ADAPTIVE HIGH-ORDER HYBRID GRIDS FOR FIELD COMPUTATIONS

Yannis Kallinderis, Alexandros Karkoulias and Panagiotis Antonellis

Department of Mechanical and Aeronautical Engineering,
The University of Patras,
Patras 26500, Greece
e-mail: kallind@otenet.gr, alex.karkoulias@gmail.com, adonel@gmail.com

Keywords: High-order grids, Adaptive hybrid grids, Computational Mechanics.

Abstract. *Curved geometries typically require a large number of low-order (straight-edge) computational elements. High-order numerical solvers have been primarily used with low-order meshes. There is a need for curved, high-order computational elements. Hybrid meshes consisting of hexahedra, prisms, tetrahedra and pyramids are in use for complex-geometry viscous flow computations.*

The present work studies the employment of second-order hybrid meshes that are relatively coarse for field simulations. These curved meshes are compared with the conventional low-order fine ones in terms of accuracy, computational time and, eventually, efficiency. The cases that are considered include volume calculations, as well as computation of the gradient of various field functions. Directionally high-order elements are also employed and evaluated. Lastly, employment and evaluation of adaptive local high-order grids is presented.

1 INTRODUCTION

Curved geometries are typically discretized with straight-edge computational elements. In three dimensions, especially, a large number of elements is required for accuracy. High-order numerical methods have gained importance (see, for example, [1], [2], [3] and [4]). However, low-order computational grids have been employed with the existing high-order solvers.

There is a need for generation and employment of curved-edge computational elements to discretize geometries that contain parts of relatively high curvature. This use could also be extended to the interior of the computational domain in order to treat local fields of high curvature. A significant task to be undertaken regards the evaluation of the efficiency / effectiveness of using high-order meshes.

Typical geometries are characterized by local regions of high curvature. The use of locally high-order meshes is therefore an approach to explore. The complexity of the current computational domains in terms of geometry and field has necessitated the use of hybrid grids consisting of different mesh topologies [5]. Mixed elements of prisms, hexahedra, tetrahedra and pyramids are quite common. Creation of high-order counterparts of those complex meshes is another challenge. The efficiency is affected by the number of high-order edges an element contains. This leads to the necessity of studying partially or *directionally* high-order elements in terms of accuracy and efficiency.

There have been a few efforts of generating high-order meshes for flow field computations as demonstrated in [6], [7], [8]. The majority of the existing methods start with a relatively coarse grid and insert the extra points needed to convert each element to high-order. There have also been methods which use a high-order (curved-edge) surface mesh, which is then marched outwards in order to create the volume elements. Robustness of the marching process is an issue in the vicinity of concave / convex regions of the geometry. A primary issue of high-order meshes is the validity of the generated elements [9].

Relatively little work has been published on the evaluation of effectiveness of high-order grids in comparison with the conventional straight-edge (low-order) ones. Further, the adaptively high-order grids have not been explored. The same holds for the use of directionally high-order computational elements. The above issues are the focus of the present work.

The following section presents the method for generating second-order grids along with the numerical integration that is performed using the high-order computational elements. Next, an evaluation of second-order hybrid grids is performed. Adaptive high-order meshes is the focus of the section after that. This includes locally high-order, as well as directionally high-order grids.

2 CURVED-EDGE 3D HYBRID GRIDS

The generation of second-order hybrid meshes in three dimensions is presented. Then, calculation of field gradients and volumes using those meshes is described.

2.1 Construction of curved-edge 3D hybrid grids

The construction of curved-edge grids is not a trivial problem as it introduces several challenges to the grid generation process: the curved grid needs to be coarse enough for reducing the computational cost of high-order discretization methods and at the same time preserve its validity (non-intersecting elements) at the vicinity of high-curvature geometric regions. Those issues become more challenging when dealing with 3D hybrid grids, where all the types of 3D elements (hexahedra, prisms, pyramids and tetrahedra) can be encountered.

For the purpose of this work, the CENTAUR grid generator [10] is employed to construct second-order 3D hybrid grids. The construction of curved-edge grids within CENTAUR extends the conventional generation of straight-edge grids, using the following general algorithm, as illustrated in Figure 1:

- Generate a coarse straight-edge hybrid grid (Step A);
- Insert an additional midpoint at every grid edge (Step B);
- Use the CAD information to map the midpoint of every boundary grid edge to the underlying CAD surface (Step C);
- Adjust the position of the interior edge midpoints based on the mapped position of the boundary edge midpoints, in order to prevent intersecting grid elements and ensure grid validity (Step D).

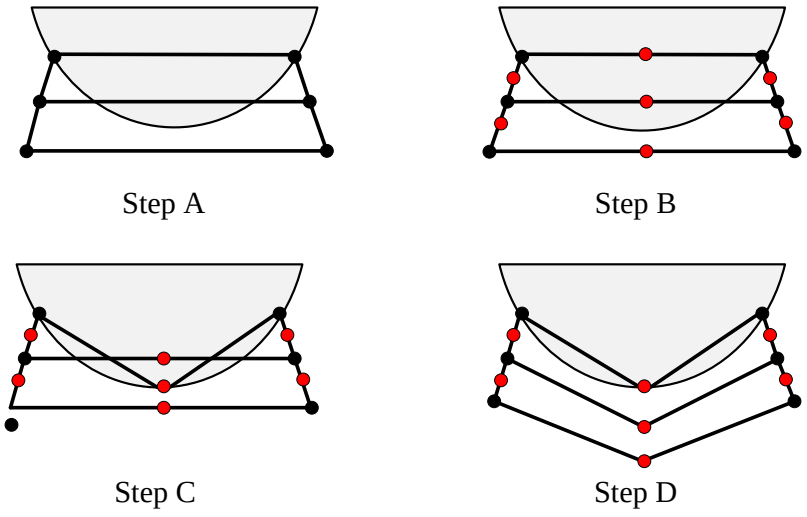


Figure 1. Sequence of steps for constructing curved-edge grids.

This approach for constructing curved-edge 3D hybrid grids prevents intersecting elements, even in high-curvature areas of the geometry. Another advantage of this approach is that it is independent of the grid element types, thus it can be applied to any 3D hybrid grid. The constructed curved-edge grid is a second-order grid, as it defines each edge with the two corner points and its midpoint.

2.2 Spatial derivative and volume calculation using high-order hybrid grid elements

Using the Finite Volume Method the evaluation of the first order spatial derivative of the variable u with respect to x can be done via use of the divergence theorem:

$$\frac{\partial u}{\partial x} = \frac{1}{V}\iiint \frac{\partial u}{\partial x}\cdot dV = \frac{1}{V}\oiint_{\partial V} u\cdot \vec{i}\cdot d\vec{S} \cong \frac{1}{V}\sum_{faces} u\cdot \vec{i}\cdot d\vec{S} \tag{1}$$

, where V is the surrounding volume of a node for a node-based scheme, and the summation is over the faces of the volume.

The mapping between the physical space $\vec{X}=(x,y,z)$ and the reference space $\vec{\xi}=(\xi,\eta,\zeta)$ can be done by implementing the Lagrange shape functions [11]:

$$\vec{X}\left(\vec{\xi}\right)=\sum_{i=1}^{n} N_i\left(\vec{\xi}\right)\cdot \vec{x_i} \tag{2}$$

, where $\vec{x_i}$ is the coordinate vector of vertex i, N_i is the shape function corresponding to vertex i, and n is the total number of vertices per element. The mapping for a second-order triangle and a second-order quadrilateral can be seen in Figures 2 and 3, respectively.

The vector $\vec{dS}$ in Eq. (1) is in the direction of the normal vector to the face and it is defined as:

$$\vec{dS} = \vec{r_\xi} \times \vec{r_\eta} \tag{3}$$

, where $\vec{r}$ is the position vector to the points lying on the surface $\vec{r} = x\,\vec{i} + y\,\vec{j} + z\,\vec{k}$ and $\vec{r_\xi}$, $\vec{r_\eta}$ are the first derivatives of $\vec{r}$ with respect to ξ and η, respectively.

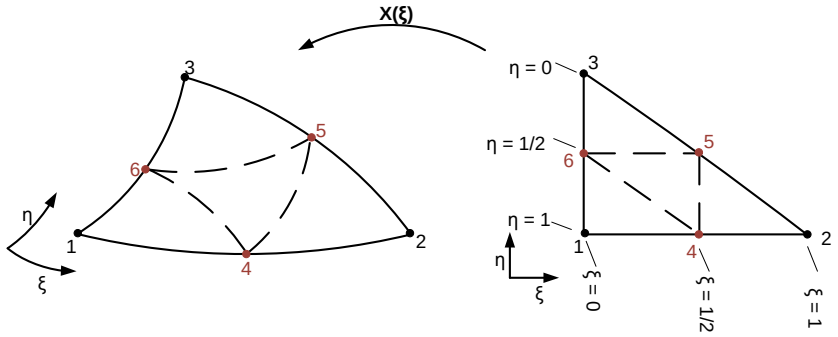


Figure 2. Mapping between physical and reference space for a second-order triangle.

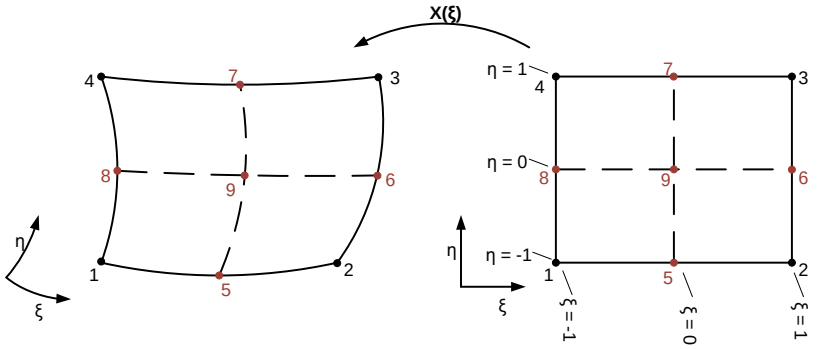


Figure 3. Mapping between physical and reference space for a second-order quadrilateral.

The variable u is defined in a similar way:

$$u\left(\vec{\xi}\right) = \sum_{i=1}^{n} N_i\left(\vec{\xi}\right) \cdot u_i \tag{4}$$

The volume V can be calculated via the divergence theorem:

$$V = \iiint dV = \frac{1}{3} \oiint_{\partial V} \vec{r} \cdot \vec{dS} \cong \frac{1}{3} \sum_{faces} \vec{r} \cdot \vec{dS} \tag{5}$$

, where the summation is over the volume's faces.

The derivative and volume calculations require the evaluation of surface integrals (Eq. (1) and Eq. (5), respectively). Those integrals are evaluated by using a Gaussian quadrature rule, for both triangular and quadrilateral faces. The order of integration can vary between 1 and 4, which affects both accuracy and computational time.

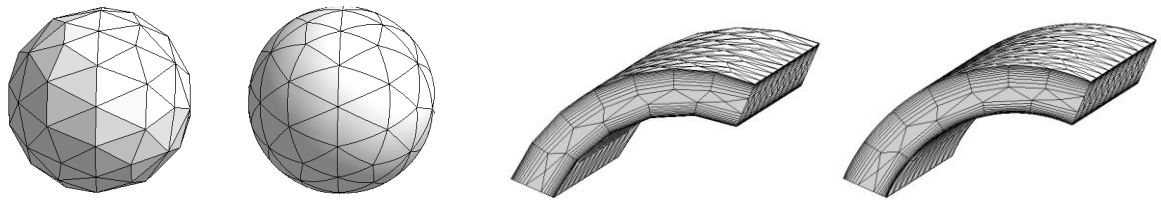
3 EVALUATION OF EFFECTIVENESS OF SECOND-ORDER HYBRID MESHES

In this section we compare the curved meshes with the conventional ones in terms of accuracy, computational time and, eventually, efficiency. Given an initial low-order coarse mesh, the decision that has to be made is either to generate and use a low-order fine mesh or a high-order coarse mesh. This decision is made upon a measure that is called efficiency. Efficiency is defined as the error reduction divided by the increase of computational time using the initial low-order coarse mesh as a reference:

$$Efficiency = -\frac{\Delta\left(\log_{10}\left(Error\right)\right)}{\Delta\left(CPU\ time\right)} \tag{6}$$

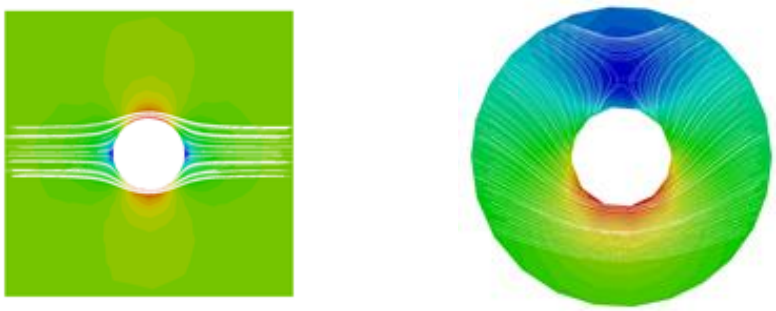
The objective is to reduce the computational error as much as possible, while keeping the computational time fairly low. It is clear that the higher the efficiency, the more appropriate is the corresponding high-order mesh.

The cases that are considered in order to make the comparison are: (i) volume calculations of geometric shapes with known volumes (the sphere shown in Figure 4(a),(b) and the cylindrical sector of Figure 4(c),(d)), and (ii) spatial gradient calculations for known potential flow velocity fields on curved geometries (the flow around a sphere shown in Figure 5(a) and the flow around a rotating cylinder of Figure 5(b)).



(a) (b) (c) (d)

Figure 4. Volume calculation of analytic shapes: (a) sphere discretized with low-order mesh, (b) sphere discretized with high-order mesh, (c) cylindrical sector discretized with low-order mesh, (d) cylindrical sector discretized with high-order mesh.



(a) (b)

Figure 5. Potential flow streamlines: (a) flow around a sphere, (b) flow around a rotating cylinder.

For each of the evaluation cases, four different grids were utilized: a low-order coarse (LOC), a low-order fine (LOF), a high-order coarse (HOC), and a high-order fine (HOF). The number of elements for the coarse/fine grids of each case is listed below:

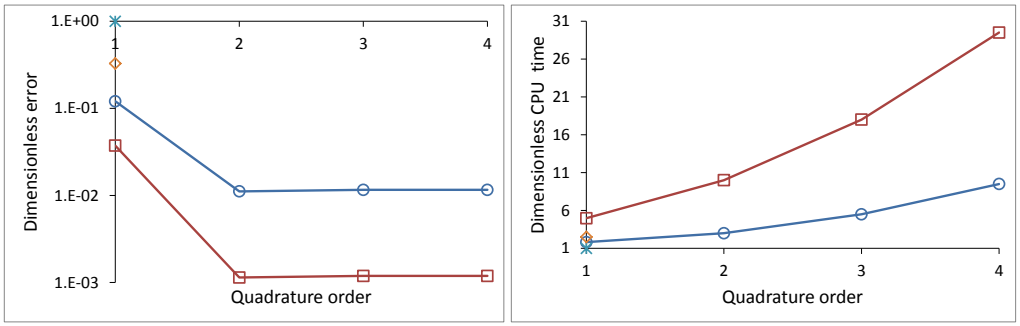
	Coarse	**Fine**
Sphere	1410	4710
Cylindrical sector	3739	14497
Cylinder	10999	23365

Table 1: Number of elements for the grids of each case.

The analytic velocity fields are listed below:

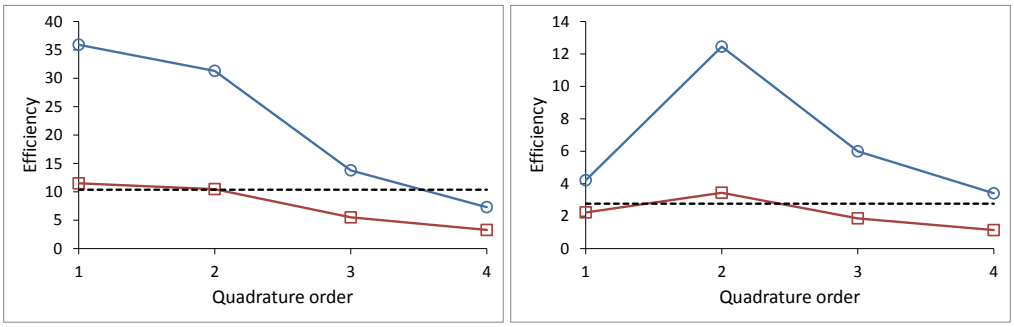
$$Flow\,around\,a\;sphere:\begin{cases} u = u_\infty\left(1+\dfrac{r_o^3}{2}\dfrac{r^2-3x^2}{r^5}\right) \\ v = -\dfrac{3}{2}u_\infty r_o^3\dfrac{xy}{r^5} \\ w = -\dfrac{3}{2}u_\infty r_o^3\dfrac{xz}{r^5} \end{cases}, \quad Flow\;around\,a\;cylinder:\begin{cases} u = u_\infty - \dfrac{\Gamma}{2\pi}\dfrac{y^2}{x^2+y^2} \\ v = \dfrac{\Gamma}{2\pi}\dfrac{x^2}{x^2+y^2} \\ w = 0 \end{cases} \tag{7}$$

Figure 6 shows the error and the computational time corresponding to the volume calculation of the sphere shown in Figure 4(a),(b). All values have been normalized with those yielded by the low–order coarse mesh.



(a) (b)

Figure 6. Sphere volume calculation: (a) error, (b) computational time. Type of mesh: HOC (○), HOF (□), LOC (×), LOF (◊).



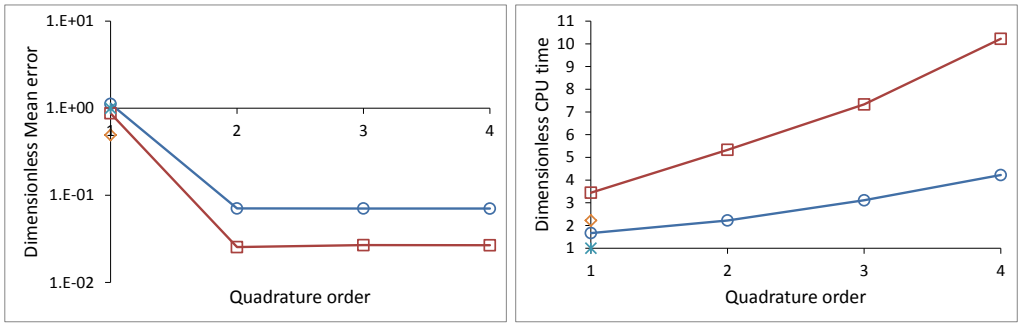
(a) (b)

Figure 7. Volume calculation Efficiency: (a) sphere, (b) cylindrical sector. Type of mesh: HOC (○), HOF (□), LOF (---).

The resulting efficiency of the volume calculation of the sphere and the cylindrical sector is shown in Figure 7(a) and Figure 7(b), respectively. The results demonstrate that the high-order meshes (HOC and HOF) achieve better efficiency compared to a fine low-order mesh (LOF). Moreover, the high-order coarse mesh (HOC) is

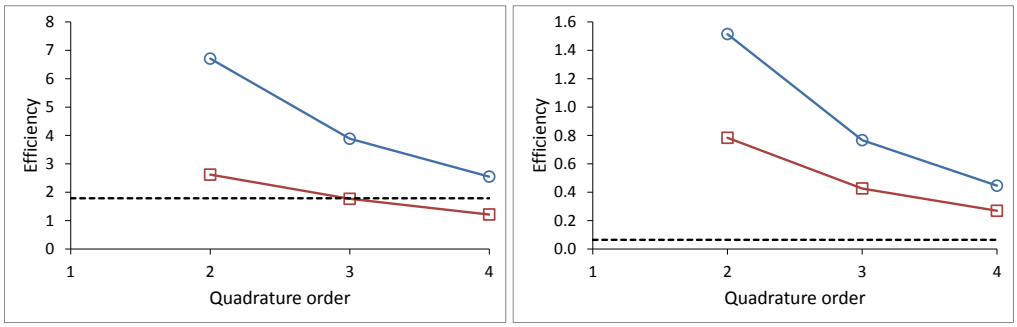
much more efficient than the high-order fine (HOF), especially when using a second-order Gaussian quadrature rule.

Figure 8 shows the error and the computational time for the spatial gradient calculation of the flow around the sphere shown in Figure 5(a). The resulting efficiency of both cases (sphere, rotating cylinder) for the spatial gradient calculation are shown in Figure 9. Again, it can be observed that high-order meshes (HOC and HOF) are more efficient than the low-order fine (LOF), with HOC being the better option when combined with a second-order Gaussian quadrature rule.



(a) (b)

Figure 8. Spatial gradient calculation for the flow around a sphere: (a) error, (b) computational time. Type of mesh: HOC (○), HOF (□), LOC (×), LOF (◊).



(a) (b)

Figure 9. Spatial gradient calculation Efficiency: (a) flow around a sphere, (b) flow around a rotating cylinder. Type of mesh: HOC (○), HOF (□), LOF (---).

The above results demonstrate that high-order grids significantly reduce the calculation error, thus resulting in better accuracy, compared to the corresponding low-order ones. Moreover, it can be observed that a coarse high-order grid (HOC) has better accuracy than a fine low-order grid (LOF). This implies that high-order grids can achieve better accuracy than low-order grids with much fewer grid elements.

The order of Gaussian quadrature rule used for the computations affects the efficiency factor, due to increased computational time. Based on the evaluation results, it is noted that a coarse high-order grid combined with a *second-order* Gauss integration achieves better efficiency for both cases.

4 ADAPTIVE HIGH-ORDER GRIDS

The main disadvantage of a high-order grid is the increased computational time required per element. So, if the computational time could be reduced, while keeping the same accuracy, then the efficiency factor of high-order grids would further increase, and thus make them more attractive for use. Two ways are proposed here in order to reduce the computational time: (i) use of *directional* high-order computational elements, and (ii) use of *adaptive* local high-order meshes.

Until now, every element in a high-order grid was considered curved and treated accordingly (using high-order discretization). However, even on a curved mesh, there are many regions with flat elements. Especially, based on the high-order construction algorithm presented in Section 2.1, the internal elements that are far from

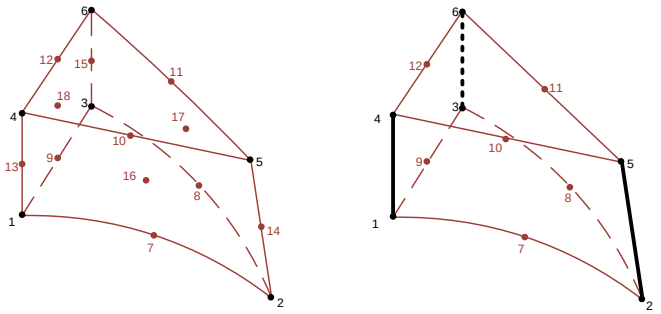
the boundary faces should be primarily flat, as the midpoint relocation is not affecting them. Such regions can be treated as if they belonged to a straight-edge mesh, thus simplifying the calculations and reducing the computational time.

The *directional* high-order (directional HO) meshes consider as curved all the lateral edges that belong to boundary layer elements (prisms / hexahedra). Thus, in case of boundary layer prisms, only the triangular faces are considered as curved, assuming that the quadrilateral faces are flat, as shown in Figure 10. The remaining internal elements are considered as low-order elements. With this technique, only a small subset of the faces is treated as curved, while the rest is treated as flat, thus the required computational time is substantially reduced.

The *adaptive* high-order (adaptive HO) meshes utilize a local-curvature criterion to identify which surface faces should be treated as curved and which should be treated as flat (low-order). In order for a face to be considered curved, at least one of its edges must satisfy the following criterion:

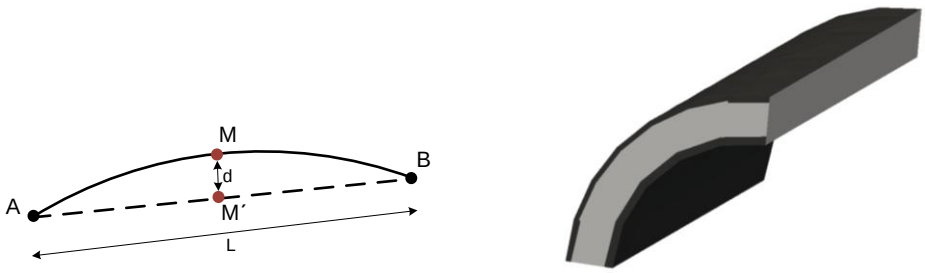
$$d/L \geq c \tag{8}$$

, where d is the distance between the actual high-order midpoint and the low-order midpoint, L is the edge length, as shown in Figure 11(a), and c is a user-defined constant. It is noted that Eq. (8) uses a dimensionless expression of local curvature, which is simple to calculate. Figure 11(b) shows application of this curvature criterion to the cylindrical sector mesh. The high-order faces (denoted with the dark color) are found in the curved parts of the geometry, only.



(a) (b)

Figure 10. Second-order prism: (a) isotropically high-order, (b) directional high-order with the bold lines denoting the low-order (straight) edges of the element.



(a) (b)

Figure 11. Adaptive high-order: (a) criterion sketch, (b) adaptive cylindrical sector mesh (high-order faces are the dark ones).

Figure 12 shows application of the criterion of Eq. (8) to a typical boundary layer hexahedron. The normal-to-the-surface lines are low-order.

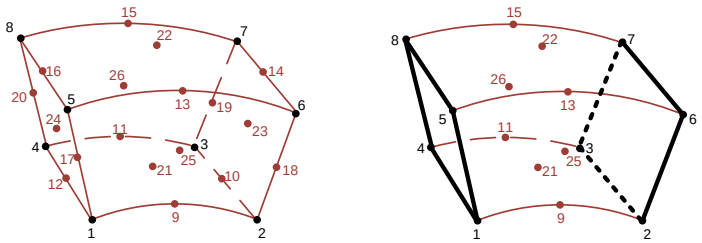


Figure 12. Second-order hexahedron: (a) isotropically high-order, (b) adaptive high-order with the bold lines denoting the low-order (straight) edges of the element.

4.1 Evaluation of effectiveness

In this section the volume calculation of the cylindrical sector and the spatial gradient calculation of the flow around the sphere are repeated employing directional HO and adaptive HO meshes.

The percentages of the curved faces in the cylindrical sector mesh (Figure 4(c)) are 54.8% using the directional HO technique and 40.4% using the adaptive HO technique. Similarly, for the case of the flow around a sphere (Figure 5(a)), the percentages are 15.8% and 13.5%, respectively. In both cases it is seen that the percentages are lower for the adaptive HO meshes.

Figure 13 shows the improved efficiency of the calculations for the case of directional HOC meshes.

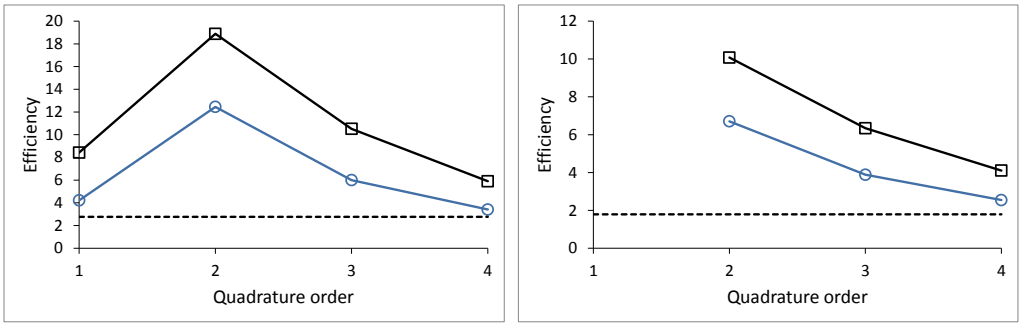


Figure 13. Efficiency of directionally high-order mesh: (a) cylindrical sector volume calculation, (b) spatial gradient calculation for the flow around a sphere. Type of mesh: HOC (○), directional HOC (□), LOF (---).

The computational error is kept the same as we switch from globally HO to directional HO and adaptive HO, as shown in Figure 14(a) for the volume calculation of the cylindrical sector. Meanwhile, the computational time is reduced for the case of directional HOC, and further reduced for the case of adaptive HOC (Figure 14(b)), increasing, in this way, the resulting efficiency in both cases. Figure 15 shows further improvement of efficiency when using an adaptive local HOC mesh. The maximum efficiency is observed if we use a second-order Gauss integration.

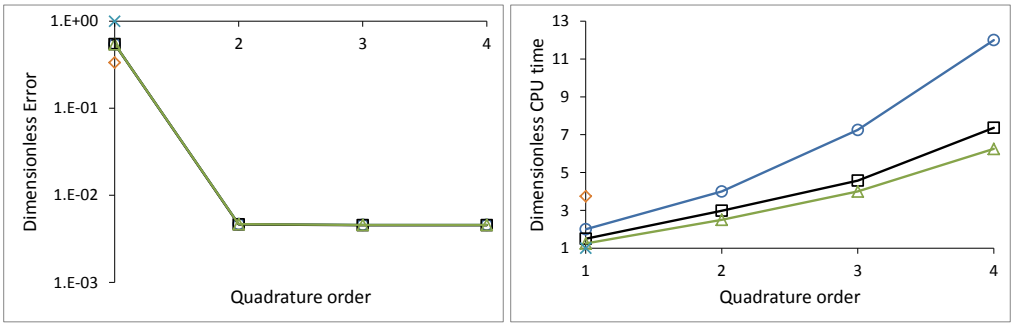
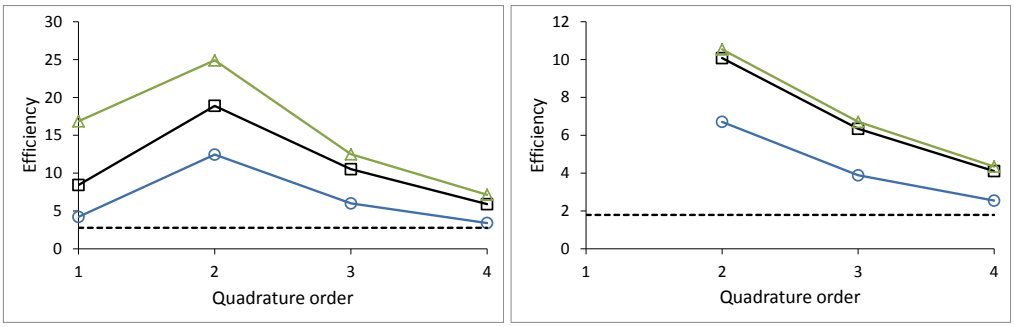


Figure 14. Cylindrical sector volume calculation: (a) error, (b) computational time. Type of mesh: HOC (○), directional HOC (□), adaptive HOC (△), LOC (×), LOF (◊).



(a) (b)

Figure 15. Efficiency of an adaptive high-order mesh: (a) cylindrical sector volume calculation, (b) spatial gradient calculation for the flow around a sphere. Type of mesh: HOC (○), directional HOC (□), adaptive HOC (△), LOF (---).

5 SUMMARY

In this study we have compared the curved (second-order) meshes with the conventional straight-edge ones. A measure of effectiveness was employed in order to compare the meshes in terms of accuracy and computational time. The cases considered included volume and spatial gradient calculations. In all cases it was shown that high-order meshes were more efficient than low-order ones. Moreover, coarse high-order meshes (HOC) were shown to be much more efficient compared to the fine high-order ones (HOF). Next, two techniques were employed in order to tackle the high computational time related to high-order meshes: the directional high-order meshes and the adaptive high-order meshes. Both techniques treated certain parts of the mesh as curved, assuming that the rest is a straight-edge mesh. Based on the results, it was shown that both directional and adaptive high-order meshes maintain the accuracy of the global counterparts, while reducing the computational time, leading to higher levels of efficiency. Specifically, the adaptive high-order meshes give the best results in terms of efficiency compared to the other types of meshes (directional high-order, global high-order, and low-order ones).

Future work involves two directions: (i) evaluation of higher order meshes (third-order and greater), and (ii) employment of a solver for realistic flow field computations using high-order grids.

REFERENCES

[1] Leicht, T. and Kroll, N. (2013), "Status of High-Order Methods for Aerospace Applications", *37th Advanced VKI CFD Lecture Series 2014-03*, Von Karman Institute for Fluid Dynamics.

[2] Wang, Z.J. (2002), "Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids", *Journal of Computational Physics*, vol. 178, pp. 210-251.

[3] Shu, C.W. (2003), "High-order Finite Difference and Finite Volume WENO Schemes and Discontinuous Galerkin Methods for CFD", *International Journal of Computational Fluid Dynamics*, vol. 17, no. 02, pp. 107-118.

[4] Cockburn, B., Karniadakis, G.E. and Shu, C.W. (2000), "The Development of Discontinuous Galerkin Methods", *Lecture Notes in Computational Science and Engineering*, pp. 3-50.

[5] Kallinderis, Y. (1998), "Hybrid Grids and Their Applications", Chapter 25, *Handbook of Grid Generation*, CRC Press, Boca Raton, FL.

[6] Geuzaine, C., Johnen, A., Lambrechts, J., Remacle, J.-F. and Toulorge, T. (2013), "The Generation of Curvilinear Meshes", *37th Advanced VKI CFD Lecture Series 2014-03*, Von Karman Institute for Fluid Dynamics.

[7] Sahni, O., Luo, X.J., Jansen, K.E. and Shephard, M.S. (2010), "Curved boundary layer meshing for adaptive viscous flow simulations", *Finite Elements in Analysis and Design*, vol. 46, no. 1-2, pp. 132-139.

[8] Xie, Z. Q., Sevilla, R., Hassan, O. and Morgan, K. (2013), "The generation of arbitrary order curved meshes for 3D finite element analysis", *Computational Mechanics*, vol. 51, no. 3, pp. 361-374.

[9] Johnen, A., Remacle, J.-F., and Geuzaine, C. (2013), "Geometrical validity of curvilinear finite elements", *Journal of Computational Physics*, vol. 233, pp. 359–372.

[10] CENTAUR Software - Computational Grids for Advanced Modelling. www.centaursoft.com

[11] Chandrupatla T. R. and Belegundu, A. D. (2002), *Introduction to Finite Elements in Engineering*, 3rd ed., Prentice Hall.