C3.5: Direct numerical simulation of the Taylor-Green vortex at $Re = 1600$

Cenaero

1 Code description

Discretization. Argo is based on the discontinuous Galerkin / interior penalty method. It has been implemented for hybrid two and threedimensional curved grids, featuring triangles, quadrangles, tetrahedra, prisms and hexahedra. The maximum order of interpolation, both in terms of geometry and solution, is 4.

Iterative methods and time integration. On meshes featuring large differences in stable time steps (eg. due to boundary layers), we use second order backward differencing time-integration, with Jacobi preconditioned Newton-GMRES iterations as the implicit solver. For isotropic meshes, explicit Runge-Kutta iterations are used.

Parallellization. Argo features a hybrid shared / distributed memory parallellisation. The shared memory parallellisation uses an internal domain partitioning per element type and operation. Distributed memory parallellisation is based on ghost elements. Weak and strong scaling tests have been done up to 512 cores on an intel based cluster, and up to 32.000 cores on a BlueGene/P machine.

Postprocessing. All types of monitors (forces, fluxes, volume integrals) are embedded in the code. For these integration rules of order 2p+1 are used (where p is the order of interpolation), for any quantity. The high-order polynomial output of solution, state functions and derivative quantities, is done in Gmsh 2.0 format; the views are then recursively refined up to 4 times with Gmsh.

2 Case summary

Computations. A grid convergence study has been performed using four grids. All computations use third order interpolants resulting in a fourth order precision. The number of degrees of freedom (i.e. the degrees of freedom at continuity), the grid size, and the number of CPUs used for the computations are summarized in table 2. The time-stepping is done using a fourth order

Mesh	Dof	Grid size	CPU
very coarse	144^3	48^3	128
coarse	192^3	64^3	256
baseline	288^3	96^3	320
fine	384^3	128^3	320

explicit Runge-Kutta with a CFL condition of unity.

Machine. Each of the nodes of the cluster is composed of 2 quad-core processors (Intel Xeon L5420). The cores are clocked at 2.5GHz, and have 6MB of level 2 cache. The nodes are organised following the *SMP (symmetric multiprocessing)* paradigm, meaning that all of the cores access the shared 16GB RAM memory through a common *(FSB) front side bus*. Hence the performance is much impacted by the memory bandwidth, in function of the number of active jobs, and in particular on their memory access requirements. This cluster will be superceded this month by a more recent architecture, this time featuring *NUMA (non-uniform memory access)* nodes (Intel Nehalem), thereby greatly alleviating memory bandwidth issues.

Performance. The taubench statistics are as a consequence rather dependent on the load of the node. During 3 tests, results ranged from 11.5 to 15.3 seconds (each time averaged over 4 runs). In all of these cases all 7 other cores were executing other jobs. As the queueing system does not permit to choose the nodes in function of their load, nor to exclusively reserve a node, it is difficult to really cover all possibilities, in particular the extreme cases (no jobs vs very memory-intensive jobs on the other cores). For these cases jobs are moreover rather light in memory usage (20% of RAM), as the benchmark itself (10%). One would expect the timings to degrade further, as the memory transfer load on the node increases.

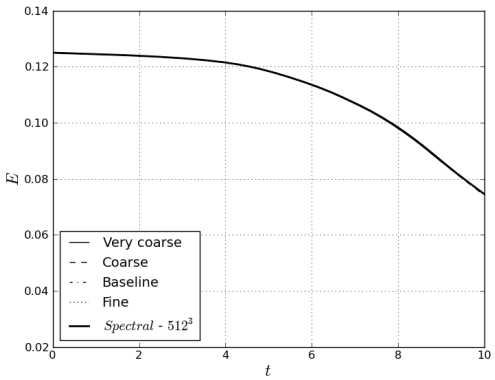
Parallel issues. In practice, it difficult to foresee let alone control the load that will be encountered when running a parallel computation. For the moment, the queuing system does not yet allow for hybrid parallel jobs, so all computations are run in pure MPI mode.

3 Meshes

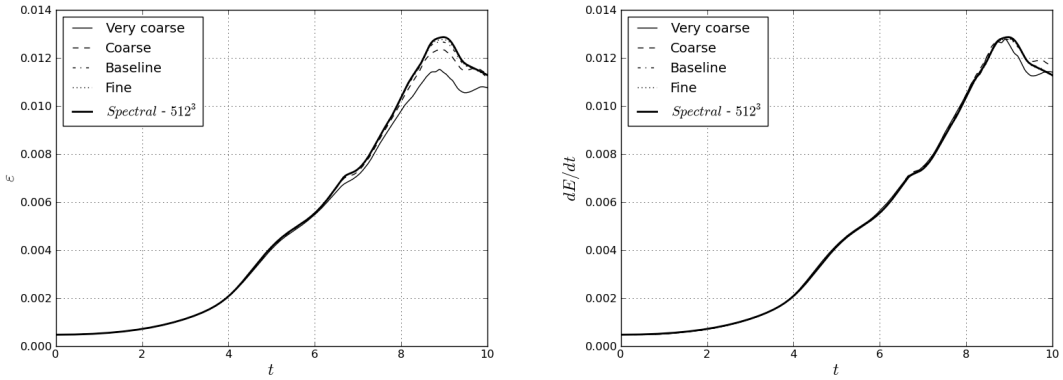
The grids are composed of regular hexahedral elements, as stated in the workshop. The meshes were generated and partitioned using GMSH.

4 Results

The evolution of the global kinetic energy is presented on figure 4. Even the very coarse computation is able to capture this quantity with good precision.



The difference between the computations is more pronounced when comparing global energy dissipation rate. This rate can be calculated either using the integral of the enstrophy integrated over the domain or by computing the temporal derivative of the energy. We can see on figure 4 that the results of those definitions are not equivalent when the computation is not resolved.



Indeed, the numerical scheme leads to jumps in the solution when the computation is under-resolved. Those jumps are not taken into account when calculating the integral of the enstrophy over the domain. The difference between the two values (enstrophy integrated and temporal derivative of the energy) can be used to mesure the accuracy of the computation. Figure 4 shows the error between the two quantities. We can see that for the fine computation, the error is of the same order of magnitude as for the spectral results. Hence we can say that a similar level of resolution has been reached.