AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
23rd AIAA Computational Fluid Dynamics Conference

AIAA 2017-3097

A Finite Volume Method Based on Variational Reconstruction for Compressible Flows on Arbitrary Grids

Lingquan Li*, Xiaodong Liu†, Jialin Lou‡, Hong Luo§
North Carolina State University, Raleigh, NC, 27695, USA

Hiroaki Nishikawa¶
National Institute of Aerospace, Hampton, VA, 23666, USA

Yuxin Ren‖
Tsinghua University, Beijing, 100084, China

A cell-centered finite volume method based on a variational reconstruction, termed FV(VR) in this paper, is developed for compressible flows on 3D arbitrary grids. In this method, a linear polynomial solution is reconstructed using a newly developed variational formulation. The solution gradients are obtained by solving an extreme value problem, which minimizes jumps of the values of the reconstructed polynomial solutions and their spatial derivatives at cell interfaces, and therefore maximizes smoothness of the reconstructed polynomial solutions. Like the least-squares reconstruction, the variational reconstruction has the property of 1-exactness. Unlike the cell-centered finite volume method based on the least-squares reconstruction, termed FV(LS) in this paper, the resulting FV(VR) method is stable even on tetrahedral grids, since its stencils are intrinsically the entire mesh. However, the data structure required by FV(VR) is as the same as FV(LS) and is thus compact and simple. A variety of the benchmark test cases are presented to assess the accuracy, efficiency, robustness and flexibility of this finite volume method. The numerical experiments demonstrate that the developed FV(VR) method is able to maintain the linear stability, attain the designed second-order of accuracy, and outperform the FV(LS) method without a significant increase in computing costs and storage requirements.

I. Introduction

Computational Fluid Dynamics (CFD) has become an indispensable tool for a variety of applications in science and engineering.Generally speaking, numerical methods used in CFD can be classified as structured grid methods, unstructured grid methods and Cartesian grid methods. The structured grid methods alone are not practical for engineering applications, as they have a disadvantage for griding complex geometries. More often, they are used in the context of Chimera or overlapping approaches [1][2] to simplify the grid generation process for a complex configuration. The difficulty of generating a structured grid for complex geometries in the engineering community aroused the interest in the development of unstructured grid method [3]-[7]. Unstructured grids provide great flexibility in dealing with the complex geometries in practice and offer a natural framework for solution-adaptive mesh refinement. However, the computational costs and memory requirements for unstructured triangular/tetrahedral grids are generally higher than for structured grids. Although the solution accuracy may not be strongly affected by element type even in the boundary layers, computational efficiency can be benefit substantially through the use of prismatic elements in the boundary layers and Cartesian cells in the inviscid regions. This is due to a simple fact that approximately, five to six times more tetrahedra than hexahedra are required to fill a given region with a fixed number of nodes. Although the boundary layer regions occupy only a small portion of the computational domain, it is not uncommon for more than half of the mesh resolution to be packed into this small region, and thus the quadrilateral elements in 2D and prismatic elements in 3D can lead to a significant saving in both memory requirements and computational costs. To ensure high computational efficiency, unstructured triangular/tetrahedral elements should be kept minimal. The advantages

*Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, AIAA Student Member. lli32@ncsu.edu
†Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, AIAA Student Member.
‡Graduate Research Assistant, Department of Mechanical and Aerospace Engineering, AIAA Student Member.
§Professor, Department of Mechanical and Aerospace Engineering, Associate Fellow AIAA.
¶Associate Research Fellow, Associate Fellow AIAA.
‖Professor, Department of Engineering Mechanics, Member AIAA.

of Cartesian grid approaches [8]-[12] include ease of grid generation, lower computational storage requirements, and significantly less operational count per cell. However, the main challenge in using Cartesian methods is how to deal with arbitrary boundaries, as the grids are not body-aligned. The cells of a Cartesian mesh near the body may intersect with boudary surfaces. A hybrid method is in need that combines the advantages and strengths of all these three grid types.

In recent years, significant progress has been made in developing finite volume methods to solve the compressible Euler and Navier-Stokes equations on unstructured grids [13]-[24]. These methods have gained widespread acceptance in recent years for their robustness, intuitive formulation, and computational advantages. There mainly exist two types of finite volume methods: cell-centered and vertex-centered finite volume methods. Although the debate that which one has an edge over the other will probably never end, the cell-centered finite volume formulation can be easily extended to higher-order (>2nd) approximation. In the cell-centered finite volume method, the mesh cells serve as control volumes and the flux integrals are evaluated on the boundary of the cells. Since the boundaries of the four most common element types (tetrahedra, prisms, pyramids and hexahedra) are consisted of either triangular or quadrilateral faces, the boundary flux integrals can be readily approximated by a numerical integration scheme, e.g., by a quadrature rule. Furthermore, the cell-centered finite volume formulation is preferred in the context of arbitrary grids, which may contain hanging nodes. The existence of several cell types in a hybrid grid poses a great challenge to numerical methods. Although it is unavoidable to treat different cell types differently during the pre-processing and post-processing stages, it is undesirable that an algorithm depends on the mesh topology during the flow solution stage. The required conditinal statements not only lead to a sophiscated code but also adversely affect the program speed. Therefore, an algorithm that treats different cell types in the same way is needed. Such an algorithm is termed grid-transparent in the literature, which does not require any information on the local cell topology. A gird-transparent scheme has a number of advantages: first of all it can significantly reduce the discretization stencils compared to a non-grid-transparent scheme; secondly it can increase the speed of the program; last it can facilitate the implementation of implicit schemes and parallelizatons.

Stability and accuracy of a finite volume mehtod are mainly determined by the evaluation of numerical fluxes, the accuracy of reconstruction schemes, and the use of slope limiters, which are needed to suppress spurious oscillations in the vicinity of discontinuities. Most efforts in the development of the finite volume methods are primarily focused on numerical fluxes with different Riemann solvers and slope limiters to avoid oscillations in the vincinity of strong discontinuities. Highly accurate and efficient reconstruction schemes remain relatively unexplored, although they can significantly improve the accuracy of the finite volume methods. Most of work [25]-[28] in this area is a continuation of the pioneering work by Barth and Frederickson [29] based on a least-squares reconstruction. Although the second-order cell-centered finite volume methods based on a least-squares reconstruction are in general stable in 2D and structured grids in 3D, they suffer from the so-called linear instablility on unstructured tetrahedral grids, when the reconstruction stencils only involve von Neumann neighborhood, i.e., adjacent face-neighboring cells [30]. In fact, the second-order cell-centered finite volume methods can hardly achieve a formal second order of accuracy for the solution of the Euler and Navier-Stokes equations in practice, which can be attributed to the lack of a highly accurate reconstruction method.

The objective of the presented work is to develop an accurate, efficient and robust cell-centered finite volume method for solving compressible flow problems on arbitrary grids. The novelty of this finite volume method is to use a newly developed variational formulation [31] to reconstruct a linear polynomial solution. This variational reconstruction can be regarded as an extension of the compact finite difference schemes [32] to the unstructured grids. The solution gradients are obtained by solving an extreme value problem, which minimizes jumps of the values of the reconstructured polynomial solutions and their spatial derivatives at cell interfaces, and therefore maximizes smoothness of the reconstructed polynomial solutions. Like the least-squares reconstruction, the variational reconstruction has the property of 1-exactness. Unlike the cell-centered finite volume method based on the least-squares reconstruction, termed FV(LS) in this paper, the resulting FV(VR) method is stable even on tetrahedral grids, since its stecils ate intrinsically the entire mesh. However, the data-structure required by FV(VR) is the same as FV(LS) and is thus compact and simple. A variety of the benchmark test cases are presented to assess the accuracy, efficiency, robustness and flexibility of this finite volume method. The numerical experiments demonstrate that the VR method is more accurate and converges faster than its LS counterpart and the developed FV(VR) method is able to maintain the linear stability, attain the designed second-order of accuracy, and outperform the FV(LS) method without a significant increase in computing costs and storage requirements. The remainder of this paper is organized as follows. The governing equations are presented in Section 2. The developed variational reconstruction based finite volume method is described in

Section 3. Extensive numerical experiments are reported in Section 4. Concluding remarks are given in Section 5.

II. Governing Equations

The Euler equations governing unsteady compressible inviscid flows over any region Ω with boundary $\Gamma = \partial\Omega$ can be expressed in integral form as

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{U} d\Omega + \int_{\Gamma} \mathbf{F}_j \cdot \mathbf{n}_j d\Gamma = 0 \tag{2.1}$$

where the summation convention is used here, and $\mathbf{n}_j$ is the outward unit normal to Γ. The conservative variable vector $\mathbf{U}$, and inviscid flux vector $\mathbf{F}$ are defined by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix} \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho e + p) \end{pmatrix} \tag{2.2}$$

Here ρ, p and e denote the density, pressure, and specific total energy of the fluid, respectively, and u_i is the velocity of the flow in the coordinate direction x_i. The pressure is computed from the equation of state

$$p = (\gamma - 1)\rho(e - \frac{1}{2}u_j u_j) \tag{2.3}$$

which is valid for perfect gas, where γ is the ratio of the specific heats.

III. Numerical Method

3.1. Finite Volume Formulation

The system of the governing compressible Euler equatons is discretized in space using a cell-centered finite volume method. In a finite volume method, the computational domain Ω is divided by a set of non-overlapping control volumes Ω_i with boundary $\Gamma = \partial\Omega_i$ that can be one or combination of four most common element types: tetrahedra, prisms, pyramids and hexahedra. On each control volume, the integral form of the governing equations is required to be satisfied.

$$\int_{\Omega_i} \frac{\partial \mathbf{U}}{\partial t} d\Omega + \int_{\Gamma_i} \mathbf{F}_j \mathbf{n}_j d\Gamma = 0 \tag{3.1}$$

Take the unknowns to be the cell-averaged conservative variable vector

$$\mathbf{U}_i = \frac{1}{V_i}\int_{\Omega_i} \mathbf{U} d\Omega \tag{3.2}$$

By summing the boundary integration over the cell interfaces between the cell Ω_i and its adjacent face-neighboring cell Ω_j, $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$, and by approximating the flux integral using one point quadrature at the midpoint of the face Γ_{ij} in Eq. (3.1), the semi-discrete form of the equations may be written as

$$V_i \frac{d\mathbf{U}_i}{dt} + \sum_j \mathbf{F}_{ij} \cdot \mathbf{n}_{ij} \Gamma_{ij} = 0 \tag{3.3}$$

where V_i is the volume of the control volume Ω_i, $\mathbf{n}_{ij}$ is the outward unit normal to Γ_{ij}, and the flux function gives the flux through the face Γ_{ij} seperating the control volume Ω_i from the adjacent control volume Ω_j. If the numerical fluxes at the interfaces in Eq. (3.3) are simply evaluated as an arithmetic average of the normal fluxes, the resulting finite volume scheme, equivalent to the classic central differencing scheme, allows for the apperearance of checker boarding modes, and thus suffers from numerical instabilities, unless some type of numerical dissipation in the form of artificial viscosity is introduced. To construct a stable scheme for the compressible Euler equations, any of the Riemann solvers can be formulated by adopting different forms for the numerical fluxes at the interface. When an upwind scheme is used to compute the numerical fluxes at the cell interface via simple cell-averaged flow variables, i.e.,

$$\mathbf{F}_{ij} \cdot \mathbf{n}_{ij} = \mathbf{F}(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) \tag{3.4}$$

the resultant finite volume method is only first-order accurate in space. A second order of accuracy can be achieved using a reconstruction scheme where a piecewise linear polynomial solution is reconstructed in the control volume using the neighboring cells. The numerical fluxes at the interfaces are then evaluated using upwind-biased interpolations of the solution via the MUSCL approach. This leads to the numerical fluxes

$$\mathbf{F}_{ij} \cdot \mathbf{n}_{ij} = \mathbf{F}(\mathbf{U}_{ij}^{-}, \mathbf{U}_{ij}^{+}, \mathbf{n}_{ij}) \tag{3.5}$$

The upwind-biased interpolations of the solutions for $\mathbf{U}_{ij}^{-}$ and $\mathbf{U}_{ij}^{+}$ are defined by

$$\mathbf{U}_{ij}^{-} = \mathbf{U}_i + \phi_i(\mathbf{x}_{ij} - \mathbf{i}) \cdot \nabla \mathbf{U}_i, \tag{3.6}$$

and

$$\mathbf{U}_{ij}^{+} = \mathbf{U}_j + \phi_j(\mathbf{x}_{ij} - \mathbf{j}) \cdot \nabla \mathbf{U}_j, \tag{3.7}$$

where $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_{ij}$ are the position vector for the centers of cell Ω_i, Ω_j and face Γ_{ij}, respectively, and ϕ is a slope limiter.

Equation (3.3) represents a system of ordinary differential equation. A fully implicit time scheme is employed to integrate Equation (3.3) to reach steady-state solutions. An approximate Newton method is used to linearize the equations arising from the implicit discretization. A fast, matrix-free implicit method, GMRES+LU-SGS method [22][23], is then used to solve the resultant system of linear equations. In this work, HLLC scheme [33] is used to compute interface fluxes. Several limiters can be chosen to suppress spurious oscillation of the numerical solutions in the vicinity of the discontinuities. As our design goal is to develop a grid transparent finite volume method on mixed element type grids with possible hanging nodes, the data structure of the flow solver is based on the face of the grid, rather than on the elements. A face based data structure has the advantage that there no limitations on the number of faces, which can be connected to an element, and thus can handle hanging nodes with ease. The computation of fluxes and their contribution to cells can be effictively implemented by loops over the faces through gather/scatter operations.

3.2. Reconstructin Schemes

As it can be seen above, the accuracy of a reconstruction method can have a big effect on the stability and accuracy of the finite volume methods. A second order finite volume method needs to reconstruct a piecewise linear polynomial solution using cell-averaged values of the flow variables in the neighboring cells. This simply requires the evaluation of the derivatives in the cell for the flow variables. One of the most commonly used and simplest reconstruction schemes is the least-squares reconstruction [29], where the computation of gradients is performed in the form of a minimization problems. This least-squares reconstructed finite volume method FV(LS) can be successfully used to solve the 2D compressible Euler equations for smooth flows on arbitrary grids and is able to achieve the designed second order of accuracy and significantly improve the accuracy of the underlying first-order FV method. However, when extended to solve the 3D compressible Euler equations on tetrahedral grids, the FV(LS) suffers from the so-called linear instability, which occurs even for the linear hyperbolic equation [30]. This linear instability is attributed to the fact that the reconstruction stencils only involve von Neumann neighborhood, i.e., adjacent face-neighboring cells. The linear stability can be achieved using extended stencils, which will unfortunately sacrifice the compactness of the underlying reconstruction method.

Alternatively, a recently developed variational reconstruction [31] can be used to obtain the solution gradients by solving an extreme value problem, which minimizes jumps of the values of the reconstructed polynomial solutions and their spatial derivatives at cell interfaces, and therefore maximizes smoothness of the reconstructed polynomial solutions. In this case, a cost function in the variational reconstruction can be defined as

$$\mathbf{I} = \sum_{iface=1}^{nface} \mathbf{I}_{iface} \tag{3.8}$$

where nface is the number of cell interfaces in a grid and $\mathbf{I}_{iface}$ denotes the interface jump integration function for a given interface. For an internal face Γ_{ij} separating the left element i and the right element j, $\mathbf{I}_{iface}$ is given by

$$\mathbf{I}_{iface} = \frac{1}{d_{ij}} \int_{\Gamma_{ij}} \sum_{p=0}^{1} \left(d_{ij}^{p} w_p [\frac{\partial^p \mathbf{U}}{\partial \mathbf{x}^p}] \right)^2 d\Gamma \tag{3.9}$$

$$= \frac{1}{d_{ij}} \int_{\Gamma_{ij}} (w_0[\mathbf{U}])^2 + \left(d_{ij} w_1[\frac{\partial^p \mathbf{U}}{\partial x}]\right)^2 + \left(d_{ij} w_1[\frac{\partial^p \mathbf{U}}{\partial y}]\right)^2 + \left(d_{ij} w_1[\frac{\partial^p \mathbf{U}}{\partial z}]\right)^2 d\Gamma$$

where d_{ij} is the distance between the centroids of the two cells, w_p is the weight, and [] denotes the jump operator, which is defined as

$$[\mathbf{U}] = \mathbf{U}^+ - \mathbf{U}^- \tag{3.10}$$

where $\mathbf{U}^-$ and $\mathbf{U}^+$ denote the values of $\mathbf{U}$ on Γ_{ij} for the two elements *i* and *j*, respectively. The face integral can be computed exactly using Gaussian quadrature formulas with sufficient precision. The constitutive relations of the variational reconstruction are derived by minimizzing the total **I** with respect to the coefficients of the reconstruction polynomial, i.e., the gradients. This leads to a system of linear equations, which is then solved using the LU-SGS method [22]. The resultant finite volume method: FV(VR), can achieve the linear stability, since the stencils of this variational reconstruction are intrinsically the entire mesh. In appendix, we prove that this VR reconstruction in 1D recovers to the compact finite difference scheme [32] on uniform grids. Consequently, the VR reconstruction can be viewed as the extension of the compact difference schemes on arbitrary grids.

When extended to k-th order reconstruction, $\mathbf{I}_{iface}$ can be written as

$$\mathbf{I}_{iface} = \frac{1}{d_{ij}} \int_{\Gamma_{ij}} \sum_{p=0}^{k} \sum_{m=0}^{p} \sum_{n=0}^{m} \left(d_{ij}^p w_{m,n,p}[\frac{\partial^p \mathbf{U}}{\partial \mathbf{x}^n \partial \mathbf{y}^m \partial \mathbf{z}^{p-m-n}}]\right)^2 d\Gamma \tag{3.11}$$

3.3. WENO Limiter

In order to maintain the so-called non-linear stability, i.e., to suppress non-physical oscillations in the vicinity of strong discontinuities, the solution gradients on cell *i* is finally obtained using a nonlinear WENO reconstruction [34][35] as a convex combination of the VR reconstructed first derivatives at the cell itself (k=0) and its face-neighboring cells (k=1, ..., njface),

$$\frac{\partial \mathbf{U}}{\partial x_n}|_i^{WENO} = \sum_{k=0}^{njface} w_k \frac{\partial \mathbf{U}}{\partial x_n}|_k \tag{3.12}$$

where *njface* is the number of the face-neighboring cells for cell *i* and the normalized nonlinear weights w_k are computed as

$$w_k = \frac{\widetilde{w}_k}{\sum_{i=0}^{nface} \widetilde{w}_k} \tag{3.13}$$

The non-normalized nonlinear weights $\widetilde{w}_k$ are functions of the linear weights λ_i and the so-called oscillation indicator o_k

$$\widetilde{w_i} = \frac{\lambda_i}{(\epsilon + o_k)^\gamma} \tag{3.14}$$

where ϵ is a small positive number used to avoid division by zero, and γ is an integer parameter to control how fast the non-linear weights decay for non-smooth stencils. The oscillation indicator for the reconstructed second order polynomials is simply defined as

$$o_k = \sqrt{(\frac{\partial U}{\partial x_n}|_k)^2} \tag{3.15}$$

The linear weights λ_i can be chosen to balance the accuracy and the non-oscillatory property of the FV method. Note that the VR reconstructed polynomial at the cell itself serves as the central stencil and the least-squares reconstructed polynomials on its face-neighboring cells act as biased stencils in this WENO reconstruction.

IV. Numerical Examples

A few examples are presented in this section to demonstrate the high accuracy and robustness of our FV(VR) method for compressible flow problems on arbitrary grids. All solutions in test case 2 to test case 4 are obtained

by the second order methods without any limiters, in an effort to ensure that the solution accuracy is developed on arbitrary grids. The code has the ability to compute 1D, 2D and 3D problems. Results for one-dimensional flows can be readily obtained by setting the number of cells in both y- and z- directions to be 1 using a hexahedral grid. Fow two-dimensional problems, the number of cells in the z-direction is simply set to be 1. The first test case is to assess the 1-exactness and the accuracy of the variational reconstruction method. Test case 2 to test case 4 are chosen to demonstrate that the developed FV(VR) method is able to achieve the designed second-order of convergence for smooth flows. The last two test cases are presented to illustrate the applicability of the FV(VR) method for solving problems of scientific and industrial interests for complex configurations. The length scale, characterizing the cell size of an unstructured grid, is defined as $1/\sqrt[2]{ncells}$ and $1/\sqrt[3]{ncells}$, for 2D and 3D problems, respectively, where $ncells$ is the number of cells. The L2-norm of the entropy production is used as the error measurement

$$\|\epsilon\|_{L_2(\Omega)} = \sqrt{\int_\Omega \epsilon^2 d\Omega} \tag{4.1}$$

where the entropy production ϵ is defined as

$$\epsilon = \frac{S - S_\infty}{s_\infty} = \frac{p}{p_\infty}(\frac{\rho_\infty}{\rho})^\gamma - 1 \tag{4.2}$$

Note that the entropy production, where the entropy is defined as $S = p/\rho^\gamma$, is a very good criterion to measure accuracy of the numerical solutions, when the flow under consideration is smooth and therefore isentropic.

Test case 1. Convergence study on reconstruction methods

This test case is chosen to assess the 1-exactness and the accuracy of the variational reconstruction method in comparison with its least-squares counterpart on both hexahedral and tetrahedral grids. The computational domain is a cube ($0 \leq x_i \leq 1$) for hexahedral grids. Tetrahedral grids are the ones used in test case 4. A fully linear polynomial function and a smooth function are used to assess the accuracy, the order of convergence, and the 1-exact property of the two reconstruction methods. Table 1 and Table 2 present numerical results obtained by the least-squares reconstruction and VR reconstruction with 1 and 4 Gauss points for $\mathbf{I}_{iface}$ on hexahedral grids and 1 and 3 Gauss points for $\mathbf{I}_{iface}$ on tetrahedral grids, which are also illustrated in Figure 1 and Figure 2. Mesh size, L2-error of the error function and the order of convergence are shown. As expected, both least-squares reconstruction and variational reconstruction methods have the 1-exact property, i.e., being able to reconstruct a linear polynomial exactly. For a generally smooth function, both the least-squares reconstruction and the variational reconstruction can achieve the designed second order of convergence. However, the variational reconstruction is more accurate and has a faster convergence rate than the least-squares reconstruction, especially on tetrahedral grids. In addition, using more Gauss points leads to a more accurate reconstruction.

Table 1. L2-error and order of convergence of reconstruction methods on hexahedral grids

	LS		VR(1 Gauss points)		VR(4 Gauss points)	
Element number	L2-error	Order	L2-error	Order	l2-error	Order
Case 1: $f(x,y,z) = 999x - 888y + 777z - 666$						
$8 \times 8 \times 8$	5.1018e-14	-	2.2413e-13	-	2.4294e-13	-
$16 \times 16 \times 16$	5.2064e-14	-	4.8416e-13	-	4.8350e-13	-
$32 \times 32 \times 32$	4.9396e-14	-	9.3799e-13	-	9.3768e-13	-
$64 \times 64 \times 64$	4.8193e-14	-	1.7902e-12	-	1.7901e-12	-
Case 2: $f(x,y,z) = sin(\pi x)sin(\pi y)sin(\pi z)$						
$8 \times 8 \times 8$	1.0467e-2	-	9.1387e-3	-	1.0841e-2	-
$16 \times 16 \times 16$	2.6027e-3	2.015	2.0723e-3	2.148	2.2795e-3	2.258
$32 \times 32 \times 32$	6.5055e-4	2.007	5.0036e-4	2.057	5.2014e-4	2.139
$64 \times 64 \times 64$	1.6265e-4	2.007	1.2375e-4	2.022	1.2570e-4	2.056

Table 2. L2-error and order of convergence of reconstruction methods on tetrahedral grids

	LS		VR(1 Gauss points)		VR(3 Gauss points)	
Element number	L2-error	Order	L2-error	Order	l2-error	Order
Case 1: $f(x,y,z) = 999x - 888y + 777z - 666$						
535	1.9738e-10	-	1.2420e-9	-	7.6205e-10	-
$2,426$	1.9285e-10	-	8.1763e-10	-	6.3624e-10	-
$16,467$	1.8852e-10	-	7.5296e-10	-	5.0341e-10	
$124,706$	1.8789e-10	-	5.1960e-10	-	4.6568e-10	-
Case 2: $f(x,y,z) = sin(\pi x/10)sin(\pi y/10)sin(\pi z/10)$						
535	33.0761	-	25.4885	-	24.3049	-
$2,426$	10.8501	1.614	6.6682	1.941	5.2408	2.221
$16,467$	2.8138	1.954	1.4078	2.252	0.8692	2.601
$124,706$	0.7031	2.008	0.3158	2.164	0.1764	2.324

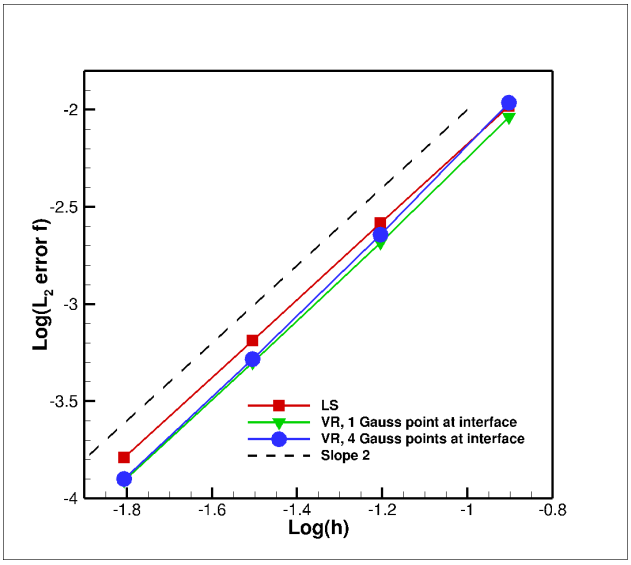


Figure 1. Convergence histories of the least-squares and variational reconstruction methods on hexahedral grids

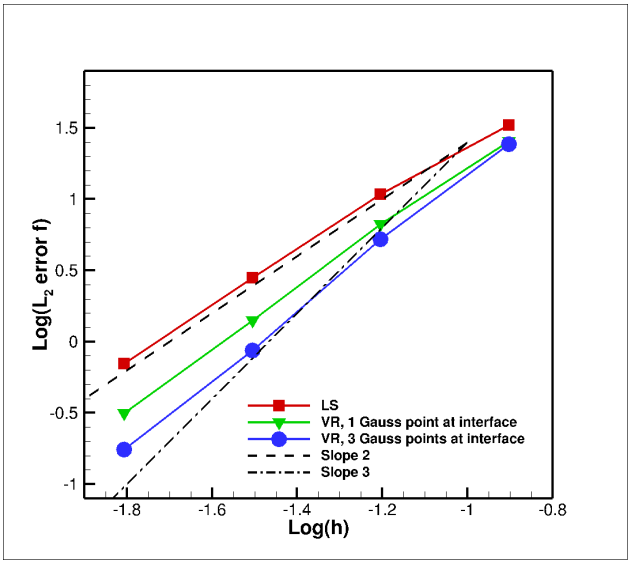


Figure 2. Convergence histories of the least-squares and variational reconstruction methods on tetrahedral grids

Test case 2. Subsonic flow through a channel with a smooth bump

This test case is chosen to demonstrate the accuracy and convergence rates of the FV(VR) and FV(LS) methods for internal flows. The problem under consideration is a subsonic flow inside a 3D channel with a smooth bump on the lower surface. The height, width and length of the channel are 0.8, 0.8 and 3, respectively. The shape of the lower wall is defined by the function $0.0625exp(-25x^2)$ from $x = -1.5$ to $x = 1.5$. The inflow condition is prescribed at a Mach number of 0.5, and an angle of attack of 0°. Figure 3 shows the three successively refined prismatic grids used in the grid convergence study. The numbers of prismatic elements, grid points and boundary faces for the three grids are (512, 914, 1024), (2048, 2850, 4096) and (8192, 9794, 16384), respectively. The cell size is haved between consecutive meshes. Numerical solutions to this problem are computed using the FV(VR) method on these three grids to obtain a quantitative measurement of the order of accuracy and discretization errors. Figure 4 illustrates the computed velocity contours in the flow field obtained by the FV(VR) method on these three grids. The errors and the convergence rates for both FV(VR) and FV(LS) methods are reported in Table 3. It shows the mesh size, the L2-error of the error function and the order of convergence. Both methods are able to attain the designed second order of the convergence in this test case, being 2.252 and 1.869, respectively. The FV(VR) method offers a better convergence and more accurate solutions on the finest grid than the FV(LS) method.

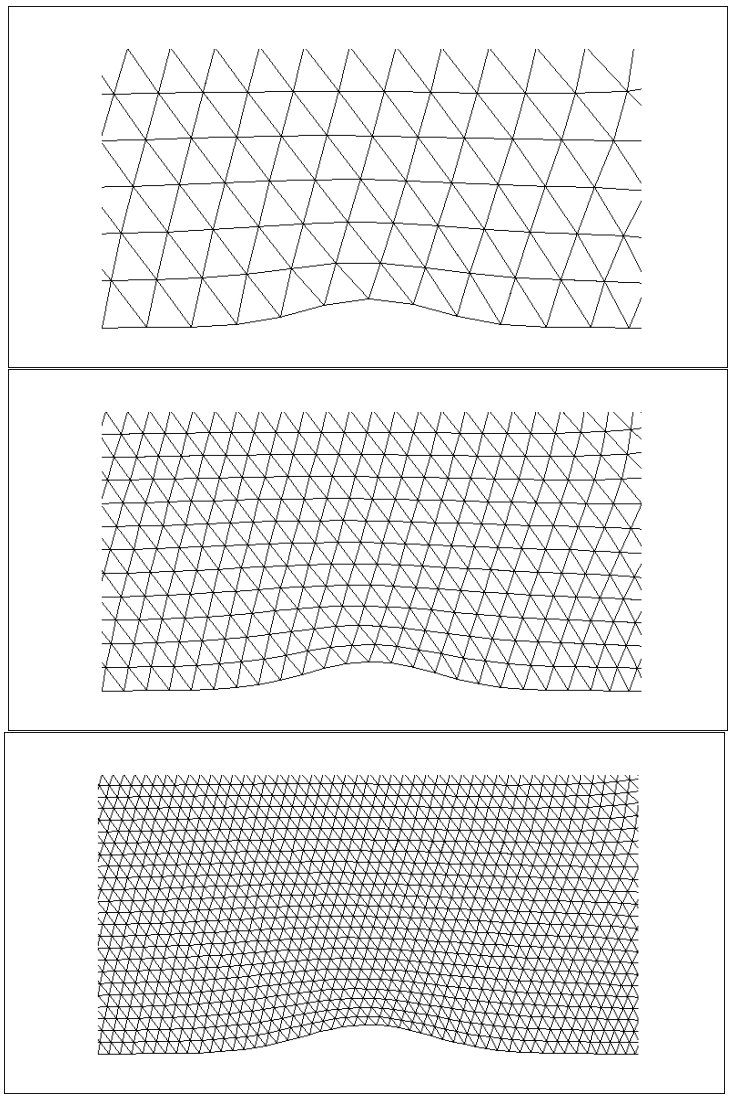


Figure 3. A sequence of three successively globally refined unstructured prismatic meshes used for computing a subsonic flow in a channel with a smooth bump

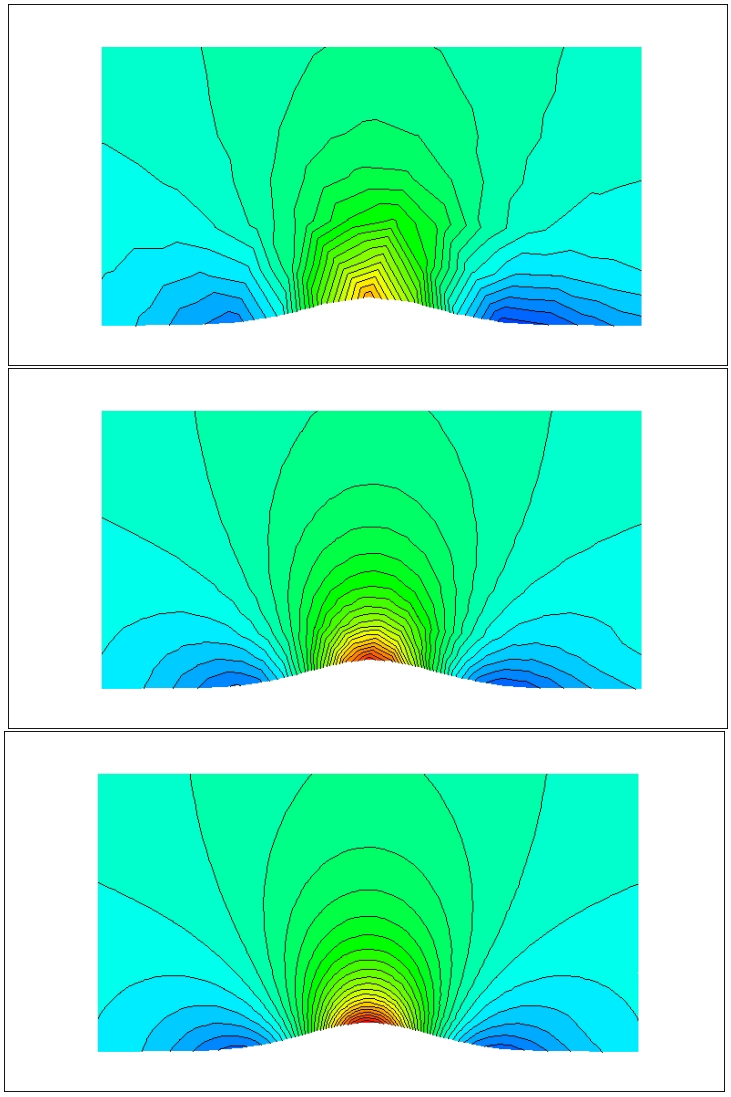


Figure 4. Computed velocity contours in the flow field obtained by the FV(VR) method on a sequence of three successively globally refined unstructured prismatic grids for a subsonic flow through a channel with a bump on the lower surface at $Ma_\infty = 0.5$

Table 3. L2-error and order of convergence for the FV(VR) and FV(LS) methods

	FV(VR)		FV(LS)	
Log(Length Scale)	Log(L2-error)	Order	Log(L2-error)	Order
-2.3046	-3.0680	-	-3.1910	-
-2.6056	-3.6956	2.085	-3.7416	1.829
-2.9066	-4.4203	2.407	-4.3160	1.908

Test case 3. Subsonic flow past a circular cylinder

An inviscid subsonic flow past a circular cylinder at a Mach number of $M_\infty = 0.38$ is considered in this test case to assess the order of accuracy and discretization error of the FV(VR) and FV(LS) methods for external flows. Computations are performed on two types of grids: one is consisted of hexahedral cells and the other is consisted of prismatic cells. Figure 5 and Figure 7 show four successively refined o-type hexahedral and prismatic grids with 16×5, 32×9, 64×17 and 128×33 points, respectively. The first number refers to the number of points in the circular direction, and the second number refers to the number of concentric circles in the mesh. The radius of the cylinder is $r_1 = 0.5$.

The domain is bounded by $r_{33} = 20$. The radii of concentric circles for 128×33 mesh are set up as

$$r_i = r_1 \left(1 + \frac{2\pi}{128} \sum_{j=0}^{i-1} \alpha^j \right), \quad i = 2, ..., 33. \tag{4.3}$$

where $\alpha = 1.1580372$. The coarser grids are generated by successively coarsing the finest mesh. Numerical solutions to this problem are computed on these four grids to obtain quantitative measurement of the order of accuracy and discretization errors of the FV(VR) and FV(LS) methods. The computed Mach number contours obtained by these two methods on the hexahedral and prismatic grids are shown in Figure 6 and Figure 8, respectively. The errors and the convergence rates for both methods are reported in Table 4 and Table 5, respectively. The mesh size, L2-error and the order of convergence are shown in the tables. The FV(VR) method is able to achieve the designed second order of the convergence on both hexahedral and prismatic grids, being 2.063 and 2.675, respectively. However, the FV(LS) method can barely attain a full second order of convergence even for this simple problem on a well-designed set of grids, being 1.704 and 2.091, respectively. The FV(VR) method offers a better convergence and more accurate solutions on the fine girds than the FV(LS) method, demonstrating the superior performance of the FV(VR) method over the FV(LS) method.

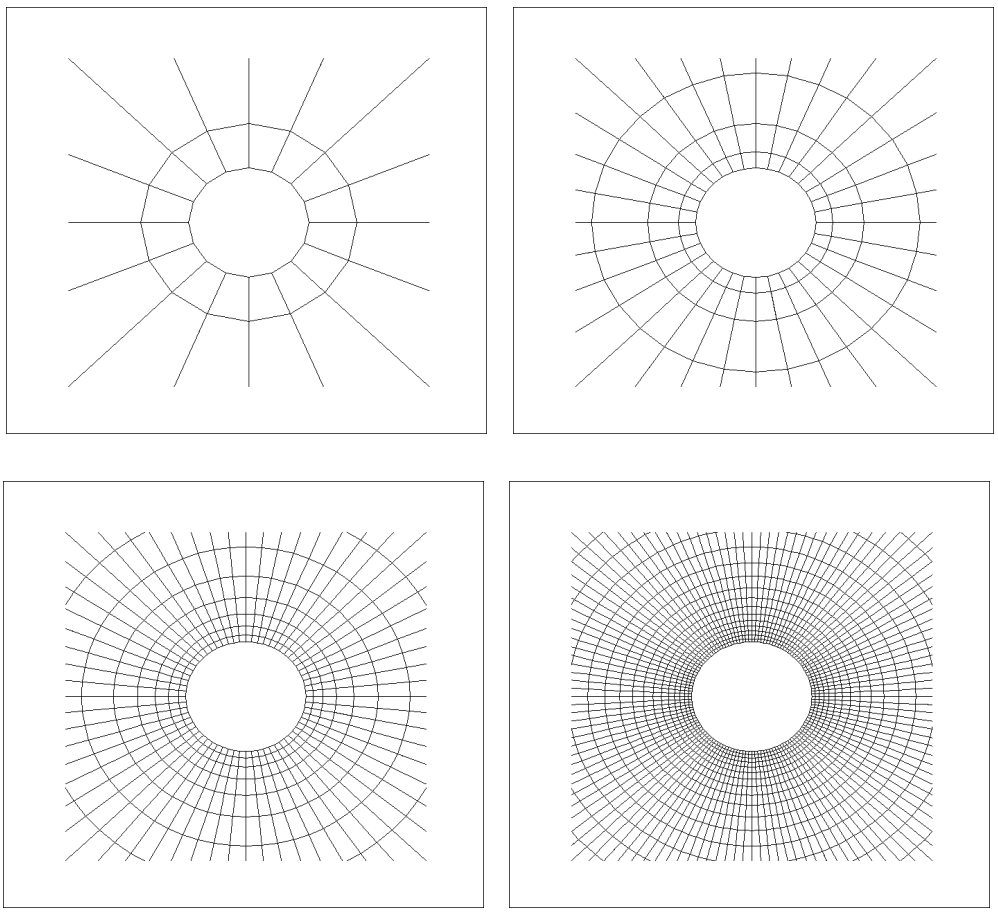


Figure 5. Sequences of four successively globally refined hexahedral meshes 16×5, 32×9, 64×17 and 128×33 for computing inviscid subsonic flow past a circular cylinder

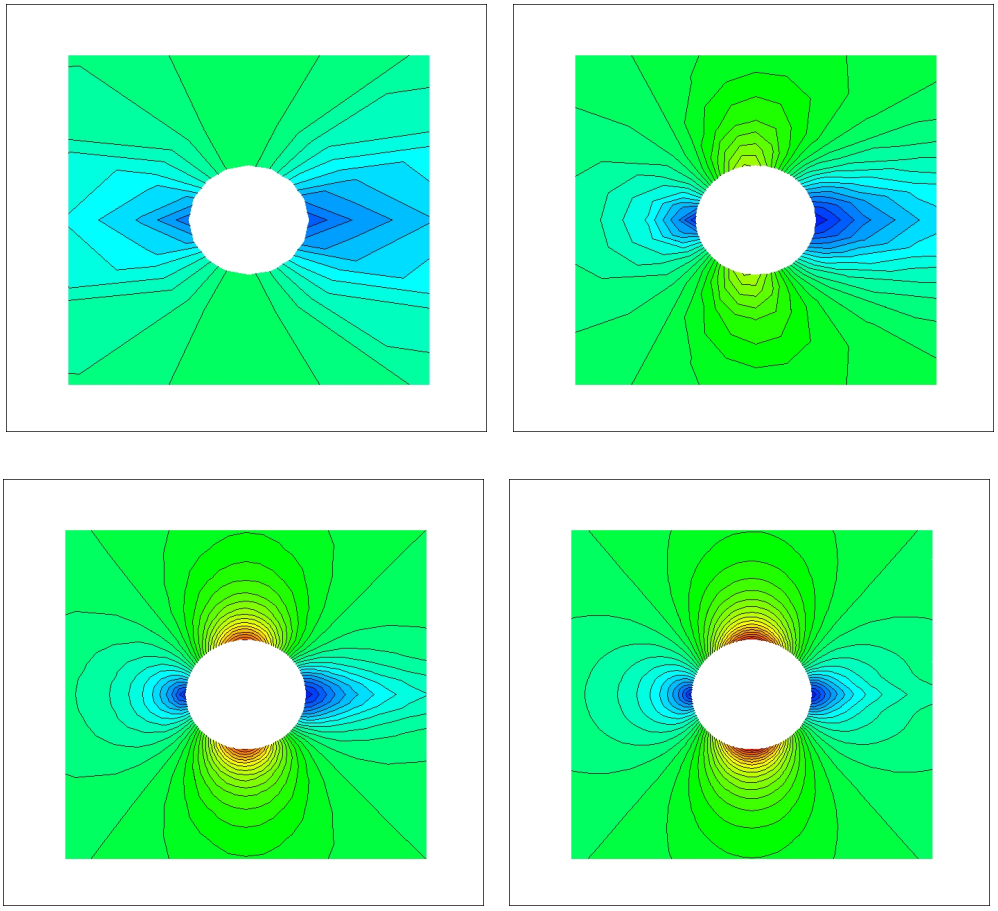
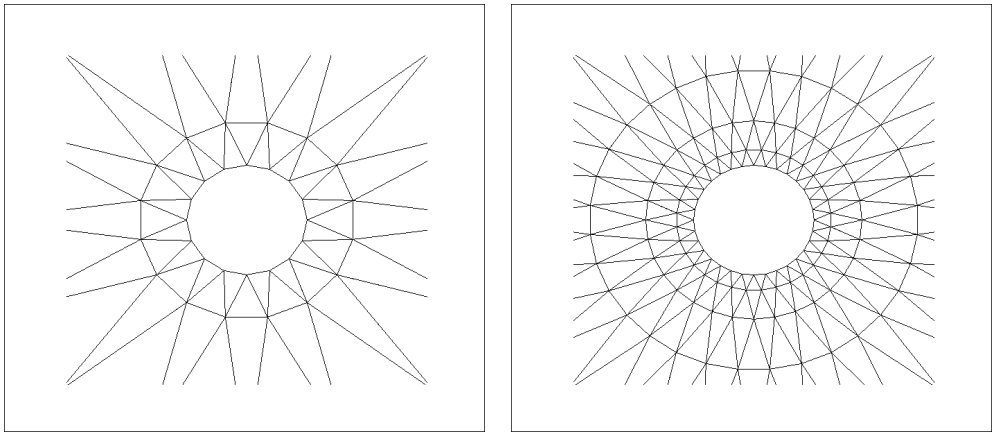


Figure 6. Computed Mach number contours on four successively refined hexahedral grids obtained using the FV(VR) scheme for subsonic flow past a circular cylinder at a Mach number of 0.38



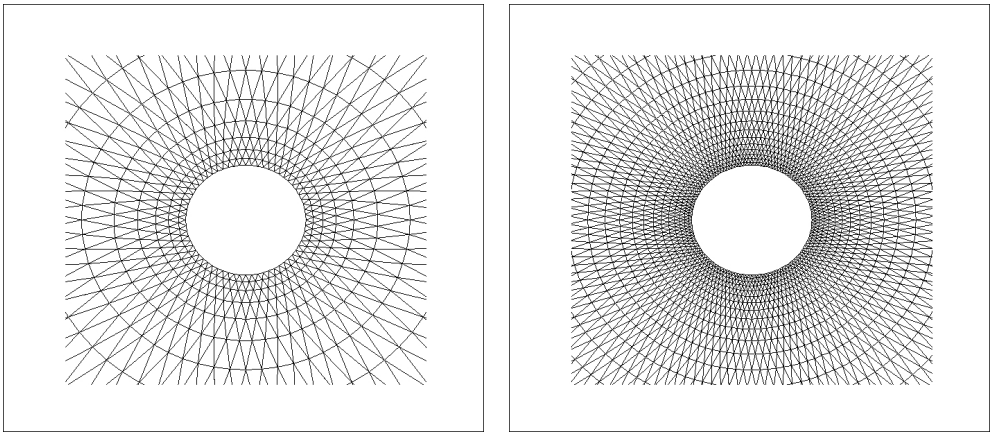


Figure 7. Sequences of four successively globally refined prismatic meshes 16×5, 32×9, 64×17 and 128×33 for computing inviscid subsonic flow past a circular cylinder

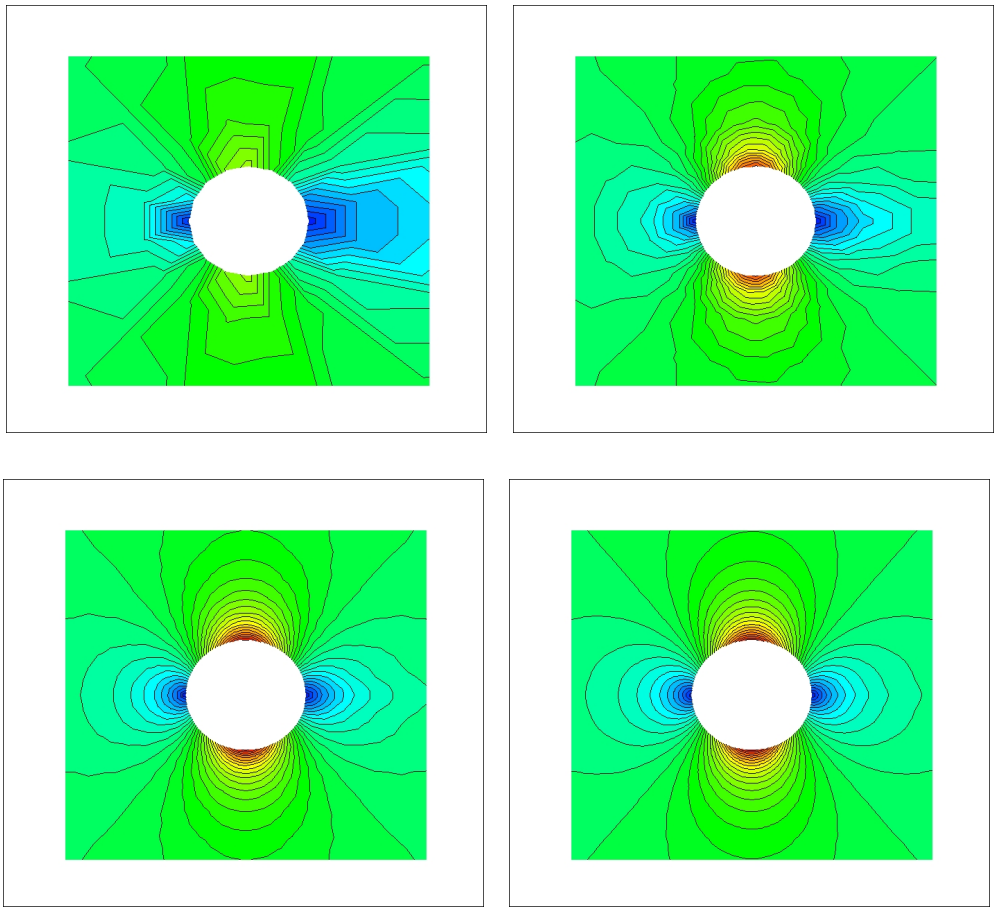


Figure 8. Computed Mach number contours on four successively refined prismatic grids obtained using the FV(VR) scheme for subsonic flow past a circular cylinder at a Mach number of 0.38

Table 4. L2-error and order of convergence for the FV(VR) and FV(LS) methods on hexahedral grids

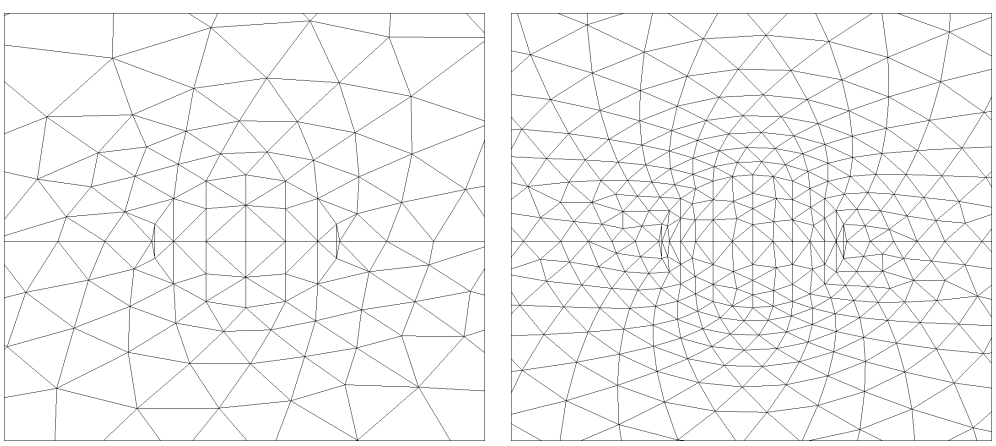
	FV(VR)		FV(LS)	
Log(Length Scale)	Log(L2-error)	Order	Log(L2-error)	Order
-0.903	-0.4931	-	-0.6982	-
-1.204	-0.9172	1.409	-1.0846	1.284
-1.505	-1.5745	2.184	-1.6368	1.834
-1.806	-2.3559	2.596	-2.2341	1.984

Table 5. L2-error and order of convergence for the FV(VR) and FV(LS) methods on prismatic grids

	FV(VR)		FV(LS)	
Log(Length Scale)	Log(L2-error)	Order	Log(L2-error)	Order
-1.054	-0.8984	-	-0.9132	-
-1.355	-1.6419	2.470	-1.5898	2.278
-1.657	-2.4842	2.798	-2.2354	2.115
-1.957	-3.3144	2.758	-2.8011	1.880

Test case 4. Subsonic flow past a sphere

In this test case, a subsonic flow past a sphere at a Mach number of $M_\infty = 0.5$ is considered to assess if the FV(VR) method can achieve a formal order of convergence rate on tetrahedral grids. A sequence of four successively refined tetrahedral grids used in the grid convergence study is shown in Figure 9. The numbers of tetrahedral elements, grid points and boundary faces for the four grids are (535, 167, 124), (2426, 598, 322), (16467, 3425, 1188) and (124706, 23462, 4538), respectively. The cell size is haved between consecutive meshes. Note that only a quarter of the configuration is modeled due to the symmetry of the problem, and that the number of elements on a successively refined mesh is not exactly eight times the coarse mesh's element number due to the nature of unstructured grid generation. The computations are conducted on the four grids using the FV(VR) method. Since the FV(LS) method is unstable for this test case, the results obtained by a second-order DG(P1) method on the first three grids are also presented for the purpose of comparison. Figure 10 illustrates the computed velocity contours in the flow field obtained by the FV(VR) method on these four grids. The errors and the convergence rates for both FV(VR) and DG(P1) methods are reported in Table 8. The mesh size, L2-error and the order of convergence are shown in this table. Figure 11 provides the details of the spatial convergence of the FV(VR) and DG(P1) methods for this numerical experiment. Both FV(VR) and DG(P1) achieve the expected second order of convergence rates, being 1.995 and 2.096, respectively. The second order DG(P1) method is more accurate than the second order FV(VR) method on the same grids. However, the FV(VR) solution is almost as accurate as the DG(P1) solution for the same number of degrees of freedom, convincingly demonstrating the high accuracy of the FV(VR) method.



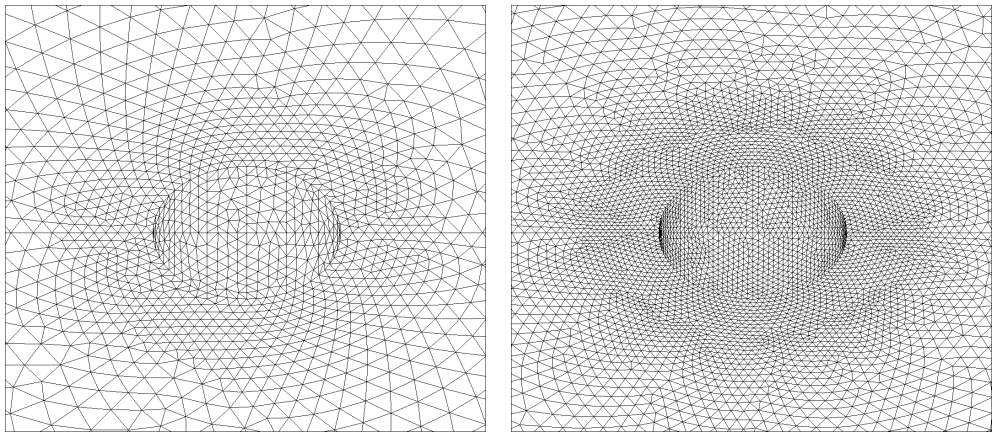


Figure 9. A series of four successively globally refined tetrahedral meshes for subsonic flow past a sphere at $M_\infty = 0.5$

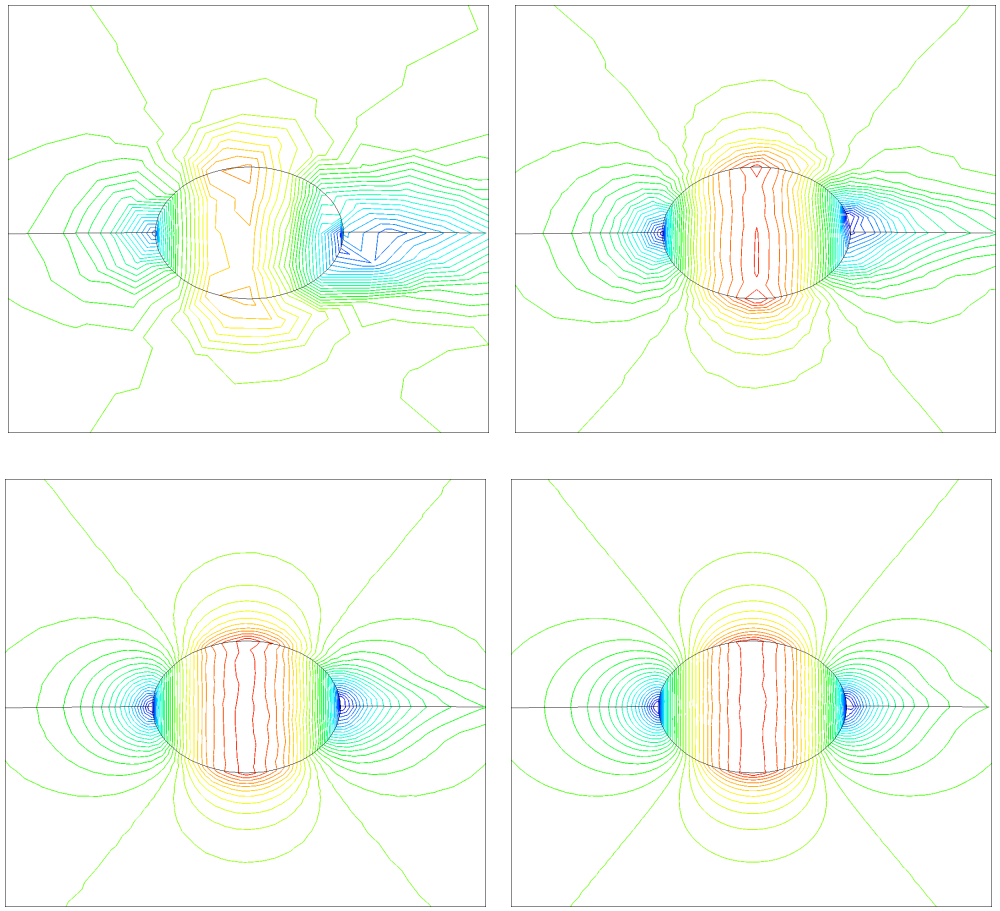


Figure 10. Computed velocity contours in the flow field obtained by the FV(VR) method for subsonic flow past a sphere at $M_\infty = 0.5$

Table 6. L2-error and order of convergence for the FV(VR) and DG(P1) methods

	FV(VR)		DG(P1)	
Log(Length Scale)	Log(L2-error)	Order	Log(L2-error)	Order
-1.1283	-2.1000	-	-2.5294	-
-1.4055	-2.6730	2.067	-3.1311	2.170
-1.6986	-3.2146	1.848	-3.7236	2.021
-1.9951	-3.8284	2.071	-	-

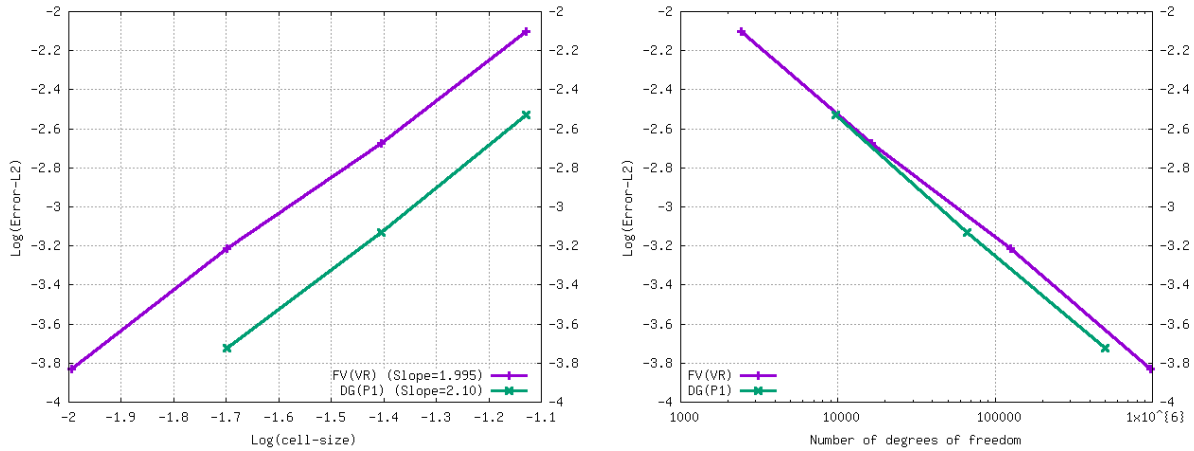


Figure 11. Accuracy summary for an inviscid subsonic flow past a sphere obtained using the FV(VR) and DG(P1) method

Test case 5. Transonic flow past an ONERA M6 wing

A transonic flow over the ONERA M6 wing at a Mach number of $M_\infty = 0.84$ and an attack angle of $\alpha = 3.06^o$ is considered here. This test case is chosen to demonstrate that the FV(VR) method is able to obtain a stable solution for transonic flows using a WENO limiter. The mesh used in this computation consists of 593,169 tetrahedral cells, 110,282 grid points and 39,770 boundary faces. Figure 12 shows the computed pressure contours on the upper wing surface obtained by the FV(VR) solutions. The computed pressure coefficients obtained by the FV(VR) method are compared with experimental data [36] at six span-wise stations in Figure 13. The results obtained by the FV(VR) method compare closely with the experimental data, except at the root stations, due to the lack of viscous effects. The shocks are captured sharply essentially without any oscillations, demonstrating the high accuracy and non-oscillatory property of our FV(VR) method, when combined with a WENO limiter.

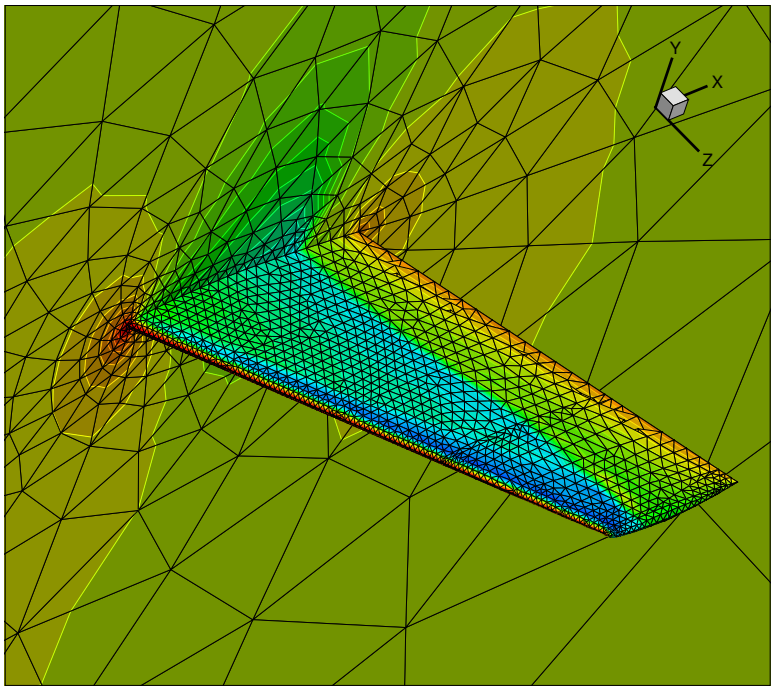


Figure 12. Computed pressure contours on the unstructured surface mesh obtained by the FV(VR) method (nelem = 593,169, npoin = 110,282, nface = 39,770) for a transonic flow past a M6 wing at $M_\infty = 0.84$

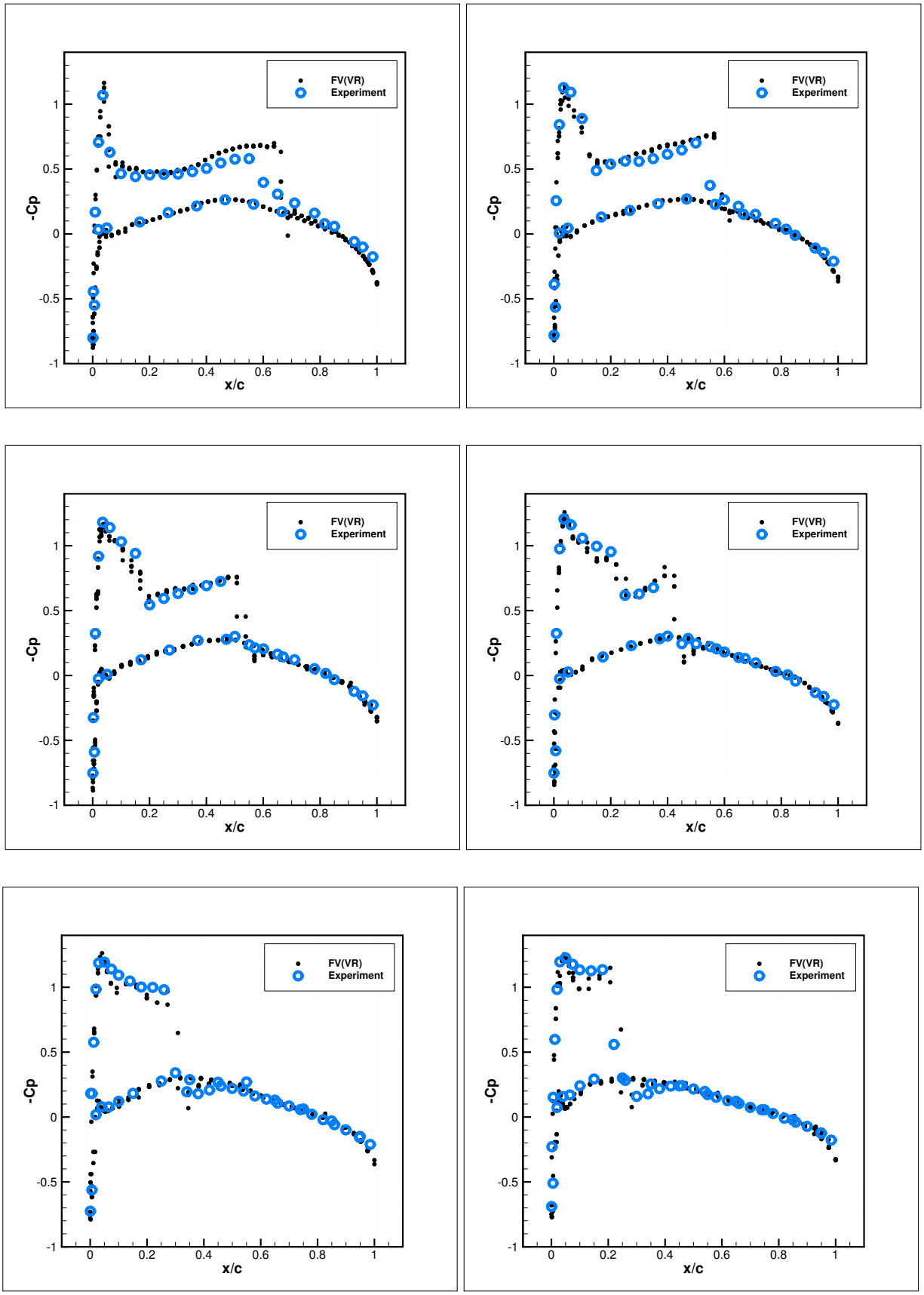


Figure 13. Pressure coefficient distributions for wing section at 20%, 44%, 65%, 80%, 90%, 95% semispan locations

Test case 6. Transonic flow past a Boeing 747 aircraft

Finally, a transonic flow past a complete Boeing 747 aircraft is presented. The 747 configuration includes the fuselage, wing, horizontal and vertical tails, under-wing pylons, and flow-through engine nacelle. The mesh used in this test case, contains 371,162 grid points, 1,025,170 tetrahedral cells, and 60,780 (double check) boundary faces for the half-span airplane. The solution is computed at a free stream of Mach number of 0.85 and an angle of attack of 2°. The computed Mach number contours on the surface of the airplane, along with the surface mesh, are shown in Figure 14. This example confirms that the developed FV(VR) method can be used to compute complicated flows of practical importance for complex configurations.

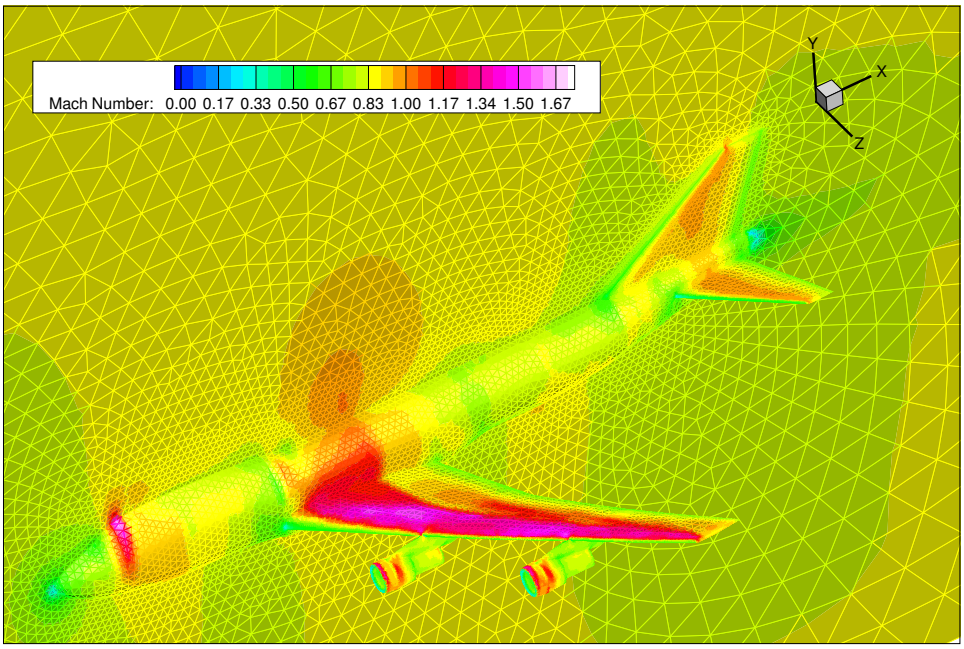


Figure 14. Computed Mach number contours and unstructured surface mesh for transonic flow past a complete B747 aircraft at $M_\infty = 0.84$

V. Conclusions

A cell-centered finite volume method based on a variational reconstruction, FV(VR), has been developed for solving the compressible Euler equations on 3D arbitrary grids. The variational reconstruction can be viewed as an extension of the compact finite difference schemes on unstructured grids and has the property of 1-exactness. The resulting FV(VR) method is linearly stable, since its stencils are intrinsically the entire mesh. However, the data structure required by the FV(VR) method is compact and simple. A variety of the benchmark test cases are presented to assess the performance of this FV(VR) method. The numerical experiments indicate that the developed FV(VR) method is able to maintain the linear stability, attain the designed second-order of accuracy, and outperform the FV(LS) method without a significant increase in computing costs and storage requirements. Future development will be focused on the extension of the second-order FV(VR) method to a higher-order FV(VR) method.

Acknowledgements

This work has been partially funded by the U.S. Army Research Office under the contract/grant number W911NF-16-1-0108 with Dr. Mattew Munson as the program manager.

References

[1] Benek, J., Buning, P., and Steger, J., "A 3-D chimera grid embedding technique," *7th Computational Physics Conference*, 1985, p. 1523.

[2] Benek, J., Donegan, T., and Suhs, N., "Extended Chimera grid embedding scheme with application to viscous flows," *8th Computational Fluid Dynamics Conference*, 1987, p. 1126.

[3] George, P. L., Hecht, F., and Saltel, É., "Automatic mesh generator with specified boundary," *Computer methods in applied mechanics and engineering*, Vol. 92, No. 3, 1991, pp. 269–288.

[4] Lohner, R., "Three-dimensional grid generation by the advancing front method," *Int. J. Numer. Meths. Fluids.*, Vol. 8, 1988, pp. 1135–1149.

[5] Pirzadeh, S., "Unstructured viscous grid generation by the advancing-layers method," *AIAA journal*, Vol. 32, No. 8, 1994, pp. 1735–1737.

[6] Sharov, D., Luo, H., Baum, J. D., and Löhner, R., "Unstructured Navier–Stokes grid generation at corners and ridges," *International Journal for Numerical Methods in Fluids*, Vol. 43, No. 6-7, 2003, pp. 717–728.

[7] Connell, S. D. and Braaten, M. E., "Semistructured mesh generation for three-dimensional Navier-Stokes calculations," *AIAA journal*, Vol. 33, No. 6, 1995, pp. 1017–1024.

[8] Clarke, D. K., Hassan, H., and Salas, M., "Euler calculations for multielement airfoils using Cartesian grids," *AIAA journal*, Vol. 24, No. 3, 1986, pp. 353–358.

[9] Coirier, W. J. and Powell, K. G., "An accuracy assessment of Cartesian-mesh approaches for the Euler equations," *Journal of Computational Physics*, Vol. 117, No. 1, 1995, pp. 121–131.

[10] Berger, M. and LEVEQUE, R., "An adaptive Cartesian mesh algorithm for the Euler equations in arbitrary geometries," *9th Computational Fluid Dynamics Conference*, 1989, p. 1930.

[11] Aftosmis, M., Melton, J., and Berger, M., "Adaptation and surface modeling for Cartesian mesh methods," *12th Computational Fluid Dynamics Conference*, 1995, p. 1725.

[12] Luo, H., Baum, J. D., and Löhner, R., "A hybrid Cartesian grid and gridless method for compressible flows," *Journal of Computational Physics*, Vol. 214, No. 2, 2006, pp. 618–632.

[13] NAKAHASHI, K., "FDM-FEM zonal approach for viscous flow computations over multiple-bodies," *25th AIAA Aerospace Sciences Meeting*, 1987, p. 604.

[14] Weatherill, N., "Mixed structured-unstructured meshes for aerodynamic flow simulation," *The Aeronautical Journal (1968)*, Vol. 94, No. 934, 1990, pp. 111–123.

[15] Mavriplis, D. and Venkatakrishnan, V., "A unified multigrid solver for the Navier-Stokes equations on mixed element meshes," *International Journal of Computational Fluid Dynamics*, Vol. 8, No. 4, 1997, pp. 247–263.

[16] Nakahashi, K., Sharov, D., Kano, S., and Kodera, M., "Applications of unstructured hybrid grid method to high-Reynolds number viscous flows," *International Journal for Numerical Methods in Fluids*, Vol. 31, No. 1, 1999, pp. 97–111.

[17] Coirier, W. and Jorgenson, P., "A mixed volume grid approach for the Euler and Navier-Stokes equations," *34th Aerospace Sciences Meeting and Exhibit*, 1996, p. 762.

[18] Haselbacher, A. and Blazek, J., "Accurate and efficient discretization of Navier-Stokes equations on mixed grids," *AIAA journal*, Vol. 38, No. 11, 2000, pp. 2094–2102.

[19] Haselbacher, A., McGuirk, J. J., and Page, G. J., "Finite volume discretization aspects for viscous flows on mixed unstructured grids," *AIAA journal*, Vol. 37, No. 2, 1999, pp. 177–184.

[20] Sbardella, L. and Imregun, M., "An efficient discretization of viscous fluxes on unstructured mixed-element grids," *International Journal for Numerical Methods in Biomedical Engineering*, Vol. 16, No. 12, 2000, pp. 839–849.

[21] Frink, N. T. and Pirzadeh, S. Z., "Tetrahedral finite-volume solutions to the Navier-Stokes equations on complex configurations," 1998.

[22] Luo, H., Baum, J. D., and Löhner, R., "A fast, matrix-free implicit method for compressible flows on unstructured grids," *Journal of Computational Physics*, Vol. 146, No. 2, 1998, pp. 664–690.

[23] Luo, H., Baum, J. D., and Löhner, R., "An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids," *Computers & fluids*, Vol. 30, No. 2, 2001, pp. 137–159.

[24] Luo, H., Sharov, D., Baum, J. D., and Loehner, R., "On the computation of compressible turbulent flows on unstructured grids," *International Journal of Computational Fluid Dynamics*, Vol. 14, No. 4, 2001, pp. 253–270.

[25] Ollivier-Gooch, C., Nejat, A., and Michalak, K., "On obtaining high-order finite-volume solutions to the Euler equations on unstructured meshes," *18th AIAA Computational Fluid Dynamics Conference*, 2007, p. 4464.

[26] Michalak, C. and Ollivier-Gooch, C., "Unstructured high-order accurate finite-volume solutions of the Navier-Stokes equations," *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, 2009, p. 954.

[27] Diskin, B. and Thomas, J. L., "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: inviscid fluxes," *AIAA journal*, Vol. 49, No. 4, 2011, pp. 836–854.

[28] Diskin, B., Thomas, J. L., Nielsen, E. J., Nishikawa, H., and White, J. A., "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes," *AIAA journal*, Vol. 48, No. 7, 2010, pp. 1326–1338.

[29] Barth, T. and FREDERICKSON, P., "Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction," *28th aerospace sciences meeting*, 1990, p. 13.

[30] Haider, F., Croisille, J.-P., and Courbet, B., "Stability analysis of the cell centered finite-volume M uscl method on unstructured grids," *Numerische Mathematik*, Vol. 113, No. 4, 2009, pp. 555–600.

[31] Wang, Q., Ren, Y.-X., Pan, J., and Li, W., "Compact high order finite volume method on unstructured grids III: Variational reconstruction," *Journal of Computational physics*, Vol. 337, 2017, pp. 1–26.

[32] Lele, S. K., "Compact finite difference schemes with spectral-like resolution," *Journal of computational physics*, Vol. 103, No. 1, 1992, pp. 16–42.

[33] Batten, P., Leschziner, M., and Goldberg, U., "Average-state Jacobians and implicit methods for compressible viscous and turbulent flows," *Journal of computational physics*, Vol. 137, No. 1, 1997, pp. 38–78.

[34] Luo, H., Xia, Y., Li, S., Nourgaliev, R., and Cai, C., "A Hermite WENO reconstruction-based discontinuous Galerkin method for the Euler equations on tetrahedral grids," *Journal of Computational Physics*, Vol. 231, No. 16, 2012, pp. 5489–5503.

[35] Luo, H., Xia, Y., Spiegel, S., Nourgaliev, R., and Jiang, Z., "A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids," *Journal of Computational Physics*, Vol. 236, 2013, pp. 477–492.

[36] Schmitt, V. and Charpin, F., "Pressure distributions on the ONERA-M6-Wing at transonic Mach numbers," *Experimental data base for computer program assessment*, Vol. 4, 1979.

Appendix: 1

In this appendix, the equivalence of the variational reconstruction to the well-known compact finite difference scheme is demonstrated on a uniform grid. The jump at interface $i+\frac{1}{2}$ is defined as

$$\mathrm{I}_{i+\frac{1}{2}} = \frac{1}{2}\left\{\left[w_0\left(u^+_{i+\frac{1}{2}} - u^-_{i+\frac{1}{2}}\right)\right]^2 + \left[w_1 h\left(\frac{du}{dx}\Big|^+_{i+\frac{1}{2}} - \frac{du}{dx}\Big|^-_{i+\frac{1}{2}}\right)\right]^2\right\}$$

where

$$u^+_{i+\frac{1}{2}} = u_{i+1} - \frac{h}{2}s_{i+1}, \quad u^-_{i+\frac{1}{2}} = u_i + \frac{h}{2}s_i, \quad \frac{du}{dx}\Big|^+_{i+\frac{1}{2}} = s_{i+1}, \quad \frac{du}{dx}\Big|^-_{i+\frac{1}{2}} = s_i.$$

The objective function is defined as the summation of the interface jumps over all the interfaces

$$I = \sum_{i=0}^{n} \mathrm{I}_{i+\frac{1}{2}}$$

The equations for the slopes s can then be derived by finding the minimizer of the objective function. Taking the derivative I with respect to s_i and setting it to be zero leads to

$$\frac{\partial I}{\partial s_i} = \sum_{i=0}^{n} \frac{\partial \mathrm{I}_{i+\frac{1}{2}}}{\partial s_i} = \frac{\partial \mathrm{I}_{i-\frac{1}{2}}}{\partial s_i} + \frac{\partial \mathrm{I}_{i+\frac{1}{2}}}{\partial s_i} = 0$$

Note that only $\mathrm{I}_{i-\frac{1}{2}}$ and $\mathrm{I}_{i+\frac{1}{2}}$ contain the coefficient s_i and

$$\begin{aligned}
\frac{\partial \mathrm{I}_{i+\frac{1}{2}}}{\partial s_i} &= w_0\left(u^+_{i+\frac{1}{2}} - u^-_{i+\frac{1}{2}}\right)(-\frac{h}{2})w_0 + w_1 h\left(\frac{du}{dx}\Big|^+_{i+\frac{1}{2}} - \frac{du}{dx}\Big|^-_{i+\frac{1}{2}}\right)(-h)w \\
&= \left(\frac{w_0^2}{4} + w_1^2\right)h^2 s_i + \left(\frac{w_0^2}{4} - w_1^2\right)h^2 s_{i+1} - \frac{w_0^2 h}{2}\left(u_{i+1} - u_i\right) \\
\frac{\partial \mathrm{I}_{i-\frac{1}{2}}}{\partial s_i} &= w_0\left(u^+_{i-\frac{1}{2}} - u^-_{i-\frac{1}{2}}\right)(-\frac{h}{2})w_0 + w_1 h\left(\frac{du}{dx}\Big|^+_{i-\frac{1}{2}} - \frac{du}{dx}\Big|^-_{i-\frac{1}{2}}\right)hw \\
&= \left(\frac{w_0^2}{4} - w_1^2\right)h^2 s_{i-1} + \left(\frac{w_0^2}{4} + w_1^2\right)h^2 s_i - \frac{w_0^2 h}{2}\left(u_i - u_{i-1}\right)
\end{aligned}$$

The resulting linear equation for the slope s becomes

$$\left(\frac{w_0^2}{4} - w_1^2\right)h^2 s_{i-1} + 2\left(\frac{w_0^2}{4} + w_1^2\right)h^2 s_i + \left(\frac{w_0^2}{4} - w_1^2\right)h^2 s_{i+1} = \frac{w_0^2 h}{2}\left(u_{i+1} - u_{i-1}\right)$$

Note that when $w_1 = 0.5w_0$, the slope is given by the central differencing scheme. When $w_1 = 0$, the above equation can be simplified as

$$\frac{s_{i-1}}{2} + s_i + \frac{s_{i+1}}{2} = \frac{u_{i+1} - u_{i-1}}{h}$$

which is the second order compact finite difference scheme[32]. Consequently, the variational reconstruction scheme can be regarded as the extension of the compact finite difference scheme on unstructured grids. The resultant linear system for the slopes is symmetric, compact and diagonally dominant. When w_1 is not equal to zero, the matrix of the linear system is strictly diagonally dominant, and therefore nonsingular. Consequently, the gradient solution always exists uniquely and easy to compute numerically.