10th ANKARA INTERNATIONAL AEROSPACE CONFERENCE AIAC-2019-110
18-20 September 2019 - METU, Ankara TURKEY

AN EFFICIENT EDGE BASED DATA STRUCTURE FOR A VERTEX BASED FINITE VOLUME ALGORITHM ON HYBRID UNSTRUCTURED MESHES

Semih Akkurt* and Mehmet Sahin†
Istanbul Technical University
Istanbul, Turkey

ABSTRACT

An efficient edge based data structure has been developed in order to implement a compressible Navier-Stokes solver based on a vertex based finite volume algorithm on unstructured hybrid meshes in two- and three-dimensions. In the present approach, the data structure is tailored to meet the requirements of the algorithm by considering data access patterns and cache efficiency. The required data are packed and allocated in a way that they are close to each other in the physical memory. Therefore, the proposed data structure increases cache efficiency and improves computation time. As a result, the explicit solver indicates a significant speed up compared to other open-source solvers in terms of CPU time. A fully implicit version of the numerical algorithm has also been implemented based on PETSc library in order to improve the robustness of the algorithm. The resulting algebraic equations due to the compressible Navier-Stokes and the one equation Spallart-Allmaras turbulence equations are solved in a monolithic manner using the restricted additive Schwarz preconditioner combined with the FGMRES(m) Krylov subspace algorithm. The numerical algorithm is validated for the viscous compressible flow around a RAE2822 airfoil and around a DLR-F6 geometry. The sequential and parallel simulations indicate significant improvements in computational performance.

INTRODUCTION

The use of computational fluid dynamics (CFD) has become a principal element for industrial aircraft design because of its flexibility to evaluate alternative designs. However, the use of more sophisticated turbulence model may increase the cost of numerical calculations. Therefore, the considerations related to cache efficiency should be taken into account for speeding up the numerical CFD algorithms. Time required for a single mathematical operation on modern CPUs is far less than a read from RAM (approximately 10 times slower). Hence, computer actually reads some chunks of memory rather than a single entry, assuming the data that is spatially close in the memory is used successively in the algorithm. However, achieving high cache efficiency is not so straightforward. First of all, the algorithm and the data structure should be compatible with each other. After examining the numerical procedure and data access patterns, a proper algorithm and a data structure can be designed together considering the key points related to cache efficiency. In this paper, a second order vertex based finite volume

*PhD. in Aeronautical Engineering Department, Email: semih@itu.edu.tr
†Prof in Astronautical Engineering Department, Email: msahin@itu.edu.tr

solver is developed in order to consider these points. Therefore, quad-edge data structure [guibas and Stolfi , 1985] and half-edge data structure [Botsch et al. , 2002] are examined, redesigned, and simplified in order to fit requirements of the cell-vertex centered finite volume methods. The present data structure is efficient mainly because it enhances the data locality and therefore improves the cache efficiency.

NUMERICAL METHOD

The compressible Navier-Stokes equations can be written in the following non-dimensional form

$$\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV+\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_i}dS-\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_v}dS=0 \tag{1}$$

where V is the control volume, S is the control volume surface area, $\mathbf{Q}$ is the state vector, $\mathbf{F_i}$ is the inviscid fluxes and $\mathbf{F_v}$ is the viscous fluxes. The numerical method based on the vertex based finite volume formulation. The inviscid fluxes are computed with a flux splitting method using Roe scheme and the right and left state vectors are computed using an unweighted second order upwind least square interpolation [Barth , 1991]. In order to improve the computational performance, the least square interpolation coefficients are stored at outgoing edges. The primitive variable gradients required for the viscous fluxes are evaluated at edge mid-points using the Green-Gauss theorem. Although the average value of the convective gradient at vertices may be directly used for the viscous fluxes, this approach is prone to odd-even decoupling. Although the present approach increases the computational cost, it significantly improves the robustness of the numerical algorithm allowing higher CFL numbers. For turbulent flows, the classical one-equation Spalart-Allmaras [Spalart and Allmaras , 1994] turbulence model without transition terms is implemented and it is solved with the Navier-Stokes equations in a fully coupled manner (monolithic). For explicit time integration, we employ Newton's method with the first-order Euler explicit in time. For the implicit implementation, the first order Euler implicit method with Newton's method is used. For a vertex based finite volume formulation, the residual vector is given by

$$\mathbf{R}(\mathbf{Q})=\frac{\partial}{\partial t}\int_{\Omega}\mathbf{Q}dV+\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_i}dS-\oint_{\partial\Omega}\mathbf{n}\cdot\mathbf{F_v}dS=0 \tag{2}$$

and the residual should be zero at time level $n+1$. If the Taylor series is used at time level $n+1$, the Euler implicit approach leads to the following Newtons's method.

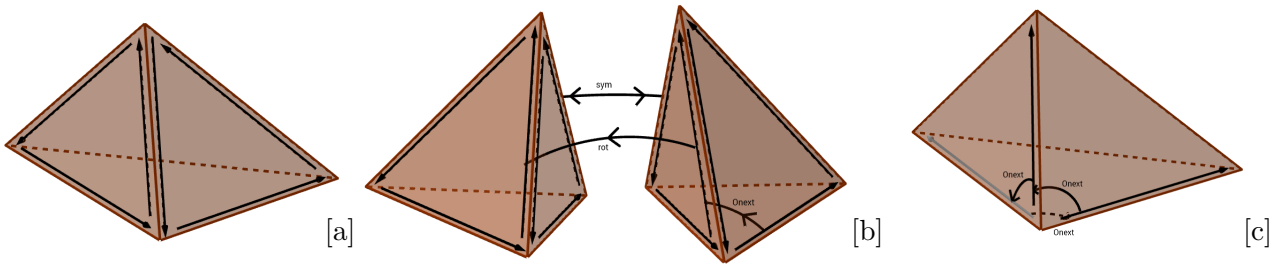
$$\frac{\partial\mathbf{R}}{\partial\mathbf{Q}}\Delta\mathbf{Q}^{m+1}=\left[\frac{M}{\Delta t}+J\right]\Delta\mathbf{Q}^{m+1}=-\mathbf{R}(\mathbf{Q}^{m})=-\left[\frac{\mathbf{Q}^{m}-\mathbf{Q}^{n}}{\Delta t}V+\sum\mathbf{n}\cdot\mathbf{F_i}(\mathbf{Q}^{m})S-\sum\mathbf{n}\cdot\mathbf{F_v}(\mathbf{Q}^{m})S\right] \tag{3}$$

where M corresponds to mass matrix, m is the mth Newton iteration and Δt is the time step. The present implicit version has been implemented within the PETSc 3.7.7 library [Balay et al. , 2018] using the restricted additive Schwarz preconditioner combined with the FGMRES(m) Krylov subspace algorithm.

DATA STRUCTURE

The above vertex based HEMLAB code has been implemented on unstructured hybrid meshes using the following edge based data structure in two- and three-dimensions. The present three-dimensional data structure is an extension to the two-dimensional version [Akkurt and Sahin , 2017]. Although the data structure is capable of managing arbitrary polyhedrons in three-dimensions, it is currently limited to tetrahedra, hexahedra, wedge, and pyramids. The main element of the 3D edge data structure is called facet-edge which is shown in Figure 1-[a]. There is a facet-edge for each edge of each face of a volume element. There are four pointers in the edge data structure, which are named as $Dest$, sym, rot, and $Onext$ as shown in Figure 1-[b] and -[c]. The $Dest$ pointer points the destination node, sym

pointer points the other half of the facet-edge. *rot* pointer is an extension due to the nature of the 3D mesh data. It makes it possible to revolve around the faces that are connected to the edge. Finally, the *Onext* pointer plays a slightly different role in 3D case compared to its 2D version. Using *Onext* pointer, it is possible to rotate around edges of an element that are outgoing from a single node.



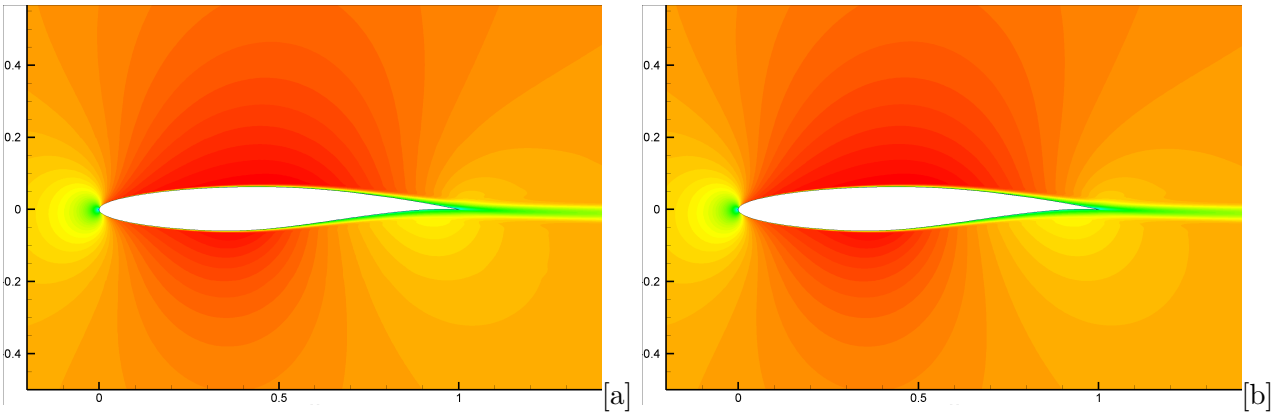
[a] [b] [c]

Figure 1: Three dimensional facet-edges [a], *sym* and *rot* pointers [b], and *Onext* pointer [c].

RESULTS AND DISCUSSION

The initial numerical results corresponds to the subsonic steady flow around the RAE2822 airfoil at $M_\infty = 0.3$, $Re = 1\times10^6$ and $\alpha = 0^\circ$. The mesh consists of 166,439 vertices and 165,188 quadrilateral elements. The mesh has a boundary layer mesh and the mesh is also highly refined in the wake. The free stream turbulence value is set to $0.005\nu_\infty$ and Spalart-Allmaras one equation model is used. The computed Mach contours are provided in Figure 2-[a] and the results are compared with the solution of Standford SU2 code in Figure 2-[b]. The result are indistinguishable one from another. In here, we should note that the Stanford SU2 code employs a weighted least square interpolation and the viscous fluxes are evaluated using the averages of the least square gradient values at vertices.

The three-dimensional validation case corresponds to the transonic flow around the DLR-F6 configuration (WBNP) at $M_\infty = 0.75$, $Re = 3 \times 10^6$ (based on mean cord) and $\alpha = 1^\circ$. The DLR-F6 geometry is proposed by the second AIAA CFD Drag Prediction Workshop and has been extensively used in the literature in both experimentally an numerically [Brodersen and Stuermer , 2001]. The benchmark mesh consists of 7,344,331 vertices and 20,359,140 mixed (tetrahedra and prism) elements (https://dpw.larc.nasa.gov/DPW2/). The computed streamtraces (skin friction lines) along with the pressure contours around the DLR-F6 configuration are provided in Figure 3-[a] and the results are compared with the solution of the SU2 code using the same Spalart-Allmaras one equation turbulence model. The results shows relatively very good agreement.



[a] [b]

Figure 2: Comparison of computed Mach contours [a] with SU2 results [b] at $M_\infty = 0.3$, $Re = 1 \times 10^6$ and $\alpha = 0^\circ$.

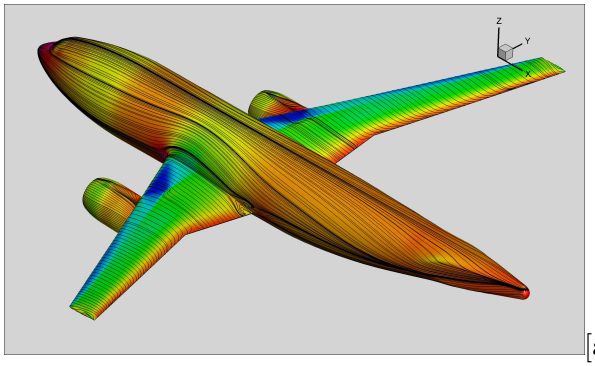
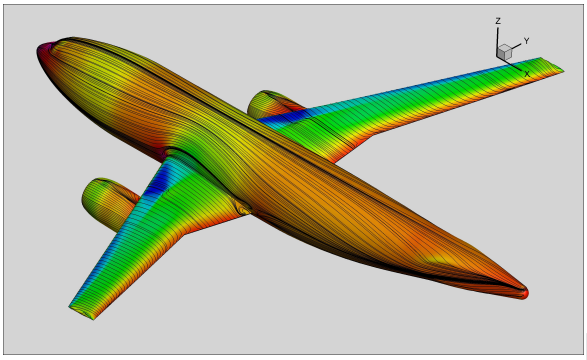

[a]

[b]

Figure 3: Comparison of computed streamtraces (skin friction lines) and pressure contours [a] with SU2 results [b] at $M_\infty = 0.75$, $Re = 3 \times 10^6$ and $\alpha = 1°$.

The edge based data structure significantly improves the CPU time due to increased cache efficiency and the results are given below in the Table 1 for Intel® Core™ i7-4720HQ CPU @ 3.60 GHz processor. Although the numerical algorithms are almost equivalent for the Euler equations (except weighted/unweighted least square interpolations), the HEMLAB solver approximately indicates up to 4.5 times speed-up in three-dimensions for the Euler explicit approach. The performance of the solver for the laminar Navier-Stokes equations is provided in Table 2. Although the performance is decreased due to the usage of a more expensive viscous flux evaluation, it is still faster than SU2 solver.

Table 1: Euler solver performance and comparison with the SU2 solver.

Tetra	Wedge	Pyramid	Hexa	HEMLAB[s]	SU2[s]	Speed-up
1.8m	-	-	-	77.59	198.2	2.55
3.01m	-	-	-	124.4	316.1	2.54
2.10m	1.21m	-	-	133.6		
-	-	-	1.00m	71.84	323.4	4.50

Table 2: Navier-Stokes solver performance and comparison with the SU2 solver.

Tetra	Wedge	Pyramid	Hexa	HEMLAB[s]	SU2[s]	Speed-up
1.8m	-	-	-	164.1	343.0	2.09
3.01m	-	-	-	285.9	588.2	2.06
2.10m	1.21m	-	-	406.4		
-	-	-	1.00m	184.7	577.1	3.12

CONCLUSIONS

An efficient halfedge data structure has been implemented for an edge based finite volume formulation in order to demonstrate its efficiency compared to the classical edge-based implementation for a vertex based finite volume formulation. In the present approach, the quad-edge and halfedge data structures are redesigned and simplified in order to fit requirements of the cell-vertex centered finite volume methods. The method also inherently allows vertex insertion and deletion, which is particularly suitable for adaptive mesh refinement or local remeshing. Together with its data structure, the numerical algorithm is designed to be cache conscious by increasing the spatial/temporal locality of the data that is being used all together for a single computation. The numerical algorithm is validated for the subsonic viscous flows around a RAE2822 airfoil and around a DLR-F6 configuration. The efficiency of the numerical algorithm has been tested by comparing the CPU times for the Stanford SU2 code and significant increase in computational performance is achieved.

References

Akkurt, S. and Sahin, M., (2017), An efficicient data structure for three-dimensional vertex based finite volume method. APS 70th Annual Meeting Division of Fluid Dynamics, Denver, CO, USA, 19-21 November 2017.

Balay, S., Abhyankar, S., Adams, M. F., et al., (2018), PETSc Users Manual. ANL-95/11 - Revision 3.10, Mathematic and Computer Science Division, Argonne National Laboratory. http://www.mcs.anl.gov/petsc

Barth, T. J., (1991), A 3-D upwind Euler solver for unstructured meshes. *AIAA Paper 91-1548-CP*.

Botsch, M., Steinberg S., Bischoff S., Kobbelt L., (2002), OpenMesh - a generic and effcient polygon mesh data structure, 1st OpenSG Symposium.

Brodersen, O. and Stuermer, A., (2001). Drag prediction of engine-airframe interference effects using unstructured Navier-Stokes calculations., 19th AIAA Applied Aerodynamics Conference, Fluid Dynamics and Co-located Conferences, (AIAA 2001-2414), Anaheim, CA, USA.

Guibas, L. and Stolfi, J., (1985), Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams, ACM Transactions on Graphics, 4(2):75-123.

Spalart, P. R. and Allmaras, S. R., (1994), A one-equation turbulence model for aerodynamic flows. La Recherche Aerospatiale 1994(1):521.