

Time-Accurate Local Time Stepping and High-Order Space-Time CESE Methods for Multi-Dimensional Flows with Unstructured Meshes

Chau-Lyan Chang^{*},
NASA Langley Research Center, Hampton, VA 23681

Balaji Venkatachari^{**} and Gary Cheng[†]
University of Alabama at Birmingham, Birmingham, AL 35294

With the wide availability of affordable multiple-core parallel supercomputers, next generation numerical simulations of flow physics are being focused on unsteady computations for problems involving multiple time scales and multiple physics. These simulations require higher solution accuracy than most algorithms and computational fluid dynamics codes currently available. This paper focuses on the developmental effort for high-fidelity multi-dimensional, unstructured-mesh flow solvers using the space-time conservation element, solution element (CESE) framework. Two approaches have been investigated in this research in order to provide high-accuracy, cross-cutting numerical simulations for a variety of flow regimes: 1) time-accurate local time stepping and 2) high-order CESE method. The first approach utilizes consistent numerical formulations in the space-time flux integration to preserve temporal conservation across the cells with different marching time steps. Such approach relieves the stringent time step constraint associated with the smallest time step in the computational domain while preserving temporal accuracy for all the cells. For flows involving multiple scales, both numerical accuracy and efficiency can be significantly enhanced. The second approach extends the current CESE solver to higher-order accuracy. Unlike other existing explicit high-order methods for unstructured meshes, the CESE framework maintains a CFL condition of one for arbitrarily high-order formulations while retaining the same compact stencil as its second-order counterpart. For large-scale unsteady computations, this feature substantially enhances numerical efficiency. Numerical formulations and validations using benchmark problems are discussed in this paper along with realistic examples.

Nomenclature

L	=	length of the cavity
f	=	frequency of instability oscillations
M	=	freestream Mach number
Re	=	Reynolds number based on length
k	=	cylinder height for both protuberance and cavity
D	=	depth of the cavity, or diameter of the cylindrical roughness element
δ	=	boundary-layer thickness
t	=	time
x	=	streamwise coordinate
y	=	wall-normal coordinate
z	=	spanwise coordinate in direction parallel to leading edge
ρ	=	density
u	=	streamwise velocity

^{*} Aerospace Technologist, MS 128, Computational AeroSciences Branch, Senior Member, AIAA

^{**} Post-doctoral Research Fellow, Department of Mechanical Engineering, Member, AIAA

[†] Associate Professor, Department of Mechanical Engineering, Senior Member, AIAA

v	=	wall-normal velocity
w	=	spanwise velocity
e	=	total energy, defined in the second section
p	=	pressure
R	=	gas constant
T	=	temperature
U	=	dependent solution vector, defined in the second section
V	=	integration volume
Ω	=	surface for the integration volume
F	=	flux vectors for general conservation laws
F_x, F_y, F_z	=	flux vectors for compressible Navier-Stokes equations in three spatial directions
S	=	source vector for general conservation laws
O, V_i	=	solution points and vertices within each solution element
V'_i	=	stretched vertices to add numerical dissipation in the EBD approach
Δ_{\min}	=	global minimum spacing between two vertices
$\sigma_{\min}, \sigma_{\max}$	=	global lower and upper limit of the numerical dissipation parameter
x_0, y_0, z_0	=	coordinates of solution point within the solution element
γ	=	ratio of the specific heat
\vec{h}	=	flux in the joint space-time domain
\vec{s}	=	surface normal vector in the joint space-time domain
A	=	area of the space-time element interface
δ_p	=	shock detection parameter
ε	=	small number 10^{-20}
θ_i	=	magnitude of derivative vectors
ζ_i	=	sorted values of θ_i
ψ_i	=	intermediate variables used to derive weighted average
w_i	=	weighting for i -th derivative
a	=	coefficient of the scalar wave equation
θ	=	wave variable used to define solutions to the scalar wave equation
A_w	=	acoustic wave amplitude
ω	=	angular frequency of acoustic wave
k_x, k_y, k_z	=	wave numbers in $x, y,$ and z direction, respectively

Subscripts

x, y, z, t = derivatives in spatial and temporal directions

Superscripts

i = number index for surrounding element and derivatives

I. Introduction

After decades of computational fluid dynamics (CFD) research, the primary development in recent years is focused on high-fidelity numerical computations for complex configurations. While confidence in predictive capabilities using CFD is relatively high in certain domains such as transonic aerodynamics using steady-state Reynolds Averaged Navier-Stokes (RANS) equations¹, available codes and numerical algorithms to-date are still deficient in handling time accurate computations over complex configurations with multiple physical scales. For example, computations of unsteady wakes behind a large roughness element in supersonic or hypersonic boundary layers²⁻⁴ remain quite challenging. The spontaneous instability waves due to the presence of the roughness element could trigger unstable, high-frequency waves downstream that eventually could lead to laminar-to-turbulent transition. Several issues are associated with such simulations. First, to handle the geometry with a reasonable grid count, an unstructured mesh is favored, despite the fact that most matured and well-used CFD codes are still structured-mesh based. Second, temporal accuracy is a main concern. To simulate such highly unstable flows, numerical dissipation has to be sufficiently reduced in order not to hamper the generation and

development of instability waves. Third, computation of transitional flows or fully turbulent flows without any modeling requires resolution of multiple temporal and spatial scales. The challenging aspect is to preserve accuracy with large disparity of mesh sizes while maintaining computational efficiency. Last, but not least, is the requirement to handle interactions of fine scale waves with flow discontinuities. Shock capturing inevitably demands larger numerical dissipation that could in turn improperly suppress the wave development. Improving numerical algorithms and developing efficient parallel computer codes to meet the above requirements are no doubt one of the most important research directions for CFD development in the next decade.

The importance of using unstructured meshes for complex configurations goes beyond doubts. Complex geometries pose great challenges for structured grid generation. Laying out topological arrangement to cope with corners, small parts, and minute details appears to be quite time-consuming. Even with the success of the topology layout, connectivity requirements among blocks inevitably add unwanted clustering that in turn substantially increases the total grid count. Unstructured mesh generation on the other hand is free of these issues. An unstructured mesh also makes grid adaptation easier to implement because it does not have topological constraints. To maintain high temporal accuracy using unstructured meshes, several high-order numerical algorithms have been investigated in recent years. The discontinuous Galerkin (DG) method formulated by a combination of the weak form and discontinuous cell-interface integrals to allow jump solutions has been actively pursued.⁵⁻¹⁰ The spectral difference method¹¹⁻¹² also offers an alternative way to attain high-order accuracy for unstructured meshes. While most of these high-order methods offer different formulations to capture shocks and treat viscous terms, they follow a similar strategy to calculate cell-interface fluxes via an approximate 1D Riemann solver. Special attention is often required to deal with multi-dimensional flows with shocks.

The space-time conservation element, solution element (CESE) method introduced by Chang¹³ in the mid 1990s offers a distinctly different numerical framework to solve general conservation laws via a consistent formulation over the space-time domain. Unlike conventional methods following the 1D Riemann approximation concept, the CESE method is a genuine multi-dimensional scheme formulated by enforcing space-time flux conservation in the discretized domain. Integration of the conservation laws is carried out on staggered discretized volumes called conservation elements (CE), whose boundaries encompass only the smooth, well-defined regions defined by a solution element (SE). Consequently, no interface reconstruction or dimensional-splitting is needed. Flux conservation in space and time offers distinctly better numerical properties, both in terms of boundary condition treatment and temporal accuracy. It has been shown that this relatively newer numerical framework handles waves and shocks with accuracy comparable to existing high-order methods.¹⁴⁻¹⁵ The CESE method consists of a family of numerical schemes for the solutions of general conservation laws. These schemes form the foundation of the CESE numerical framework to be discussed in this paper.

With the wide availability of parallel supercomputers, numerical simulations of flow physics will be focused on time accurate computations with multiple time scales and physics. This new demand stems from the fact that most outstanding flow problems involve unsteady governing equations along with a large disparity in time scales in different parts of the flowfield. While modeling such intricate physical phenomena remains important and in fact, is still the only feasible way for many cases, direct time-accurate computations play an increasingly important role. This paper is focused on developing unstructured mesh based numerical algorithms and computer software framework with high temporal accuracy for unsteady flow simulations. To this end, numerical algorithm studies are carried out in this research to provide high-fidelity, cross-cutting numerical simulations for a variety of flow regimes via two main approaches: 1) time-accurate local time stepping and 2) high-order CESE method. Both investigations are part of the continual development of the NASA Langley Research Center's unstructured-mesh Navier-Stokes solver *ez4d* as part of the research effort under the Revolutionary Computational Aerosciences (RCA) subproject of the Fundamental Aeronautical Program (FAP).

Time-accurate numerical computations can be done with either implicit or explicit methods. Explicit methods have a time step constraint with a CFL number around one as a direct outcome of the finite-size domain of dependence. Implicit methods, on the other hand, maintain numerical stability at a much larger CFL number. Depending on the time scale of the flow physics, implicit methods may or may not offer

higher computational efficiency. For low speed problems with a time scale much larger than the acoustic wave time scale, implicit methods are much more efficient. For supersonic flows, the instability waves and near wall structures have a time scale comparable to or even smaller than the acoustic wave scale. Accuracy requirements thus dictate a small time step around a CFL number of one. Many direct numerical simulations of high-speed flows are performed with an explicit time-marching algorithm. For these kinds of problems, an implicit method may still be preferred due to its robustness but not accuracy considerations. The CESE framework is formulated as an explicit method and the stable CFL limit is one, similar to other widely used explicit methods for unstructured meshes. For the CESE method, numerical dissipation increases as the CFL number decrease.¹³ For a computational domain with multiple temporal-spatial scales, a constant time step results in a large disparity in CFL number. Regions with large grid size become less accurate if a constant time step is used throughout the domain. A time-accurate local time-stepping algorithm with a uniform CFL number while preserving time-accuracy could significantly improve both numerical accuracy and computational efficiency. Parallel to the development of local time-stepping, a high-order CESE method could also increase solution accuracy for time-accurate computations with multiple scales. Numerical algorithm advancement in these two areas is aimed at providing a more accurate flow solver for high-fidelity flow physics simulations.

The time accurate local time-stepping implementation in the full 3D Navier-Stokes CESE solver follows the concept proposed by Chang et al.¹⁶ for one-dimensional flows. The key idea is to derive the local time-stepping algorithm by enforcing flux conservation both in space and in time. Since the CESE method is built upon flux conservation over the entire space and time domain, the implementation of different time steps at different cells is relatively straightforward. A book keeping procedure similar to what was done by Yen¹⁷ is adopted. The time steps across different cells are divided in a set of pre-determined discrete values. Depending on the time steps of the neighboring cells, the solutions are stored at multiple levels. These stored solutions are then used for the flux integration along the CE interfaces to make sure flux conservation is enforced at all times. For the high-order CESE development, the method proposed by Chang¹⁸ is extended for 3D Navier-Stokes equations. In contrast to conventional high-order DG methods, the high-order CESE schemes proposed by Chang maintain a CFL limit of one regardless of the order of accuracy. In contrast, the stable CFL number limit decreases as the order of polynomials increases for explicit DG methods. For example, the stable CFL limit is only 0.1 for the p4 scheme.¹⁹ A CFL limit of one for the high-order CESE method offers much relief from this stringent time step limitation as compared to other explicit methods. The implementation of Chang's high-order CESE method by Bilyeu et al.²⁰⁻²¹ verifies the high CFL limit for 2D Euler equations. They also show that it is possible to construct higher than the 4th-order accurate CESE schemes for 1D wave equations. Both time accurate local time-stepping schemes and the 4th-order accurate CESE framework are implemented in the *ez4d* code for large-scale 3D Navier-Stokes Computations. Numerical algorithms implemented are discussed in the next section, followed by results and discussion. Some concluding remarks are included at the end of this paper.

II. Numerical Method

Three-dimensional general conservation laws can be written as

$$\frac{\partial U}{\partial t} + \nabla \cdot F = S \quad (1)$$

where U is the dependent variable, F is the flux vector, and S is the source term. Both F and S are functions of dependent variables. For the special case of compressible Navier-Stokes equations, the dependent variables are $U = (\rho, \rho u, \rho v, \rho w, e)$ and the flux vectors $F = (F_x, F_y, F_z)$ satisfy the following differential equation

$$\frac{\partial U}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} = 0 \quad (2)$$

where x , y , z , and t represent spatial coordinates and time, respectively. Flow variables ρ , u , v , w , and e represent density, three velocity components, and total energy ($e = p/(\gamma - 1) + \rho(u^2 + v^2 + w^2)/2$, where p denotes the pressure), respectively. Definitions of the total flux vectors (including an Euler part and a viscous part) F_x , F_y , and F_z , can be found in standard text books and will not be included here. The source vector, S , contains all external forcing or other energy-related source terms; and it vanishes for compressible Navier-Stokes equations. To close the system, the perfect gas relation, $p = \rho RT$, with T representing the temperature and R the gas constant, is used in conjunction with eq. (1).

Numerical solutions of eq. (1) can be obtained either by a differential or integral form. The integral form is preferred for fluid dynamics equations because of two main reasons. First, the integral form enforces flux conservation in the discretized space. Second, numerical issues associated with derivatives across flow discontinuities can be treated consistently only in the integral form. Most modern CFD solvers are formulated in the integral form but with distinct differences in the way eq. (1) is integrated. Most existing upwind or DG formulations treat the temporal derivative in eq. (1) as a separate identity and a finite-difference form is assumed for temporal derivatives. As a result, flux conservation is only enforced in the spatial domain. In the discretized space, this is equivalent to introducing a temporal source term in the conservation law. This source term depends on temporal discretization as well as the time step employed. While this in general works for solving steady state, large time asymptotic problems because the unsteady term vanishes at large time, undesired numerical issues may arise for time-accurate computations. One major issue is the error due to stiffness could accumulate after long time integration. The boundary conditions must also be treated consistently so that any error or reflection would not produce spurious waves that propagate into the computational domain.

The other issue for most existing finite volume or DG formulations is associated with the Riemann solver used to render the flux vector unique at cell interfaces. From the fundamental standpoint, formulations for computing flux vectors at the presumed discontinuous cell interface are ad hoc at best. Correct (and thus unique) flux vectors at the flow discontinuity can only be approximated accurately for any multi-dimensional cell interfaces if and only if the jump solution is known a priori. Any one-dimensional based approximation breaks down when the cell interface is not aligned with the discontinuity. For this reason, dimensional splitting along with grid adaptation is often needed to capture shocks. Treating shock waves at the cell (and flux-integration) interface is apparently departing from fundamental gas dynamics. In it, the flux vector is integrated before and after the discontinuity. A control volume contains the shock with the boundaries only in the smooth region is used to derive correct jump conditions. While such practices are forgotten or neglected by most CFD algorithms, the space-time CESE method is perhaps the only method that follows similar concept of flux conservation, i.e., the conservation or control volume should be without any discontinuities at the boundaries like most finite-volume family of schemes.

Following the above-mentioned fundamental concepts, the space-time CESE method is formulated by the integral form of the conservation laws with a space-time flux \vec{h} , is defined as:

$$\vec{h} = (F_x, F_y, F_z, U) \quad (3)$$

Based on above-mentioned physical consideration, general conservation laws should be rewritten in the following integral form

$$\oint_{\sigma(V)} \vec{h} \cdot d\vec{s} = \int_V S dV \quad (4)$$

where the space-time flux vector is integrated over the surface Ω of an arbitrary space-time volume V . Strictly speaking, the differential form, eq. (1), is only valid for smooth solutions. The surface normal vector is defined by $d\vec{s} = \vec{n} dA$, where dA is the area increment on Ω , and \vec{n} is the outward unit normal vector. Equation (4) is quite general, and in fact, any conservation law with or without a source vector can be cast in this form. Thus, the numerical algorithm devised for eq. (4) can be easily extended to other physical problems. Theoretically, the space-time domain can be discretized with time being treated as one coordinate. With this unstructured or structured temporal-spatial mesh, a time accurate solution can be obtained by integrating eq. (4) using the CESE numerical framework. However, this is not done in most of

the CESE implementations due to topological difficulties in generating a four-dimensional mesh for time-accurate 3D problem. Alternatively, the time coordinate is organized in a structured, layered fashion with a constant increment for either structured or unstructured meshes.

Given a discretized space-time domain with an unstructured mesh, the dependent variables can be assumed to vary according to the following Taylor series expansion truncated to the third order,

$$\begin{aligned}
U(x, y, z, t) = & U_0 + U_t(t-t_0) + U_x(x-x_0) + U_y(y-y_0) + U_z(z-z_0) \\
& + U_{xx}(x-x_0)^2/2 + U_{xy}(x-x_0)(y-y_0) + U_{yy}(y-y_0)^2/2 + \\
& U_{xz}(x-x_0)(z-z_0) + U_{yz}(y-y_0)(z-z_0) + U_{zz}(z-z_0)^2/2 + \\
& U_{xt}(x-x_0)(t-t_0) + U_{yt}(y-y_0)(t-t_0) + U_{zt}(z-z_0)(t-t_0) + U_{tt}(t-t_0)^2/2 + \\
& U_{xxx}(x-x_0)^3/6 + U_{yyy}(y-y_0)^3/6 + U_{zzz}(z-z_0)^3/6 + U_{ttt}(t-t_0)^3/6 + \\
& U_{xxy}(x-x_0)^2(y-y_0)/2 + U_{xyy}(x-x_0)(y-y_0)^2/2 + U_{xxz}(x-x_0)^2(z-z_0)/2 + \\
& U_{xzz}(x-x_0)(z-z_0)^2/2 + U_{yyz}(y-y_0)^2(z-z_0)/2 + U_{yzz}(y-y_0)(z-z_0)^2/2 + \\
& U_{xxt}(x-x_0)^2(t-t_0)/2 + U_{xyt}(x-x_0)(y-y_0)(t-t_0) + U_{yyt}(y-y_0)^2(t-t_0)/2 + \\
& U_{zxt}(z-z_0)^2(t-t_0)/2 + U_{yzt}(y-y_0)(z-z_0)(t-t_0) + U_{xzt}(x-x_0)(z-z_0)(t-t_0) + \\
& + U_{xyz}(x-x_0)(y-y_0)(z-z_0) + U_{xxt}(x-x_0)(t-t_0)^2/2 + U_{yyt}(y-y_0)(t-t_0)^2/2 + \\
& U_{ztt}(z-z_0)(t-t_0)^2/2
\end{aligned} \tag{5}$$

within each solution element (SE), which roughly coincides with the discretized cell element. The coordinate (x_0, y_0, z_0) is a reference point within an SE called solution point in the CESE method. The coefficients of the expansion are not equal but analogous to their counterparts in derivatives. These coefficients are unknowns to be solved for at each element. Dependent variables vary smoothly within each solution element. Jumps are allowed across solution elements. The piecewise smooth assumption allows flow discontinuities to be captured while retaining the designed approximations for each element. To solve for unknown coefficients, the conservation laws, eq. (4), must be integrated numerically in the computational domain. Most high-order unstructured mesh methods carry out numerical integration along the element interfaces (which coincides with the cell boundaries) only in the spatial domain. To handle jumps across these element interfaces, numerical approximations must be employed to determine unique fluxes based on solutions from either side of the interface. In contrast, for the CESE method, integration of eq. (4) is not carried out along the boundaries of the solution element, but along a staggered temporal-spatial conservation element (CE). The conservation elements are staggered with existing solution elements such that all CE interfaces only pass through the smooth region of any solution element. Integrating fluxes across a flow discontinuity, similar to what is done to derive Rankine-Hugoniot relations, poses no numerical issues at all and no ad hoc Riemann solver is needed. Figure 1 shows the CE arrangement around a triangular element. For two-dimensional CESE solvers, one quadrilateral CE is constructed for each neighboring element and for 3D tetrahedral elements, the CE for each neighbor is a double pyramid. Flux integration using eq. (4) around all surrounding CEs results in surface integrals on three different group of faces: i) top faces ($GBAF$, $GFED$, and $GDCB$), ii) bottom faces ($G'B'A'F'$, $G'F'E'D'$, and $G'D'C'B'$), and iii) side faces (all remaining faces). Bottom and side face integrals depend only on solutions from previous time level. Top face integration requires dependent variables at the new time level.

If the Taylor series approximations shown in eq. (5) are truncated to first order, the resulting numerical method is formally second-order accurate. In the CESE numerical framework, the solution point of each SE is located at the geometric center of all surrounding CEs (the union of $GBAF$, $GFED$, and $GDCB$). The solution procedure for the second-order accurate schemes involves three steps.

- S1. Integration of eq. (4) can be simplified to algebraic equations containing only unknowns U_o , at the new time level. Due to the choice of solution point, all derivatives (U_x , U_y , and U_z) vanish in these equations. Solutions of these algebraic equations can be easily calculated for all elements without any matrix inversion.
- S2. Coefficients U_x , U_y and U_z for each element are calculated by using generalized finite difference formulations for unstructured neighbors. For each neighbor, a set of derivative coefficients is calculated by solutions around each edge.¹⁵ A weighted average procedure in favor of smaller values (discussed below) is used to determine a set of unique derivative coefficients for each element.
- S3. Update temporal derivative, U_t , by using the original governing equations. This completes one time iteration.

Procedure S2 involves spatial derivatives calculations using finite-difference formulations. Unlike conventional structured-mesh algorithms, these spatial derivatives are unknowns to be determined at each time step in the CESE method. Theoretically, these spatial derivatives can be calculated for each solution element by solving flux-conservation equations for each quadrilateral CE. These theoretical base line solutions for spatial derivatives form the non-dissipative a-scheme and numerical dissipation can be added by modifying them.¹³ Several methods can be used to add numerical dissipation including the c-scheme,¹³ Courant number insensitive scheme (CNI),¹⁸ and the edge-based derivative (EBD) method.¹⁵ For highly non-uniform or high-aspect ratio meshes, the EBD method has been shown to produce the most robust results.

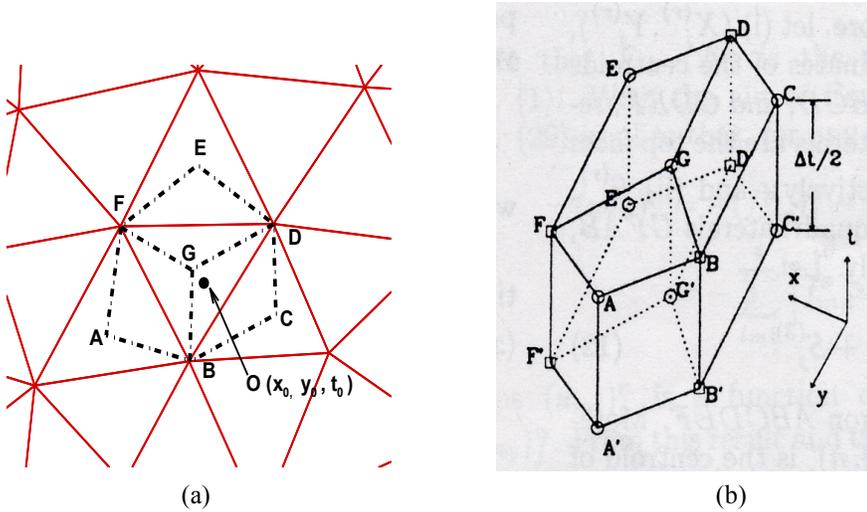


Figure 1. Conservation elements for a 2D triangular mesh: (a) top view, (b) space time CE volume.

For an unstructured mesh, solutions at three points are needed to determine U_x and U_y in a two-dimensional domain and four points are needed for U_x , U_y , and U_z in a three-dimensional domain. For a triangular element with a solution point O and three vertices, V_1 , V_2 , and V_3 , three sets of derivatives based on three edges can be computed. As discussed in Ref.,¹⁵ these derivatives form an equivalent of the non-dissipative scheme. Numerical dissipation can be added by growing the triangles to new vertices V'_1 , V'_2 , and V'_3 with a stretching parameter σ according to

$$V'_i = O + \sigma(V_i - O) \quad (6)$$

for $i = 1, 2$, and 3 . The parameter, σ , has a value greater than one. Larger σ implies more numerical dissipation. For the edge-based CNI scheme, the σ value decreases with the local CFL number. The optimum solution accuracy can be obtained with a σ value small enough to maintain numerical stability. Numerical dissipation control can thus be achieved by decreasing the σ value for large cells. For high

aspect ratio meshes, the value of σ has to be increased significantly to maintain numerical stability. The shape of the triangle affects both solution accuracy and numerical stability. For an elongated cell, the edge length can vary by orders of magnitude. The shortest side requires a very large σ (in the order of $10^3 - 10^5$, depending on the aspect ratio) to maintain stability. Stretching by the same ratio for the longest side would add excessive amount of numerical dissipation. The combined effect often causes inaccurate viscous solution near the wall. Better solution accuracy could be obtained while maintaining numerical stability by using a different σ for different vertices. Thus, eq. (6) is modified to

$$V'_i = O + \sigma_i (V_i - O) \quad (7)$$

The stretching parameter, σ_i is determined by

$$\sigma_i = \max(\sigma_{\min}, \sigma \Delta_{\min} / \|V_i - O\| + 1) \quad (8)$$

Additional parameters are introduced to fine tune the control. The global minimum spacing between any two vertices is denoted by Δ_{\min} and σ_{\min} is a pre-set lower limit of the stretching parameter. The value σ in eq. (8) is a constant within an element. Its value can be chosen as a global constant or could be a simple function of grid sizes (for example, $\sigma = \text{CFL} \sigma_{\max}$, where σ_{\max} is the maximum stretching desired). For a high-aspect ratio mesh, a very large value of σ would increase σ_i significantly only for the short side (smaller $\|V_i - O\|$ values). Numerical experiments indicate that the use of eqs. (7) and (8) allows accurate solutions of wall shear stress for a flat-plate boundary layer with an aspect ratio as high as 10^5 . For a relatively uniform mesh, $\sigma < 2$ is sufficient to maintain both accuracy and stability. For a high-aspect ratio mesh, on the other hand, σ needs to be increased to 10^3 or higher. Nevertheless, such a high value is only applied at the vertices with a very small spacing.

In general, more numerical dissipation (larger σ) is needed in the vicinity of flow discontinuities. A shock detection scheme helps locate the flow discontinuities. More numerical dissipation can thus be added accordingly. The shock sensor proposed by Ref.²⁵ is used:

$$\delta_p = \nabla p \cdot (M\vec{V}) \quad (9)$$

where M is the local Mach number and \vec{V} is the local velocity vector. If δ_p is greater than a threshold value, the element is considered to contain a shock; and proper numerical dissipation can be added accordingly. Using non-dimensional values (normalized by freestream values), the value of δ_p is greater than 10 in the vicinity of shocks generally, although this threshold value varies from problem to problem. The use of the above shock sensor avoids adding excessive numerical dissipation away from the shock. For the high-order CESE method, it can also be used to reduce approximation orders around flow discontinuities.

For triangular and tetrahedral elements, each neighbor provides sufficient information to construct all spatial derivatives. Suppose there are n derivatives around an element. The n value varies from 3 for a triangle to 12 for a hexahedral element. The weighted averaging procedure by Chang²⁶ is used. The process begin by first calculating

$$\theta_i = \sqrt{(U_x^i)^2 + (U_y^i)^2 + (U_z^i)^2 + \varepsilon}$$

where superscript i indicates the i -th derivative and $\varepsilon = 10^{-20}$ is added to avoid zero derivatives. Sort all θ_i and define ζ_i as the sorted array arranged from the largest to the smallest value (i.e., $\zeta_1 = \max(\theta_i)$).

$$\psi_i = \prod_{j=1}^{n-1} \zeta_j / \prod_{j=1, j \neq i}^n \theta_j - 1$$

By definition, all ψ_i are positive numbers. The weighting factor for the i -th derivative is then defined as

$$w_i = (1 + \alpha / \sqrt{\text{CFL}} \psi_i) g_i \quad (10)$$

where CFL is the local CFL number at each element and α is a global parameter with a range from 0 to about 2. Larger α values imply larger numerical dissipation. For acoustic wave simulations, a very small α value can be used to increase the phase accuracy. For shock capturing, the α value can be set to 1 or 2.

The parameter g_i accounts for the geometrical effect of different neighboring cell sizes. The CE volume can be assigned to g_i such that larger CE carries more weight in the derivative calculations. The unique derivatives at each element can be then calculated by

$$\begin{aligned} U_x &= \sum_{i=1}^n w_i U_x^i / \sum_{i=1}^n w_i \\ U_y &= \sum_{i=1}^n w_i U_y^i / \sum_{i=1}^n w_i \\ U_z &= \sum_{i=1}^n w_i U_z^i / \sum_{i=1}^n w_i \end{aligned} \quad (11)$$

The above weighted average procedure is used universally for all the computations with or without discontinuities. No adjustment or fine-tuning of the weighted averaging is needed to handle bow shocks or even flows with very complex shock patterns. This weighted averaging procedure is in a way similar to conventional flux limiters used in upwind formulations. Nonetheless, the weighted average procedure in the CESE framework does not alter flux vectors directly. The weighted derivative coefficients calculated simply modify numerical dissipation near flow discontinuities to maintain numerical stability.

The second-order CESE methods have been used for steady-state as well as time-accurate computations extensively. To further improve numerical efficiency and solution accuracy without significantly increasing the total grid counts, the time-accurate local time-stepping and the high-order CESE methods discussed below have been investigated in this paper.

Time-accurate Local Time-stepping CESE Method

In the initial stage of the CESE method development, an ad hoc local time stepping has been implemented for steady state computations. In this attempt, each element is advanced at a different time step satisfying the stable CFL constraint ($CFL < 1$). Flux conservation in time is not preserved. It was found that for simple flow geometries such as flat-plate boundary layers, this ad hoc local time-stepping procedure does yield rapidly converging and accurate steady-state solutions. However, for flows with strong shocks such as blunt body configurations, the calculated shock stand-off distance was wrong compared to exact solutions or solutions obtained by constant time step. This is not surprising because when flux conservation is not preserved, there is no guarantee for correct steady-state solutions. Chang et al.,¹⁶ propose a time-accurate local time stepping framework for 1D flows by using interface patches. In this formulation, flux conservation in both space and time is ensured by book keeping solutions at multiple time steps. It has been shown that solution accuracy is preserved even with a large step size jump across the cell interface. In this investigation, we follow similar procedure to implement time-accurate local time-stepping (TALTS) for multi-dimensional flows.

The fundamental concept of TALTS is based on space-time flux conservation. Conservation in the spatial domain alone does not guarantee flux conservation in time, a key ingredient for temporal accuracy with disparities in time steps. The CESE method is constructed based on flux conservation in the temporal-spatial domain; and, thus is fully equipped to implement TALTS. The following algorithms are taken to implement TALTS in the Navier-Stokes code:

- L1. Sorting time steps: local time steps are first calculated based on a prescribed CFL number (normally between 0.5 and 1). Assume the smallest time step found is Δt , an array of time steps, $f(k) = 2^k \Delta t$ is constructed. The local time step for each element is adjusted to the nearest $f(k)$ value. After the adjustment, time steps vary as a power of 2 throughout the entire domain. For a relatively uniform mesh, the maximum k value needed is no more than 3 or 4. For viscous flow computations, the maximum k value could be as large as 10 or more. A maximum k value of 10 amounts to three orders of magnitude variation in time steps (or element sizes). RANS calculations could require a maximum k of 15 or more.

- L2. Determining solution levels: base on the time steps of neighbors, each element must determine how many solution levels must be stored for flux integration. Normally 2 or 4 levels of solutions need to be stored. If the cell size jumps abruptly, more levels are required. For a smooth mesh, no more than two levels are needed.
- L3. Integrating flux with patches: for cells with smaller time-step neighbors, flux integration on the side faces must be done with patches. Solutions stored with multiple time levels are used to construct flux vectors on the divided side-face patches. Summing flux vectors for all the patches completes the side-face integration to ensure flux conservation in time and space.
- L4. Timing iterations: a clock to monitor physical time is added for time marching. An element will march to the next time level only when the local cell time after the iteration matches the current physical time. Caution must also be taken to avoid advancing time before the neighbors catch up.
- L5. Communicating multiple solution levels: for parallel computations, the intermediate solutions must be communicated to neighboring domains for accurate flux integrations.

The baseline algorithm in the *ez4d* code has been modified slightly for viscous flow computations.¹⁵ The original strong flux conservation formulation in conjunction with a modified edge-base derivatives approach is used to effectively compute viscous flow with highly stretched unstructured meshes inside the boundary layer. Solution accuracy has been carefully validated for time-accurate inviscid and viscous flows. The *ez4d* code is developed in the C++ language entirely by following modern software engineering procedures. Object-oriented and generic programming using templates have been employed extensively in the software framework for *ez4d*. Sound software architecture ensures easy maintenance and agile future development. Both message passing interface (MPI) and multi-thread capabilities are integrated in the software framework to take advantage of multi-core, distributed memory cluster environments. Currently, the *ez4d* code can solve Euler or laminar Navier-Stokes equations for triangular, quadrilateral, tetrahedral, or hexahedral unstructured meshes.

The 4th-Order Accurate CESE method for 3D Navier-Stokes Equations

The implementation of the 4th-order CESE method for 3D Navier-Stokes equations follows the theoretical framework introduced by Chang for 1D scalar equations¹⁸. Traditionally, high-order methods are implemented by increasing the order of the truncated Taylor series representation of a solution element. These high order polynomials are used for the integration of the governing equations. While the treatment of integrations varies from finite volume, spectral element, spectral differences to DG, the unknown high-order coefficients are obtained by solving a matrix equation for all flow dependent variables. Increasing the order of the polynomials is equivalent to sub-dividing an element into smaller sub-elements. The additional polynomial coefficients, or quadrature points, decrease the size of the domain of dependence. As a result, the allowable time step decreases significantly as the order of polynomials increases. For example, the maximum CFL number is only around 0.1 for a typical 4th-order DG method.¹⁹ The reduced time step size partially offsets the increasing accuracy of the solutions with fewer elements. To avoid such a stringent time step requirement, an implicit integration technique can be used. However, such implicit method is difficult to implement for unstructured meshes and only works for steady-state problems. The 4th-order accurate CESE method introduced by Chang, on the other hand, preserves the maximum CFL number around 1 for explicit formulations. The formulation of this 4th-order scheme for Navier-Stokes equations will be discussed below with emphasizes on multi-dimensional and general flux formulations.

The fundamental building block for preserving the CFL limit in the high-order CESE method is the introduction of additional equations for higher derivatives. For a 4th-order scheme, equations for the second derivatives of the dependent variables can be obtained by differentiating the governing equations twice. Six second-derivative equations for Navier-Stokes equations can be derived as follows.

$$\begin{aligned}
\frac{\partial U_{xx}}{\partial t} + \frac{\partial F_{xxx}}{\partial x} + \frac{\partial F_{xxy}}{\partial y} + \frac{\partial F_{xxz}}{\partial z} &= 0 \\
\frac{\partial U_{xy}}{\partial t} + \frac{\partial F_{xxy}}{\partial x} + \frac{\partial F_{xyy}}{\partial y} + \frac{\partial F_{xyz}}{\partial z} &= 0 \\
\frac{\partial U_{yy}}{\partial t} + \frac{\partial F_{xyy}}{\partial x} + \frac{\partial F_{yyy}}{\partial y} + \frac{\partial F_{yyz}}{\partial z} &= 0 \\
\frac{\partial U_{xz}}{\partial t} + \frac{\partial F_{xxz}}{\partial x} + \frac{\partial F_{xyz}}{\partial y} + \frac{\partial F_{xzz}}{\partial z} &= 0 \\
\frac{\partial U_{yz}}{\partial t} + \frac{\partial F_{xyz}}{\partial x} + \frac{\partial F_{yyz}}{\partial y} + \frac{\partial F_{yzz}}{\partial z} &= 0 \\
\frac{\partial U_{zz}}{\partial t} + \frac{\partial F_{xzz}}{\partial x} + \frac{\partial F_{yzz}}{\partial y} + \frac{\partial F_{zzz}}{\partial z} &= 0
\end{aligned} \tag{12}$$

For simple conservation laws such as the linear wave equation, the second-derivative equations take an identical form as the original governing equation. To solve these second-derivative equations, additional space-time flux vectors are introduced:

$$\begin{aligned}
\vec{h}_{xx} &= (F_{xxx}, F_{xxy}, F_{xxz}, U_{xx}) \\
\vec{h}_{xy} &= (F_{xxy}, F_{xyy}, F_{xyz}, U_{xy}) \\
\vec{h}_{yy} &= (F_{xyy}, F_{yyy}, F_{yyz}, U_{yy}) \\
\vec{h}_{xz} &= (F_{xxz}, F_{xyz}, F_{xzz}, U_{xz}) \\
\vec{h}_{yz} &= (F_{xyz}, F_{yyz}, F_{yzz}, U_{yz}) \\
\vec{h}_{zz} &= (F_{xzz}, F_{yzz}, F_{zzz}, U_{zz})
\end{aligned} \tag{13}$$

For compressible Navier-Stokes equations, a total of six conservation laws listed below need to be integrated at each time level in addition to eq. (4).

$$\begin{aligned}
\oint_{\sigma(V)} \vec{h}_{xx} \cdot d\vec{s} &= 0 \\
\oint_{\sigma(V)} \vec{h}_{xy} \cdot d\vec{s} &= 0 \\
\oint_{\sigma(V)} \vec{h}_{yy} \cdot d\vec{s} &= 0 \\
\oint_{\sigma(V)} \vec{h}_{xz} \cdot d\vec{s} &= 0 \\
\oint_{\sigma(V)} \vec{h}_{yz} \cdot d\vec{s} &= 0 \\
\oint_{\sigma(V)} \vec{h}_{zz} \cdot d\vec{s} &= 0
\end{aligned} \tag{14}$$

The Mathematica^{®27} commercial software is used in this paper to derive all high-derivative governing equations and to generate corresponding C++ code to be included in the *ez4d* code. For 3D flows, six double derivatives, U_{xx} , U_{xy} , U_{yy} , U_{yz} , U_{xz} , and U_{zz} , need to be solved in addition to U . The solution procedure for the 4th-order scheme can be summarized as follows:

- H1. Solving second derivatives: using eq.(14) and the procedure S1 used for the second-order method, all six second derivatives are calculated for each element.

- H2. Computing third spatial derivatives: using the computed second derivatives and the procedure S2 for the second-order method, third spatial derivatives are computed for each element.
- H3. Integrating high-order flux equations: using the computed second and third derivatives, solution U_0 can be computed by integrating eq.(4) by using eq.(5). Note here that with the selection of the solution point being the geometric center of the combined CEs, the first derivatives still vanish in the top faces flux integrals. Second and third derivatives on the top faces are already determined in H1 and H2. This results in a simple and easy way to solve algebraic equations for U_0 .
- H4. Calculating first spatial derivatives: first derivatives are calculated using a procedure similar to S2 but with high-order finite difference formulations. The same weighted averaging as in the second-order scheme is applied (eq. (11)).
- H5. Updating temporal derivatives: using all spatial derivative coefficients computed, all derivatives with temporal differentials are calculated using the governing equations. This completes the iteration.

The high-order scheme requires some additional equations and geometric information to be computed. Temporal partial derivatives (calculated in H5) are obtained by differentiating the governing equations, for example,

$$U_{tt} = \frac{\partial F_{xt}}{\partial x} + \frac{\partial F_{yt}}{\partial y} + \frac{\partial F_{zt}}{\partial z} \quad (15)$$

All corresponding derivatives on the right-hand side can be obtained by analytical expressions by using the derivative coefficients defined in eq. (5). Again, all these temporal partial derivatives are derived by using the Mathematica® software to avoid errors. In procedures H3 and H4 above, the integration of the top and bottom faces requires high-order moments associated with integrations of the second and third order terms in eq. (5). These high-order moments are pre-computed and stored for each CE with respect to the CE center before the iteration starts. There are total of seven moments required for 2D calculations and 16 for 3D geometries.

In the procedure H4, eq. (4) contains integrations of flux vectors over the side faces. These flux vectors are functions of the high-order polynomials defined in eq. (5). To maintain high-order accuracy, Jacobians of these flux vectors with respect to U , and all derivatives must be derived and used for flux integration over space and time if the flux vectors are evaluated as a function of polynomials directly. Deriving these Jacobian terms for 3D Navier-Stokes equations with variable viscosity involves tedious book keeping processes and is thus error-prone. In this paper, an alternative procedure is used. All high-order flux integrations on the side faces are carried out using quadrature integrals.²⁸ Flux vectors are calculated at quadrature points within the side face by using the high-order solution element already computed. The flux integral is then computed using high-order quadrature formulae. No Jacobian terms are needed in this approach. The nine-point Gaussian quadrature is used for 2D elements and a combined triangular 4-point quadrature in space and 3-point in time is used for 3D elements. These quadrature points are designed to carry out integrals over the quadrilateral and triangular prism side faces, respectively.

III. Results and Discussion

The time-accurate local time-stepping and the 4th order CESE methods discussed above are aimed at providing higher-fidelity numerical solutions for unsteady flow computations. Several test cases are investigated in this paper to assess the effectiveness of both approaches. In the next two subsections, these test cases are described in detail.

Time-Accurate Local Times-stepping for Unsteady Flows

The TALTS method is designed to handle large grid size disparities in a discretized domain. One frequently encountered problem is to simulate acoustic waves propagating through a viscous mesh. Computing acoustic waves using such a non-uniform mesh is quite challenging. Figure 2 shows a

rectangular domain with a clustered mesh near the bottom boundary. A constant time step computation results in significant phase error due to the large CFL number disparities. The wave amplitude is also incorrectly damped. In contrast, the TALTS computation maintains a rather uniform CFL number (around 0.8) and thus results in much improved accuracy in both amplitude and phase throughout the domain.

Flux conservation plays a crucial role in shock capturing schemes. It is well known that failing to conserve the flux in the numerical computations could result in a wrong solution. An ad hoc local time-stepping method uses different time steps at different cells and flux conservation in time is violated. Conventional schemes use local time stepping for steady-state flow computations in which the residual norm is driven to a small value at convergence. For the CESE method, such an approach may work differently for steady state computations. The violation of flux conservation in time persists for long time integration. As a result, the converged solutions only guarantee a balance between temporal and spatial fluxes and may not preserve the desired flux conservation at large time. One example investigated here is a Mach 6 flow over a cylinder computed with a typical wall-clustered viscous mesh. The mesh has a maximum aspect ratio of about 10^3 and tenth-power local time steps ($k = 10$) were used. Figure 3 shows the computed Mach number contours compared against inviscid shock-fitted solutions²⁹ using two different methods of time marching. The ad hoc local time-stepping results clearly over-predict the shock stand-off distance. The TALTS results on the other hand correctly compute the shock locations. The Mach number contours in the inviscid region also agree well with shock-fitting Euler solutions. The ad hoc local time-stepping method, despite its drawbacks in shock capturing, handles smooth steady-state solutions with good accuracy. It also has the characteristics of quick convergence to large-time asymptotic solutions. The best strategy for steady-state computations is to first run ad hoc local time-stepping to quickly settle with a nearly correct solution. The TALTS method is then used to converge the solution to the correct large-time asymptotes.

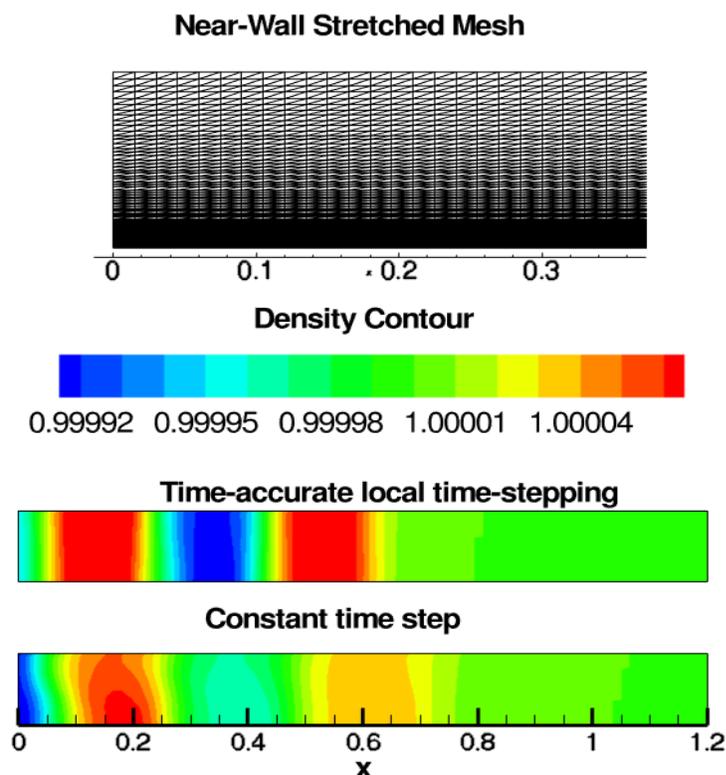


Figure 2. Acoustic waves propagating through a domain with clustered mesh, showing density contours at a time instant.

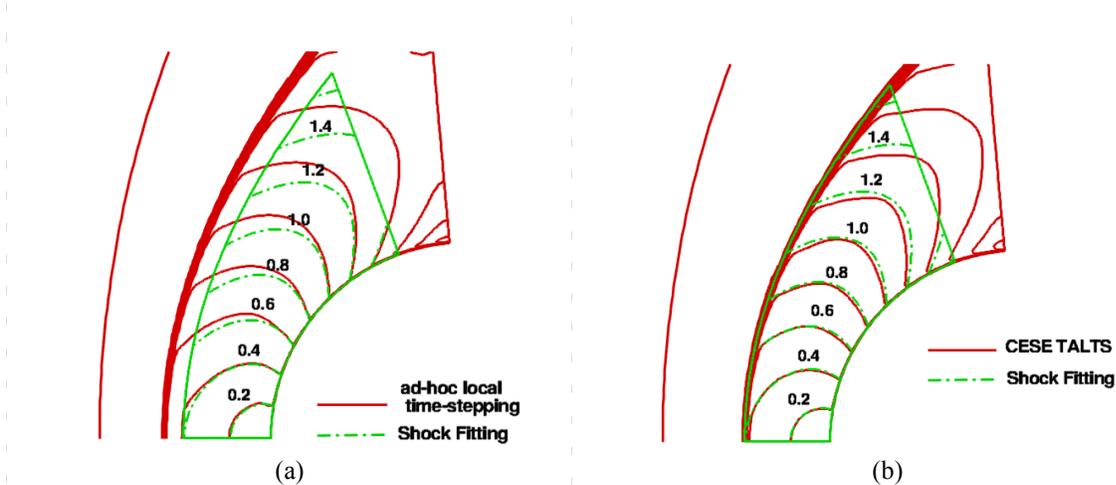


Figure 3. Mach 6 viscous flow over a cylinder with a wall-clustered viscous mesh, Mach number contours compared inviscid solutions obtained by shock-fitting method:²⁹(a) ad hoc local time-stepping (b) TALTS.

In more practical applications, a large mesh-size disparity is unavoidable. Flow physics dictates the required mesh size. The ideal mesh should have a grid distribution that is commensurate with the physical length scale. The development of the TALTS scheme is aimed at coping with a computational domain with various physical length scales. In the next test case, a cone with a counterflowing jet at the leading edge submerged in a Mach 1.6 free stream is computed. The shear layer and jet instability waves alter the leading edge shock. To accurately compute the flowfield for sonic boom applications, both the near-field instability waves and far-field shocks need to be captured with proper resolutions. To reduce the total grid counts, the mesh has to transition from the near field fine grid to the farfield coarse grid like that shown in Fig. 4. In the constant time step approach, the finest mesh near the jet exit determines the time step size used in the simulations by using the constraint of $CFL < 1$. In the farfield, the same time step implies a very small local CFL number. Numerical dissipation increases significantly when the CFL number decreases for the CESE method. The computed normalized pressure disturbance contours (based on the freestream pressure value) shown in Fig. 5(a) exhibit a twist near the grid transition region. Further downstream, the shock appears to be diffused due to excessive numerical dissipation. On the contrary, the computed solutions using TALTS displayed in Fig. 5(b) show consistent shock resolutions across both regions. In this particular example, the grid size varies several orders of magnitude. The TALTS simulations use $k = 11$, i.e., the ratio between the largest and the smallest time step is 2,048.

Another application of the TALTS method is to compute steady-state supersonic flow over a large blunt body as shown in Fig. 6. To resolve the viscous boundary layer and the strong bow-shock simultaneously, the mesh size varies by several orders of magnitude. The TALTS scheme can significantly improve the accuracy both in shock-capturing and resolving boundary-layer profiles. In this particular example, ninth-power time levels ($k = 9$) for the 32 million tetrahedral elements were used. Parallel efficiency for the TALTS schemes needs some careful consideration. The mesh in this example is decomposed into 120 blocks running in parallel using 120 cores on NASA Langley Research Center’s K-cluster with Sandy Bridge Intel CPUs. The average physical time required is about 1.38 seconds per time iteration. Similar computations using constant time step took about 4.2 seconds physical time per time iteration. This amounts to about a factor of 3 saving in computational time by using the TALTS method. The speed-up in computational efficiency is reasonable. However, load balancing is suboptimal because many blocks only contain relatively large cells. These blocks would spend significant amount of time waiting for other blocks with small time steps to catch up. To further improve parallel efficiency, the domain decomposition must be done with due consideration of equal grid size distribution. A “chunk” style grid with each chunk

including roughly equal amount of fine and coarse elements can significantly improve overall throughput for large-scale computations.

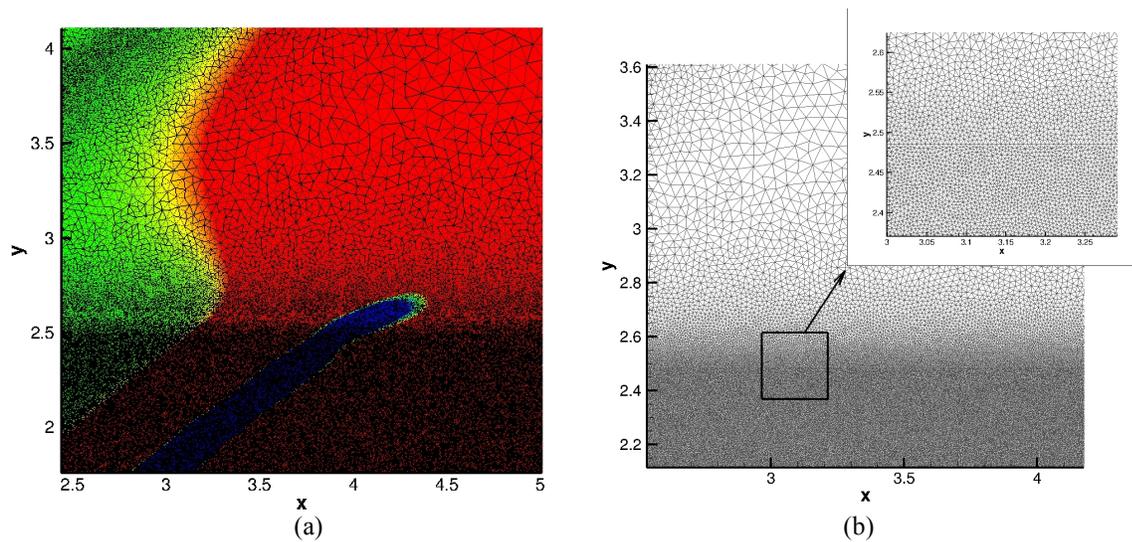


Figure 4. Mesh transition from fine near-field to coarse farfield used for sonic boom applications: (a) mesh interface between two zones (pressure contours levels the same as Fig. 5(a) below (b) enlarged view of the triangular mesh.

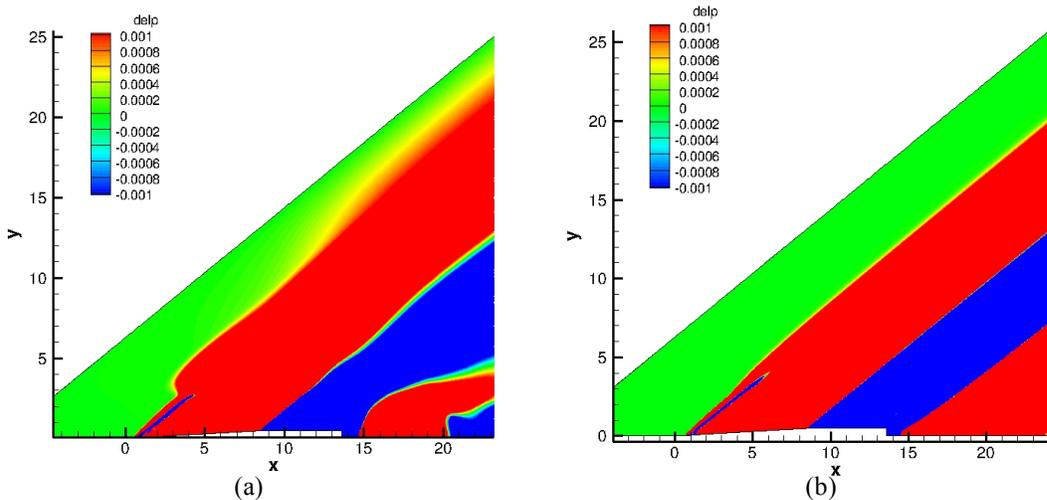


Figure 5. Computed pressure contours for a Mach 1.6 flow over a cone with a counterflowing jet located at $x = 1$: (a) constant time step (b) TALTS.

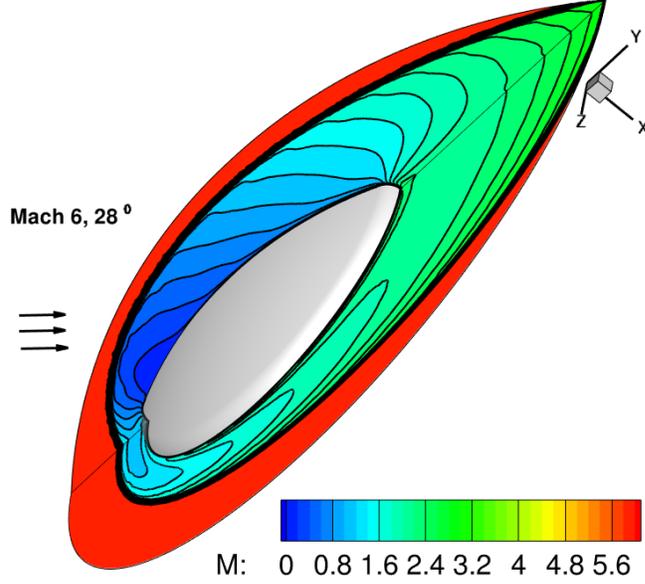


Figure 6. Computed Mach number contours at two planes for a Mach 6 flow over a blunt body with a 28° angle of attack, computed using TALTS.

The 4th-order CESE Method for Scalar Equations

To verify the order of accuracy for the 4th-order CESE numerical framework, the following scalar wave equation is used for extensive tests.

$$\frac{\partial u}{\partial t} + a \left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \right) = 0 \quad (16)$$

The corresponding flux vector is $\vec{h} = (au, au, au, u)$. Without loss of generality, the parameter value for a is set to be 1. Due to the simplicity of this equation form, the high-order equations (eq. (13)) for the above scalar equation take the same form as the original equation. Equation (16) allows a variety of exact solutions to be constructed for testing purposes. A polynomial solution

$$u = f(\theta), \quad \theta = t - \frac{x}{3} - \frac{y}{3} - \frac{z}{3} \quad (17)$$

is used for validation. For simplicity, the function is assumed to satisfy $f(\theta) = \theta^n$, where n is a positive integer. A $20 \times 20 \times 20$ cubic domain is discretized using four different meshes: 10^3 , 20^3 , 40^3 , and 80^3 . The resulting structured meshes are further sliced into tetrahedrons. The above exact solutions are imposed as initial and boundary conditions. Numerical dissipation parameters are fixed at $\alpha = 0$ and $\sigma = 1.5$. Theoretically, when the polynomial order satisfies $n < 4$, the 4th-order scheme should give exact solutions. And indeed, the computed solutions retained exact solutions with an error norm hovering around 10^{-11} - 10^{-15} for a large number of iterations. When $n = 4$, the error norm versus number of grid counts in each direction is shown in Fig. 7. As expected, the error norms converge at 4th-, 3rd-, and 2nd-order rate for U_0 , U_x , U_{xx} , and U_{xxx} , respectively. The error norms for y -derivatives converge at an identical rate.

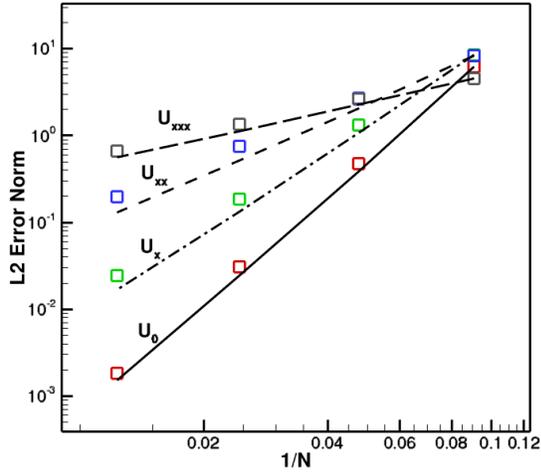


Figure 7. L2 error norms of U_0 and derivatives versus $1/N$ (N is the number of points in each direction) for $f(\theta)=\theta^4$ (Lines correspond to theoretical 4th-, 3rd-, and 2nd-order convergence).

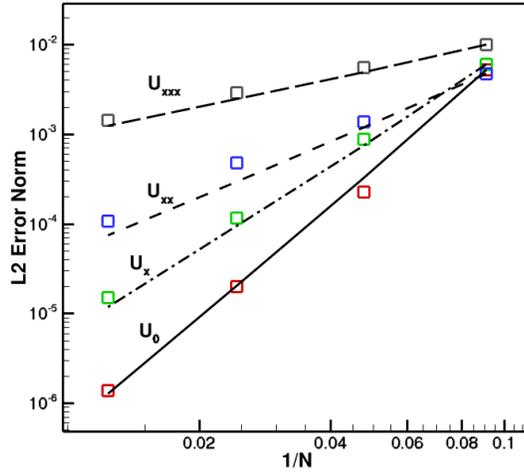


Figure 8. L2 error norms of U_0 and derivatives versus $1/N$ (N is the number of points in each direction) for $f(\theta)=A\sin(\theta)$ (Lines correspond to theoretical 4th-, 3rd-, and 2nd-order convergence).

The second test is done for an exact wave solution with the following form:

$$f(\theta)=A_w \sin(\theta), \quad \theta=\omega t-k_x x-k_y y-k_z z$$

The relation between the frequency and wave numbers is $\omega-k_x-k_y-k_z=0$. To fully test the multi-dimensional formulations, the wave angle is 45° with respect to the x -axis. The computed L2 error norms for dependent variables and their derivatives are shown in Fig. 8. Similar 4th-order convergence is verified for the wave solutions.

Acoustic wave propagation through a square 2D domain is used to validate the 4th-order accurate algorithms for 2D Euler equations. Isotropic triangular meshes as shown in Fig. 9(a) are used for accuracy studies. Four different mesh levels with 462, 1,880, 7,378, and 29,276 elements are used for computations. These meshes correspond to an average cell size decreases of factors 2, 4, and 8, respectively. Exact solutions of linear acoustic waves are imposed both as initial and boundary conditions. The wave propagates along the diagonal direction as shown by the contour plots in Fig. 9(b). In each case, the L2 error norm is computed after 20 periods of waves have passed through the domain. A CFL number of 0.8 is used for all four levels of meshes. Numerical dissipation parameters are set to $\alpha=0$ and $\sigma=2$. Lower numerical dissipation by decreasing σ can increase the accuracy and lower the error norm. However, for order of accuracy studies, a constant value of 2 is used.

Exact solutions of acoustic waves are obtained by assuming linear perturbations. Error norms for all four meshes are first calculated using linear Euler equations with a density acoustic wave amplitude of 10^{-4} . Figure 10(a) shows the expected slightly better than the 4th-, 3rd-, and 2nd-order convergence for dependent variables and their derivatives, respectively. The same set of computations is repeated using the full Euler equations, and the results are shown in Fig. 10(b). Surprisingly, except for the third derivatives, the error norms of all other variables and derivatives level off around 10^{-8} . A literature search confirmed that most, if not all, acoustic wave validation cases for high-order methods had been performed for the linearized Euler equations because the exact acoustic wave solutions are only valid for linear equations and solutions calculated by full Euler equations have nonlinear effects. However, a simple estimate can be made. For a wave amplitude of 10^{-4} , quadratic nonlinear terms modify the linear solutions at a value around 10^{-8} . Thus, the error norm can never be smaller than 10^{-8} . To verify this, the acoustic wave

amplitude is changed to 10^{-6} and the resulting norm is shown in Fig. 11(a). The L2 error norms now converge at the correct rate for dependent variables and all derivatives. Nonlinear terms only affect the 10^{-12} level, lower than the error of the finest mesh error norm. If the amplitude is further dropped to 10^{-8} , the nonlinear terms affect only at the machine accuracy level at 10^{-16} . Consequently, the error norms converge at the expected rate without any problems as shown in Fig. 11(b).

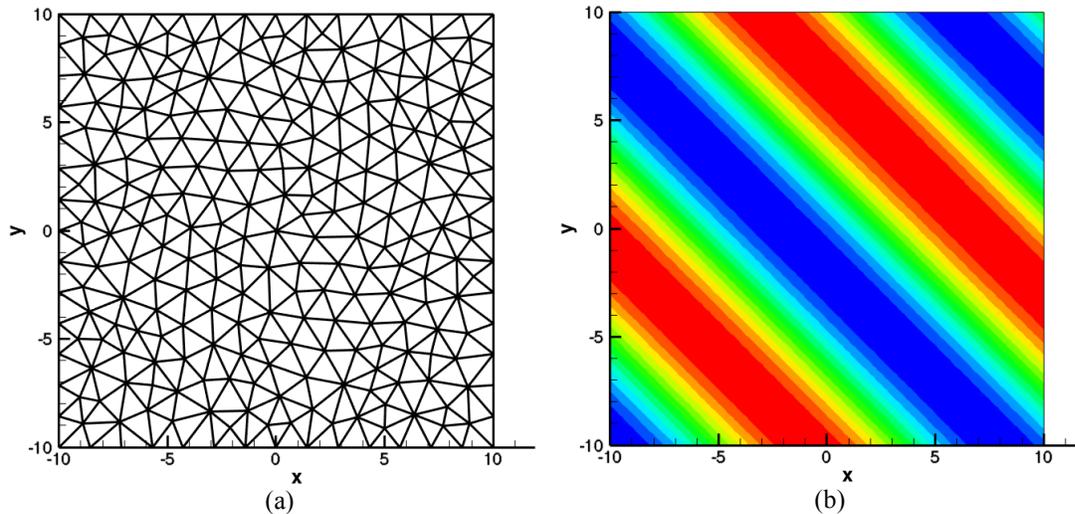


Figure 9. Acoustic waves propagating diagonally through a 20×20 square domain using isotropic triangular meshes (a) the coarsest mesh (b) velocity contours indicating wave direction.

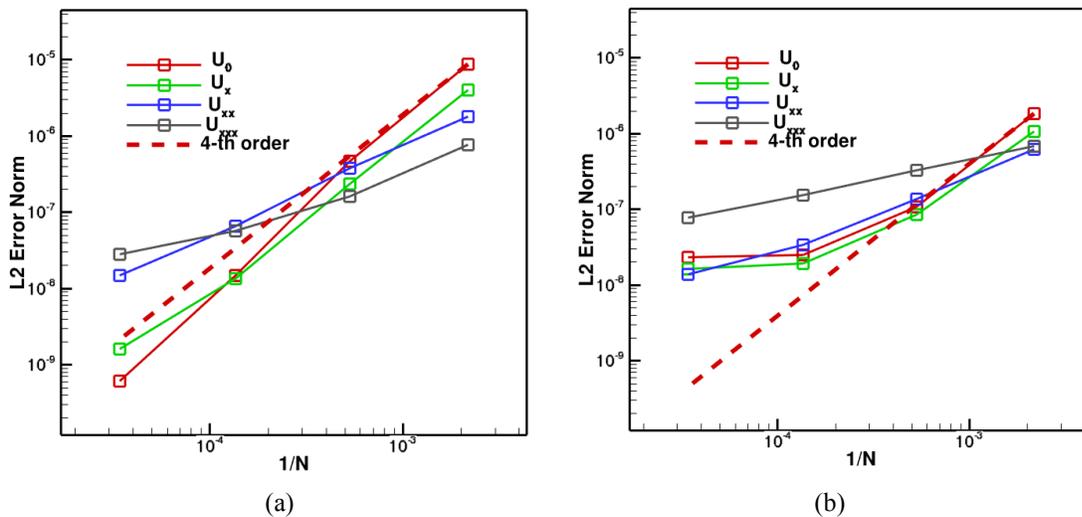


Figure 10. L2 error norm versus number of elements, N , for acoustic waves propagating through a square domain with a density amplitude of 10^{-4} : (a) linear Euler solutions (b) full Euler solutions. Lines indicate theoretical 4th-order convergence.

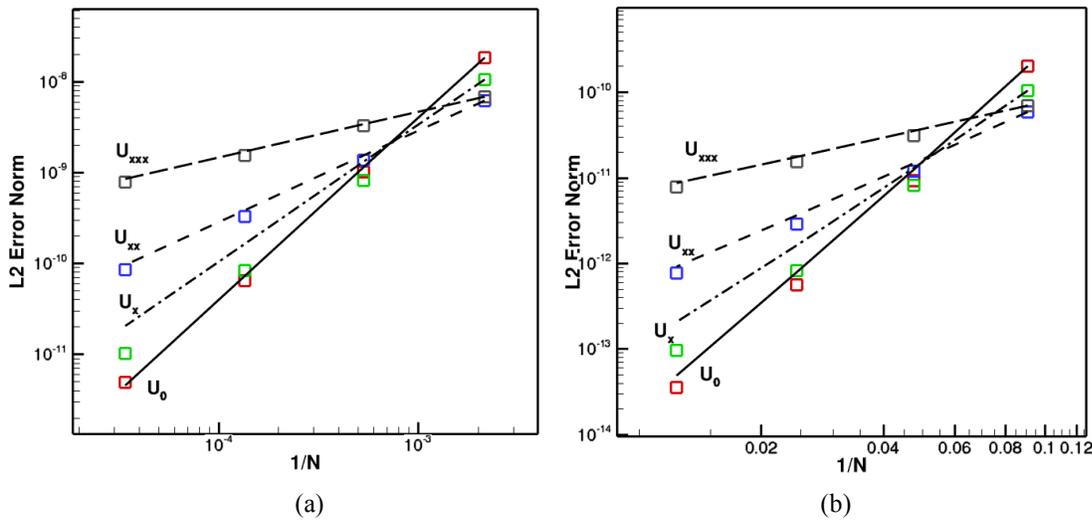


Figure 11. L2 error norm versus number of elements N for acoustic waves propagating through a square domain with different density amplitude: (a) full Euler solutions with amplitude = 10^{-6} (b) full Euler solutions with amplitude = 10^{-8} . Lines indicate theoretical 4th-, 3rd-, and 2nd-order convergence.

The test cases established by Shu and Osher²² for 1D acoustic and shock wave interaction has been used extensively for testing high-order algorithms. The shock impinges on an acoustic wave and generates short-scale oscillations along with shocklets upstream. Numerical computations must capture the shock and preserve the penetrated acoustic wave without excessive damping. The shock detection method is used to switch to second-order schemes in the vicinity of shocks (the dissipation parameter σ is increased to 5 at the shock). The computational results using 1,600 points by Shu and Osher has been treated as the grid-converged solution in many code validation cases. The 4th-order CESE method is also tested with the same number of points. Figure 12(a) compares the computed density distribution at one time instant. The good agreement is evident. Both the shorter scale acoustic waves near the shock and N-shaped shocklets further upstream are properly resolved by the CESE method. The agreement in phase (locations of peaks and valleys) is also quite remarkable. However, the CESE results predict slightly larger (about 5-10%) near shock acoustic waves amplitudes. Figure 12(b) shows a similar calculation but with a lower weighted-average dissipation parameter, $\alpha = 0.5$. The computed penetrated acoustic wave amplitudes increase even further. Additional computations using 3,200 points indicate similar phenomena. Results calculated by the 4th-order CESE method do not appear to converge at the consensus grid-converged solution for this benchmark problem. The consensus grid-converged solution represents only one discretized solution using a relatively larger amount of numerical dissipation for the CESE method. Reducing this numerical dissipation results in an increase of the penetrated acoustic wave amplitudes. Due to the lack of experimental data, the true converged solution is not yet determined. Further computations are needed to shed more light on this benchmark problem.

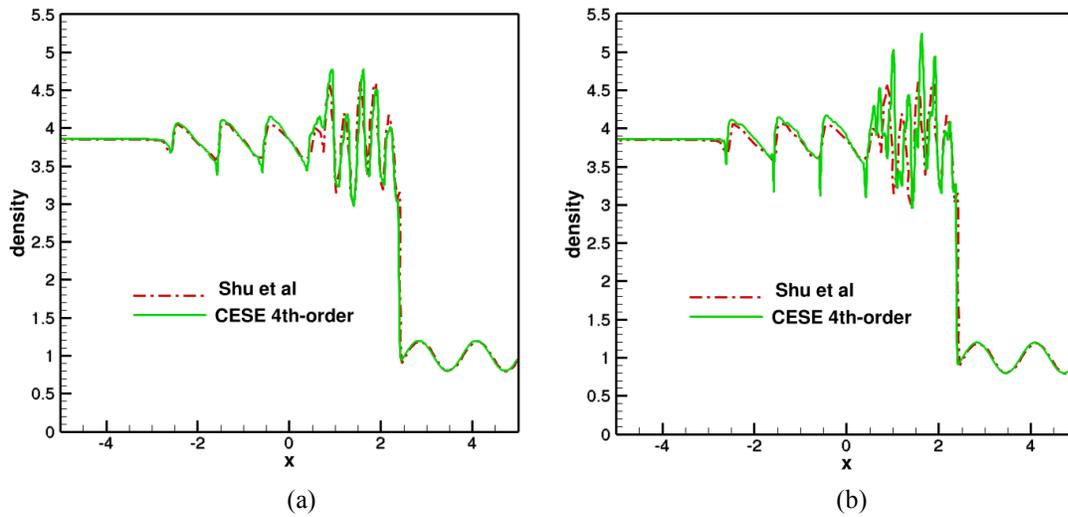
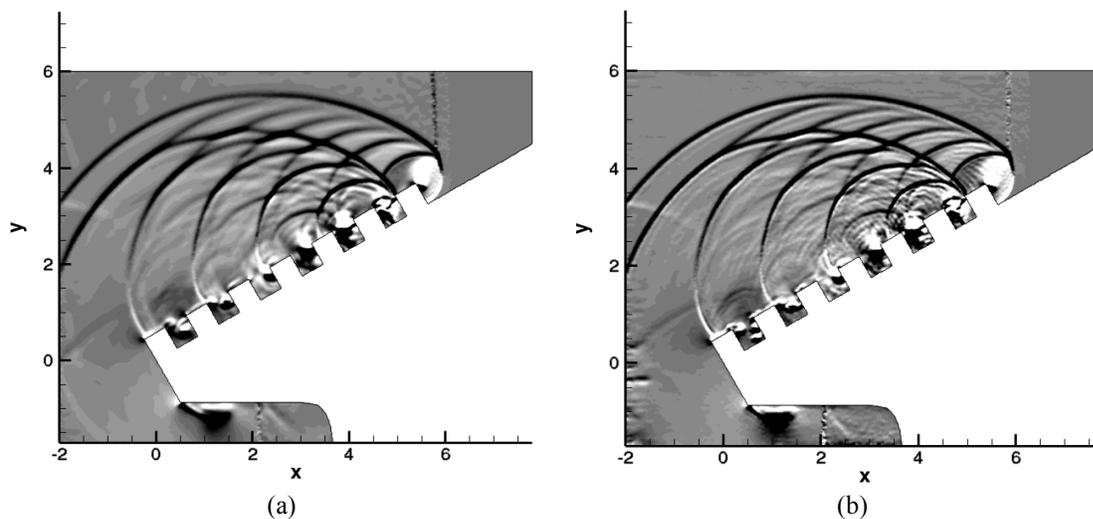
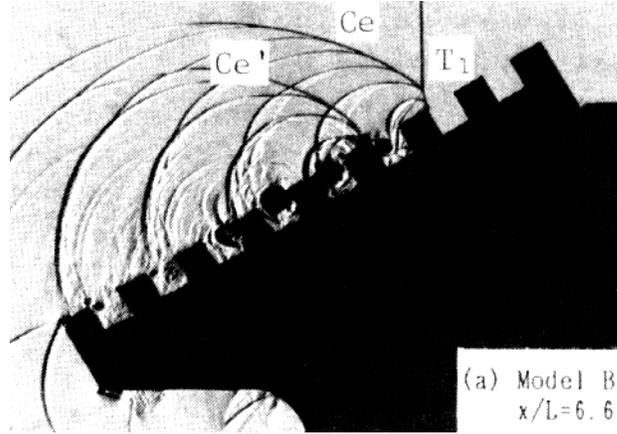


Figure 12. Comparison of computed density distribution using 4th-order CESE with Shu and Osher²² by using a σ value of 1.5 and different numerical dissipation: (a) $\alpha = 1$ (b) $\alpha = 0.5$.

As a more difficult test case involving complex geometry and shock patterns, the Mach 1.4 flow over a multi-gutter wall, previously investigated experimentally by Suzuki et al.²⁴ and computationally by Wang et al.,²⁵ is tested with the high-order CESE method. A total of about 125,000 triangular elements were used in the computations. Figure 13 compares results obtained by using both second- and 4th-order CESE methods using the same mesh with the experiments. The shock sensor defined in eq. (9) is used to detect shocks. For both computations, slightly more numerical dissipation ($\sigma = 2$) is used in the vicinity of shocks and a uniform dissipation parameter ($\sigma = 1.1$) is used elsewhere. In the 4th-order computations, the discretization is also switched to second-order in the vicinity of shocks. Keeping high-order derivatives across the shock is numerically unnecessary. Switching to second-order help resolve the shocks better. As can be seen from Fig. 13, both computations give quite satisfactory shock resolutions. The 4th-order solutions provide more details around the gutter wall and near the complex reflected shocklets, additional short-scale waves reflecting from the wall is also visible. This example demonstrates the use of high-order method in the region where small-scale physics are present. Numerical computations are also underway to detect the minimum grid requirement for the high-order CESE method.

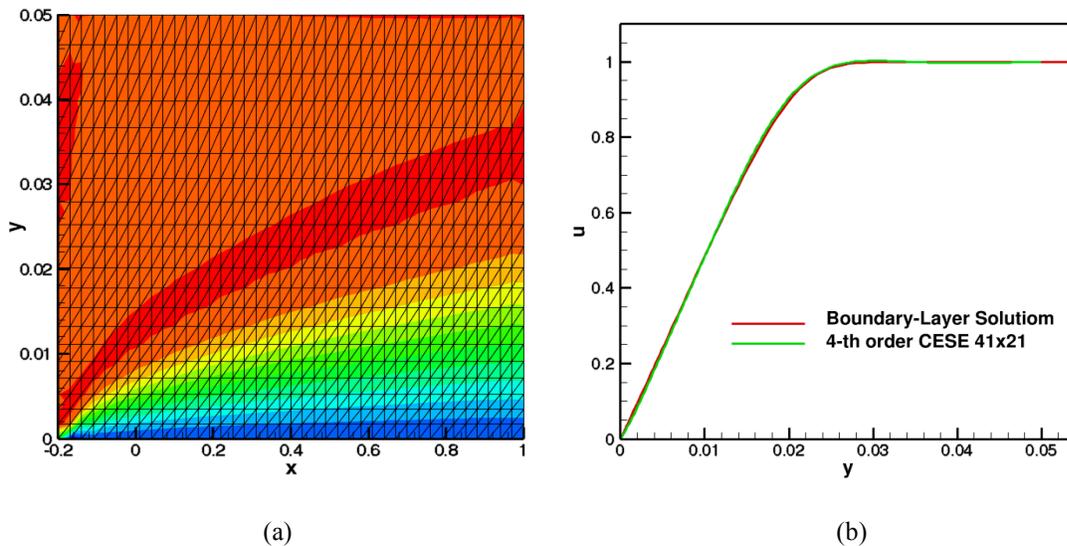




(c)

Figure 13. Normal density gradient contours for a Mach 1.41 flow over a multi-gutter wall: (a) results with second-order CESE (b) results with 4th-order CESE (c) experimental Schlieren photography by Suzuki et al.²⁴ at a slightly different time.

As a preliminary step for validating the developed Navier-Stokes solvers, a Mach 3 flow over a flat plate is computed by using a triangular mesh sliced from a structured 41×21 quadrilateral grid. The inflow and top boundaries are fixed at a Mach number of 3, the bottom wall boundary is set to be at adiabatic conditions, and the downstream boundary uses non-reflecting conditions. The computed velocity contours overlay by the mesh used is shown in Fig. 14(a). The velocity profiles at a Reynolds number of 10^5 agree well with the compressible boundary solutions, as shown in Fig. 14(b). With only about 10 mesh points within the boundary layer, the agreement is satisfactory for a 4th-order Navier-Stokes solver. More detailed validations including order of accuracy are under investigation and will be the subject of a separate paper.



(a)

(b)

Figure 14. Mach 3 viscous flow over an adiabatic flat-plate boundary layer computed by high-order CESE: (a) velocity contours and grid distribution (b) velocity profiles at $Re = 10^5$ compared with compressible boundary-layer solutions.

IV. Concluding Remarks

As an research effort toward developing next-generation, high-fidelity time-accurate flow solvers for complex geometries, the space-time CESE numerical framework has been extended for two new numerical algorithms. The TALTS method allows different time steps to be employed at different cells while maintaining flux conservation over the entire space-time domain. Traditional explicit time-accurate solvers suffer from stringent time step restrictions caused by the finest mesh near either the viscous wall or minute geometrical details. A great deal of computational time is wasted at large cells during the time advancement. For schemes like CESE that are sensitive to the local CFL number, the solution accuracy also degrades at large cells due to extremely small time steps. The TALTS scheme eliminates these numerical issues for problems with a large grid size disparity. Numerical results indicate that in addition to better computational time compared to the conventional constant time-step simulations, the solution accuracy is greatly enhanced for the CESE framework. This new capability has the potential to help future computations for problems with multiple scales and physics such as transitional or turbulent flow simulations. Load balancing could be a potential issue for the TALTS schemes if the decomposed domains do not contain evenly distributed grid size distributions. However, more careful domain decompositions could alleviate this issue.

The CESE flow numerical framework has also been extended to allow 4th-order accurate Navier-Stokes computations. Solutions obtained for all 2D test cases show excellent accuracy with designed order of convergence. The stable CFL number limit remains around one for the 4th-order accurate CESE method. This feature eliminates the need to decrease the time step size substantially as the order of accuracy increases. Scalar wave, Euler, and Navier-Stokes equations for scalar waves, acoustic waves, and boundary-layer flows are validated for simple test cases. The flow variables, first-, second-, and third-derivatives have L2 error norms that converge at 4th-, 3rd-, and 2nd-order, respectively. The high-order framework can thus be used in the regions where derivatives are more important. The test case for a supersonic flow over a multi-gutter wall shows improved resolution away from the shock by using the high-order CESE method. The combination of the TALTS schemes and the high-order CESE framework could potentially provide a new computational paradigm for future large-scale time-accurate simulations. More validations for 3D cases and the combination of TALTS with high-order methods are underway, and results will be reported in a future paper.

Acknowledgments

The authors thank Dr. Sin-Chung Chang of the NASA Glenn Research Center for all the great technical discussions. The research funding for this research has been provided by the Revolutionary Computational Aerosciences subproject under the Aeronautical Sciences Project of the Fundamental Aeronautics Program (FAP).

References

1. Johnson, F. T., Tinoco, E. N., and Yu, N. J., "Thirty Years of Development and Application of CFD at Boeing Commercial Airplanes, Seattle," *Computers & Fluids*, Vol. 34, pp. 1115-1151, 2005.
2. Chang, C.-L., Choudhari, M., "Hypersonic viscous flow over large roughness elements. *Theor Comput Fluid Dynam*, in press doi:10.1007/s00162-010-0191-9.
3. Chang, C.-L., Choudhari, M. M., and Li, F., "Numerical Computations of Hypersonic Boundary-Layer over Surface Irregularities," AIAA Paper 2010-1572, 2010.
4. Chang, C.-L., Choudhari, M., Li, F., and Venkatachari, B. S., "Effects of Cavities and Protuberances on Transition over Hypersonic Vehicles," AIAA Paper 2011-3245, 2011.
5. Bassi, F. and Rebay, S., "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," *J. Comput. Physics*, Vol. 138, No. 2, pp. 251-285, 1997.
6. Cockburn, B., and Shu, C.-W., "Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *SIAM Journal of Scientific Computing*, Vol. 16, No. 3, pp. 173-261, 2001.
7. Fidkowski, K. J. and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, pp. 673-694, 2011.

8. Burgess, N. K. and Mavriplis, D. J., "hp-Adaptive Discontinuous Galerkin Methods for the Navier-Stokes Equations," *AIAA Journal*, Vol. 50, No. 12, pp. 2682-2694, 2012.
9. Atkins, H. L. and Pampell, A., "Robust and Accurate Shock Capturing Method for High-Order Discontinuous Galerkin Methods," AIAA Paper 2011-3058, 2011.
10. H.T. Huynh, A Flux Reconstruction Approach to High-Order Schemes Including Discontinuous Galerkin Methods, AIAA paper 2007-4079, 2007.
11. Premasathan, S., Liang, C. and Jameson, A., "Computation of Flows with Shocks Using Spectral Difference Scheme with Artificial Viscosity," AIAA Paper 2010-1449, 2010.
12. Y. Liu, M. Vinokur, & Z. J. Wang, Spectral difference method for unstructured grids I: Basic formulation, *J. of Comput. Phys.*, Vol. 216, pp. 780-801, 2006.
13. Chang, S.-C., "The Method of Space-Time Conservation Element and Solution Element—A New Approach for Solving the Navier-Stokes and Euler Equations," *J. Comput. Physics*, Vol. 119, pp. 295-324, 1995.
14. Chang, C.-L., "Time-Accurate, Unstructured-Mesh Navier-Stokes Computations with the Space-Time CESE Method," AIAA Paper 2006-4780, 2006.
15. Chang, C.-L., "Three-Dimensional Navier-Stokes Calculations Using the Modified Space-Time CESE Method," AIAA Paper 2007-5818, 2007.
16. Chang, S.C., Wu, Y., Yang, V., and Wang, X.Y., "Local Time-Stepping Procedures for the Space-Time Conservation Element and Solution Element Method," *Inter. J. of Comput. Fluid Dyn.*, Vol. 19, No. 5, pp. 359-380, 2005.
17. Yen, J. C., "Demonstration of a Multi-Dimensional Time-Accurate Local Time Stepping CESE Method," AIAA Paper 2011-2755, 2011.
18. Chang, S., "A New Approach for Constructing Highly Stable High Order CESE Schemes," AIAA-2010-543, 2010.
19. Bi, C. and Ginting, V., "A Posteriori Error Estimates of Discontinuous Galerkin Method for Nonmonotone Quasi-linear Elliptic Problems," *J. Scientific Computing*, Vol. 55, Issue 3, pp. 659-687, 2013.
20. Bilyeu, D., Chen, Y., and Yu, S. J., "High-Order CESE Methods for the Euler Equations," 49th AIAA Paper 2011-1065, 2011.
21. Bilyeu, D., Chen, Y. and Yu, S. J., "A Two-dimensional 4th-Order CESE Method for the Euler Equations on Triangular Unstructured Meshes," AIAA Paper 2012-0293, 2012.
22. Shu, C.-W. and Osher, S., "Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, II," *J. Comput. Physics*, Vol. 83, pp. 32-78, 1989.
23. Suzuki, T., Adachi, T., and Kobayashi, S. "An Experimental Analysis on Shock Reflection over the Two-Dimensional Model on a Dust Layer," in 17th International symposium on Shock Waves & Shock Tubes," edited by Yong W. Kim, Bethlehem, PA, 1989.
24. Wang, X. Y., Chow, C. Y., and Chang, S. C., "The Space-Time Conservation Element and Solution Element Method—A New High-resolution and Genuinely Multidimensional Paradigm for Solving Conservation Laws – II. Numerical Simulation of Shock Waves and Contact Discontinuities," NASA/TM—1998-208844, 1998.
25. Lovely, D., and Haimes, R., "Shock Detection from Computational Fluid Dynamics Results," AIAA Paper 1999-3285, 1999.
26. Chang, S.-C. and Wang, X.-Y., "Multi-Dimensional Courant Number Insensitive CE/SE Euler Solvers for Applications Involving Highly Non-uniform Meshes," AIAA Paper 2003-5285, 2003.
27. "Mathematic", Wolfram Research, <http://www.wolfram.com/mathematic/>, last accessed May 11, 2013.
28. Abramowitz, M. and Stegun, I. A., "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables," *New York: Dover*, 1965.
29. Salas, M. D. and Atkins, H. L., "On Problems Associated with Grid Convergence of Functionals," *Computers & Fluids*, Vol. 38, Issue 7, pp. 1145-1454, 2009.