Journal of Scientific Computing (2023) 94:48
https://doi.org/10.1007/s10915-022-02068-3

High-Order CENO Finite-Volume Scheme with Anisotropic Adaptive Mesh Refinement: Efficient Inexact Newton Method for Steady Three-Dimensional Flows

L. Freret[1] · C. N. Ngigi[1] · T. B. Nguyen[2] · H. De Sterck[3] · C. P. T. Groth[1]

Received: 4 March 2022 / Revised: 1 November 2022 / Accepted: 26 November 2022

Abstract
A high-order finite-volume scheme with anisotropic adaptive mesh refinement (AMR) is combined with a parallel inexact Newton method for the solution of steady compressible fluid flows governed by the Euler and Navier–Stokes equations on three-dimensional multi-block body-fitted hexahedral meshes. The proposed steady flow solution method combines a family of robust and accurate high-order central essentially non-oscillatory (CENO) spatial discretization schemes with both a scalable and efficient Newton–Krylov–Schwarz (NKS) algorithm and a block-based anisotropic AMR method. The CENO scheme is based on a hybrid solution reconstruction procedure that provides high-order accuracy in smooth regions (even for smooth extrema) and non-oscillatory transitions at discontinuities and makes use of a high-order representation of the mesh and a high-order treatment of boundary conditions. In the proposed Newton method, the resulting linear systems of equations are solved using the generalized minimal residual (GMRES) algorithm preconditioned by a domain-based additive Schwarz technique. The latter uses the domain decomposition provided by the block-based AMR scheme leading to a fully parallel implicit approach with an efficient scalability of the overall scheme. The anisotropic AMR method is based on a binary tree data structure and permits local anisotropic refinement of the grid in preferred directions as directed by appropriately specified physics-based refinement criteria. Numerical results are presented for a range of inviscid and viscous steady problems and the computational performance of the combined scheme is demonstrated and assessed.

Keywords Steady inviscid and viscous flows · Parallel anisotropic AMR · Newton–Krylov–Schwarz method · High-order finite volume scheme

✉ L. Freret
lfreret@utias.utoronto.ca

[1] Institute for Aerospace Studies, University of Toronto, Toronto, ON M3H 5T6, Canada

[2] Computational Engineering Study Program, Vietnamese–German University, Le Lai Street, Hoa Phu Ward, Thu Dau Mot City, Binh Duong, Vietnam

[3] Department of Applied Mathematics, University of Waterloo, Waterloo, ON N2L 3GI, Canada

Published online: 21 January 2023

1 Introduction and Scope

1.1 Introduction

One highly successful approach for reducing the computational cost of computational fluid dynamics (CFD) solutions is to make use of solution-directed mesh adaptation where the underlying computational mesh automatically adapts according to the solution, adding mesh resolution only where required. Methods based on adaptive mesh refinement (AMR) have proved to be particularly effective for the solution of conservation equations on structured Cartesian and body-fitted meshes and have been developed for a wide range of problems [1–8]. The non-uniform block-based AMR scheme of Freret and Groth [9] and Freret et al. [10] was more recently proposed for the solution of inviscid and viscous flows, allowing anisotropic refinement of the mesh with standard low-order (i.e., second-order accurate) finite-volume schemes. This anisotropic block-based AMR approach is also well suited for use with high-order spatial schemes [11, 12] and has been coupled for this purpose to the solution of the ideal magnetohydrodynamics (MHD) equations using the high-order central essentially non-oscillatory (CENO) finite-volume scheme of Ivan et al. [13, 14]. The latter overcomes many of the traditional computational issues associated with essentially non-oscillatory (ENO) and weighted ENO (WENO) schemes [15, 16] by using a hybrid reconstruction approach based on a fixed central stencil, the same for each solution variable. An unlimited high-order K-exact reconstruction is then performed in the cells where the solution is well resolved while the scheme reverts to a standard low-order piecewise limited linear approach for cells with under-resolved or discontinuous solution content. Switching in the hybrid procedure is determined by a smoothness indicator. The CENO high-order scheme has been successfully applied to a broad range of flows on uniform Cartesian meshes including inviscid flows [13], viscous flows [13, 17], large-eddy simulation (LES) of turbulent premixed flames [18], and MHD problems [14, 19]. The efficiency of the CENO scheme has also been assessed on cubed-sphere meshes [14, 20], in applications with high-order implicit time-marching schemes [21], and has also been extended to unstructured meshes for laminar viscous flows [17] and turbulent reactive flows [18]. For many applications, the high-order CENO scheme has been shown to be more efficient than a second-order method in terms of execution time and number of computational cells required to achieve a target discretization error [13, 17, 18]. Note that other ENO/WENO variants have also been proposed in the literature with the goal of overcoming the computational bottlenecks of such schemes. Recent examples include the so-called targeted ENO (TENO) and central-weighted ENO (CWENO) methods [22–26].

For steady flow problems, the inherent disparate temporal scales of the governing conservation equations and the varying mesh spacing arising from the AMR strategy give rise to the need for solving a coupled system of non-linear algebraic equations with a high degree of numerical stiffness. One common approach for the efficient solution of non-linear algebraic equations is Newton's method [27–34]. In this study, an inexact Newton method is considered for the solution of the stiff non-linear equations arising from the high-order spatial discretization of the partial differential equations. In this method, the non-linear system is solved by applying a Newton linearization and using an iterative linear solver to solve the resulting linear systems for each Newton iteration [28, 34]. The linear system arising from the linearization is both large and sparse and is solved by a right-preconditioned generalized minimal residual (GMRES) method [35–38]. A domain-based additive Schwarz preconditioning technique is used as the global preconditioner and incomplete lower-upper factorization with fill is used as the local preconditioner to improve the convergence rate.

The Schwarz preconditioner provides the advantage of using the same domain decomposition procedure as used by the anisotropic block-based AMR scheme, making parallel implementation rather straightforward. The combination of the strengths of block-based AMR with an implicit Newton–Krylov–Schwarz (NKS) time-stepping scheme to achieve a scalable parallel framework has been considered previously for standard lower-order spatial discretization methods of two-dimensional (2D) and three-dimensional (3D) steady and unsteady as well as non-reactive and reactive flows [9, 10, 28, 34, 39]. The parallel NKS method is reconsidered here for application in conjunction with the high-order CENO finite-volume scheme so as to achieve efficient high-order predictions of 2D and 3D steady flows.

1.2 Scope

The aim of this study is then to combine the high-order CENO finite-volume scheme of Ivan et al. [13, 14] and the anisotropic block-based AMR technique of Freret and Groth [9] and Freret et al. [10] with an efficient and scalable inexact Newton method [28, 34, 39] for computing steady-state solutions of the Euler and Navier–Stokes equations governing inviscid and viscous compressible gaseous flows. Furthermore, as correct high-order treatment of boundary conditions is a crucial element for developing accurate numerical schemes, an effort has been made to develop a high-order geometry through the use of a tricubic representation of the mesh elements as well as imposing high-order boundary conditions by constraining the least-squares reconstruction in cells adjacent to the boundary. The benefits and computational performance of this proposed combination are demonstrated by considering numerical results for a range of representative flow problems.

The remainder of the paper is organized as follows. The high-order CENO finite-volume scheme with high-order mesh and boundary conditions is described in Sect. 2. Description of the anisotropic block-based AMR then follows in Sect. 3. The proposed inexact Newton's method for the solution of steady flow problems is then presented in Sect. 4. Qualitative and quantitative comparisons are subsequently described in Sect. 5 to demonstrate the rapid convergence properties and potential of the combined high-order AMR discretization procedure with implicit NKS method for a range of flow problems.

2 High-Order Central ENO Finite-Volume Scheme

The proposed high-order AMR scheme is applied herein to the solution of Euler and Navier–Stokes equations governing inviscid and viscous compressible three-dimensional flows of a polytropic gas. For viscous applications considered herein, the flow Reynolds numbers are restricted to ensure that the flows remain laminar. Applications to turbulent flows and any associated additional modelling of any unresolved solution content are not considered herein. The latter will be the subject of future follow-on studies.

2.1 Governing Equations

The conservation form of the governing equations for inviscid or viscous flows of a compressible polytropic gas can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \vec{\nabla} \cdot \vec{\mathbf{F}} = \frac{\partial \mathbf{U}}{\partial t} + \vec{\nabla} \cdot \left(\vec{\mathbf{F}}_{\mathrm{H}} - \vec{\mathbf{F}}_{\mathrm{E}} \right) = \mathbf{0}, \tag{2.1}$$

where $\mathbf{U}$ is the vector of conserved solution variables and $\vec{\mathbf{F}}$ is the solution flux dyad. For a three dimensional Cartesian coordinate system, $\mathbf{U}$ is given by

$$\mathbf{U} = [\rho, \rho u, \rho v, \rho w, \rho e]^{\mathrm{T}}, \tag{2.2}$$

where ρ is the gas density, u, v, w are the components of the velocity vector in the x, y, z directions, $e = p/(\rho(\gamma - 1)) + u^2/2$ is the specific total energy, $p = \rho RT$ is the gas pressure, T is the gas temperature, R is the ideal gas constant, and γ is the ratio of specific heats and $h = e + p/\rho$ is the specific enthalpy. The corresponding solution total flux dyad, $\vec{\mathbf{F}}$, is the sum of the inviscid or hyperbolic flux and the viscous or elliptic flux dyads, $\vec{\mathbf{F}}_{\mathrm{H}}$ and $\vec{\mathbf{F}}_{\mathrm{E}}$, respectively, and given by $\vec{\mathbf{F}} = \vec{\mathbf{F}}_{\mathrm{H}} - \vec{\mathbf{F}}_{\mathrm{E}} = [\mathbf{F}_h - \mathbf{F}_v, \mathbf{G}_h - \mathbf{G}_v, \mathbf{G}_h - \mathbf{G}_v]$ where $\mathbf{F}_h, \mathbf{G}_h, \mathbf{H}_h$ are the inviscid or hyperbolic flux vectors associated with the x, y, z coordinate directions, respectively, and $\mathbf{F}_e, \mathbf{G}_e, \mathbf{H}_e$ are the corresponding viscous flux vectors in the x, y, and z coordinate directions, respectively. Definitions of the latter can be found in the textbooks by Schlichting and Gersten [40] or Hirsch [41]. In the case of inviscid flows, $\mathbf{F}_e = \mathbf{G}_e = \mathbf{H}_e = \mathbf{0}$ and $\vec{\mathbf{F}} = \vec{\mathbf{F}}_{\mathrm{H}}$.

2.2 Finite-Volume Method and Semi-discrete Form

A semi-discrete form of the integral form of Eq. (2.1) results from the application of a standard high-order finite-volume method applied for a hexahedral computational cell, (i, j, k), of a three-dimensional grid. The resulting semi-discrete form is given by

$$\frac{d\overline{\mathbf{U}}_{ijk}}{dt} = -\frac{1}{V_{ijk}} \sum_{f=1}^{6} \sum_{m=1}^{N_{\mathrm{Gface}}} \left(\tilde{\omega} (\vec{\mathbf{F}}_{\mathrm{H}} - \vec{\mathbf{F}}_{\mathrm{E}}) \cdot \vec{n} \right)_{i,j,k,f,m} = \overline{\mathbf{R}}_{ijk}(\overline{\mathbf{U}}), \tag{2.3}$$

where here $\mathbf{R}(\overline{\mathbf{U}})$ is the solution residual, N_{Gface} is the number of Gauss quadrature points per face, and $\vec{n}$ is the local normal of the face f at each of the N_{Gface} Gauss quadrature points. The hexahedral cells are contained within logically Cartesian blocks that form a multi-block body-fitted mesh with general unstructured connectivity between blocks as shown in Fig. 1. In general, the total number of Gauss integration points, N_{Gface}, at which the numerical flux is evaluated is dictated by the targeted rate of spatial solution accuracy of the discretization. In this study, standard tensor-product quadrature with four Gauss quadrature points is used for cell faces, $N_{\mathrm{Gface}} = 4$, providing fourth-order accurate spatial accuracy. The latter is the target accuracy for the high-order CENO finite-volume scheme considered here.

The numerical values of the hyperbolic fluxes, $\vec{\mathbf{F}}_{\mathrm{H}} \cdot \vec{n}$, at each Gaussian quadrature point on each face of the cell (i, j, k) are determined from the solution of a Riemann problem. Given the high-order reconstructed values of the solution vectors to the left and right of the cell interface, $\mathbf{U}_l$ and $\mathbf{U}_r$, an upwind numerical flux is evaluated by making use of the so-called HLLE flux function [42, 43], which is based on an approximate solution to the Riemann problem in the direction defined by the normal to the cell face. Note that a wide variety of Riemann-solver based flux functions have been proposed and are described in the literature (see, for example, [44–46]), providing varying degrees of accuracy and robustness. However, unlike for low-order spatial discretization schemes, the relative importance of the choice of flux function on solution accuracy is generally less important for high-order methods and the HLLE flux function has been found to provide good performance in terms of reliability and robustness for the high-order finite-volume applications of interest here [11–14, 19]. Numerical values for the elliptic fluxes $\vec{\mathbf{F}}_{\mathrm{E}} \cdot \vec{n} = \mathcal{F}_{\mathrm{E}}(\mathbf{U}_c, \vec{\nabla}\mathbf{U}_c, \vec{n})$ are determined by using the centrally-weighted approach proposed by Ivan and Groth [13] in terms of values of

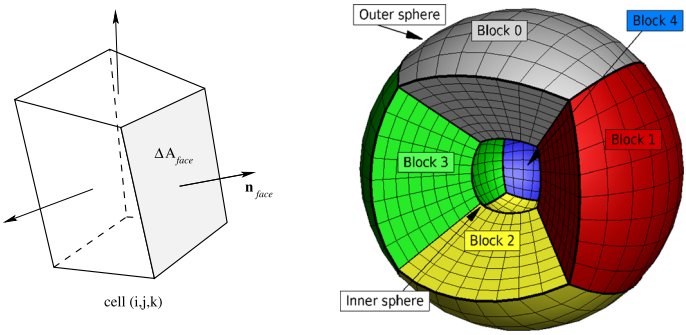


Fig. 1 **a** Three-dimensional hexahedral cell, (i,j,k), of multi-block body-fitted mesh showing face normals and **b** example of a cubed-sphere mesh with one of the six root blocks removed for clarity

the cell interface solution, $\mathbf{U}_c$, its gradient, $\vec{\nabla}\mathbf{U}_c$, and the normal vector, $\vec{n}$. The values of $\mathbf{U}_l$, $\mathbf{U}_r$, $\mathbf{U}_c$ are determined by performing K-exact CENO polynomial reconstruction [13] as detailed in the next section. In particular, K-exact polynomial reconstruction is used to reconstruct the solution, $\mathbf{U}$, to $(K+1)$-order of accuracy. Direct differentiation of this K-exact reconstructed solution is then used to obtain a K-order accurate value of the solution gradient $\vec{\nabla}\mathbf{U}$. The resulting scheme for the Navier–Stokes equations is then overall K-order accurate. An accuracy of order $K+1$ is achieved for the Euler equations. Practically, for the three-dimensional simulations of viscous flows examined in Sects. 5.6 and 5.7, just one K-exact polynomial reconstruction is performed for each variable when evaluating both the hyperbolic and elliptic fluxes. Details of the high-order CENO reconstruction procedure and spatial discretization scheme now follow.

2.3 High-Order CENO Spatial Discretization Scheme

2.3.1 *K*-Exact Least-Square Reconstruction

The hybrid CENO finite-volume method for conservation laws originally proposed by Ivan and Groth [13] is used to discretize the governing equations on a hexahedral computational grid. The hybrid CENO procedure uses the multidimensional unlimited K-exact reconstruction of Barth [47] in smooth regions and reverts to a piecewise limited linear reconstruction algorithm in regions deemed as non-smooth or under-resolved by a solution smoothness indicator, $\mathcal{S}$, thus providing monotone solutions near discontinuities.

The K^{th}-order Taylor polynomial representing a K-exact reconstruction of a scalar solution quantity, U_{ijk}, within a cell (i, j, k) about the cell centroid, $(x_{ijk}, y_{ijk}.z_{ijk})$, can be expressed as:

$$U^K_{ijk}(x,y,z) = \sum_{\substack{p_1=0 \\ p_1+p_2+p_3\le K}}^{K} \sum_{p_2=0}^{K} \sum_{p_3=0}^{K} (x-x_{ijk})^{p_1}(y-y_{ijk})^{p_2}(z-z_{ijk})^{p_3} D_{p_1p_2p_3}. \tag{2.4}$$

The coefficients, $D_{p_1p_2p_3}$, of the Taylor polynomial are the unknown derivatives (so-called Taylor derivatives or coefficients) of the expansion and their number, N, for a target fourth-order accuracy is equal to $N = 20$ for the Euler equations ($K = 3$ piecewise cubic

reconstruction) and $N = 35$ for the Navier-Stokes equations ($K = 4$ piecewise quartic reconstruction). The evaluation of $D_{p_1 p_2 p_3}$ requires the least-squares solution of an over-determined system of linear equations of the form

$$\mathbf{A}\mathbf{x} - \mathbf{c} = \mathbf{e}, \tag{2.5}$$

where the coefficient matrix, $\mathbf{A}$, of the linear system depends only on the mesh geometry and can be calculated a priori in a preprocessing step. The current average values of the conserved solution within each cell, $\overline{\mathbf{U}}$, are contained in the vector $\mathbf{c}$ and the error in the reconstructed solution in each control volume is represented by vector $\mathbf{e}$ which has its norm minimized in a least-squares sense. The desired coefficients, $D_{p_1 p_2 p_3}$, are contained in the solution vector x. Singular value decomposition (SVD) is used to solve the weighted least-squares problem associated with the CENO reconstruction [13]. This approach permits the computation of a pseudo-inverse matrix after which the solution of the least-squares problem is given by a simple matrix–vector product. The use of a single fixed stencil, the same for all dependent variables, allows the pseudo-inverse matrix to be stored and re-used in the reconstruction of all solution variables, thereby avoiding the repeated evaluation of the pseudo inverse. This was found to reduce significantly the computational costs of performing the CENO reconstruction without requiring substantial additional storage [13]. Additionally, it is noted that issues often arise with high-order polynomial reconstruction related to conditioning and/or invertibility that generally increase in severity with the order of the scheme and can be highly dependent on mesh features such as cell size, aspect ratio, and topology. However, a combination of inverse-distance weighting and column-scaling procedure is applied here to the least-squares problem of Eq. (2.5) which significantly improves its conditioning, making the problem virtually independent of mesh geometry, and thereby affording robust and reliable solutions of the least-squares problem [13]. The resulting solutions of the over-determined system provide the values of the Taylor derivatives, $D_{p_1 p_2 p_3}$, defining the unlimited high-order reconstruction polynomials valid for regions of smooth solution content.

The mechanism used to switch between the preceding unlimited high-order representation and a piecewise limited linear reconstruction of the solution relies on a solution smoothness indicator, as described fully by Ivan et al. [13, 14]. The solution smoothness indicator is evaluated as follows

$$\mathcal{S} = \frac{\alpha}{\max(1-\alpha, \epsilon)} \frac{\mathcal{N}_{\mathrm{SOS}} - \mathcal{N}_D}{\mathcal{N}_D - 1}, \tag{2.6}$$

where the smoothness parameter, α, is given by

$$\alpha = 1 - \frac{\sum_{\gamma}\sum_{\delta}\sum_{\zeta}(u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - u^K_{ijk}(\vec{X}_{\gamma\delta\zeta}))^2}{\sum_{\gamma}\sum_{\delta}\sum_{\zeta}(u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - \bar{u}_{ijk})^2}, \tag{2.7}$$

and where $\mathcal{N}_{\mathrm{SOS}}$ is an integer value representing the size of the stencil, $\mathcal{N}_D$ is the number of degrees of freedom in the K-exact polynomial reconstruction (i.e., $\mathcal{N}_D = 20$ for the solution of the Euler equations and $\mathcal{N}_D = 35$ for the solution of the Navier–Stokes equations), the ranges of the indices (γ, δ, ξ) are taken to include either the whole or a subset of the supporting reconstruction stencil for cell (i, j, k), and the parameter, $\epsilon = 10^{-8}$, is introduced to avoid division by zero. Based on the magnitude relative to a chosen cut-off value, $\mathcal{S}_C$, a value of $\mathcal{S}$ larger than $\mathcal{S}_C$ indicates a smooth representation of the solution while a value of $\mathcal{S}$ less than $\mathcal{S}_C$ indicates the presence of non-smooth or under-resolved solution content. In the case of the latter, limited linear reconstruction is performed with $K = 1$ using the slope limiter of

Venkatakrishnan [48]. For all of the simulations presented in Sect. 5 to follow, the smoothness indicator cut-off value was $\mathcal{S}_C = 1,500$.

It is noted that in the hybrid CENO reconstruction scheme outlined above, the unlimited high-order reconstruction is performed in terms of the conserved solution variables, $\mathbf{U}$, and a primitive variable set, $\mathbf{W} = [\rho, \mathbf{u}, p]^T$, is used where the scheme reverts to limited linear reconstruction for the treatment of non-smooth solution content. In this way, tight control is maintained on the monotonicity and positivity of key flow variables while preserving the conservation properties and consistent high-order accuracy of the finite-volume scheme as discussed by Freret et al. [12]. This procedure involving the change in reconstruction variables also incurs virtually no additional computational overhead.

2.3.2 Reconstruction Stencil

Stencil selection can be a crucial aspect of ENO-type schemes [49]. In the CENO approach, central high-order stencils are used exclusively which, for a fixed stencil size, are known to be the most accurate stencils. In selecting the appropriate CENO stencil for a given solution accuracy, stencil symmetry and ensuring the over-determinedness of the least-squares reconstruction problem of minimum size are the key considerations.

In order to arrive at an over-determined system of equations to solve for the unknown Taylor derivatives associated with the solution reconstruction, a central stencil including the first ring of 27 nearest neighbours and 6 additional next-to-nearest neighbours is used here in the solution reconstruction of the Euler equations. This reconstruction stencil, $\mathcal{S}_{33}$, is defined by

$$\mathcal{S}_{33} = \{(i, j, k)|i, j, k \in \{-1, 0, 1\}\} \cup \{(l, 0, 0) \cup (0, l, 0) \cup (0, 0, l)|l \in \{-2, 2\}\}, \quad (2.8)$$

and, for a non-adapted body-fitted mesh, consists of 33 cells, ensuring an over-determined stencil for the 20 unknown derivatives required in a $K = 3$ or cubic reconstruction of the solution for the Euler equations. The latter yields a fourth-order accurate spatial-discretization of these hyperbolic equations. In the case of the Navier–Stokes equations, $K = 4$ quartic reconstruction is required to achieve fourth-order accuracy and a central reconstruction stencil, $\mathcal{S}_{57}$, consisting of the first ring plus $6 \times 5 = 30$ additional next-to-nearest or second-ring cells is instead used. This stencil can be defined by

$$\begin{aligned}\mathcal{S}_{57} =& \{(i, j, k)|i, j, k \in \{-1, 0, 1\}\} \\ &\cup \{(l, m, n) \cup (m, l, n) \cup (m, n, l)||l| = 2, m, n \in \{-1, 0, 1\}, |m + n| \leq 1\}.\end{aligned} \quad (2.9)$$

For a non-adapted body-fitted mesh, this stencil contains 57 cells and ensures an over-determined system for the 35 unknown Taylor derivatives of the linear system of Eq. (2.5). Figure 2a and b depicts the reconstruction stencils for the cubic and quartic representation of the solution, $\mathcal{S}_{33}$ and $\mathcal{S}_{57}$, respectively, in the case of a uniform Cartesian mesh. In the case of a cubed-sphere grid, which contains corner and edge degeneracies in the mesh [14, 20], the 33-cell $\mathcal{S}_{33}$ stencil degenerates to 29-cell and the 57-cell $\mathcal{S}_{57}$ stencil degenerates to a 51-cell stencil at the grid block edges, respectively, as shown in the cubic and quartic reconstruction stencils of Fig. 2c and d. Nevertheless, both reconstruction stencils still ensure an over-determined linear system for the Taylor derivatives of Eq. (2.5). Note that the reconstruction defined here are valid for non-adapted or nominally uniform body-fitted mesh. In the case of the anisotropic block-based AMR considered here, modified stencils must be considered as described in Sect. 3 to follow.

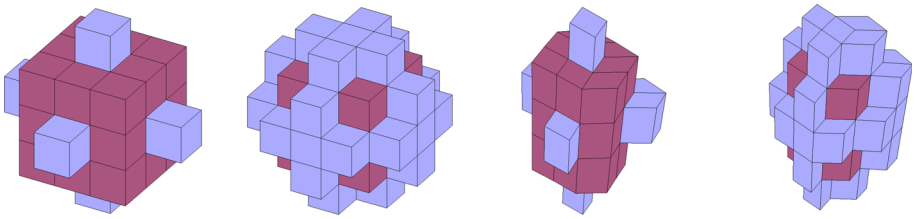


Fig. 2 Illustration of the nominally 33- and 57-cell stencils associated with a uniform Cartesian mesh (**a, b**) and a cubed-sphere mesh (**c, d**). The 33-cell stencils of (**a**) and (**c**) is used for fourth-order accurate inviscid flow predictions while the 57-cell stencils (**b**) and (**d**) are used for fourth-order accurate solutions of the Navier–Stokes equations. Purple coloured cells coloured are located in the first ring (nearest neighbours) and light blue coloured cells are added from the second ring (next to nearest neighbours). Due to the edge degeneracy of the cubed-sphere mesh, the 33-cell stencil of (**c**) contains only 29 cells and the 57-cell stencil of (**d**) has only 51 cells (Color figure online)

For the limited linear reconstruction used in regions deemed to be non-smooth, a standard central stencil of consisting of the 27 nearest neighbours is used, thus providing an over-determined set of equations to solve for the three unknown first-order spatial derivatives associated with the reconstruction. This linear reconstruction stencil, $\mathcal{S}_{27}$, is defined simply by

$$\mathcal{S}_{27} = \big\{(i, j, k) | i, j, k \in \{-1, 0, 1\}\big\}. \tag{2.10}$$

2.4 High-Order Representation of Geometry

Typical flow problems of interest in CFD may involve domains with curved boundaries consisting of non-planar computational surfaces. In order to facilitate the accurate treatment of such boundaries with high-order solution accuracy, a mapping or coordinate transformation procedure is often used to map the geometrical properties of the computational cells in the physical space to a regular Cartesian reference space. The mapping is done using shape functions and mapping Jacobians [50]. Such approaches have been adopted in other high-order finite-volume methods [32, 51, 52]. The mapping procedure allows accurate evaluation of mesh quantities such as the positions of volumetric and face Gauss integration points and more precise evaluation of geometrical properties such as cell volume, face areas, moments, and the position of the cell centroid. It also facilitates more accurate enforcement of boundary conditions and evaluation of numerical fluxes and source terms. The type of mapping approach determines the order of accuracy of the geometry representation. To reduce the effects of geometrical modelling errors, particular care must be taken to represent curved boundaries using a transformation that provides an order of accuracy matching that of the underlying finite-volume spatial discretization scheme. In the event that the numerical scheme has an order of accuracy greater than that of the coordinate transformation, geometrical mapping errors can pollute numerical results and dominate the error, effectively reducing the prediction accuracy.

The mapping approach adopted herein extends the approach presented by Ivan et al. [14] in which a trilinear mapping was to represent the hexahedral elements of a body fitted mesh and map them to a reference cubic cell of a regular Cartesian coordinate system. The latter is appropriate for a second-order (i.e., linear) representation of the domain boundary. In the

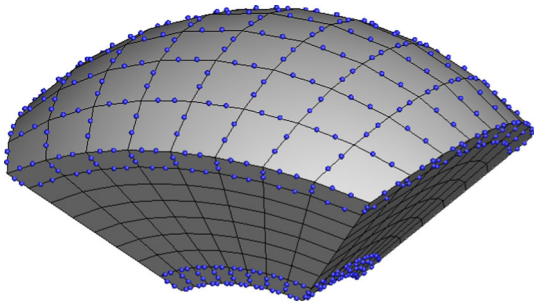


Fig. 3 Grid block 0 of the cubed-sphere mesh of Fig. 1b showing how the high-order geometry representation is enabled in boundary cells to enable a 4th-order representation of the geometry. The additional nodes shown in blue are added to the boundary cells to permit the tricubic mapping (Color figure online)

present study, an enhanced tricubic formulation is used to map the hexahedral elements to a serendipity-type reference cube as described by Lapidus and Pinder [53] defined on the three-dimensional domain $[-1, 1] \times [-1, 1] \times [-1, 1]$ with nodal basis functions defined at prescribed node positions within the reference element. The proposed tricubic mapping procedure is capable of providing 4th-order accurate (i.e., cubic) representation of domain boundaries, consistent with the desired accuracy of the spatial discretization scheme.

The original second-order trilinear mapping scheme adopted by Ivan et al. [14] made use of shape functions defined only at the vertices of the reference element. Hence, eight basis functions are used for the trilinear mapping approach, one defined at each vertex of the reference cube. The expressions for the trilinear transformation and its associated coefficients are re-summarized in Appendix "A.1 Trilinear Transformation". In the case of the tricubic mapping approach, two additional nodes are introduced along the edges of the serendipity-type reference element [50]. This results in a total of 32 shape functions for the tricubic mapping as summarized in Appendix "A.2 Tricubic Transformation". The tricubic high-order geometry representation comes at a non-negligible computational cost in terms of memory/storage. In order to reduce the latter, the high-order representation can be enabled only for cells adjacent to the domain boundary, which typically represent just a small fraction of the total number of cells. An illustration of this hybrid approach (tricubic mapping for high-order cells at the physical boundary and trilinear mapping for the remaining interior cells) is depicted in Fig. 3 where the high-order nodes are shown for the grey coloured grid block of a cubed-sphere grid shown in Fig. 1. The high-order nodes depicted in blue are used only within cells adjacent to the boundary. A trilinear transformation is used for all other cells. For such a coarse mesh, the high-order cells represents 25% of the total number of cells and as the mesh is refined this percentage can decrease substantially. For example, in the convergence study to be considered in Fig. 4, the high-order cells represent respectively 25%, 12.5%, 6.25%, 3.125% and finally just 1.5% of the total number of computational cells on each successively refined mesh, respectively.

2.4.1 Definition of High-Order Quadrature Nodes

The location of the high-order nodes is determined using a standard computational mapping in terms of grid metrics [54, 55]. In particular, the mapping between the physical coordinates, $\vec{x} = (x, y, z)$, of the mesh and a corresponding uniformly spaced Cartesian computational space, (ξ, η, ζ), is approximated by the following third-order Taylor series expansion

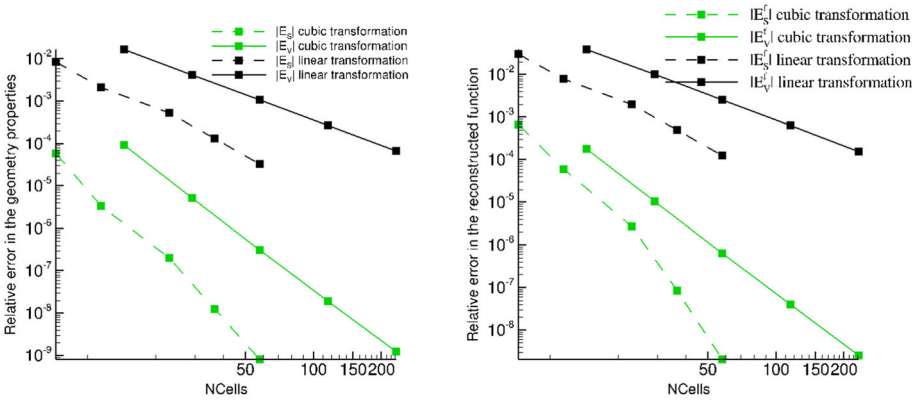


Fig. 4 Accuracy of 3D geometry representation showing **a** the relative computed error in the volume, $|E_V|$, and surfaces (inner and outer boundary), $|E_S|$, of a cubed-sphere grid, $\mathcal{D}$, using both the high-order tricubic representation of the boundary cells with trilinear elsewhere and the trilinear representation for all cells; and **b** the high-order reconstruction of the functional $f(x, y, z) = x^4 + y^4 + z^4$ on the cubed-sphere grid, $\mathcal{D}$, showing the relative error in the integrated value of the function evaluated over the volume, $|E_V^f|$, and over the inner and outer surfaces, $|E_S^f|$, for fourth-order ($K = 3$) CENO reconstruction again with high-order tricubic representation of the boundary cells with trilinear elsewhere and the trilinear representation for all cells. In both cases, NCells represents the equivalent number of cells for a one-dimensional domain

$$
\begin{aligned}
&\vec{x}(\xi + \Delta\xi, \eta + \Delta\eta, \zeta + \Delta\zeta) \\
&\quad = \vec{x}(\xi, \eta, \zeta) + \left.\frac{\partial \vec{x}}{\partial \xi}\right|_{\xi,\eta,\zeta} \Delta\xi + \left.\frac{\partial \vec{x}}{\partial \eta}\right|_{\xi,\eta,\zeta} \Delta\eta + \left.\frac{\partial \vec{x}}{\partial \zeta}\right|_{\xi,\eta,\zeta} \Delta\zeta \\
&\qquad + \frac{1}{2}\left(\left.\frac{\partial^2 \vec{x}}{\partial \xi^2}\right|_{\xi,\eta,\zeta} (\Delta\xi)^2 + 2\left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \eta}\right|_{\xi,\eta,\zeta} (\Delta\xi\Delta\eta) + 2\left.\frac{\partial^2 \vec{x}}{\partial \xi \partial \zeta}\right|_{\xi,\eta,\zeta} (\Delta\xi\Delta\zeta) \right. \\
&\qquad \left. + \left.\frac{\partial^2 \vec{x}}{\partial \eta^2}\right|_{\xi,\eta,\zeta} (\Delta\eta)^2 + 2\left.\frac{\partial^2 \vec{x}}{\partial \eta \partial \zeta}\right|_{\xi,\eta,\zeta} (\Delta\eta\Delta\zeta) + \left.\frac{\partial^2 \vec{x}}{\partial \zeta^2}\right|_{\xi,\eta,\zeta} (\Delta\zeta)^2 \right) \\
&\qquad + \frac{1}{6}\left(\left.\frac{\partial^3 \vec{x}}{\partial \xi^3}\right|_{\xi,\eta,\zeta} (\Delta\xi)^3 + 3\left.\frac{\partial^3 \vec{x}}{\partial \xi^2 \partial \eta}\right|_{\xi,\eta,\zeta} (\Delta\xi)^2\Delta\eta + 3\left.\frac{\partial^3 \vec{x}}{\partial \xi^2 \partial \zeta}\right|_{\xi,\eta,\zeta} (\Delta\xi)^2\Delta\zeta \right. \\
&\qquad + \left.\frac{\partial^3 \vec{x}}{\partial \eta^3}\right|_{\xi,\eta,\zeta} (\Delta\eta)^3 + 3\left.\frac{\partial^3 \vec{x}}{\partial \eta^2 \partial \zeta}\right|_{\xi,\eta,\zeta} (\Delta\eta)^2\Delta\zeta + 6\left.\frac{\partial^3 \vec{x}}{\partial \xi \partial \eta \partial \zeta}\right|_{\xi,\eta,\zeta} \Delta\xi\Delta\eta\Delta\zeta \\
&\qquad \left. + \left.\frac{\partial^3 \vec{x}}{\partial \zeta^3}\right|_{\xi,\eta,\zeta} (\Delta\zeta)^3 \right),
\end{aligned}
\tag{2.11}
$$

where the mapping has been expressed in terms of the grid metrics. The values of the latter (i.e., the derivatives) are computed here using fourth-order central-difference approximations within the interior of the grid blocks and third-order forward/backward differences at grid block boundaries. The locations of the high-order nodes are then determined by combining the Taylor series expansions associated with the two nearest vertices using an averaging procedure.

2.4.2 Volumetric and Surface Integration

To evaluate the volumetric integral,

$$\mathcal{I} = \iiint_{V_{ijk}} g(\vec{X})\, \mathrm{d}V, \tag{2.12}$$

the variables and integration domain are transformed to those of a reference unit cube by making use of the mathematical transformation, $\vec{X} = \vec{X}(p, q, r)$, and its transformation Jacobian determinant, $\det \mathbf{J}$ [13]. Thus, the volumetric integral, $\mathcal{I}$, is evaluated in the canonical space, (p, q, r), as

$$\begin{aligned} \mathcal{I} &= \iiint_{[-1,1]^3} g(\vec{X}(p,q,r))\ \det \mathbf{J}\, \mathrm{d}p\mathrm{d}q\mathrm{d}r \\ &\approx \sum_{1}^{N_v} g(\vec{X}_m)\, \omega_m, \end{aligned} \tag{2.13}$$

where N_v is the number of volumetric Gaussian quadrature points. Tensor-product curbature is used here and, for fourth-order accuracy, $N_v = 8$ is used where the abscissa and weights for each direction of the transformed domain on $[-1, 1]$ are $[-0.577, 0.577]$ and $[1., 1.]$, respectively.

High-order estimates of surface integrals are evaluated here in a similar fashion. To evaluate a surface integral along a surface with a constant p-coordinate, the following expression is used

$$\begin{aligned} \mathcal{A}_p &= \iint_{A_p} g(\vec{X})\mathrm{d}A = \iint_{[-1,1]^2} g(\vec{X}(p,q,r))\ \det \mathbf{J}_p \mathrm{d}q\mathrm{d}r \\ &\approx \sum_{m=1}^{N_g} g(\vec{X}_m)\ \det \mathbf{J}_p \omega_m, \end{aligned} \tag{2.14}$$

where N_g is the number of Gaussian quadrature points on the surface. Again, for fourth-order accuracy, $N_g = 4$ is used where again the abscissa and weights for each direction of the transformed domain on $[-1, 1]$ are $[-0.577, 0.577]$ and $[1., 1.]$, respectively. Similar expressions for integrals along surfaces $\mathcal{A}_q$ and $\mathcal{A}_s$ follow by cyclic permutation of Eq. (2.14). Expressions for the Jacobians of the surface transformations are provided in Appendix "A.3 Transformation Jacobians".

To assess the accuracy of the preceding high-order definitions of the 3D mesh geometry, a domain, $\mathcal{D}$, defined by a cubed-sphere mesh [14, 20] with an inner radius of $R_i = 1$ and an outer radius of $R_o = 3$ is now considered. The volume of $\mathcal{D}$ and the total surface area associated with the boundaries of $\mathcal{D}$ are compared to the exact known values of the volume and surface. The relative errors in the computed values of the volume are here defined using

$$|E_V| = \frac{1}{V_T} \left| \sum_{i,j,k} \iiint_{V_{ijk}} \mathrm{d}V - V_T \right|, \tag{2.15}$$

where $V_T = 104\pi/3$ is the exact total volume of $\mathcal{D}$ and, similarly, the relative error for the total boundary surface is determined using

$$|E_S| = \frac{1}{A_T} \left| \left(\sum_{i,j} \iint_{A_{ij},R_o} \mathrm{d}A + \sum_{i,j} \iint_{A_{ij},R_i} \mathrm{d}A \right) - A_T \right|, \tag{2.16}$$

where $A_T = 4\pi + 36\pi = 40\pi$ is the sum of the exact outer and inner boundary surface areas of $\mathcal{D}$. Figure 4a shows convergence of the error in the volume and surface area for $\mathcal{D}$ for a sequence of meshes of increasing resolution. The initial mesh is made up of 6 blocks with $8 \times 8 \times 8$ cells and 3,072 total cells and the final mesh has 24,576 blocks and a total of $N = 12{,}582{,}912$ computational cells. The expected theoretical asymptotic convergence rates are clearly achieved for both trilinear and tricubic transformations as the mesh is refined with the slopes for the estimated error approaching -1.99 for the trilinear mapping procedure applied everywhere and -3.99 for the trilinear mapping approach applied at the boundary cells as described above.

Reconstruction of a smooth function, $f(x, y, z) = x^4 + y^4 + z^4$, are now also compared to the exact solution on the same cubed-sphere domain, $\mathcal{D}$. The norm of the numerical error in the reconstructed solution over the domain is computed as follows:

$$|E_V^f| = \frac{1}{V_T}\left|\sum_{i,j,k}\iiint_{V_{ijk}} f_{i,j,k}^K(\vec{X})\mathrm{d}V - \iiint_{V_T} f(\vec{X})\mathrm{d}V\right| \\ \frac{1}{V_T}\left|\sum_{i,j,k}\iiint_{V_{ijk}} f_{i,j,k}^K(\vec{X})\mathrm{d}V - V_T^f\right|, \tag{2.17}$$

where the exact value of the function, f, integrated over the volume is $V_T^f = (26232\pi)/35$ and $f_{i,j,k}^K(\vec{X})$ is the numerical reconstruction from cell averages of function f within cell, (i, j, k), with K^{th}-order reconstruction accuracy. The norm of the numerical solution error on the boundary of the domain, $\mathcal{D}$, is estimated using

$$|E_S^f| = \frac{1}{A_T}\left|\sum_{i,j}\left(\iint_{A_{ij},R_i} f_{i,j}^K(\vec{X})\mathrm{d}A\right) + \sum_{i,j}\left(\iint_{A_{ij},R_o} f_{i,j}^K(\vec{X})\mathrm{d}A\right) - A_T^f\right|, \tag{2.18}$$

where the analytical integrated function f over the inner and outer surfaces, $A_T^f = (12\pi \times 3^6)/5 + (12\pi)/5$. Figure 4b shows the convergence of the reconstructed function obtained using the fourth-order CENO method on the same series of meshes as examined in Fig. 4a. The expected theoretical asymptotic convergence rate of the fourth-order accurate method is achieved with the tricubic transformation applied to the boundary cells for integration over the volume (green solid lines) and over the surfaces (green dashed lines). Second-order accuracy is obtained with the fourth-order CENO reconstruction when trilinear transformation is used everywhere within $\mathcal{D}$ for both the volume integration (black solid lines) and the surface integration (black dashed lines). The results of Fig. 4b exemplify the typical geometrical representation errors that can pollute and reduce the accuracy of the predictions of high-order spatial discretization methods. As the mesh for domain $\mathcal{D}$ is refined, the slopes of the reconstruction error approach -3.99 and -5.4 for the tricubic approach for the reconstructed function measured over the volume and surfaces, respectively, providing validation of the proposed tricubic-based CENO reconstruction procedure applied to general hexahedral cells with non-planar faces.

2.5 High-Order Treatment of Boundary Conditions

Treatment is also required for the high-order imposition of boundary data. One approach to imposing boundary conditions is to make use of extra rows of cells, so-called ghost cells,

which are added outside and beyond the geometric boundary of the computational domain. Solution values are then imposed in the ghost cells in such a way that the reconstructed solution approximates the solutions associated with the particular boundary condition. Such a method has been used in the previous studies by Freret et al. [11, 12]; however, it presents two distinct disadvantages. First, it requires that there are sufficient additional layers of ghost cells in order to carry out the reconstruction and evaluation of the smoothness indicator in all necessary cells. It also requires the accurate specification of the data within each ghost cell, which is not always possible, depending on the boundary condition. An alternative approach is to enforce the boundary data by constraining the CENO reconstruction in control volumes adjacent to the boundary as previously described by Ollivier-Gooch and Van Altena [56] and later by Ivan and Groth [13]. In the current study, constrained reconstruction procedure is considered and this represents the extension of the approach of Ivan and Groth [13] for two-dimensional domains to the three-dimensional case.

Constrained reconstruction involves the addition of equality constraints to the reconstruction procedure to ensure that the polynomial approximations of the CENO reconstruction within the interior cells next to the boundary exactly satisfy the particular boundary condition at the Gauss quadrature integration points. By constraining the least-squares reconstruction in the control volumes adjacent to the boundary, complex boundary conditions can be enforced exactly by the reconstruction. In the current study, solid wall boundary conditions have been implemented for reflecting (slip) boundaries with $\vec{V} \cdot \vec{n} = 0$. In order to solve the system of equations arising from the constrained least squares reconstruction (i.e., exact boundary constraints plus reconstruction conditions), a linear equality-constrained least squares problem is solved. In other words, a least-squares solution is sought that minimizes $||\mathbf{Fx} = \mathbf{y}||_2$ such that $\mathbf{Bx} = \mathbf{d}$ where $\mathbf{B}$ is a $P \times M$ matrix and $\mathbf{d}$ is a P-vector with P the number of constraints.

For the $\vec{V} \cdot \vec{n} = 0$ reflective boundary condition, the additional constraints that need to be satisfied by the reconstruction coefficients for the x-, y- and z-velocity components, $(D_{p_1p_2p_3})_u$, $(D_{p_1p_2p_3})_v$, and $(D_{p_1p_2p_3})_w$, respectively, is

$$\sum_{\substack{p_1=0 \\ p_1+p_2+p_3 \le K}}^{K} \sum_{p_2=0}^{K} \sum_{p_3=0}^{K} \Delta X_g^{p_1} \Delta Y_g^{p_2} \Delta Z_g^{p_3} \left[n_x^g (D_{p_1p_2p_3})_u + n_y^g (D_{p_1p_2p_3})_v + n_z^g (D_{p_1p_2p_3})_w \right] = 0, \tag{2.19}$$

where $\Delta X_g^{p_1} = (x_g - x_{ijk})^{p_1}$, $\Delta Y_g^{p_2} = (y_g - y_{ijk})^{p_2}$ and $\Delta Z_g^{p_3} = (z_g - z_{ijk})^{p_3}$. The number of constraints in this case is $P = 7$ which represents one constraint for each of the $N_g = 4$ Gauss quadrature points on the boundary face, n_g, and the mean conservation constraint for each component of the velocity. The second dimension of the matrix, $\mathbf{B}$, is $M = 3N$ with N being the number of unknown Taylor derivatives. The $M \times M$ matrix, $\mathbf{F}$, the M-component vector, $\mathbf{y}$, and the M-component solution vector, $\mathbf{x}$, are defined as

$$\mathbf{F} = \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} (c)_u \\ (c)_v \\ (c)_w \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} (D_{p_1p_2p_3})_u \\ (D_{p_1p_2p_3})_v \\ (D_{p_1p_2p_3})_w \end{pmatrix}, \tag{2.20}$$

where $\mathbf{A}$ is defined in Eq. (2.5) and $(c)_u$, $(c)_v$ and $(c)_w$ form the right-hand side vector c of Eq (2.5) for the x-, y- and z-velocity components, respectively. Numerical assessment of the constrained reconstruction associated with a high-order mesh representation is considered in Sects. 5.1 and 5.2 for an inviscid subsonic flow past a cylinder and past a sphere, respectively. Constrained reconstruction for other boundary conditions, including no-slip velocity

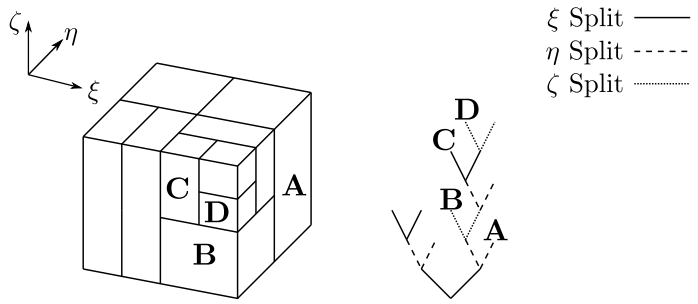


Fig. 5 Three-dimensional binary tree and the corresponding grid blocks after several levels of anisotropic refinement of the mesh

boundary conditions with $\vec{V} = \vec{0}$ can also be applied by following a similar procedure to that outlined above.

3 Anisotropic Block-Based Adaptive Mesh Refinement (AMR)

In block-based methods the computational domain is divided into grid blocks containing groupings of cells. In the current anisotropic block-based approach, each grid block is taken to have the same number of cells in order to more readily achieve an optimal load balancing. Mesh adaptation is then accomplished by refining and coarsening of the grid blocks. Each refinement produces new blocks, called "children" from a "parent" block and the children can be refined further. This refinement process can be associated with a preferred direction and can be reversed in regions that are deemed over-resolved and two, four or eight children can coarsen or merge into a single parent block. In order to keep track of the connectivity between blocks, a binary tree data structure is used. Figure 5 depicts the resulting binary tree after several refinements of an initial mesh consisting of a single block.

In the present work, heuristic or physics-based refinement criteria involving the gradients of the solution are used to direct the mesh refinement. For anisotropic refinement of the mesh, directionally dependent refinement criteria are proposed here based on the individual components of the gradient vectors. Expressions for the directional refinement indicators, r_γ, r_η, and r_ζ, are as follows:

$$r_\gamma = \frac{1}{u}\left(\vec{\nabla} u \cdot \Delta\tilde{\boldsymbol{\gamma}}\right), \quad r_\eta = \frac{1}{u}\left(\vec{\nabla} u \cdot \Delta\tilde{\boldsymbol{\eta}}\right), \quad r_\zeta = \frac{1}{u}\left(\vec{\nabla} u \cdot \Delta\tilde{\boldsymbol{\zeta}}\right), \tag{3.1}$$

where $\Delta\tilde{\boldsymbol{\gamma}}$, $\Delta\tilde{\boldsymbol{\eta}}$ and $\Delta\tilde{\boldsymbol{\zeta}}$ are the vector differences between the midpoints of the faces in the γ, η, and ζ logical coordinate directions of the body-fitted multi-block mesh, respectively. In Eq. (3.1), the variable, u, here represents any solution quantity of interest. In the present study, the flow density, $u = \rho$, Mach number, $u = \mathrm{M}$, and the velocity field magnitude, $u = ||\mathbf{u}||_{L_2}$, are considered for directing the mesh refinement in various flow problems. Additionally, as the high-order CENO scheme provides a measure of solution smoothness, an anisotropic refinement indicator based on the CENO smoothness indicator is also adopted here, as originally proposed by Freret et al. [12]. For direction γ, the anisotropic refinement indicator takes the form

$$r_\gamma = \exp\left(-\mathcal{S}_\gamma\right), \tag{3.2}$$

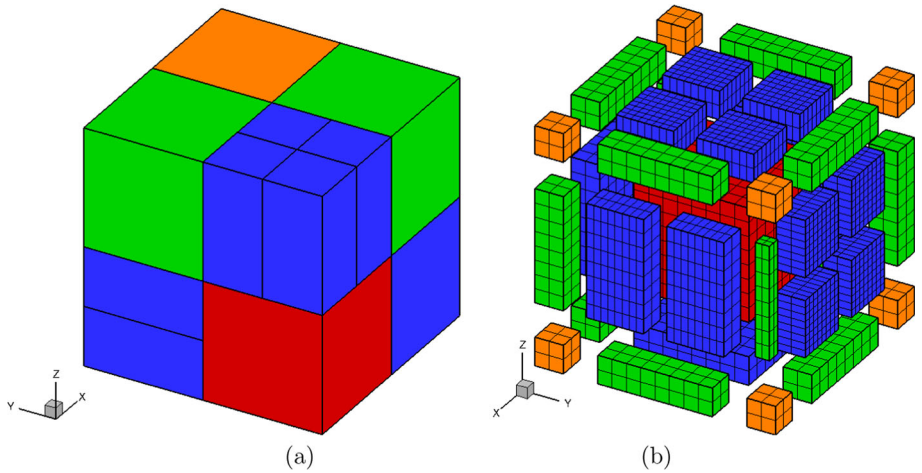


Fig. 6 Example of **a** a multi-block anisotropic-refined AMR mesh illustrating **b** the non-uniform representation of the grid block where the red-coloured block is the interior block of interest for which the non-uniform structure is represented and the blue-, green-, and orange-coloured blocks share a face, an edge and a block, respectively, and together the cells of the latter makeup the ghost cells of the interior block in the non-uniform representation (Color figure online)

with

$$\mathcal{S}_\gamma = \frac{\alpha_\gamma}{\max(1-\alpha_\gamma, \epsilon)}, \tag{3.3}$$

where

$$\alpha_\gamma = 1 - \frac{\sum_\delta \sum_\zeta (u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - u^K_{ijk}(\vec{X}_{\gamma\delta\zeta}))^2}{\sum_\delta \sum_\zeta (u^K_{\gamma\delta\zeta}(\vec{X}_{\gamma\delta\zeta}) - \bar{u}_{ijk})^2}. \tag{3.4}$$

The smoothness based refinement measures, r_η and r_ζ, for the η and ζ coordinate directions, respectively, have similar forms. In the proposed AMR scheme, regions with large values of r_γ, r_η, and r_ζ are refined preferentially in the appropriate coordinate directions and regions with small measures of these values are flagged for mesh coarsening in the appropriate coordinate direction.

The general anisotropic AMR framework of Freret and Groth [9] and Freret et al. [10] was originally formulated for a limited second-order spatial discretization scheme and adopts a non-uniform representation of the computational cells within each grid block. As discussed by Freret et al. [10], the non-uniform block approach readily allows and is directly compatible with the use of high-order spatial discretization methods. In particular, two primary benefits provided by the non-uniform block treatment, among several others, is the elimination of the need for the restriction and prolongation of high-order solution content in the ghost cells of the grid blocks, which can be computationally expensive [11, 12]. Figure 6a depicts a multi-block anisotropic refined AMR mesh obtained after several refinements, showing a block of interest coloured in red and the neighbouring grid blocks. The blue-coloured blocks share a face with the red-coloured block while the green- and orange-coloured blocks share an edge and a corner, respectively, with the red block. The non-uniform block representation or treatment is illustrated in Fig. 6b where an exploded view of the non-uniform grid block permits a view of

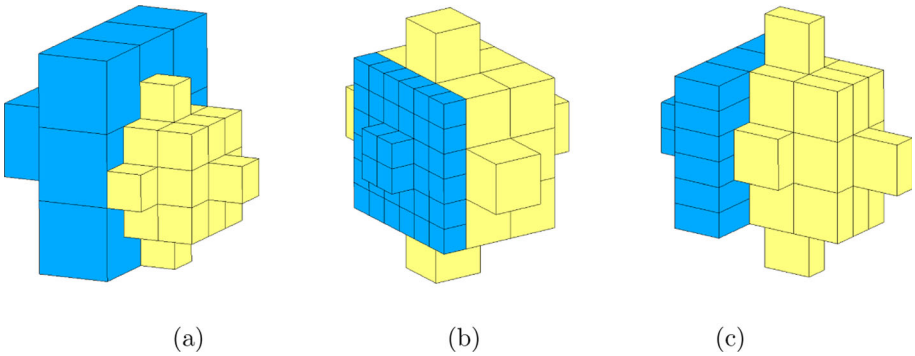


Fig. 7 Illustration of the modified nominally 33-cell fourth-order inviscid flow reconstruction stencils for a Cartesian cell, (i, j, k), located at block boundaries such that the stencils include the ghost cells from neighbouring blocks with **a** neighbouring block having a coarser level of refinement, **b** neighbouring block having a finer level of refinement, and **c** neighbouring block having both finer and coarser levels of refinement in the tangential directions. The cells associated with the interior domain of the grid block are coloured yellow and the ghost cells are coloured blue (Color figure online)

the interior grid block of interest in this case, again coloured in red. The ghost cells associated with neighbouring blocks are also shown, which are at differing levels of refinement in the various coordinate directions. The blue-, green-, and orange-coloured blocks share a face, an edge and a block, respectively, with the red block. Together the cells of the blue, green, and orange blocks makeup the ghost cells of the interior block in the non-uniform representation. It can be seen that ghost cell blocks contain directly the cells of the neighbouring blocks, even those at different levels of refinement as found at mesh resolution changes. This non-uniform structure removes the need for the high-order restriction and prolongation of the solution to the ghost cells; however, hanging nodes may now be present and the definition of the high-order reconstruction stencil for the CENO scheme becomes more challenging with anisotropic AMR. Freret et al. [10] have proposed an effective neighbour search algorithm in this case and it is used here. Figure 7 shows examples of the nominally 33-cell fourth-order inviscid flow stencil defined by Eq. (2.8) for computational cells located next to a block boundary with a mesh resolution change across the interface. The shaded cells of Fig. 6 represent such boundary cells. Figure 7a depicts the stencil for the boundary cell of a block having a finer neighbouring block, Fig. 7b is illustrative of the stencil for a boundary cell associated with a block having a coarser neighbouring block, and finally Fig. 7c shows the stencil for the case in which the neighbour block is both coarser and finer in the two tangential directions. Additionally, in the case of the nominally 57-cell fourth-order viscous flow stencil defined by Eq. (2.9), Fig. 8a, b, and 7c depict the corresponding stencils for a similar boundary cells of a block having finer and coarser neighbouring blocks, as well as a neighbour block that is both coarser and finer in the two tangential directions, respectively. Note that the number of cells associated with the stencils of the boundary cells now varies considerably, depending on the mesh refinement, but in each case the system of equations for the reconstruction remains over determined.

Finally following refinement and coarsening of the mesh, high-order accurate solution transfer from the originally coarse to newly introduced fine cells are required to distribute the solution among offspring cells with a consistent order of accuracy. To do this, the high-order CENO reconstruction polynomials for all solution variables, u, within coarse cells are

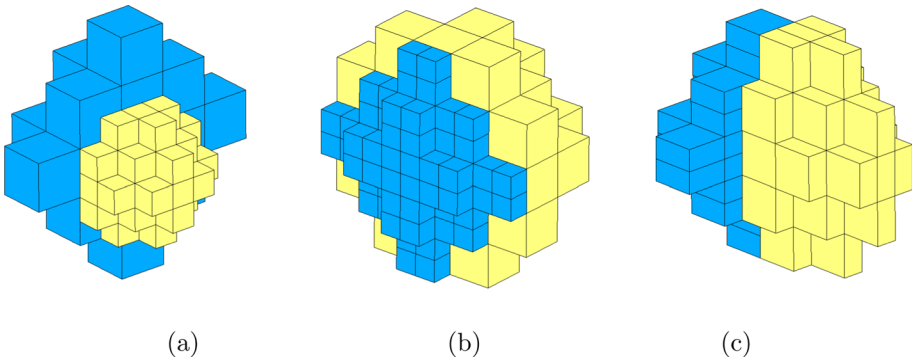


Fig. 8 Illustration of the modified nominally 57-cell fourth-order inviscid flow reconstruction stencils for a Cartesian cell, (i, j, k), located at block boundaries such that the stencils include the ghost cells from neighbouring blocks with **a** neighbouring block having a coarser level of refinement, **b** neighbouring block having a finer level of refinement, and **c** neighbouring block having both finer and coarser levels of refinement in the tangential directions. The cells associated with the interior domain of the grid block are coloured yellow and the ghost cells are coloured blue (Color figure online)

integrated over the domains associated with each new fine cell having a volume, V_{fine}, such that

$$\bar{u}_{\text{fine}} = \frac{1}{V_{\text{fine}}} \iiint_{V_{\text{fine}}} u^k_{\text{coarse}}(\vec{X}) \mathrm{d}V = \frac{1}{V_{\text{fine}}} \sum_{m=1}^{N_v} \omega_m \, u^k_{\text{coarse}}(\vec{X}_m), \tag{3.5}$$

where $\bar{u}_{\text{fine}}$ is the new cell average solution quantity for the fine cell and the volume integral is computed exactly for the reconstruction polynomial with an appropriate-order tensor-product Gaussian quadrature volumetric integration technique ($N_v = 8$ quadrature points are used for fourth-order spatial accuracy). Here, ω_m and $\vec{X}_m$ are the Gaussian weights and quadrature points associated with the fine cell.

It should be noted that the combination of the high-order CENO finite-volume spatial discretization scheme and reconstruction procedure adopted herein for the arbitrary hexahedral mesh elements of anisotropically refined grids is an unsplit approach in which the flux evaluation and reconstruction in each direction are explicitly coupled. The use of directional splitting and/or curvilinear coordinate mappings may simplify the implementation of high-order methods for three-dimensional applications; however, the proposed unsplit procedure, while possibly somewhat more computationally expensive, has been shown in previous studies to provide important added reliability and robustness as well as more consistent accuracy for the complex mesh topologies of both highly anisotropically refined body-fitted and degenerate cubed-sphere meshes [11, 12].

4 Inexact Newton Method

4.1 Non-linear Equations and Newton Update Scheme

The proposed Jacobian-free inexact Newton method adopted in this study follows the algorithm developed and applied previously by Groth and co-workers [28, 34, 39]. The approach is well adapted for computations on large multi-processor distributed-memory parallel clusters. For steady-state problems, the semi-discrete form of the governing equations given by

Eq. (2.3) above reduces to

$$\frac{\mathrm{d}\overline{\mathbf{U}}}{\mathrm{d}t} = \mathbf{R}(\overline{\mathbf{U}}) = \mathbf{0}. \tag{4.1}$$

Given an initial estimate of the steady state solution, $\mathbf{U}^0$, the solution of the coupled system of non-linear algebraic equations defined by Eq. (4.1) is then obtained here via Newton's method by iteratively solving a sequence of linear systems of equations for successively improved k^{th} estimates, $\mathbf{U}^k$, satisfying

$$\left(\frac{\partial \mathbf{R}}{\partial \overline{\mathbf{U}}}\right)^k \Delta\overline{\mathbf{U}}^k = \mathbf{J}(\overline{\mathbf{U}}^k)\Delta\overline{\mathbf{U}}^k = -\mathbf{R}(\overline{\mathbf{U}}^k), \tag{4.2}$$

where $\mathbf{J} = \partial\mathbf{R}/\partial\overline{\mathbf{U}}$ is the residual Jacobian and $\Delta\overline{\mathbf{U}}^k = \mathbf{U}^{k+1} - \mathbf{U}^k$. The improved solution at the k^{th} Newton step iteration is subsequently obtained using

$$\overline{\mathbf{U}}^{k+1} = \overline{\mathbf{U}}^k + \Delta\overline{\mathbf{U}}^k. \tag{4.3}$$

The Newton iterations are performed until a desired reduction of the residual norm is achieved, that is

$$||\mathbf{R}(\overline{\mathbf{U}}^k)|| < \epsilon \, ||\mathbf{R}(\overline{\mathbf{U}}^0)||, \tag{4.4}$$

where here a tolerance, ϵ, in the range $[10^{-10}, 10^{-7}]$ is used for steady inviscid and laminar flows considered in this study.

4.2 Sparse Non-symmetric Linear Equations and Preconditioned GMRES Method

As noted above, each step of Newton's method requires the solution of a linear problem of the form

$$\mathbf{J}\mathbf{x} = \mathbf{b}, \tag{4.5}$$

where $\mathbf{x} = \Delta\overline{\mathbf{U}}$ and $\mathbf{b} = -\mathbf{R}(\overline{\mathbf{U}})$. This linear system is large, sparse, and non-symmetric and is solved herein using the GMRES iterative algorithm of Saad and co-workers [35–38]. A combination of additive Schwarz global and incomplete lower-upper (ILU) local preconditioning is used to both allow a parallel implementation of the Newton method and to speed up the convergence of the GMRES algorithm. ILU preconditioning with a specified level of fill, f, is used here and, for the problems of interest, ILU(2) with a fill level $f = 2$ is found to work well for the high-order CENO scheme and steady flow problems considered herein. As discussed by Dembo et al. [57], determining the exact solution of Eq. (4.5) is not necessary for rapid convergence of Newton's method and partial convergence of the linear problem can prove to be computationally more efficient. A convergence tolerance of between 0.01 and 0.001 is used here for the iterative solution of t he linear problem in the resulting inexact Newton method.

4.3 Globalization of Newton Method via Implicit-Euler Startup

In order to ensure that the proposed inexact Newton's method is globally convergent for the solution of the steady-state problems of interest here, a so-called switched-evolution-relaxation (SER) startup strategy is used. As proposed by Mulder and Van Leer [58], based on the implicit Euler time marching scheme, the SER approach provides a smooth switching between time-integration of the governing equations and Newton's method. The application

of this startup procedure leads to the solution of a modified linear system of equations of the form

$$\left[\frac{\mathbf{I}}{\Delta\tau^k}+\left(\frac{\partial\mathbf{R}}{\partial\overline{\mathbf{U}}}\right)^k\right]\Delta\overline{\mathbf{U}}^k=-\mathbf{R}^k, \tag{4.6}$$

where $\mathbf{I}$ is the identity matrix and $\Delta\tau^k$ represents a so-called artificial or pseudo time-step. As $\Delta\tau^k\rightarrow\infty$, Newton's method of Eq. (4.2) is recovered. In the SER startup procedure, the time-step $\Delta\tau^k$ varies. Starting from a relatively small and finite value, $\Delta\tau^k$ is gradually increased and made very large as the desired steady solution is approached. As the time step becomes large, quadratic convergence of Newton's method is recovered. The time-step size is determined by considering a combination of the inviscid Courant-Friedrichs-Lewy (CFL) and viscous Neumann stability conditions for the discretized governing equations. The time step, $\Delta\tau^k$, for inexact Newton step, k, is then specified in terms of a time-step multiplier, C^k_{CFLN}, using

$$\Delta\tau^k\leq C^k_{\text{CFLN}}\ \min\left(\frac{\Delta x}{\max(|\mathbf{u}|+a)},\frac{\Delta x^2}{\nu}\right), \tag{4.7}$$

where $\Delta x=V^{1/3}$ is a measure of the grid size, $a=\sqrt{\gamma RT}$ is the sound speed, and ν is the kinematic viscosity of the gas. The time-step multiplier for the k^{th} iteration is then evaluated in terms of the level of convergence using the relation

$$C^k_{\text{CFLN}}=C^0_{\text{CFLN}}\frac{||\mathbf{R}(\overline{\mathbf{U}}^0)||}{||\mathbf{R}(\overline{\mathbf{U}}^k)||}, \tag{4.8}$$

for which the value of C^0_{rmCFLN} is case dependent but selected here to be in the range [0.1, 1]. Additionally, as discussed in Sect. 5 to follow, "freezing" of the smoothness indicator is also applied to aid in the convergence of the Newton method for flow problems with strong shocks.

5 Numerical Results for Inviscid and Viscous Flows

To demonstrate the capabilities of the high-order CENO finite-Volume with high-order boundary and boundary condition treatment and anisotropic AMR schemes used in combination with the efficient inexact Newton method, the application the proposed high-order solution method is now considered for various three-dimensional inviscid and viscous flow problems governed by the Euler and Navier-Stokes equations. In particular, the high-order geometry representation and constrained reconstruction procedure are assessed for steady-state inviscid subsonic flows past both a cylinder and sphere. Numerical results are then presented for three inviscid supersonic steady-state flow cases: a radial outflow problem, flow past a sphere, and flow over a bump in a channel. Two additional subsonic laminar steady-state flow problems are also considered: flow past a circular cylinder and flow past a NACA 0012 symmetric airfoil. Of the five flow problems considered, four involve curved flow geometry for which the high-order geometry treatment of Sect. 2.4 was used. All of the flow problems have been chosen to illustrate the potential benefits of proposed high-order finite-volume scheme with AMR for significantly reducing the computational costs associated with solving steady 3D flows in a robust manner with both smooth and non-smooth solution content. Unless specified, the HLLE flux function with the Venkatakrishnan limiter were used together with the Newton–Krylov–Schwarz algorithm with a GMRES tolerance of 0.001 and ILU(2) local preconditioning to achieve steady-state solutions in all of the sim-

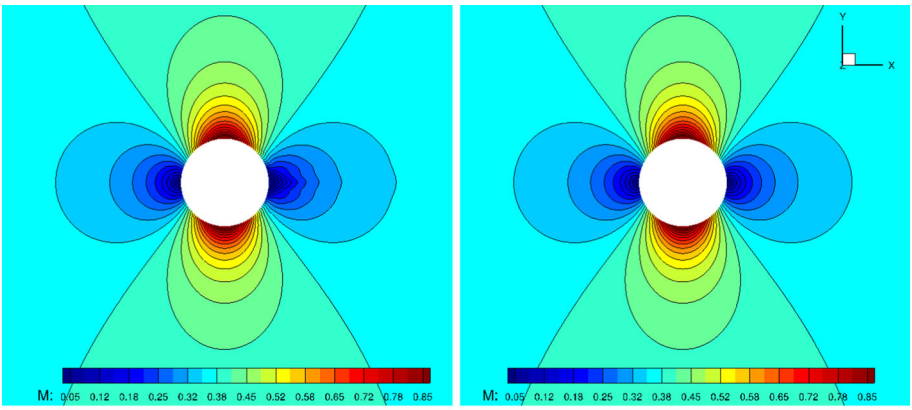


Fig. 9 Inviscid subsonic $M_\infty = 0.38$ flow past a circular cylinder showing the predicted Mach number distributions in the $z = 0$ plane of mesh M_3 obtained using tricubic representation of the hexahedral boundary elements with **a** high-order boundary treatment using ghost cells and **b** high-order constrained reconstruction boundary treatment

ulations. The Newton iterations were stopped once the non-linear residual was reduced by nine orders of magnitude with $\epsilon = 10^{-9}$ and a cut-off value of the smoothness indicator of the CENO scheme used to switch between unlimited fourth-order reconstruction and limited linear reconstruction was chosen to be $\mathcal{S}_\mathcal{C} = 1{,}500$.

5.1 Inviscid Subsonic Flow Past a Circular Cylinder

The accuracy of the high-order geometry and high-order constrained reconstruction is first examined by considering a steady, inviscid, subsonic flow past a circular cylinder with a free-stream Mach number of $M_\infty = 0.38$ where the free-stream flow is in the x-coordinate direction. The radii of the inner and outer boundaries of the cylindrical domain are 1 m and 100 m, respectively. A reflection boundary condition was imposed at the inner circular surface and specified or fixed flow boundary conditions corresponding to the free-stream conditions were imposed on the outer boundary of the domain. The initial solution was assumed to be the uniform free-stream conditions everywhere. A sequence of converged steady-flow solutions were obtained on a series of four uniformly refined meshes with an initial mesh consisting of $B_1 = 32$ blocks with $6 \times 20 \times 8$ cells per block and a total of $M_1 = 30{,}720$ cells. As the flow is 2D in nature, the successive uniformly-refined meshes were refined only in the x- and y-coordinate directions, leading to finer meshes with $B_2 = 128$, $B_3 = 512$, and $B_4 = 2{,}048$ blocks and $M_2 = 122{,}880$, $M_3 = 491{,}520$, and $M_4 = 1{,}966{,}080$ computational cells, respectively. The proposed inexact Newton method provided fully converged solutions of the inviscid subsonic cylinder flow on each of the four grids.

Predicted Mach number distributions for the inviscid subsonic cylinder flow the obtained using the tricubic representation of the hexahedral boundary elements are given in Fig. 9 for mesh M_3. In Fig. 9a, high-order constrained reconstruction is used in the imposition of the boundary conditions at the cylinder boundary and, for the results of Fig. 9b, the ghost-cell approach is used. It is quite noticeable that, for the same mesh resolution, the constrained reconstruction provides a significantly more accurate result than the ghost-cell boundary

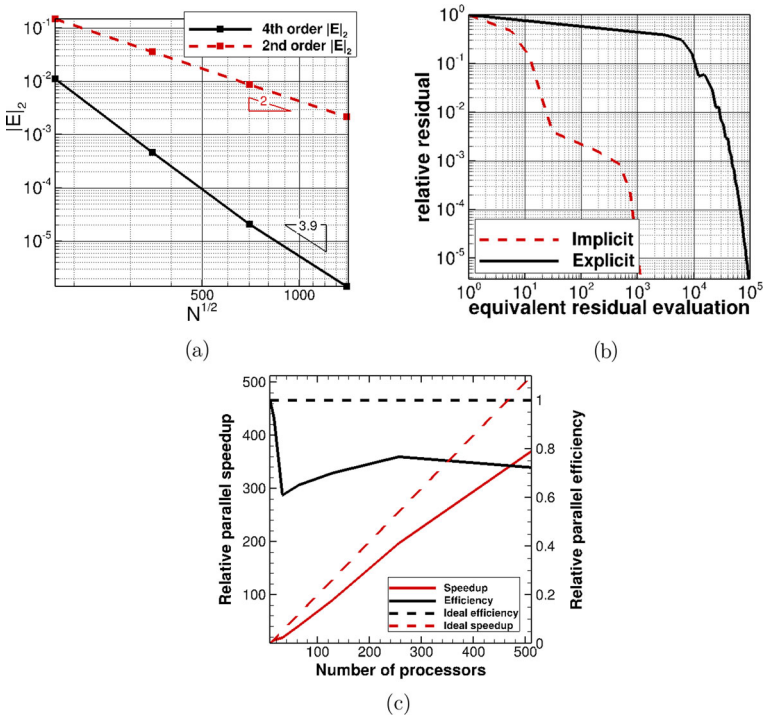


Fig. 10 Inviscid subsonic $M_\infty = 0.38$ flow past a circular cylinder obtained using the high-order geometry representation for boundary elements and constrained reconstruction for boundary-condition evaluation showing the: **a** L_2 norm of the error in the entropy difference, $||S - S_\infty||_2$, obtained with the CENO 4th-order scheme (black colored curve) compared to a standard second-order scheme with piecewise linear reconstruction (red colored curve) as a function of mesh size; **b** convergence history of the inexact Newton method (red colored curve) to the steady-state solution on the third mesh, M_3, compared to that obtained using an explicit, 4th-order, Runge–Kutta, time-marching method (grey dashed line) as measured by the density residual as a function of the equivalent number of residual evaluations; and **c** the relative parallel speed-up, s_p, and parallel efficiency, ε_p, as a function of the number of processors used in the calculation

treatment. In particular, the expected symmetry properties of the Mach number distribution are accurately reproduced using the constrained boundary treatment whereas, for the ghost-cell approach, there are noticeable asymmetries in the upstream and downstream solutions with the downstream solution exhibiting predictions generally associated with excessive numerical dissipation.

As the subsonic cylinder flow is a relatively low-speed inviscid flow that does not contain any discontinuities, the entropy is constant throughout the domain and, therefore, the L_2 norm of difference between the local value of the entropy and its free-stream value, $||S - S_\infty||_2$, is expected to be zero. This entropy difference was used to assess the accuracy of the high-order predictions for this case. The convergence of $||S - S_\infty||_2$ as the computational mesh is refined is given in Fig. 10a for the set of results in which constrained reconstruction was used in the imposition of the boundary conditions on the cylinder surface. The error convergence of the high-order scheme is also compared here to that of a standard second-order scheme with piecewise linear reconstruction. It is evident that the $K = 3$ CENO finite-volume scheme converges at its expected rate of four (black colored curve), thus providing validation of the high-order geometry and constrained reconstruction boundary treatment within the

CENO scheme. Additionally, the second-order method converges at its expected rate of two (red colored curve). The results of Fig. 10a also highlight the computational benefit of the high-order CENO scheme for this case. For the same solution accuracy, many more cells are required by the standard second-order scheme. For example to reach an error of 10^{-2}, just 30,720 cells are required by the 4th-order CENO scheme whereas 491,520 cells are required by the second-order method to achieve this same accuracy. Similar findings have been demonstrated more extensively for the CENO scheme in other previous studies of both inviscid and viscous flows [13, 17, 18].

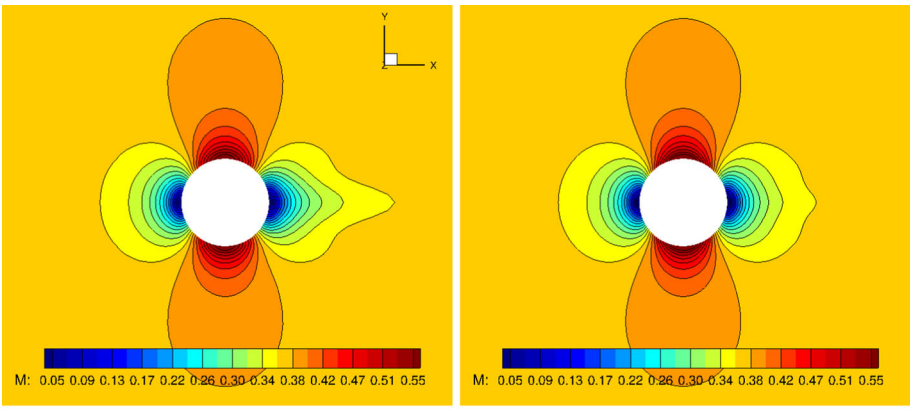
The convergence history of the inexact Newton implicit method used in conjunction with the $K = 3$ CENO finite-volume scheme to the desired steady-state solution of the inviscid subsonic cylinder flow on the third mesh, M_3, is depicted in Fig. 10b as measured by the density residual as a function of the equivalent number of residual evaluations. For comparison, the convergence history obtained using the conditionally-stable, explicit, 4th-order, Runge–Kutta, time-marching method in conjunction with the CENO 4th-order scheme is also shown in the figure. Note that the equivalent number of residual evaluations is related directly to the required computational time for the simulation via the processor time required to evaluate the high-order solution residual, $\mathbf{R}(\overline{\mathbf{U}})$, defined by Eq. (2.3) on the given mesh and is a convenient way to measure and compare the computational effort of various methods across different high-performance computing architectures. The rapid convergence to steady state offered by the Newton method is clearly evident and, compared to the explicit approach, the implicit treatment offers significant computational savings. For the same level of convergence with a tolerance of 10^{-5}, just 900 equivalent residual evaluations, corresponding to approximately 1.9 min of elapsed time using 512 processors, are required to obtain the steady state solution using the inexact Newton method whereas about 90,000 residual evaluations are required when using the explicit time-marching scheme. This represents a substantial reduction in the computational effort by a factor of slightly more than 100. It should also be noted that the proposed high-order CENO scheme combined with inexact Newton method and additive Schwarz preconditioning exhibits relatively high parallel scalability across a large number of processors as illustrated in Fig. 10c, where the relative parallel speed-up, s_p, and parallel efficiency, ε_p, as a function of the number of processors used for inviscid subsonic cylinder flow problem on mesh M_3 is given. The relative parallel speed-up, s_p, and relative parallel efficiency, ε_p, are defined by

$$s_p = \frac{t_1}{t_p} p, \qquad \varepsilon_p = \frac{s_p}{p}, \tag{5.1}$$

where t_p is the total processor time required to solve the problem using p processors and t_1 is the processor time required to solve the problem using a single processor. It can be observed that the relative parallel speedup is nearly linear and the parallel efficiency generally remains above 70–75% for the calculation up to 512 processors, which is rather good for a fully implicit treatment. While it is felt that further investigation of the influences of the various solution parameters is required to fully optimize the parallel performance of the algorithm as a function of mesh size and number of processors, these results are very representative of the capabilities of the proposed high-order solution methodology in terms of parallel performance.

5.2 Inviscid Subsonic Flow Past a Sphere

The second validation flow problem considered here is that of steady, inviscid, subsonic flow around a sphere. The flow geometry in this case is fully three dimensional. As for the previous



(a) (b)

Fig. 11 Inviscid subsonic $M_\infty = 0.38$ flow past a sphere showing the predicted Mach number distributions in the $z = 0$ symmetry plane obtained using tricubic representation of the hexahedral boundary elements with high-order constrained reconstruction boundary treatment on **a** mesh M_3 and **b** M_4 containing 393,216 and 3,145,728 computational cells, respectively

subsonic cylinder flow case, the free-stream Mach number was taken to be $M_\infty = 0.38$ with the flow direction aligned with the x-coordinate direction. The radii of the inner and outer boundaries of the spherical domain were similarly taken to be 1 m and 100 m, respectively. A spherical computational domain was created using the cubed-sphere meshing procedure of Ronchi et al. [59] and implemented here within the block-based AMR framework by Ivan et al. [14, 20]. The spherical domain of the cubed-sphere mesh consists of six sectors (or root blocks) connected with degenerate edges, forming an inner hollow sphere and an outer spherical shell as depicted in Fig. 1b. Reflection and fixed boundary conditions were imposed on the inner and outer surfaces of the spherical shell. For this case, the initial cubed-sphere grid consisted of the six root blocks with $8 \times 8 \times 16$ cells in each block for a total of 6,144 computational cells. The mesh lines were stretched towards the inner sphere. Predicted solutions for the subsonic sphere were then obtained on a series of four uniformly-refined meshes beginning with the initial mesh. The number of blocks and cells associated with each of the four uniformly-refined meshes were $B_1 = 6$, $B_2 = 48$, $B_3 = 384$, and $B_4 = 3{,}072$ and $M_1 = 6{,}144$, $M_2 = 49{,}152$, $M_3 = 393{,}216$, and $M_4 = 3{,}145{,}728$ cells, respectively.

The predicted Mach number distributions obtained using the 4th-order CENO finite-volume scheme with $K = 3$ reconstruction, tricubic representation of the hexahedral boundary elements, and constrained reconstruction for boundary condition treatment are given in Fig. 11a and b for meshes M_3 and M_4, respectively. While the presence of an artificial wake downstream of the sphere can be observed in the inviscid solutions, the wake decreases in size as the mesh is refined and the predictions are also in good agreement, at least qualitatively, with those of Hoshyari [60]. Furthermore, like the subsonic cylinder flow, this spherical flow is also expected to be homentropic with $S - S_\infty = 0$ and predicted relative error norm of the entropy difference, $||S - S_\infty||_2$, is given in Fig. 12 as a function of mesh density. The expected theoretical rate of convergence is again obtained, providing further validation of the high-order geometry and constrained reconstruction boundary treatment with the CENO scheme.

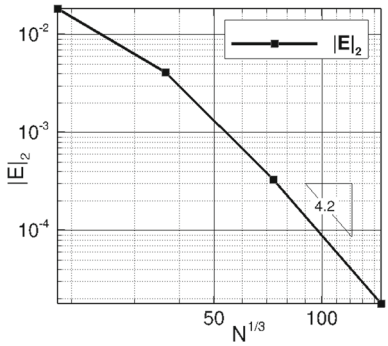


Fig. 12 L_2 norm of the error in the entropy difference, $||S - S_\infty||_2$, versus mesh size for inviscid subsonic $M_\infty = 0.38$ flow past a sphere obtained using the high-order geometry representation for boundary elements and constrained reconstruction for boundary-condition treatment

5.3 Inviscid Supersonic Radial Outflow

Steady, inviscid, supersonic, radial outflow on a 3D spherical domain is considered next with an inflow boundary located at a radius of $R_i = 1$ m and the outflow boundary at a radius of $R_o = 4$ m. In this case, air enters through the inner boundary of the spherical domain with a supersonic velocity in a purely radial direction with magnitude, V_r, subsequently expands, and exits supersonically through the outer boundary of the spherical domain. The inflow is held fixed with a flow density of $\rho_i = 10$ kg/m^3, a radial velocity of $V_{r,i} = 4.5$ m/s, and a pressure of $p_i = 26$ Pa. A ratio of specific heats of $\gamma = 1.4$ is assumed. Note that the flow remains supersonic through the domain and, at the outflow boundary, the gas exits at supersonic speed. An analytical solution for the radial velocity, $V_r(r)$, can be obtained for this steady outflow problem (see Ivan et al. [20]). The solution is given implicitly by the expression

$$C_3 - \frac{1}{r^2 V_r \left[\left(C_2 - V_r^2 \right)^{\frac{1}{\gamma - 1}} \right]} = 0, \tag{5.2}$$

where the constants C_3 and C_2 are given by

$$C_3 = \frac{1}{\left(\frac{2\gamma}{\gamma - 1} \frac{p_i}{\rho_i}^{\frac{1}{\gamma - 1}} \right) R_i^2 V_{r,i}}, \quad C_2 = \frac{2\gamma}{\gamma - 1} \frac{p_i}{\rho_i} + V_{r,i}^2, \tag{5.3}$$

and depend on the inflow conditions. Other flow properties, such as density and pressure, can be obtained assuming steady isentropic flow.

The initial 3D cubed-sphere mesh for the supersonic outflow problem consisted of 4 blocks in the radial direction and 1,536 blocks in each radial block shell for a total of $B_1 = 6{,}144$ blocks each containing $8 \times 8 \times 8$ computational cells. This initial 3D mesh contained $M_1 = 3{,}145{,}728$ hexahedral elements. After the steady numerical solution was obtained on the initial mesh using the inexact Newton, a sequence of four anisotropic AMR procedures were subsequently carried and the steady flow solution was again obtained via the inexact Newton method on each of the refined meshes. The mesh refinement was based on the gradient of the flow density. In this way, numerical predictions of the steady supersonic outflow were obtain on five AMR multi-block hexahedral meshes consisting of $B_1 = 6{,}144$, $B_2 = 7{,}680$, $B_3 = 13{,}824$, $B_4 = 23{,}040$, and $B_5 = 36{,}864$ grid blocks and containing $M_1 = 3{,}145{,}728$, $M_2 = 3{,}932{,}160$, $M_3 = 7{,}077{,}888$, $M_4 = 11{,}796{,}480$ and $M_5 = 18{,}874{,}368$ computational cells. A useful measure of the efficiency of the block-based AMR scheme can

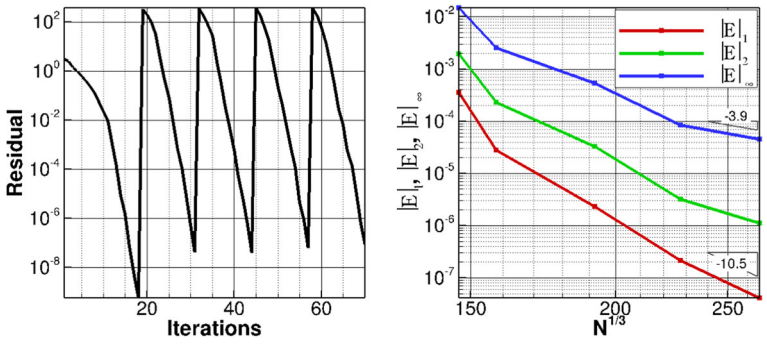


Fig. 13 Steady, inviscid, supersonic, radial outflow showing **a** the convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the 5 anisotropic AMR meshes M_1, M_2, M_3, M_4, and M_5 and **b** the L_1, L_2, L_∞ norms of the error in the flow density compared to the exact solution as a function of mesh size

be defined by a refinement efficiency parameter, ε_Δ, given by

$$\varepsilon_\Delta = 1 - \frac{N_{\text{cells}}}{N_{\text{uniform}}}, \tag{5.4}$$

where N_{cells} is the actual number of cells in the anisotropic AMR mesh and $\text{N}_{\text{uniform}}$ is the total number of cells that would be present in a hypothetical uniform mesh having the same resolution as the finest level of the AMR mesh. While not a direct measure of overall computational efficiency, as it does not directly involve the solution error, refinement efficiency is still a useful indicator of the AMR efficiency. The corresponding refinement efficiencies for the sequence of five anisotropic AMR meshes, M_1, M_2, M_3, M_4, and M_5 are as follows: $\varepsilon_{\Delta,1} = 0.0000$ $\varepsilon_{\Delta,2} = 0.8438$, $\varepsilon_{\Delta,3} = 0.9648$, $\varepsilon_{\Delta,4} = 0.9927$, and $\varepsilon_{\Delta,5} = 0.9985$.

The convergence history of the Newton method for the supersonic radial outflow problem in terms of the density residual as a function of the number of Newton steps required on the initial coarse mesh and the subsequent sequence of adaptively refined anisotropic meshes is given in Fig. 13a. The results shown in the figure demonstrate the rapid convergence of the solution to the steady state on each mesh and that the steady state solution was obtained using less than 20 Newton steps on each mesh. The inexact Newton method is clearly very effective in obtaining the converged steady solutions for this case in an efficient manner. The corresponding predictions of the density distributions for the supersonic outflow obtained using the initial mesh M_1 and mesh M_5 is shown in Fig. 14. The grid blocks are also shown for each mesh. It is evident that anisotropic AMR procedure is able to efficiently refine the 3D mesh in only the radial direction for this purely radial outflow. Indeed, the anisotropic AMR procedure is able to take advantage of the flow varying only in the radial direction and provide convergence rates for the error in the density based on the exact solution that are greater than the theoretical fourth-order accuracy of the underlying CENO finite-volume scheme, as least for L_1 and L_2 norms of the solution error, as shown in Fig. 13b. The high computational efficiency of the combined anisotropic AMR and high-order CENO schemes is also reflected in the values of the refinement efficiency, ε_Δ, reported above for each mesh. As the mesh is adaptively refined, high values of refinement efficiency are achieved in excess of 99% indicating the potential large reduction in mesh size relatively to a uniformly refined approach with equivalent accuracy.

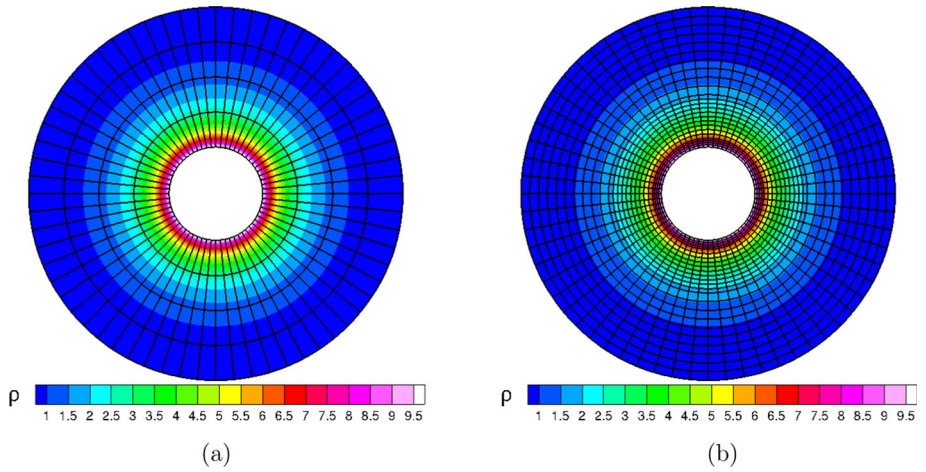
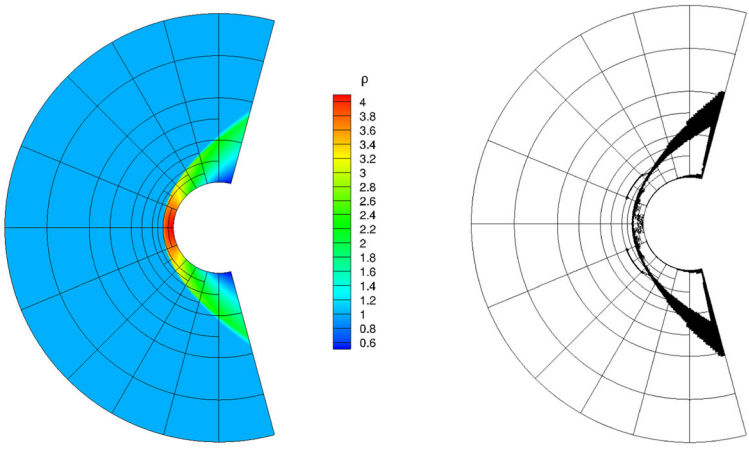


Fig. 14 Steady, inviscid, supersonic, radial outflow showing the predicted density distributions on **a** the initial mesh M_1 and **b** the final anisotropic AMR mesh M_5. The black lines represent edges of the grid blocks and mesh M_1 has $B_1 = 6{,}144$ while mesh M_5 has $B_5 = 36{,}864$ blocks, with each block having $8 \times 8 \times 8 = 512$ cells (Color figure online)

5.4 Inviscid Supersonic Flow Past a Sphere

Steady inviscid flow past a sphere is now again revisited; however, in this next case the free-stream flow is taken to be supersonic. For the case of interest, the free-stream air enters the far-field at a Mach number of $\mathrm{M}_\infty = 2$ with the flow direction in the positive x-coordinate direction. The free-stream values of the pressure and density are $p_\infty = 101.325$ kPa and $\rho_\infty = 1.225$ kg/m^3 and $\gamma = 1.4$ is assumed. The flow geometry for this flow problem consists of a sphere of radius 1 m with a far-field boundary located at a radius of 5 m. A cubed-sphere mesh was again used to represent the domain; however, to restrict the problem size, a cubed-sphere mesh with only 5 sectors, each block containing $16 \times 16 \times 16$ cells was used. Two initial uniform refinements were then applied leading to an initial mesh for the problem having $B_1 = 320$ grid blocks. Reflection boundary conditions were again imposed on the surface of the inner sphere and, at the far-field boundary, the constant or fixed free-stream values were imposed as inflow conditions and a simple linear extrapolation procedure was adopted in the application of outflow boundary conditions for the supersonic air that exits the domain.

The steady $\mathrm{M}_\infty = 2$ supersonic flow past the sphere leads to the formation of a bow shock located upstream of the cylinder with a subsonic flow region following the shock in the vicinity of the upstream stagnation point. The capabilities of the anisotropic AMR scheme to accurate resolve the bow shock was examined here by performing three consecutive anisotropic AMR procedures once a steady solution was first obtained on the initial mesh using the high-order CENO finite-volume scheme in combination with the proposed inexact Newton method. After each level of mesh refinement, a converged solution was obtained on each newly created mesh prior to performing the next refinement of the mesh. Figure 15a depicts the resulting predicted density distribution obtained using the combination of the 4th-order CENO finite-volume and Newton methods on the final anisotropic AMR mesh composed of $B_4 = 1{,}008$ grid blocks and $M_4 = 4{,}128{,}768$ hexahedral cells. Additionally, Fig. 15b shows the corresponding cells of mesh M_4 shaded in blue in which the smoothness



(a) Density field on anisotropic mesh M_4 1,008 blocks (4,128,768 cells)

(b) Smoothness indicator for density, ρ

Fig. 15 Inviscid supersonic $M_\infty = 2$ flow past a sphere showing **a** the predicted density distribution on the final anisotropic AMR mesh containing $B_4 = 1{,}008$ grid blocks and $M_4 = 4{,}128{,}768$ cells and **b** regions (shaded in blue) where the smoothness indicator has deemed the predicted solution for the density to be non-smooth on the final final anisotropic AMR mesh M_4 (Color figure online)

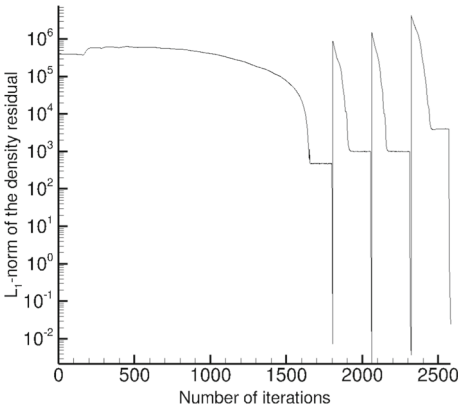


Fig. 16 Inviscid supersonic $M_\infty = 2$ flow past a sphere showing convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the 4 anisotropic AMR meshes M_1, M_2, M_3, and M_4

indicator of the CENO method indicates non-smooth solution content and the reconstruction scheme reverts to limited linear reconstruction. It is quite apparent that the anisotropic AMR mesh for this case is well aligned with the bow shock near the stagnation point and the mesh refinement occurs primarily across the shock. Furthermore, the solution smoothness indicator is also clearly well designed to detect the presence of the shock and low-order reconstruction was performed largely only in cells in the vicinity of the shock and these were found to make up less than 30% of the total number of cells. After four anisotropic AMR procedures, the mesh saving is 99% compared to an equivalent sequence of uniform refinements. Lastly. the convergence of the solution to steady state showing the density residual as a function of the number of Newton steps on all five meshes is depicted in Fig. 16. It is evident that, even for this challenging non-linear problem with a bow shock, rapid convergence of the

solution is obtained on each mesh following a slightly prolonged SER startup phase on the initial mesh. On each successive mesh, the residual decreases by several orders of magnitude before reaching a plateau. For this flow case with strong shocks, the switching associated with the highly non-linear smoothness indicator, whereby the solution content of some cells are alternatively tagged as being smooth or non-smooth (and back and forth), leads to convergence stall. At this point, fully converged solutions are then obtained by "freezing" the values of the the smoothness indicator, which leads to the rapid convergence of the solutions as shown in Fig. 16.

5.5 Inviscid Supersonic Flow Over a Bump in a Channel

The last inviscid flow problem consider herein involves a uniform supersonic flow of air at standard atmospheric density $\rho = 1.225$ kg/m^3 and pressure $p = 101.325$ kPa conditions ($\gamma = 1.4$) with a Mach number of M $= 1.4$ that enters a 2D rectangular channel that contains a sinusoidal bump along the lower surface of the channel. The length and height of the channel are 5.5 m and 2 m, respectively, and the bump is located at a distance of 1 m from the inlet of the channel. The profile of the sinusoidal channel bump along the lower surface is $h(x) = 0.042 \sin^2(\pi x)$ for $x \in [1, 2]$ m with the maximum height of the bump being 4.2 cm. At the inlet to the channel, the the incoming supersonic flow properties are held fixed whereas at the channel outlet, linear extrapolation boundary conditions are imposed for the outgoing supersonic flow. Reflective boundary conditions are imposed along the upper and lower boundaries of the channel, including the bump. Note that the presence of the sinusoidal bump results in the formation of compression waves upstream of the bump that coalesce to form a strong oblique shock. Similarly, a strong oblique shock also forms downstream of the trailing edge of the bump. These shocks subsequently reflect from both the upper and lower boundaries of the channel. The resulting set of complex non-linear wave interactions presents a significant computational challenge to resolve both accurately and efficiently.

The initial mesh for the supersonic channel flow case was composed of 32 grid blocks along the length of the channel and 16 blocks along its height for a total of $B_1 = 512$ blocks and each block contained $8 \times 8 \times 8 = 512$ cells. The resulting initial mesh consisted of $M_1 = 262{,}144$ hexahedral cells. The high-order CENO finite-volume scheme and inexact Newton method were used to obtain a steady-state solution on the initial mesh and then six anisotropic AMR procedures were performed in which a converged steady solution was obtained on each mesh. The gradient of density was used as the criteria for directing the mesh refinement. The resulting seven successive anisotropic AMR meshes, including the initial mesh, contained $M_1 = 262{,}144$, $M_2 = 513{,}024$, $M_3 = 879{,}104$, $M_4 = 1{,}809{,}408$, $M_5 = 3{,}538{,}432$, $M_6 = 6{,}959{,}616$, and $M_7 = 13{,}643{,}776$ computational cells. The corresponding refinement efficiencies for the sequence of seven anisotropic AMR meshes are $\varepsilon_{\Delta,1} = 0.0000$, $\varepsilon_{\Delta,2} = 0.7554$, $\varepsilon_{\Delta,3} = 0.9476$, $\varepsilon_{\Delta,4} = 0.9865$, $\varepsilon_{\Delta,5} = 0.9967$, $\varepsilon_{\Delta,6} = 0.9992$, and $\varepsilon_{\Delta,7} = 0.9998$, respectively.

Figure 17a provides the predicted density distribution obtained for the supersonic bump flow on the final mesh M_7. The grid blocks of the anisotropic AMR mesh M_7 are also shown in the figure. Additionally, Fig. 17b shows the corresponding cells of mesh M_7 shaded in white in which the smoothness indicator is flagging non-smooth solution content and the reconstruction scheme reverts to the lower-order reconstruction. From Fig. 17, it is evident that the anisotropic AMR procedure enriches the mesh for both the shocks emanating from the leading edge as well as the trailing edge of the bump, as well as for the reflected shocks emanating from the upper and lower channel boundaries. From the values of the refinement

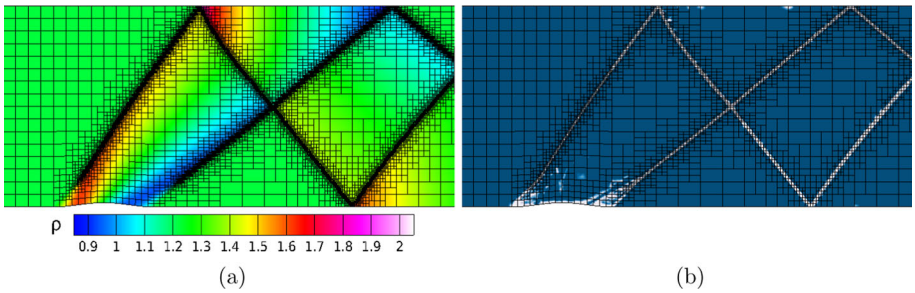


Fig. 17 Inviscid supersonic M = 1.4 flow over a bump showing **a** the predicted density distributions obtained on the final anisotropic AMR mesh composed of $M_7 = 13{,}643{,}776$ computational cells (grid block boundaries are represented by black lines) and **b** the regions (shaded in white) where the smoothness indicator has deemed the predicted solution for the density to be non-smooth on the same final final mesh M_7 (Color figure online)

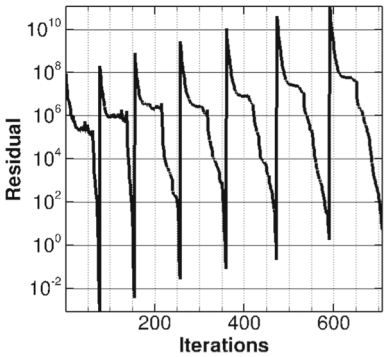


Fig. 18 Inviscid supersonic M = 1.4 flow over a bump showing the convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the seven anisotropic AMR meshes M_1, M_2, M_3, M_4, M_5, M_6, and M_7

efficiency, ε_Δ, for the seven AMR meshes given above, this enrichment of the mesh is introduced only locally and such that the refinement efficiency rapidly exceeds 99% as the mesh is refined, suggesting very efficient treatment of the discontinuous solution content compared to a uniform mesh approach. It can also be seen that, as desired, the spatial discretization procedure is fully high-order everywhere within the computational domain except for the narrow regions surrounding the shocks. Finally, the convergence history for the simulations showing the density residual as a function of the number of Newton steps for each of the seven meshes is provided in Fig. 18. It can be observed that converged steady high-order solutions are obtained quite efficiently on all seven meshes in less than 100–125 Newton steps for this compressible channel flow with multiple shocks.

5.6 Subsonic Laminar Flow Past a Circular Cylinder

The application of the high-order CENO finite-volume scheme combined with anisotropic AMR and inexact Newton schemes to steady subsonic laminar flow past a circular cylinder is also examined here. The free-stream Mach and Reynolds numbers for this laminar flow are $\mathrm{M}_\infty = 0.1$ and $\mathrm{Re} = 30$. In this case, the flow-field domain consists of the region between two concentric cylinders where the inner and outer cylinder diameters are $D_i = 0.0001$ m and $F_o = 0.004$ m, respectively. The length of the extruded domain in the third direction was selected to be $\ell = 0.0001$ m. Fixed values for the free-stream flow properties were imposed on the outer cylindrical boundary of the domain and the imposed boundary conditions for

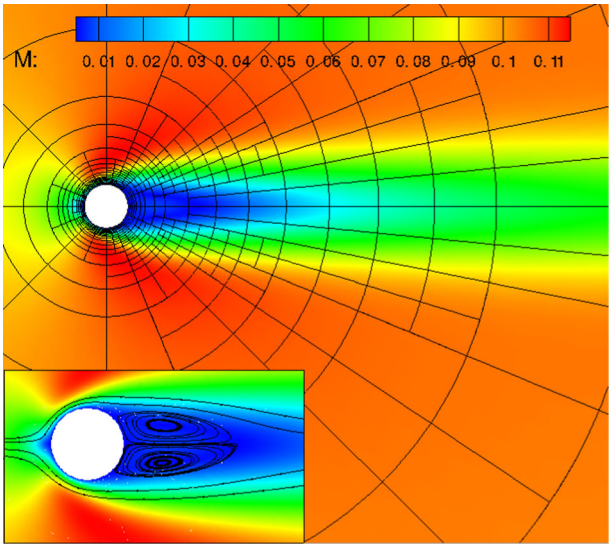


Fig. 19 Laminar subsonic flow past a circular cylinder with free-stream Mach number of $M_\infty = 0.1$ and Reynolds number of $Re = 30$ showing the predicted distribution of the flow Mach number obtained using the 4th-order CENO scheme on mesh M_5 with 518,144 cells. A close-up view of the solution is also given depicting the streamlines in the trailing edge region downstream of the cylinder

the inner cylinder were standard no slip boundary conditions for an adiabatic wall. A body-fitted O-type grid with non-uniform spacing and clustering of the grid lines toward the inner cylinder was used for the initial mesh. This initial mesh consisted of $B_1 = 70$ grid blocks and $M_1 = 71,680$ computational cells. Four successive anisotropic AMR procedures were then applied to the converged solution of the CENO finite-volume scheme and inexact Newton method obtained on this initial O-type mesh. The five AMR grids in the refinement sequence consisted of $M_1 = 71{,}680$, $M_2 = 79{,}872$, $M_3 = 112{,}640$, $M_4 = 227{,}328$, $M_5 = 518{,}144$ computational cells, respectively. On each mesh, converged steady-state solution was obtained using the 4th-order CENO finite-volume scheme with $K = 4$ quartic reconstruction.

The predicted Mach number distribution for the steady subsonic laminar flow past the cylinder obtained using the 4th-order CENO finite-volume scheme on body-fitted mesh M_5 is given in Fig. 19. Only the inner cylinder is shown in the figure. The re-circulating flow regions behind the cylinder following separation can be clearly identified by the streamlines depicted in Fig. 19. Furthermore, as indicated in Fig. 20a, the predicted drag coefficients, C_D, obtained on the sequence of five computational meshes by the high-order solution method were $C_D = 1.707$ for mesh M_1, $C_D = 1.733$ for mesh M_2, $C_D = 1.743$ for mesh M_3, $C_D = 1.749$ for mesh M_4, and $C_D = 1.752$ for mesh M_5. The predicted drag coefficient can be seen to converging as the mesh is refined and predicted value of C_D is in good agreement with the curve fits of Henderson [61], which yield a value of $C_D = 1.737$ for this case. The convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the 5 anisotropic AMR meshes is given in Fig. 20b. It can be seen that the Newton method remains effective for viscous laminar flows, providing the converged steady solution in less than 50 Newton steps even on the finest grid.

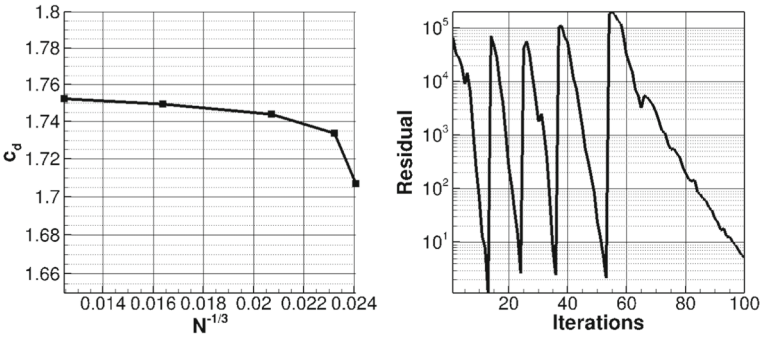


Fig. 20 Laminar subsonic flow past a circular cylinder with free-stream Mach number of $M_\infty = 0.1$ and Reynolds number of $Re = 30$ showing **a** the predicted drag coefficients, C_D, obtained using the 4th-order CENO scheme as a function of the mesh density, $\Delta x = N^{-1/3}$, on the 5 anisotropic AMR meshes M_1, M_2, M_3, M_4 and M_5 and **b** the convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the 5 meshes

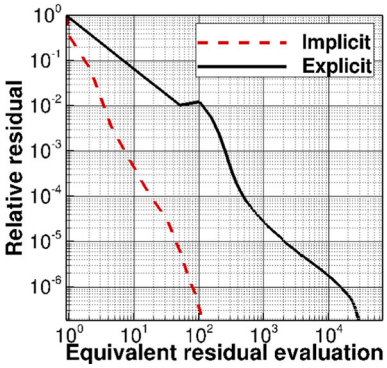


Fig. 21 Laminar subsonic flow past a circular cylinder with free-stream Mach number of $M_\infty = 0.1$ and Reynolds number of $Re = 30$ showing the convergence history of the inexact Newton method (red colored curve) to the steady-state solution on the third mesh, M_3, compared to that obtained using an explicit, 4th-order, Runge–Kutta, time-marching method (grey dashed line) as measured by the density residual as a function of the equivalent number of residual evaluations (Color figure online)

A comparison of the convergence history obtained using the proposed inexact Newton method to that of the 4th-order, explicit, time-marching scheme is also given in Fig. 21 for this subsonic laminar flow case. The convergence to steady state as measured by the density residual as a function of the equivalent number of residual evaluations for the third mesh, M_3, is shown. It can be seen that only about 105 equivalent residual evaluations, corresponding to approximately 2.6 min of elapsed time using 78 processors, are required by the Newton method to reach a convergence tolerance of $5(10)^{-7}$ while more than 30,000 residual evaluations are needed to reach this same convergence level using the explicit time-marching scheme. As for the inviscid flow cases, the inexact Newton method provides substantial computational savings when obtaining steady solutions with the high-order CENO scheme. A rapidly converged, steady, high-order solution of the subsonic laminar cylinder flow on grid M_3 is obtained in using the implicit Newton method with a reduction in computational time by a factor of more than 285 compared to that required by the explicit time-marching scheme. Furthermore, the parallel performance of the proposed Newton method with CENO finite-volume scheme is also high for this viscous flow case with a parallel efficiency similar to that demonstrated for the inviscid subsonic cylinder flow case.

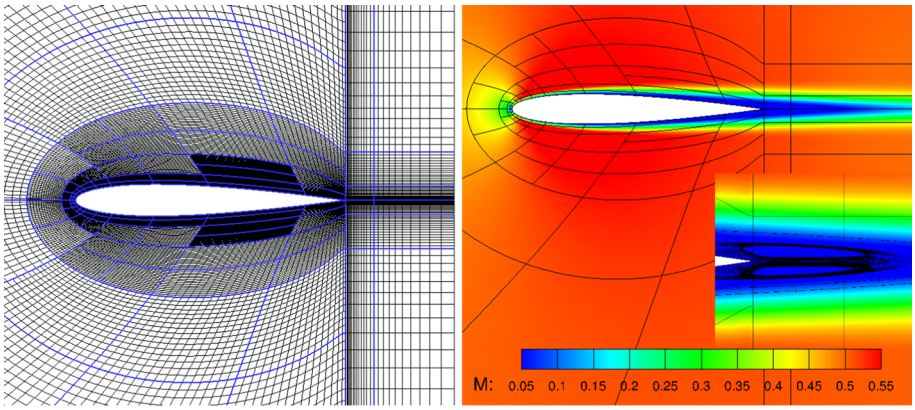


Fig. 22 Laminar subsonic flow past a NACA 0012 symmetric airfoil at zero angle of attack with free-stream Mach number of $M_\infty = 0.5$ and Reynolds number of $Re = 5{,}000$ showing **a** the final anisotropic AMR mesh with $B_4 = 144$ blocks and $M_4 = 294{,}912$ computational cells (thicker blue lines represent the grid block boundaries) and **b** the predicted flow Mach number distribution obtained with the 4th-order CENO scheme on the final anisotropic AMR mesh M_4 (Color figure online)

5.7 Subsonic Laminar Flow Past a NACA 0012 Symmetric Airfoil

A second laminar flow problem is also considered here. In this last test case, steady, subsonic, laminar flow past a NACA 0012 symmetric airfoil at zero angle of attack is examined. The free-stream Mach and Reynolds numbers for this laminar flow are $M_\infty = 0.5$ and $Re = 5,000$, respectively. This subsonic airfoil flow has been examined in previous studies [13, 62] and it has been well established that flow separation occurs near the trailing edge of the airfoil resulting in the formation of small re-circulation bubbles on the upper and lower surfaces of the airfoil.

The initial computational mesh for the airfoil computations was a C-type grid. The outer boundary of the domain was positioned at about 24 chord lengths from the airfoil surface and the length of the domain in the perpendicular direction of the airfoil was one chord length. Fixed free-stream flow conditions were imposed at the outer boundary and no slip, adiabatic wall, boundary conditions were imposed on the airfoil surface. The initial grid consisted of 128 cells along the upper and lower surface of the airfoil (64 cells for each side), 64 cells in the direction normal to the surface and just 8 cells in the direction normal to 2D plane of the airfoil. The cells of the initial grid were grouped into $B_1 = 64$ grid blocks, each consisting of $16 \times 16 \times 8$ cells, for a total of $M_1 = 131,072$ cells. The grid lines in the direction normal to the airfoil were clustered such that cell aspect ratios near the airfoil surface was around five.

After obtaining a steady solution on mesh M_1 using the CENO finite-volume scheme, a sequence of three higher-resolution meshes for the airfoil were generated by applying the anisotropic AMR procedure to arriving at three meshes having $B_2 = 74$, $B_3 = 88$, and $B_4 = 144$ grid blocks and $M_2 = 151{,}152$, $M_3 = 180{,}224$, and $M_4 = 294{,}912$ computational cells, respectively. Converged steady solutions were obtained using the proposed 4th-order CENO finite-volume scheme with quartic reconstruction in combination with the inexact Newton method. Figure 22a depicts the final anisotropic AMR grid with $M_4 = 294{,}912$ cells, showing the grid structure in the vicinity of the airfoil. The trailing edge recirculation

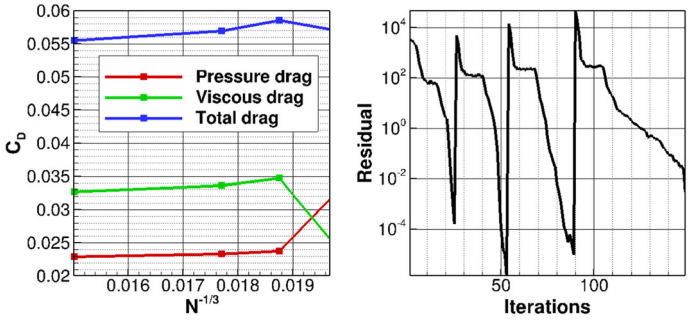


Fig. 23 Laminar subsonic flow past a NACA 0012 symmetric airfoil at zero angle of attack with free-stream Mach number of $M_\infty = 0.5$ and Reynolds number of $Re = 5,000$ showing **a** the predicted pressure drag, $C_{D,p}$, viscous drag, $C_{D,f}$ and total drag, C_D, coefficients obtained using the 4th-order CENO scheme in combination with the anisotropic AMR and inexact Newton methods as a function of the mesh density ($\Delta x = N^{-1/3}$) on meshes M_1, M_2, M_3, and M_4 and **b** the convergence history of the inexact Newton method to the steady-state solution as measured by the density residual as a function of the number of Newton steps on the 4 meshes

bubbles are also shown. Figure 22b provides the corresponding predicted spatial distribution of the flow Mach number obtained on the final mesh M_4. The convergence of the predicted coefficients of pressure drag, $C_{D,p}$, viscous drag, $C_{D,f}$, and total drag, C_D, as a function of the mesh resolution are given in Fig. 23a for each of the four grids. All of these quantities appear to be converging as the mesh is refined and the value of the total drag coefficient, C_D, predicted by the 4th-order CENO scheme on mesh M_4 is approximately 0.055541, which compares well with the value reported in similar previous studies [13, 62]. Lastly, the solution history of the Newton method on each of the four meshes is shown in Fig. 23b and rapid solution convergence was obtained in 25, 28, 36 and 60 Newton iterations in each case, respectively.

6 Conclusions

A fourth-order CENO finite-volume method combined with an efficient anisotropic block-based AMR scheme and an inexact Newton method has been proposed and described for significantly reducing the computational costs associated with the solution of steady flows governed by the Euler and Navier–Stokes equations on 3D, multi-block, body-fitted, hexahedral meshes. High-order representation of the mesh geometry has been introduced in combination with high-order constrained reconstruction for the imposition of boundary data, thus enabling the overall proposed solution method to account accurately for 3D and curved flow geometries. The proposed fourth-order AMR finite-volume scheme is readily applicable to cubed-sphere meshes and numerical results have been presented and compared to analytical solutions for several benchmark problems. For these cases, the formal accuracy of the high-order CENO method has been established and the high-order scheme was shown to outperform standard second-order methods in terms of mesh size for a given solution accuracy. Furthermore, rapid, high parallel scalability, and efficient convergence of the Newton method for steady problems has been demonstrated relative to more commonly applied explicit time-marching methods, along with the ability of the anisotropic AMR technique to

provide efficient local refinement of multi-block grids, for both smooth steady flow problems as well as steady flows with strong shocks. In terms of AMR performance, high values of refinement efficiency were achieved for a number of cases. Future research will consider the application of the proposed methodology to fully turbulent flows in conjunction with appropriate modelling for unresolved turbulence.

Acknowledgements This work was supported by the Canadian Space Agency and by the Natural Sciences and Engineering Research Council (NSERC) of Canada. In particular, the authors would like to acknowledge the financial support received from the Canadian Space Agency through the Geospace Observatory Canada program under grant number 14SUGOAMSM. Computational resources for performing all of the calculations reported herein were provided by the SciNet High Performance Computing Consortium at the University of Toronto and Compute/Calcul Canada through funding from the Canada Foundation for Innovation (CFI) and the Province of Ontario, Canada.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Appendix A. Appendix

A.1 Trilinear Transformation

The trilinear transformation is defined by

$$\overrightarrow{\mathbf{X}}(p,q,r) = \mathbf{T_1} + \mathbf{T_2}p + \mathbf{T_3}q + \mathbf{T_4}r + \mathbf{T_5}pq + \mathbf{T_6}pr + \mathbf{T_7}qr + \mathbf{T_8}pqr, \tag{A.1}$$

where p, q and r are the Cartesian coordinates in the canonical space of the reference cube. The transformation vector coefficients for the trilinear transformation are defined by $\mathbf{T_1}$, $\mathbf{T_2}$, $\mathbf{T_3}$, $\mathbf{T_4}$, $\mathbf{T_5}$, $\mathbf{T_6}$, $\mathbf{T_7}$ and $\mathbf{T_8}$ which are in turn given by

$$\begin{aligned}
\mathbf{T_1} &= 1/8\ (\mathbf{N_1} + \mathbf{N_2} + \mathbf{N_3} + \mathbf{N_4} + \mathbf{N_5} + \mathbf{N_6} + \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_2} &= 1/8\ (\mathbf{N_2} - \mathbf{N_1} - \mathbf{N_3} - \mathbf{N_4} + \mathbf{N_5} + \mathbf{N_6} - \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_3} &= 1/8\ (\mathbf{N_3} - \mathbf{N_2} - \mathbf{N_1} - \mathbf{N_4} + \mathbf{N_5} - \mathbf{N_6} + \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_4} &= 1/8\ (\mathbf{N_4} - \mathbf{N_2} - \mathbf{N_3} - \mathbf{N_1} - \mathbf{N_5} + \mathbf{N_6} + \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_5} &= 1/8\ (\mathbf{N_1} - \mathbf{N_2} - \mathbf{N_3} + \mathbf{N_4} + \mathbf{N_5} - \mathbf{N_6} - \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_6} &= 1/8\ (\mathbf{N_1} - \mathbf{N_2} + \mathbf{N_3} - \mathbf{N_4} - \mathbf{N_5} + \mathbf{N_6} - \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_7} &= 1/8\ (\mathbf{N_1} + \mathbf{N_2} - \mathbf{N_3} - \mathbf{N_4} - \mathbf{N_5} - \mathbf{N_6} + \mathbf{N_7} + \mathbf{N_8}),\\
\mathbf{T_8} &= 1/8\ (\mathbf{N_2} - \mathbf{N_1} + \mathbf{N_3} + \mathbf{N_4} - \mathbf{N_5} - \mathbf{N_6} - \mathbf{N_7} + \mathbf{N_8}),
\end{aligned}$$

where $\mathbf{N_1}$, $\mathbf{N_2}$, $\mathbf{N_3}$, $\mathbf{N_4}$, $\mathbf{N_5}$, $\mathbf{N_6}$, $\mathbf{N_7}$ and $\mathbf{N_8}$ are the vertices of the hexahedron as depicted in Fig. 24a. The tangent vectors to the coordinate lines are defined by

$$\begin{aligned}
\frac{\partial \overrightarrow{\mathbf{X}}(p,q,r)}{\partial p} &\equiv \overrightarrow{\mathbf{X}}_p(q,r) = \mathbf{T_2} + \mathbf{T_5}q + \mathbf{T_6}r + \mathbf{T_8}qr,\\
\frac{\partial \overrightarrow{\mathbf{X}}(p,q,r)}{\partial q} &\equiv \overrightarrow{\mathbf{X}}_q(p,r) = \mathbf{T_3} + \mathbf{T_5}p + \mathbf{T_7}r + \mathbf{T_8}pr,\\
\frac{\partial \overrightarrow{\mathbf{X}}(p,q,r)}{\partial r} &\equiv \overrightarrow{\mathbf{X}}_r(p,q) = \mathbf{T_4} + \mathbf{T_6}p + \mathbf{T_7}q + \mathbf{T_8}pq.
\end{aligned}$$

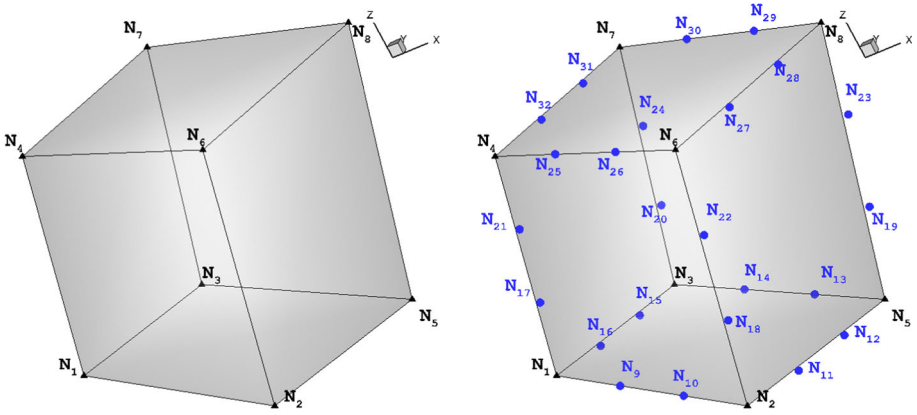


Fig. 24 **a** Second-order accurate trilinear representation of generic curved geometry. The black nodes at the vertices represent the second-order accurate basis functions. **b** Fourth-order accurate tricubic representation of generic curved geometry. The blue nodes denote the extra nodes (Color figure online)

A.2 Tricubic Transformation

The tricubic transformation is defined by

$$\begin{aligned}\vec{\mathbf{X}}(p,q,r) = {} & \mathbf{T_1} + \mathbf{T_2}p + \mathbf{T_3}q + \mathbf{T_4}r + \mathbf{T_5}p^2 + \mathbf{T_6}q^2 + \mathbf{T_7}r^2 + \mathbf{T_8}p^3 \\ & + \mathbf{T9}q^3 + \mathbf{T10}r^3 + \mathbf{T11}pq + \mathbf{T12}pr + \mathbf{T13}qr + \mathbf{T14}p^2q \\ & + \mathbf{T15}pq^2 + \mathbf{T16}p^2r + \mathbf{T17}q^2r + \mathbf{T18}pr^2 + \mathbf{T19}qr^2 + \mathbf{T20}p^3q \\ & + \mathbf{T21}pq^3 + \mathbf{T22}p^3r + \mathbf{T23}q^3r + \mathbf{T24}pr^3 + \mathbf{T25}qr^3 + \mathbf{T26}p^2qr \\ & + \mathbf{T27}p^2qr + \mathbf{T28}pq^2r + \mathbf{T29}pqr^2 + \mathbf{T30}p^3qr + \mathbf{T31}pq^3r + \mathbf{T32}pqr^3,\end{aligned} \tag{A.2}$$

and the corresponding tangent vectors to the coordinate lines are defined by

$$\begin{aligned}\frac{\partial \vec{\mathbf{X}}(p,q,r)}{\partial p} \equiv \vec{\mathbf{X}}_p(q,r) = {} & \mathbf{T_2} + 2\mathbf{T_5}p + 3\mathbf{T_8}p^2 + \mathbf{T11}q + \mathbf{T12}r + 2\mathbf{T14}pq + \mathbf{T15}q^2 + 2\mathbf{T16}pr \\ & + \mathbf{T18}r^2 + 3\mathbf{T20}p^2q + \mathbf{T21}q^3 + 3\mathbf{T22}p^2r + \mathbf{T24}r^3 + 2\mathbf{T26}pqr \\ & + 2\mathbf{T27}pqr + \mathbf{T28}q^2r + \mathbf{T29}qr^2 + 3\mathbf{T30}p^2qr + \mathbf{T31}q^3r + \mathbf{T32}qr^3, \\ \frac{\partial \vec{\mathbf{X}}(p,q,r)}{\partial q} \equiv \vec{\mathbf{X}}_q(p,r) = {} & \mathbf{T_3} + 2\mathbf{T_6}q + 3\mathbf{T9}q^2 + \mathbf{T11}p + \mathbf{T13}r + \mathbf{T14}p^2 + 2\mathbf{T15}pq + 2\mathbf{T17}qr \\ & + \mathbf{T19}r^2 + \mathbf{T20}p^3 + 3\mathbf{T21}pq^2 + 3\mathbf{T23}q^2r + \mathbf{T25}r^3 + \mathbf{T26}p^2r \\ & + \mathbf{T27}p^2r + 2\mathbf{T28}pqr + \mathbf{T29}pr^2 + \mathbf{T30}p^3r + 3\mathbf{T31}pq^2r + \mathbf{T32}pr^3, \\ \frac{\partial \vec{\mathbf{X}}(p,q,r)}{\partial r} \equiv \vec{\mathbf{X}}_r(p,q) = {} & \mathbf{T_4} + 2\mathbf{T_7}r + 3\mathbf{T10}r^2 + \mathbf{T12}p + \mathbf{T13}q + \mathbf{T16}p^2 + \mathbf{T17}q^2 + 2\mathbf{T18}pr \\ & + 2\mathbf{T19}qr + \mathbf{T22}p^3 + \mathbf{T23}q^3 + 3\mathbf{T24}pr^2 + 3\mathbf{T25}qr^2 + \mathbf{T26}p^2q \\ & + \mathbf{T27}p^2q + \mathbf{T28}pq^2 + 2\mathbf{T29}pqr + \mathbf{T30}p^3q + \mathbf{T31}pq^3 + 3\mathbf{T32}pqr^2.\end{aligned}$$

The transformation vector coefficients is this case are given by

$$\begin{aligned}
\mathbf{T_1} &= 1/64\ (9\mathbf{N_9} - 19\mathbf{N_2} - 19\mathbf{N_3} - 19\mathbf{N_4} - 19\mathbf{N_5} - 19\mathbf{N_6} - 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad -19\mathbf{N_1} + 9\mathbf{N_{10}} + 9\mathbf{N_{11}} + 9\mathbf{N_{12}} + 9\mathbf{N_{13}} + 9\mathbf{N_{14}} + 9\mathbf{N_{15}} + 9\mathbf{N_{16}} \\
&\quad +9\mathbf{N_{17}} + 9\mathbf{N_{18}} + 9\mathbf{N_{19}} + 9\mathbf{N_{20}} + 9\mathbf{N_{21}} + 9\mathbf{N_{22}} + 9\mathbf{N_{23}} + 9\mathbf{N_{24}} \\
&\quad +9\mathbf{N_{25}} + 9\mathbf{N_{26}} + 9\mathbf{N_{27}} + 9\mathbf{N_{28}} + 9\mathbf{N_{29}} + 9\mathbf{N_{30}} + 9\mathbf{N_{31}} + 9\mathbf{N_{32}}) \\
\mathbf{T_2} &= 1/64\ (19\mathbf{N_1} - 19\mathbf{N_2} + 19\mathbf{N_3} + 19\mathbf{N_4} - 19\mathbf{N_5} - 19\mathbf{N_6} + 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad -27\mathbf{N_9} + 27\mathbf{N_{10}} + 9\mathbf{N_{11}} + 9\mathbf{N_{12}} + 27\mathbf{N_{13}} - 27\mathbf{N_{14}} - 9\mathbf{N_{15}} - 9\mathbf{N_{16}} \\
&\quad -9\mathbf{N_{17}} + 9\mathbf{N_{18}} + 9\mathbf{N_{19}} - 9\mathbf{N_{20}} - 9\mathbf{N_{21}} + 9\mathbf{N_{22}} + 9\mathbf{N_{23}} - 9\mathbf{N_{24}} \\
&\quad -27\mathbf{N_{25}} + 27\mathbf{N_{26}} + 9\mathbf{N_{27}} + 9\mathbf{N_{28}} + 27\mathbf{N_{29}} - 27\mathbf{N_{30}} - 9\mathbf{N_{31}} - 9\mathbf{N_{32}}) \\
\mathbf{T_3} &= 1/64\ (19\mathbf{N_1} + 19\mathbf{N_2} - 19\mathbf{N_3} + 19\mathbf{N_4} - 19\mathbf{N_5} + 19\mathbf{N_6} - 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad -9\mathbf{N_9} - 9\mathbf{N_{10}} - 27\mathbf{N_{11}} + 27\mathbf{N_{12}} + 9\mathbf{N_{13}} + 9\mathbf{N_{14}} + 27\mathbf{N_{15}} - 27\mathbf{N_{16}} \\
&\quad -9\mathbf{N_{17}} - 9\mathbf{N_{18}} + 9\mathbf{N_{19}} + 9\mathbf{N_{20}} - 9\mathbf{N_{21}} - 9\mathbf{N_{22}} + 9\mathbf{N_{23}} + 9\mathbf{N_{24}} \\
&\quad -9\mathbf{N_{25}} - 9\mathbf{N_{26}} - 27\mathbf{N_{27}} + 27\mathbf{N_{28}} + 9\mathbf{N_{29}} + 9\mathbf{N_{30}} + 27\mathbf{N_{31}} - 27\mathbf{N_{32}}) \\
\mathbf{T_4} &= 1/64\ (9\mathbf{N_1} + 9\mathbf{N_2} + 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad -9\mathbf{N_9} - 9\mathbf{N_{10}} - 9\mathbf{N_{13}} - 9\mathbf{N_{14}} - 9\mathbf{N_{25}} - 9\mathbf{N_{26}} - 9\mathbf{N_{29}} - 9\mathbf{N_{30}}) \\
\mathbf{T_5} &= 1/64\ (19\mathbf{N_1} + 19\mathbf{N_2} + 19\mathbf{N_3} - 19\mathbf{N_4} + 19\mathbf{N_5} - 19\mathbf{N_6} - 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad -9\mathbf{N_9} - 9\mathbf{N_{10}} - 9\mathbf{N_{11}} - 9\mathbf{N_{12}} - 9\mathbf{N_{13}} - 9\mathbf{N_{14}} - 9\mathbf{N_{15}} - 9\mathbf{N_{16}} \\
&\quad -27\mathbf{N_{17}} - 27\mathbf{N_{18}} - 27\mathbf{N_{19}} - 27\mathbf{N_{20}} + 27\mathbf{N_{21}} + 27\mathbf{N_{22}} + 27\mathbf{N_{23}} + 27\mathbf{N_{24}} \\
&\quad +9\mathbf{N_{25}} + 9\mathbf{N_{26}} + 9\mathbf{N_{27}} + 9\mathbf{N_{28}} + 9\mathbf{N_{29}} + 9\mathbf{N_{30}} + 9\mathbf{N_{31}} + 9\mathbf{N_{32}}) \\
\mathbf{T_6} &= 1/64\ (9\mathbf{N_1} + 9\mathbf{N_2} + 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad -9\mathbf{N_{11}} - 9\mathbf{N_{12}} - 9\mathbf{N_{15}} - 9\mathbf{N_{16}} - 9\mathbf{N_{27}} - 9\mathbf{N_{28}} - 9\mathbf{N_{31}} - 9\mathbf{N_{32}}) \\
\mathbf{T_7} &= 1/64\ (9\mathbf{N_1} + 9\mathbf{N_2} + 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad -9\mathbf{N_{17}} - 9\mathbf{N_{18}} - 9\mathbf{N_{19}} - 9\mathbf{N_{20}} - 9\mathbf{N_{21}} - 9\mathbf{N_{22}} - 9\mathbf{N_{23}} - 9\mathbf{N_{24}}) \\
\mathbf{T_8} &= 1/64\ (9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_3} - 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad +27\mathbf{N_9} - 27\mathbf{N_{10}} - 27\mathbf{N_{13}} + 27\mathbf{N_{14}} + 27\mathbf{N_{25}} - 27\mathbf{N_{26}} - 27\mathbf{N_{29}} + 27\mathbf{N_{30}}) \\
\mathbf{T_9} &= 1/64\ (9\mathbf{N_3} - 9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad +27\mathbf{N_{11}} - 27\mathbf{N_{12}} - 27\mathbf{N_{15}} + 27\mathbf{N_{16}} + 27\mathbf{N_{27}} - 27\mathbf{N_{28}} - 27\mathbf{N_{31}} + 27\mathbf{N_{32}}) \\
\mathbf{T_{10}} &= 1/64\ (9\mathbf{N_4} - 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_1} - 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8} \\
&\quad +27\mathbf{N_{17}} + 27\mathbf{N_{18}} + 27\mathbf{N_{19}} + 27\mathbf{N_{20}} - 27\mathbf{N_{21}} - 27\mathbf{N_{22}} - 27\mathbf{N_{23}} - 27\mathbf{N_{24}}) \\
\mathbf{T_{11}} &= 1/64\ (19\mathbf{N_2} - 19\mathbf{N_1} + 19\mathbf{N_3} - 19\mathbf{N_4} - 19\mathbf{N_5} + 19\mathbf{N_6} + 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad +27\mathbf{N_9} - 27\mathbf{N_{10}} - 27\mathbf{N_{11}} + 27\mathbf{N_{12}} + 27\mathbf{N_{13}} - 27\mathbf{N_{14}} - 27\mathbf{N_{15}} + 27\mathbf{N_{16}} \\
&\quad +9\mathbf{N_{17}} - 9\mathbf{N_{18}} + 9\mathbf{N_{19}} - 9\mathbf{N_{20}} + 9\mathbf{N_{21}} - 9\mathbf{N_{22}} + 9\mathbf{N_{23}} - 9\mathbf{N_{24}} \\
&\quad +27\mathbf{N_{25}} - 27\mathbf{N_{26}} - 27\mathbf{N_{27}} + 27\mathbf{N_{28}} + 27\mathbf{N_{29}} - 27\mathbf{N_{30}} - 27\mathbf{N_{31}} + 27\mathbf{N_{32}}) \\
\mathbf{T_{12}} &= 1/64\ (19\mathbf{N_2} - 19\mathbf{N_1} - 19\mathbf{N_3} + 19\mathbf{N_4} + 19\mathbf{N_5} - 19\mathbf{N_6} + 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad +27\mathbf{N_9} - 27\mathbf{N_{10}} - 9\mathbf{N_{11}} - 9\mathbf{N_{12}} - 27\mathbf{N_{13}} + 27\mathbf{N_{14}} + 9\mathbf{N_{15}} + 9\mathbf{N_{16}} \\
&\quad +27\mathbf{N_{17}} - 27\mathbf{N_{18}} - 27\mathbf{N_{19}} + 27\mathbf{N_{20}} - 27\mathbf{N_{21}} + 27\mathbf{N_{22}} + 27\mathbf{N_{23}} - 27\mathbf{N_{24}} \\
&\quad -27\mathbf{N_{25}} + 27\mathbf{N_{26}} + 9\mathbf{N_{27}} + 9\mathbf{N_{28}} + 27\mathbf{N_{29}} - 27\mathbf{N_{30}} - 9\mathbf{N_{31}} - 9\mathbf{N_{32}}) \\
\mathbf{T_{13}} &= 1/64\ (19\mathbf{N_3} - 19\mathbf{N_2} - 19\mathbf{N_1} + 19\mathbf{N_4} + 19\mathbf{N_5} + 19\mathbf{N_6} - 19\mathbf{N_7} - 19\mathbf{N_8} \\
&\quad +9\mathbf{N_9} + 9\mathbf{N_{10}} + 27\mathbf{N_{11}} - 27\mathbf{N_{12}} - 9\mathbf{N_{13}} - 9\mathbf{N_{14}} - 27\mathbf{N_{15}} + 27\mathbf{N_{16}} \\
&\quad +27\mathbf{N_{17}} + 27\mathbf{N_{18}} - 27\mathbf{N_{19}} - 27\mathbf{N_{20}} - 27\mathbf{N_{21}} - 27\mathbf{N_{22}} + 27\mathbf{N_{23}} + 27\mathbf{N_{24}} \\
&\quad -9\mathbf{N_{25}} - 9\mathbf{N_{26}} - 27\mathbf{N_{27}} + 27\mathbf{N_{28}} + 9\mathbf{N_{29}} + 9\mathbf{N_{30}} + 27\mathbf{N_{31}} - 27\mathbf{N_{32}})
\end{aligned}$$

$$\begin{aligned}\mathbf{T_{14}} = 1/64\ (&9\mathbf{N_3} - 9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_9} + 9\mathbf{N_{10}} - 9\mathbf{N_{13}} - 9\mathbf{N_{14}} + 9\mathbf{N_{25}} + 9\mathbf{N_{26}} - 9\mathbf{N_{29}} - 9\mathbf{N_{30}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{15}} = 1/64\ (&9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_3} - 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-9\mathbf{N_{11}} - 9\mathbf{N_{12}} + 9\mathbf{N_{15}} + 9\mathbf{N_{16}} - 9\mathbf{N_{27}} - 9\mathbf{N_{28}} + 9\mathbf{N_{31}} + 9\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{16}} = 1/64\ (&9\mathbf{N_4} - 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_1} - 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_9} + 9\mathbf{N_{10}} + 9\mathbf{N_{13}} + 9\mathbf{N_{14}} - 9\mathbf{N_{25}} - 9\mathbf{N_{26}} - 9\mathbf{N_{29}} - 9\mathbf{N_{30}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{17}} = 1/64\ (&9\mathbf{N_4} - 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_1} - 9\mathbf{N_5} + 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_{11}} + 9\mathbf{N_{12}} + 9\mathbf{N_{15}} + 9\mathbf{N_{16}} - 9\mathbf{N_{27}} - 9\mathbf{N_{28}} - 9\mathbf{N_{31}} - 9\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{18}} = 1/64\ (&9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_3} - 9\mathbf{N_4} + 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_{17}} - 9\mathbf{N_{18}} - 9\mathbf{N_{19}} + 9\mathbf{N_{20}} + 9\mathbf{N_{21}} - 9\mathbf{N_{22}} - 9\mathbf{N_{23}} + 9\mathbf{N_{24}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{19}} = 1/64\ (&9\mathbf{N_3} - 9\mathbf{N_2} - 9\mathbf{N_1} - 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_{17}} + 9\mathbf{N_{18}} - 9\mathbf{N_{19}} - 9\mathbf{N_{20}} + 9\mathbf{N_{21}} + 9\mathbf{N_{22}} - 9\mathbf{N_{23}} - 9\mathbf{N_{24}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{20}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} - 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-27\mathbf{N_9} + 27\mathbf{N_{10}} - 27\mathbf{N_{13}} + 27\mathbf{N_{14}} - 27\mathbf{N_{25}} + 27\mathbf{N_{26}} - 27\mathbf{N_{29}} + 27\mathbf{N_{30}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{21}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} - 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+27\mathbf{N_{11}} - 27\mathbf{N_{12}} + 27\mathbf{N_{15}} - 27\mathbf{N_{16}} + 27\mathbf{N_{27}} - 27\mathbf{N_{28}} + 27\mathbf{N_{31}} - 27\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{22}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} + 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-27\mathbf{N_9} + 27\mathbf{N_{10}} + 27\mathbf{N_{13}} - 27\mathbf{N_{14}} + 27\mathbf{N_{25}} - 27\mathbf{N_{26}} - 27\mathbf{N_{29}} + 27\mathbf{N_{30}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{23}} = 1/64\ (&9\mathbf{N_1} + 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-27\mathbf{N_{11}} + 27\mathbf{N_{12}} + 27\mathbf{N_{15}} - 27\mathbf{N_{16}} + 27\mathbf{N_{27}} - 27\mathbf{N_{28}} - 27\mathbf{N_{31}} + 27\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{24}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} + 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-27\mathbf{N_{17}} + 27\mathbf{N_{18}} + 27\mathbf{N_{19}} - 27\mathbf{N_{20}} + 27\mathbf{N_{21}} - 27\mathbf{N_{22}} - 27\mathbf{N_{23}} + 27\mathbf{N_{24}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{25}} = 1/64\ (&9\mathbf{N_1} + 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-27\mathbf{N_{17}} - 27\mathbf{N_{18}} + 27\mathbf{N_{19}} + 27\mathbf{N_{20}} + 27\mathbf{N_{21}} + 27\mathbf{N_{22}} - 27\mathbf{N_{23}} - 27\mathbf{N_{24}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{26}} = 1/64\ (&19\mathbf{N_1} - 19\mathbf{N_2} - 19\mathbf{N_3} - 19\mathbf{N_4} + 19\mathbf{N_5} + 19\mathbf{N_6} + 19\mathbf{N_7} - 19\mathbf{N_8}\\ &-27\mathbf{N_9} + 27\mathbf{N_{10}} + 27\mathbf{N_{11}} - 27\mathbf{N_{12}} - 27\mathbf{N_{13}} + 27\mathbf{N_{14}} + 27\mathbf{N_{15}} - 27\mathbf{N_{16}}\\ &-27\mathbf{N_{17}} + 27\mathbf{N_{18}} - 27\mathbf{N_{19}} + 27\mathbf{N_{20}} + 27\mathbf{N_{21}} - 27\mathbf{N_{22}} + 27\mathbf{N_{23}} - 27\mathbf{N_{24}}\\ &+27\mathbf{N_{25}} - 27\mathbf{N_{26}} - 27\mathbf{N_{27}} + 27\mathbf{N_{28}} + 27\mathbf{N_{29}} - 27\mathbf{N_{30}} - 27\mathbf{N_{31}} + 27\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{27}} = 1/64\ (&9\mathbf{N_1} + 9\mathbf{N_2} - 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} + 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-9\mathbf{N_9} - 9\mathbf{N_{10}} + 9\mathbf{N_{13}} + 9\mathbf{N_{14}} + 9\mathbf{N_{25}} + 9\mathbf{N_{26}} - 9\mathbf{N_{29}} - 9\mathbf{N_{30}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{28}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} + 9\mathbf{N_3} - 9\mathbf{N_4} - 9\mathbf{N_5} + 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+9\mathbf{N_{11}} + 9\mathbf{N_{12}} - 9\mathbf{N_{15}} - 9\mathbf{N_{16}} - 9\mathbf{N_{27}} - 9\mathbf{N_{28}} + 9\mathbf{N_{31}} + 9\mathbf{N_{32}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{29}} = 1/64\ (&9\mathbf{N_1} - 9\mathbf{N_2} - 9\mathbf{N_3} + 9\mathbf{N_4} + 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &-9\mathbf{N_{17}} + 9\mathbf{N_{18}} - 9\mathbf{N_{19}} + 9\mathbf{N_{20}} - 9\mathbf{N_{21}} + 9\mathbf{N_{22}} - 9\mathbf{N_{23}} + 9\mathbf{N_{24}})\end{aligned}$$

$$\begin{aligned}\mathbf{T_{30}} = 1/64\ (&9\mathbf{N_2} - 9\mathbf{N_1} + 9\mathbf{N_3} + 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8}\\ &+27\mathbf{N_9} - 27\mathbf{N_{10}} + 27\mathbf{N_{13}} - 27\mathbf{N_{14}} - 27\mathbf{N_{25}} + 27\mathbf{N_{26}} - 27\mathbf{N_{29}} + 27\mathbf{N_{30}})\end{aligned}$$

$$\mathbf{T_{31}} = 1/64\ (9\mathbf{N_2} - 9\mathbf{N_1} + 9\mathbf{N_3} + 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8} \\ -27\mathbf{N_{11}} + 27\mathbf{N_{12}} - 27\mathbf{N_{15}} + 27\mathbf{N_{16}} + 27\mathbf{N_{27}} - 27\mathbf{N_{28}} + 27\mathbf{N_{31}} - 27\mathbf{N_{32}})$$

$$\mathbf{T_{32}} = 1/64\ (9\mathbf{N_2} - 9\mathbf{N_1} + 9\mathbf{N_3} + 9\mathbf{N_4} - 9\mathbf{N_5} - 9\mathbf{N_6} - 9\mathbf{N_7} + 9\mathbf{N_8} \\ +27\mathbf{N_{17}} - 27\mathbf{N_{18}} + 27\mathbf{N_{19}} - 27\mathbf{N_{20}} - 27\mathbf{N_{21}} + 27\mathbf{N_{22}} - 27\mathbf{N_{23}} + 27\mathbf{N_{24}}).$$

Here $\mathbf{N_1}$–$\mathbf{N_8}$ are the vertices of the hexahedron and $\mathbf{N_9}$–$\mathbf{N_32}$ are the high-order nodes as depicted in Fig. 24b.

A.3 Transformation Jacobians

The determinants of the Jacobians for volume and surface integration are given by

$$\begin{aligned} \det \mathbf{J}(p,q,r) &= \left| \frac{\partial(x,y,z)}{\partial(p,q,r)} \right| = \overrightarrow{\mathbf{X}}_p \cdot (\overrightarrow{\mathbf{X}}_q \times \overrightarrow{\mathbf{X}}_r), \\ \det \mathbf{J}_p(q,r) &= ||\overrightarrow{\mathbf{X}}_q \times \overrightarrow{\mathbf{X}}_r||_{p=ct}, \\ \det \mathbf{J}_q(p,r) &= ||\overrightarrow{\mathbf{X}}_p \times \overrightarrow{\mathbf{X}}_r||_{q=ct}, \\ \det \mathbf{J}_r(p,q) &= ||\overrightarrow{\mathbf{X}}_p \times \overrightarrow{\mathbf{X}}_q||_{r=ct}. \end{aligned} \tag{A.3}$$

References

1. Groth, C.P.T., De Zeeuw, D.L., Gombosi, T.I., Powell, K.G.: Global three-dimensional MHD simulation of a space weather event: CME formation, interplanetary propagation, and interaction with the magnetosphere. J. Geophys. Res. **105**(A11), 25053–25078 (2000)
2. De Zeeuw, D.L., Gombosi, T.I., Groth, C.P.T., Powell, K.G., Stout, Q.F.: An adaptive MHD method for global space weather simulations. IEEE Trans. Plasma Sci. **28**(6), 1956–1965 (2000)
3. Sachdev, J.S., Groth, C.P.T., Gottlieb, J.J.: A parallel solution-adaptive scheme for predicting multi-phase core flows in solid propellant rocket motors. Int. J. Comput. Fluid Dyn. **19**(2), 159–177 (2005)
4. Northrup, S.A., Groth, C.P.T.: Solution of laminar diffusion flames using a parallel adaptive mesh refinement algorithm. Paper 2005-0547, AIAA, January 2005
5. Gao, X., Groth, C.P.T.: A parallel adaptive mesh refinement algorithm for predicting turbulent non-premixed combusting flows. Int. J. Comput. Fluid Dyn. **20**(5), 349–357 (2006)
6. Gao, X., Groth, C.P.T.: A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows. J. Comput. Phys. **229**(5), 3250–3275 (2010)
7. Gao, X., Northrup, S.A., Groth, C.P.T.: Parallel solution-adaptive method for two-dimensional non-premixed combusting flows. Prog. Comput. Fluid Dyn. **11**(2), 76–95 (2011)
8. McDonald, J.G., Sachdev, J.S., Groth, C.P.T.: Application of Gaussian moment closure to micro-scale flows with moving and embedded boundaries. AIAA J. **51**(9), 1839–1857 (2014)
9. Freret, L., Groth, C.P.T.: Anisotropic non-uniform block-based adaptive mesh refinement for three-dimensional inviscid and viscous flows. Paper 2015-2613, AIAA, June 2015
10. Freret, L., Williamschen, M., Groth, C.P.T.: Enhanced anisotropic block-based adaptive mesh refinement for three-dimensional inviscid and viscous compressible flows. J. Comput. Phys. **458**, 111092 (2022)
11. Freret, L., Ivan, L., De Sterck, H., Groth, C.P.T.: A high-order finite-volume method with anisotropic AMR for ideal MHD flows. Paper 2017-0845, AIAA, January 2017
12. Freret, L., Ivan, L., De Sterck, H., Groth, C.P.T.: High-order finite-volume method with block-based AMR for magnetohydrodynamics flows. J. Sci. Comput. **79**(1), 176–208 (2019)
13. Ivan, L., Groth, C.P.T.: High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. J. Comput. Phys. **257**, 830–862 (2014)
14. Ivan, L., De Sterck, H., Susanto, A., Groth, C.P.T.: High-order central ENO finite-volume scheme for hyperbolic conservation laws on three-dimensional cubed-sphere grids. J. Comput. Phys. **282**, 157–182 (2015)

15. Harten, A., Enquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, III. J. Comput. Phys. **71**, 231–303 (1987)
16. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
17. Charest, M.R.J., Groth, C.P.T., Gauthier, P.Q.: A high-order central ENO finite-volume scheme for three-dimensional low-speed viscous flows on unstructured mesh. Commun. Comput. Phys. **17**(3), 615–656 (2015)
18. Charest, M.R.J., Groth, C.P.T.: A high-order central ENO finite-volume scheme for three-dimensional turbulent reactive flows on unstructured mesh. Paper 2013-2567, AIAA, June 2013
19. Susanto, A., Ivan, L., Sterck, H.D., Groth, C.P.T.: High-order central ENO finite-volume scheme for ideal MHD. J. Comput. Phys. **250**, 141–164 (2013)
20. Ivan, L., De Sterck, H., Northrup, S.A., Groth, C.P.T.: Hyperbolic conservation laws on three-dimensional cubed-sphere grids: a parallel solution-adaptive simulation framework. J. Comput. Phys. **255**, 205–227 (2013)
21. Nguyen, T.B., De Sterck, H., Freret, L., Groth, C.P.T.: High-order implicit time-stepping with high-order CENO methods for unsteady three-dimensional CFD simulations. Int. J. Numer. Methods Fluids **94**, 121–151 (2022)
22. Fu, L., Hu, X.Y., Adams, N.A.: Targeted ENO schemes with tailored resolution property for hyperbolic conservation laws. J. Comput. Phys. **349**, 97–121 (2017)
23. Fu, L.: A very-high-order TENO scheme for all-speed gas dynamics and turbulence. Comput. Phys. Commun. **244**, 117–131 (2019)
24. Boscheri, W., Semplice, M., Dumbser, M.: Central WENO subcell finite volume limiters for ADER discontinuous Galerkin schemes on fixed and moving unstructured meshes. Commun. Comput. Phys. **25**(2), 311–346 (2019)
25. Tsoutsanis, P., Dumbser, M.: Arbitrary high order central non-oscillatory schemes on mixed-element unstructured meshes. Comput. Fluids **225**, 2021 (2021)
26. Tsoutsanis, P., Adebayo, E.M., Carriba Merino, A., Perez Arjona, A., Skote, M.: CWENO finite-volume interface capturing schemes for multicomponent flows using unstructured meshes. J. Sci. Comput. **89**, 2021 (2021)
27. Pueyo, A., Zingg, D.W.: An efficient Newton-GMRES solver for aerodynamic computations. Paper 97-1955, AIAA, June 1997
28. Groth, C.P.T., Northrup, S.A.: Parallel implicit adaptive mesh refinement scheme for body-fitted multi-block mesh. Paper 2005-5333, AIAA, June 2005
29. Nejat, A., Ollivier-Gooch, C.: Effect of discretization order on preconditioning and convergence of a high-order unstructured finite-volume Newton-Krylov solver for inviscid compressible flows. Paper 2007-719, AIAA, January 2007
30. Wong, P., Zingg, D.W.: Three-dimensional aerodynamic computations on unstructured grids using a Newton–Krylov approach. Comput. Fluids **37**(2), 107–120 (2008)
31. Hicken, J.E., Zingg, D.W.: Parallel Newton–Krylov solver for the Euler equations discretized using simultaneous approximation terms. AIAA J. **46**(11), 2773–2786 (2008)
32. Nejat, A., Ollivier-Gooch, C.: A high-order accurate unstructured finite-volume Newton–Krylov algorithm for inviscid compressible flows. J. Comput. Phys. **227**, 2582–2609 (2008)
33. Dias, S.C., Zingg, D.W.: A high-order parallel Newton–Krylov flow solver for the Euler equations. Paper 2009-3657, AIAA, June 2009
34. Northrup, S.A., Groth, C.P.T.: Parallel implicit adaptive mesh refinement scheme for unsteady fully-compressible reactive flows. Paper 2013-2433, AIAA, June 2013
35. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear equations. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
36. Saad, Y.: Krylov subspace methods on supercomputers. SIAM J. Sci. Stat. Comput. **10**(6), 1200–1232 (1989)
37. Brown, P.N., Saad, Y.: Hybrid Krylov methods for nonlinear systems of equations. SIAM J. Sci. Stat. Comput. **11**(3), 450–481 (1990)
38. Saad, Y.: Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston (1996)
39. Charest, M.R.J., Groth, C.P.T., Gülder, Ö.L.: Solution of the equation of radiative transfer using a Newton–Krylov approach and adaptive mesh refinement. J. Comput. Phys. **231**, 3023–3040 (2012)
40. Schlichting, H., Gersten, K.: Boundary-Layer Theory, 8th edn. Springer, New York (2000)
41. Hirsch, C.: Numerical Computation of Internal and External Flows, Volume 1, Fundamentals of Numerical Discretization. Wiley, Toronto (1989)
42. Harten, A., Lax, P.D., van Leer, B.: On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev. **25**(1), 35–61 (1983)

43. Einfeldt, B.: On Godunov-type methods for gas dynamics. SIAM J. Numer. Anal. **25**, 294–318 (1988)
44. Gottlieb, J.J., Groth, C.P.T.: Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases. J. Comput. Phys. **78**, 437–458 (1988)
45. Toro, E.F.: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer, New York (1999)
46. Petrov, M.N., Tambova, A.A., Titarev, V., Utyuzhnikov, S., Chikitkin, A.: FlowModellium software package for calculating high-speed flows of compressible fluid. Comput. Math. Math. Phys. **58**(11), 1865–1886 (2018)
47. Barth, T.J.: Recent developments in high order k-exact reconstruction on unstructured meshes.' Paper 93-0668, AIAA, January 1993
48. Venkatakrishnan, V.: On the accuracy of limiters and convergence to steady state solutions. Paper 93-0880, AIAA, January 1993
49. Tsoutsanis, P.: Stencil selection algorithms for WENO schemes on unstructured meshes. J. Comput. Phys. X **4**, 2019 (2019)
50. Zienkiewicz, O.C., Taylor, R.L., Zhu, Z.Z.: The Finite Element Method: Its Basis and Fundamentals, 7th edn. Elsevier, New York (2013)
51. Jalali, A.: An Adaptive Higher-Order Unstructured Finite Volume Solver for Turbulent Compressible Flows. PhD thesis, University of British Columbia (2017)
52. Costa, R., Nobrega, J.M., Clain, S., Machado, G.: Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts. Comput. Methods Appl. Mech. Eng. **117**(2), 112560 (2019)
53. Lapidus, L., Pinder, G.F.: Numerical Solution of Partial Differential Equations in Science and Engineering. Wiley, Toronto (1999)
54. Anderson, D.A., Tannehill, J.C., Pletcher, R.H.: Computational Fluid Mechanics and Heat Transfer. McGraw-Hill, Toronto (1984)
55. Thompson, J.F., Warsi, Z.U.A., Mastin, C.W.: Numerical Grid Generation-Foundations and Applications. North-Holland, New York (1985)
56. Ollivier-Gooch, C.F., Altena, M.V.: A high-order accurate unstructured mesh finite-volume scheme for the advection–diffusion equation. J. Comput. Phys. **181**(2), 729–752 (2002)
57. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact newton methods. SIAM J. Numer. Anal. **19**(2), 400–408 (1982)
58. Mulder, W.A., van Leer, B.: Experiments with implicit upwind methods for the Euler equations. J. Comput. Phys. **59**, 232–246 (1985)
59. Ronchi, C., Iacono, R., Paolucci, P.S.: The "Cubed Sphere": A new method for the solution of partial differential equations in spherical geometry. J. Comput. Phys. **124**, 93–114 (1996)
60. Hoshyari, S.: A high-order unstructured finite volume solver for three-dimensional compressible flows. Master's thesis, University of British Columbia (2017)
61. Henderson, R.D.: Details of the drag curve near the onset of vortex shedding. Phys. Fluids **7**(9), 2102–2104 (1995)
62. Kannan, R., Wang, Z.J.: A study of viscous flux formulations for a p-multigrid spectral volume Navier Stokes solver. J. Sci. Comput. **41**(2), 165–199 (2009)