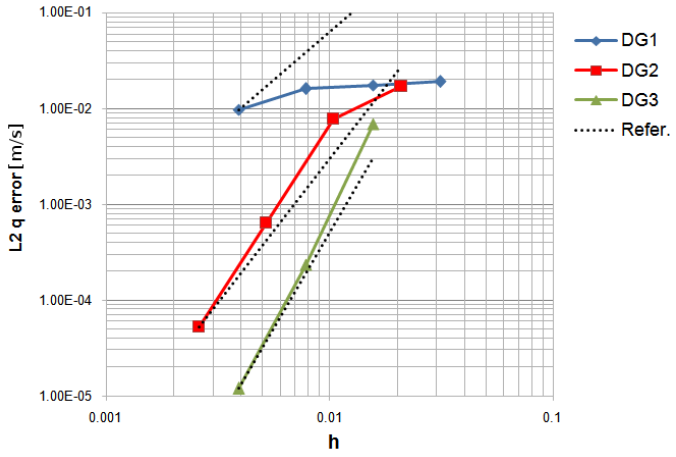

Figure 2: Velocity magnitude L2 error [m/s] vs length scale (Regular meshes)

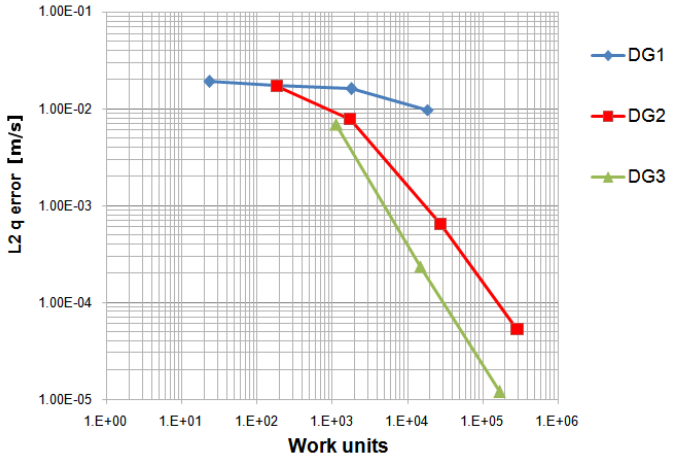

Figure 3: Velocity magnitude L2 error [m/s] vs work units (Regular meshes)

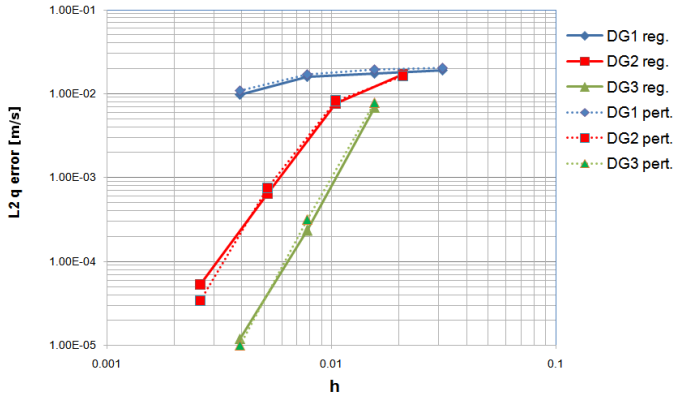

Figure 4: Velocity magnitude L2 error [m/s] vs length scale (Perturbed and regular meshes)

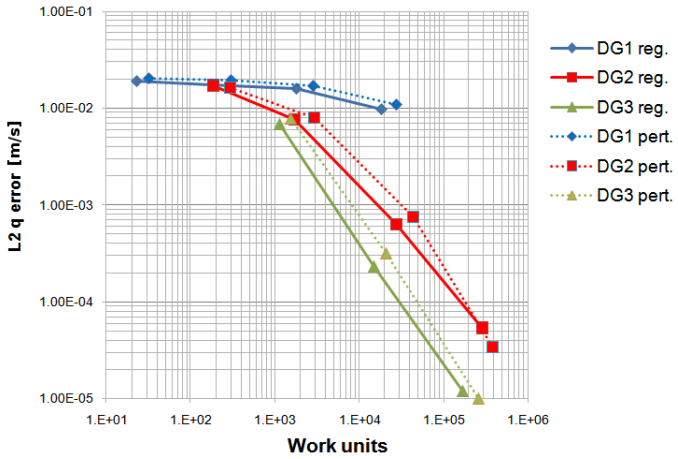

Figure 5: Velocity magnitude L2 error [m/s] vs work units (Perturbed and regular meshes)