11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
July 20–25, 2014, Barcelona, Spain

EDGE-BASED SOLVERS FOR THE COMPRESSIBLE EULER EQUATIONS ON MULTICORES AND GPUS

Matthias Möller[1]

[1] Delft University of Technology, Delft Institute of Applied Mathematics,
Mekelweg 4, 2628 CD Delft, The Netherlands, m.moller@tudelft.nl

Key words: *Compressible Euler Equations, Discretization Methods, Parallel Computing.*

A high-resolution finite element scheme for the efficient solution of the compressible Euler equations on adaptive unstructured meshes is discussed. The underlying continuous Galerkin approximation is stabilized using the *algebraic flux correction* technique [2, 3, 4]. Fletcher's group finite element formulation [1] is adopted which makes it possible to cast the (semi-)discretized problem into an edge-based formulation. In essence, the residual and/or right-hand side vectors are assembled in a loop over pairs of 'neighboring' degrees of freedom for which the associated basis functions have overlapping supports. That is,

$$\text{rhs}_i = \sum_{j\neq i} \mathcal{F}_{ij} + \text{boundary terms}$$

where $\mathcal{F}_{ij} = A_{ij}(\dot{U}_i - \dot{U}_j) + B_{ij}(U_i - U_j) + \mathbf{c}_{ij}\cdot\mathbf{F}(U_i) - \mathbf{c}_{ji}\cdot\mathbf{F}(U_j)$. The local matrices A_{ij} and B_{ij} depend on U_i and U_j and on the constant coefficients $\mathbf{c}_{ij}$ and $\mathbf{c}_{ji}$ which are evaluated and stored during preprocessing. For (semi-)implicit time-stepping schemes, the global system matrix or parts of it are also assembled edge-by-edge from the same (precomputed) data. Thus, the entire assembly process can be implemented without on-the-fly numerical integrating at the cost of storing the vector of coefficients $\mathbf{c}_{ij}$ in main/device memory.

In this paper we discuss different parallelization strategies: a) one single multi-threaded loop over all edges ij with synchronized access to the entries i and j; b) reorganization of edges into groups of independent edges which are processed asynchronously with global synchronization between different groups. The computational efficiency of both approaches is studied numerically for different OpenMP-implementations on CPUs and CUDA-kernels on GPUs. Amongst other influence factors the overall efficiency depends on the (cache-optimized) ordering of edges and – especially for efficient GPU-implementations – on the number of edges that can be processed simultaneously. We address the benefits of using nonconforming finite elements where the degrees of freedom are associated with edges (in 2D) and faces (in 3D). The resulting matrix sparsity pattern is regular even on unstructured meshes which makes it possible to adopt optimized data structures [5].

REFERENCES

[1] C.A.J. Fletcher. The group finite element formulation. *Comput. Methods Appl. Mech. Engrg.* **37** (1983) 225–243.

[2] D. Kuzmin, S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.* **175** (2002) 525–558.

[3] D. Kuzmin, M. Möller. Multidimensional FEM-FCT schemes for arbitrary time-stepping. *Int. J. Numer. Meth. Fluids* **42** (2003) 265–295.

[4] D. Kuzmin, M. Möller, J.N. Shadid, M. Shashkov. Failsafe flux limiting and constrained data projection for equations of gas dynamics. *J. Comput. Phys.* **229** (2010) 8766–8779.

[5] M. Möller. Algebraic flux correction for nonconforming finite element discretizations of scalar transport problems. *Computing* **95** (2013) 425–448.