Case C1.3: Flat plate boundary layer

Andrea Ferrero*and Francesco Larocca†
Department of Mechanical and Aerospace Engineering
Politecnico di Torino, Italy

1 Code description

Numerical simulations were performed with a discontinuous Galerkin code written in Fortran 90 which is currently under development. The code can solve Euler equations, Navier-Stokes equations or RANS equations with different turbulence models (Spalart-Allmaras, Wilcox k-omega, k-omega+Laminar Kinetic Energy) in 2D.
Several approximate Riemann problem solvers and numerical fluxes (Osher, Roe, AUSM+, Rotated-RHLL, Lax-Friedrichs) are available for the computation of convective fluxes. In particular in this test case the AUSM+ flux [5] is used.
Diffusive fluxes are computed by means of a recovery based approach [1]. The implemented method is inspired to the original recovery approach proposed by Nomura and van Leer [6] but it makes use of a different recovery basis and a different boundary procedure.
The numerical solution inside the element is represented through an orthonormal modal basis obtained by the modified Gram-Schmidt procedure. Both physical space defined and element space defined basis functions can be chosen. In the first case a set of monomials defined in the physical space is used to start the orthonormalization procedure, following the approach of [2]. In the second case, the orthonormalization is initialized with a tensor product of Legendre polynomials defined on the reference element. In this test case the second approach is used.
Curvilinear elements are implemented up to fourth order for quadrilateral and third order for triangles.
Both explicit (RK-TVD and SSP-RK) and implicit (backward Euler) time integration schemes have been implemented. For this steady test case the implicit backward Euler method is used. The jacobian is evaluated numerically. The implementation of an analytically evaluated jacobian is under development.
As far as parallelization is concerned, the explicit version of the code is fully parallelized through OpenMP directives. In the implicit version of the code,

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

the computation of fluxes and the linear solver are parallelized by OpenMP. In particular, the GMRES method with ILU(0) preconditioner from the library PARALUTION [4] is used. The numerical evaluation of the jacobian is performed in serial.
As far as postprocessing is concerned, the code can generate output files in which each mesh element is subdivided in several elements depending on the number of degrees of freedom of the reconstruction. This makes it possible to obtain a visualization which takes into account all the information related to the high order reconstruction.

2 Case summary

As far as boundary conditions are concerned total pressure and total temperature are imposed at the left inflow boundary and static pressure is imposed at the upper and exit boundaries. All simulations are stopped when L_2R/L_2R_0 drops down 10^{-10} and the difference between the *cd* evaluated at two consecutive steps is less than 10^{-11}. Here L_2R and L_2R_0 are the L2-norm of the x-momentum residual at the current iteration and at the first time step. The residual refers to the zero order modal coefficient. The pseudo-transient continuation technique is used and a CFL number evolution strategy is implemented according to [3]. The minimum CFL number is 10^4 and the maximum CFL number is 10^{10}. The GMRES iterative solver (with ILU0 preconditioner) is stopped when the relative error reaches 10^{-2} or when the number of iterations exceeds 250.

A Linux machine with an Intel i7-3930x processor and 32 Gigabytes of RAM is used. The machine produces a Taubench time of 6.5 seconds. All simulations are performed in serial. In Table 1 the work units required to perform 100 residual evaluations with 250000 DOFs are reported. The data refer to a Navier-Stokes discretization. The results for the implicit integration are dominated by the cost of the numerical evaluation of the jacobian.

p	Explicit	Implicit
1	8.6	124.4
2	13.6	575.6
3	25.1	1815.5

Table 1: Work Units for 100 residual evaluations in a viscous problem with 250000 DOFs

3 Meshes

The provided quadrilateral meshes (`flatplate_quad_refx.msh`) were used for the simulations.

4 Results

In Figure 1 and 2 the drag coefficient error is reported as a function of the equivalent length scale and the work units. The reference drag coefficient value used for the error computation is $Cd = 0.0013111835 \pm 10^{-10}$.

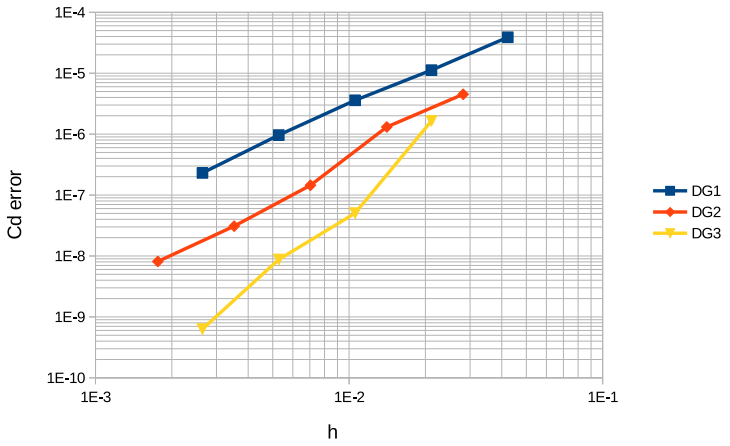

Figure 1: Cd error vs length scale

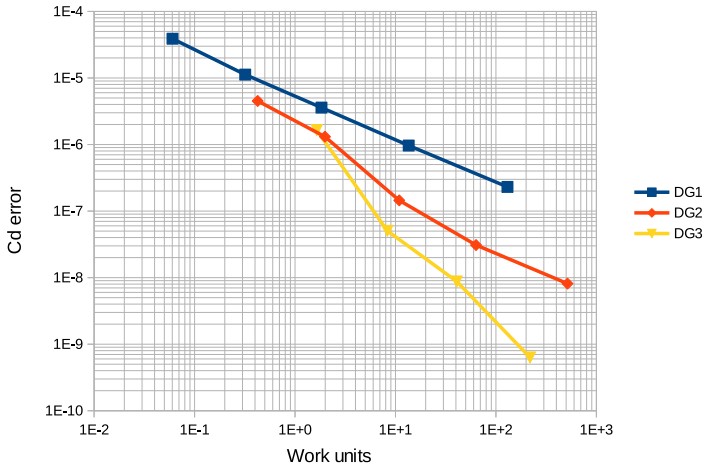

Figure 2: Cd error vs work units

References

[1] Ferrero A., Larocca F., Puppo G. A robust and adaptive recovery-based discontinuous Galerkin method for the numerical solution of convection-diffusion equations, *Int. J. Numer. Meth. Fluids*, 77:63-91, 2015.

[2] Bassi F, Botti L, Colombo A, Di Pietro DA, Tesini P. (2012) 'On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations', Journal of Computational Physics; 231 : 45-65.

[3] Colombo, A. (2011) 'An agglomeration-based discontinuous Galerkin method for compressible flows', Universit degli studi di Bergamo, PhD Thesis.

[4] Dimitar Lukarski, Nico Trost, 'PARALUTION Project', http://www.paralution.com/

[5] Liou, M.S. (1996) 'A sequel to AUSM: AUSM+', Journal of Computational Physics, Vol. 129, No. 2, pp.364382.

[6] van Leer B, Nomura S. (2005) 'Discontinuous Galerkin for Diffusion', AIAA paper 2005-5108.