Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 00 (2017) 000–000

Procedia Engineering

www.elsevier.com/locate/procedia

26th International Meshing Roundtable, IMR26, 18-21 September 2017, Barcelona, Spain

Robust and efficient validation of the linear hexahedral element

A. Johnen[a], J.-C. Weill[b], J.-F. Remacle[a]

[a] *Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Avenue Georges Lemaitre 4, 1348 Louvain-la-Neuve, Belgium*

[b] *CEA, DAM, DIF, F-91297 Arpajon, France*

Abstract

Checking mesh validity is a mandatory step before doing any finite element analysis. If checking the validity of tetrahedra is trivial, checking the validity of hexahedral elements is far from being obvious. In this paper, a method that robustly and efficiently compute the validity of standard linear hexahedral elements is presented. This method is a significant improvement of a previous work on the validity of curvilinear elements [1]. The new implementation is simple and computationally efficient. The key of the algorithm is still to compute Bézier coefficients of the Jacobian determinant. We show that only 20 Jacobian determinants are necessary to compute the 27 Bézier coefficients. Those 20 Jacobians can be efficiently computed by calculating the volume of 20 tetrahedra. The new implementation is able to check the validity of about 6 million hexahedra per second on one core of a personal computer. Through the paper, all the necessary information is provided that allow to easily reproduce the results, *i.e.* write a simple code that takes the coordinates of 8 points as input and outputs the validity of the hexahedron.

Peer-review under responsibility of the scientific committee of the 26th International Meshing Roundtable.

Keywords: Finite Element Method, Hexahedral Meshes, Hexahedron Validity, Bézier Basis

1. Introduction

Hexahedral meshes are often preferred to tetrahedral meshes as they offer excellent numerical properties: faster assembly [2], orthogonal grids in the wall-normal direction for wall-bounded flows, high accuracy in solid mechanics, both for statics [3] and dynamics, or for quasi-incompressible materials [4] [1]. Generating hex-meshes is however still an open problem for general 3D domains. Finite element meshes should ideally fill the 3D domain in a conformal fashion but should also respect some size and quality constraints in order to be suitable for finite element formulations. The validity of elements is usually the most important constraint and can be checked by verifying the local injectivity of their mapping; in the usual finite element language, one should check the positivity of the Jacobian determinant. While checking the validity of a linear tetrahedron just consists in ensuring its volume positivity, checking the validity of a linear hexahedron is not trivial.

E-mail address: amaury.johnen@uclouvain.be

[1] In many references, the accuracy of linear hexahedra is shown to be equivalent to the accuracy of quadratic tetrahedra with the same mesh density. Note that quadratic tetrahedra have one extra node per edge of the mesh, which multiplies the number of degrees of freedom by 7.

Peer-review under responsibility of the scientific committee of the 26th International Meshing Roundtable.

Testing hexahedron validity is of particular interest when generating hex meshes with an indirect method [5–7]. In these methods, a huge set of hexahedral elements whose cardinality can be as high as 40 times the number of vertices of the mesh is computed [8]. Computing the validity robustly and rapidly is then essential for the efficiency of these methods. Many algorithms have been proposed in the literature for checking the validity of hexahedra, however they do not provide any strong guarantees except method of [1]. In this paper, we particularize this method for the linear hexahedron and propose an efficient and simple implementation.

Previous works. Knupp[9] has shown that the positivity of the Jacobian determinant at the 8 corners of a linear hexahedron, as well as on its edges, is not sufficient to ensure its validity. He conjectured that any hexahedra having a positive Jacobian determinant on its boundary is valid. However, the Jacobian determinant on the faces are biquadratic functions; verifying their positivity is complex and, to our knowledge, no practical algorithm has been presented.

Some authors have proposed to check the validity by ensuring the positivity of sets of tetrahedra constructed from the 8 nodes of the hexahedron [10–14]. The number of tetrahedra ranges from 8 to 64. Ushakova[15] compiled and empirically studied these tests. It is known that the positivity of the 8 corner tetrahedra is a necessary condition [9,10]. Ushakova[15] showed that none of the tests that consider less than 58 tetrahedral volumes constitute a sufficient condition. The volume of the hexahedron is sometimes used in commercial packages [12]. It can be expressed from the volume of 10 tetrahedra. It is a poor test alone but gives a sharper necessary condition when combined with the 8 corner tetrahedra.

Another original method for checking the validity of linear hexahedra has been proposed by Knabner et al.[16]. The Jacobian determinant of the hexahedron is expanded into the monomial basis. Positivity conditions are derived from the monomial coefficients of respectively a quadratic one-dimensional polynomial, a biquadratic polynomial and a triquadratic polynomial. The latter enables to check the positivity of the Jacobian determinant of the hexahedron. However, it is needed to linearize inequalities containing a square root which implies this approach to be only a sufficient condition. A parameter provided by the user allows to determine the precision of this linearization.

A method for checking the validity of curved finite element of any type has been proposed by Johnen et al.[1]. This method consists in expanding the Jacobian determinant into the Bézier basis of order 2. Thanks to the convex hull property of Bézier expansion, the minimum of these coefficients gives a lower bound of the Jacobian determinant. Moreover, the minimum of specific coefficients gives an upper bound of the minimum of the Jacobian determinant. These bounds are subsequently sharpened by "subdividing" in a recursive and adaptive manner which allows to compute the minimum of the Jacobian determinant with any prescribed tolerance. This method can be employed for the validity of the linear hexahedron since it is a particular case of the curved hexahedron.

Contribution. To the best of our knowledge, the method [1] is the only method to robustly check the validity of linear hexahedra. However, the general framework used for curved elements is not well-adapted for an efficient computation of the validity of one specific type of element. In this work, this method is optimized for to the specific case of the linear hexahedron. We start by introducing the validity of the linear quadrangle and hexahedron (§2), and the Bézier expansion of the Jacobian determinant (§3). Then, two substantial improvements are presented: we show that only 20 quantities have to be computed instead of 27 (§4) and that those quantities can be computed as the volume of tetrahedra (§5). Finally, we present the complete algorithm (§6) and demonstrate that this new algorithm is robust and efficient (§7). The C++ code implementing the algorithm will be available in Gmsh [17] (`www.gmsh.info`).

2. Validity of finite elements

Let us consider a d-dimensional physical linear finite element which is geometrically defined by a set of N points $\boldsymbol{n}_k \in \mathbb{R}^d,\ k = 1,\dots,N$, called nodes, and a set of Lagrange shape functions $L_k(\boldsymbol{\xi}) : \Omega_{\text{ref}} \subset \mathbb{R}^d \to \mathbb{R},\ k = 1,\dots,N$. These polynomial functions allow to map a reference unit element, represented by the domain of definition Ω_{ref}, to the physical element (see Figure 1):

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{k=1}^{N} L_k(\boldsymbol{\xi})\, \boldsymbol{n}_k. \tag{1}$$

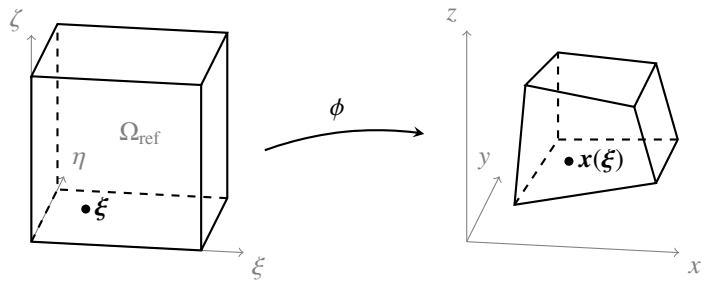

Fig. 1. Mapping between the reference and the physical hexahedron.

The Jacobian matrix of this mapping, denoted $\boldsymbol{J} : \Omega_{\text{ref}} \to \mathbb{R}^{d\times d} : \boldsymbol{\xi} \mapsto \boldsymbol{J}(\boldsymbol{\xi})$, is by definition the matrix of the first-order partial derivatives of $\boldsymbol{x}$, *i.e.* $(\boldsymbol{J})_{ij} = \frac{\partial x_i}{\partial \xi_j}$. Since the mapping is polynomial, each element of $\boldsymbol{J}$ is polynomial. To be well-defined, finite element formulations require the mapping between the reference and any physical element to be injective [18]. This imposes to the determinant of the Jacobian matrix (the Jacobian determinant) to be non-zero for every point of Ω_{ref} [19] and we conventionally impose it to be strictly positive. A physical element is valid if its Jacobian determinant is positive everywhere on the reference domain, otherwise it is invalid. The validity of linear simplices (*i.e.* linear triangles and tetrahedra) is easy to check: since the Jacobian determinant is constant for these elements, it is sufficient to compute it at any point $\boldsymbol{\xi} \in \Omega_{\text{ref}}$ and verify that it is positive. In practice, it is equivalent to compute the signed area of linear triangle since it is equal to the Jacobian determinant divided by 2. Similarly, the signed volume of linear tetrahedra is equal to the Jacobian determinant divided by 6 and can equivalently be computed to check their validity. The Jacobian determinant of linear quadrangles and hexahedra, on the other hand, is not constant over their reference domain. It is necessary to compute the minimum of their Jacobian determinant in order to check their validity. The two following sections are dedicated to explaining how to achieve it.

2.1. Validity control of a linear quadrangle

In finite element codes, the domain of definition Ω_{ref} of the quadrangular element is taken as the domain $[-1, 1] \times [-1, 1]$ due to better numerical properties. This choice has no impact on the validity criterion and we will consider $\Omega_{\text{ref}} \equiv [0, 1] \times [0, 1]$ in this paper for clarity reasons. Consequently, the Lagrange shape functions for a linear quadrangle reads:

$$\begin{cases} L_1(\xi,\eta) = (1-\xi)\,(1-\eta) \\ L_2(\xi,\eta) = \quad \xi \quad (1-\eta) \\ L_3(\xi,\eta) = \quad \xi \quad\quad \eta \\ L_4(\xi,\eta) = (1-\xi) \quad \eta. \end{cases}$$

This implies that the mapping of a quadrangle (cf. equation (1)), is bilinear. Let (x_k, y_k) denotes the coordinates of the node $\boldsymbol{n}_k$, and let us write shortly any difference $(x_j - x_i)$ as x_{ij} (and similarly for the y coordinate). The partial derivative of x with respect to ξ is noted $x_{,\xi}$. The Jacobian matrix is given by:

$$\boldsymbol{J}(\xi,\eta) = \begin{pmatrix} x_{,\xi} & x_{,\eta} \\ y_{,\xi} & y_{,\eta} \end{pmatrix} = \begin{pmatrix} x_{12}\,(1-\eta) + x_{43}\,\eta & x_{14}\,(1-\xi) + x_{23}\,\xi \\ y_{12}\,(1-\eta) + y_{43}\,\eta & y_{14}\,(1-\xi) + y_{23}\,\xi \end{pmatrix}$$

and the Jacobian determinant is given by:

$$\begin{aligned} J(\xi,\eta) = \det(\boldsymbol{J}) = \quad & L_1(\xi,\eta)\,[x_{12}\,y_{14} - y_{12}\,x_{14}] + L_2(\xi,\eta)\,[x_{12}\,y_{23} - y_{12}\,x_{23}] \\ & + L_3(\xi,\eta)\,[x_{43}\,y_{14} - y_{43}\,x_{14}] + L_4(\xi,\eta)\,[x_{43}\,y_{23} - y_{43}\,x_{23}] \\ = \quad & \sum_{k=1}^{4} L_k(\xi,\eta)\,J_k \end{aligned} \tag{2}$$

where the coefficient J_k is the value taken by the Jacobian determinant at corner k. As a consequence, the Jacobian determinant is also bilinear and its minimum is reached at one of the four corners. The validity control of linear quadrangle thus consists in computing the Jacobian determinant at each corner and in verifying that none is negative. An equivalent, but computationally more expensive test would be to compute the angles of the four corners and to check if they lie between 0° and 180°.

The four quantities to compute (either the angles or the coefficients J_k) are not linearly independent. Indeed, concerning the angles, the existing linear relation is that the four angles of a quadrangle sum up to 360°. Now, from equation (2), we can deduce that the Jacobian determinant at *e.g.* the first corner is equal to the third component of the vector $\boldsymbol{v}_{12} \times \boldsymbol{v}_{14}$, where $\boldsymbol{v}_{ij} = \boldsymbol{n}_j - \boldsymbol{n}_i = (x_{ij}, y_{ij})$ is the vector that goes from node i to node j. But, for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ of the xy-plane, it is well-known that the value of the third component of their cross product $\boldsymbol{a} \times \boldsymbol{b}$ is equal to the signed area of the parallelogram they span. In consequence, the Jacobian determinant at corner 1 is equal to two times the signed area of the triangle defined by $\boldsymbol{n}_1$, $\boldsymbol{n}_2$ and $\boldsymbol{n}_4$. Let us note A_k the signed area of the triangle of corner k. Since the total area of the quadrangle is equal to $A_1 + A_3$ or $A_2 + A_4$, we have the following relation concerning the Jacobian determinant: $J_1 + J_3 = J_2 + J_4$ (see Figure 2).

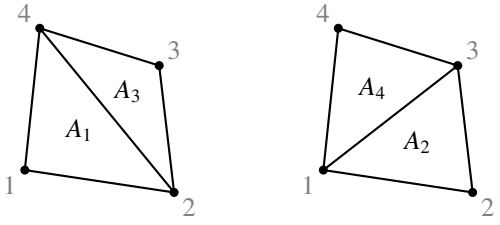

Fig. 2. The linear relationship between the areas of the triangles in a quadrangle ($A_1 + A_3 = A_2 + A_4$) implies an equivalent linear relationship between the four coefficients of the Jacobian determinant J_k: $J_1 + J_3 = J_2 + J_4$, where J_k is the value taken by the Jacobian determinant at corner k.

2.2. Validity control of a linear hexahedron

Let $(x_1(\boldsymbol{\xi}),\ x_2(\boldsymbol{\xi}),\ x_3(\boldsymbol{\xi}))$ be the trilinear mapping of the hexahedron. The 3D Jacobian determinant is by definition:

$$J(\xi, \eta, \zeta) = \sum_{i,\,j,\,k\,=1}^{3} \varepsilon_{i,\,j,\,k}\ \left(x_i\right)_{,\xi} \left(x_j\right)_{,\eta} \left(x_k\right)_{,\zeta} \tag{3}$$

where $\varepsilon_{i,\,j,\,k}$ is the permutation symbol. We have that $(x_i)_{,\xi}$ is a bilinear function in η and ζ, and similarly for $(x_i)_{,\eta}$ and $(x_i)_{,\zeta}$. This means that each term of the sum in equation (3) is triquadratic and so is the Jacobian determinant of the linear hexahedron. As a consequence, the minimum of the Jacobian determinant is not necessarily located at one of the eight corners. A more sophisticated validity test for hexahedra would be to compute the minimum of J on the edges. This can be easily implemented since the Jacobian determinant restricted to an edge is a quadratic function in one of the reference variables. However, it has been proved in [9] that this test is not sufficient. One step further would be the "face test" that would consist in computing the global minimum of a biquadratic function (defined on a square domain) for the 6 faces of the hexahedron. However, there is, to the best of our knowledge, no proof that it would be sufficient, *i.e.* that the global minimum cannot be exclusively located in the volume.

Currently, the only existing technique to robustly compute the validity of linear hexahedra is the method proposed in [1]. This method computes bounds on the minimum of the Jacobian determinant that can be sharpened as much as desired. The main drawback of the proposed algorithm is the general framework used for curved elements that is not well-adapted for an efficient computation for the linear hexahedron. We thus propose to adapt this method to the particular case that concerns us.

In the next section, we introduce the Bézier formulation that allows to compute the bounds and subsequently accurately compute the minimum of J.

3. Bézier expansion of hexahedra Jacobian determinant

Polynomial quantities can be expanded into the so-called Bézier basis in order to make use of Bézier expansion properties. In this section, we first introduce the Bézier expansion, then we derive the transformation matrix that computes the Bézier coefficients from the Lagrange coefficients.

3.1. Definition of Bézier expansion

Let B_k^n be the Bernstein polynomial function whose expression is:

$$B_k^n(t) = \binom{k}{n} t^k (1-t)^{n-k} \qquad t \in [0,1], \quad k = 0, \dots, n$$

where $\binom{k}{n} = \frac{n!}{k!(n-k)!}$ is the binomial coefficient. These functions allow to construct the hexahedral Bézier functions in term of the tensor product of three Bernstein polynomials:

$$B_{ijk}^n(\xi,\eta,\zeta) = B_i^n(\xi)\, B_j^n(\eta)\, B_k^n(\zeta). \tag{4}$$

These functions, $\{B_{ijk}^n\}_{(0\le i,j,k\le n)}$, defines the *Bézier basis* of the hexahedral polynomial space of order n. Since the Jacobian determinant of the linear hexahedron is a triquadratic function, it is included in the hexahedral polynomial space of order 2 and it can be expanded into the Bézier basis of order 2. There exists thus a unique set of coefficients b_{ijk} (also known as control values) such that we have:

$$J(\boldsymbol{\xi}) = \sum_{i,j,k=0}^{2} b_{ijk}\, B_{ijk}^2(\boldsymbol{\xi}) \tag{5}$$

where the right member of the above expression is the *Bézier expansion* of the Jacobian determinant. The number of coefficients is 27 since every index can take three values.

Bézier bases have the property that the basis functions are positive over their domain of definition and sum up to 1. This implies the well-known convex hull property which, in our case, gives that $\min_{ijk} b_{ijk} \le \min_{\boldsymbol{\xi}} J$. In addition to that, some Bézier coefficients are actual values of the Jacobian determinant. Those are the one "located" at the corners of the element. For example, we have: $b_{000} = J(0,0,0)$ and $b_{200} = J(1,0,0)$. The minimum of these corner coefficients constitutes an upper bound for $\min_{\boldsymbol{\xi}} J$. In other words, the control values allow to bound the minimum of the Jacobian determinant from below and above. A positive lower bound implies the positivity of the Jacobian determinant and the validity of the element. On the other hand, a negative upper bound implies that the element is invalid. In the third and last case, when the lower bound is negative and the upper bound is positive, nothing can be told concerning the validity of the element. Those bounds are not necessarily sharp. However, they can be sharpened as much as desired by "subdividing", *i.e.* by expanding the same function defined on a smaller domain, called a subdomain [1]. It is proven in [20,21] that such subdivision algorithm always stops and that it can be used to check the positivity of a multivariate polynomials. Moreover, the bounds converge quadratically with the size of the subdomains [22]. The subdivision algorithm can be implemented in a recursive and adaptive manner making the validity check very efficient [1].

In the following section, we explain how to compute the 27 coefficients b_{ijk} of the Bézier expansion (5).

3.2. Computation of the Bézier coefficients

In order to compute the 27 Bézier coefficients we have to write a linear system of equations. Let us consider a different indexing for Bézier coefficients and Bézier functions for which the order is given in Figure 3. This permits to gather the 27 Bézier coefficients into a vector $\boldsymbol{b}$ for which we have, for example, $b_1 = b_{000}$, $b_2 = b_{200}$ and $b_9 = b_{100}$. We will use a greek letter to refer to this new indexing. In the same way, B_α will refer to a certain function B_{ijk} such that to respect the order defined in Figure 3. Let $\boldsymbol{\xi}_\alpha, \alpha = 1, \dots, 27$ be different points of the reference domain. In practice, these points are taken as the uniformly spaced nodes of the order 2 hexahedron, which limits numerical errors. We

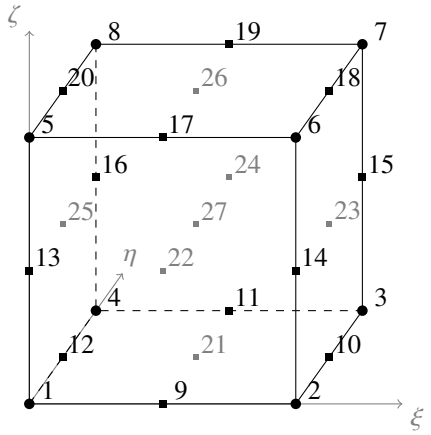

Fig. 3. Ordering of the nodes. Low order nodes are in black while high order nodes are in gray.

order them in the same way, such that we have $\boldsymbol{\xi}_1 = (0,0,0)$, $\boldsymbol{\xi}_2 = (1,0,0)$ and $\boldsymbol{\xi}_9 = (1/2,0,0)$ for example. Let $\boldsymbol{c}$ be the vector of the Jacobian determinant computed at those points, *i.e.* $c_\alpha = J(\boldsymbol{\xi}_\alpha)$.

From the definition of the Bézier expansion (5), we can write the following linear system:

$$J(\boldsymbol{\xi}_\alpha) = \sum_{\beta=1}^{27} b_\beta \, B_\beta(\boldsymbol{\xi}_\alpha) \qquad \forall \alpha \in \{1,\dots,27\}$$
$$\Leftrightarrow \qquad \boldsymbol{c} = \boldsymbol{A}\boldsymbol{b}$$

where $\boldsymbol{A}$ is a transformation matrix[2] in which each element $A_{\alpha\beta}$ is equal to $B_\beta(\boldsymbol{\xi}_\alpha)$. The inverse of $\boldsymbol{A}$, denoted $\boldsymbol{T}$, is the matrix that computes the Bézier coefficients from the computed values of the Jacobian determinant, *i.e.* $\boldsymbol{b} = \boldsymbol{T}\,\boldsymbol{c}$. Matrix $\boldsymbol{T}$ is given in Table 1. To calculate the vector $\boldsymbol{c}$, one may derive the analytical expression of the Jacobian determinant, as we did in 2D (see equation (2)). But we will see in Section 5 that it can be performed by computing the volume of tetrahedra. Moreover, we show in the next section that only a small part of $\boldsymbol{c}$ has to be computed.

4. Linear dependency of the coefficients

Like for quadrangles, Bézier coefficients of the hexahedral elements are not all linearly independent. This is linked to the fact that the Taylor series expansion of the Jacobian determinant contains only 20 non-zero coefficients, as demonstrated in [9]. In this section, we formulate the dependency between the coefficients through a similar reasoning. We then construct a transformation matrix between the 20 linearly independent Jacobian determinant values and the 27 Bézier coefficients.

The Jacobian determinant can be written as the triple scalar product:

$$J = \left(\boldsymbol{x}_{,\xi} \times \boldsymbol{x}_{,\eta}\right) \cdot \boldsymbol{x}_{,\zeta}.$$

This permits to compute the derivatives of the Jacobian determinant in terms of derivatives of the mapping. Given that the mapping is trilinear, the only non-zero derivatives of $\boldsymbol{x}$ are $\boldsymbol{x}_{,\xi}$, $\boldsymbol{x}_{,\eta}$, $\boldsymbol{x}_{,\zeta}$, $\boldsymbol{x}_{,\xi\eta}$, $\boldsymbol{x}_{,\xi\zeta}$, $\boldsymbol{x}_{,\eta\zeta}$ and $\boldsymbol{x}_{,\xi\eta\zeta}$. The derivatives of J can be found in [9] and result in the following observation:

[2] Indeed, we can expand the Jacobian determinant into the traditional Lagrange functions of order 2 for the hexahedral element, in which case we have: $J(\boldsymbol{\xi}) = \sum_{j=1}^{27} c_j \, L_j(\boldsymbol{\xi})$. The sets $\{L_j\}$ and $\{B_j\}$ are two different bases of the same functional space for which $\boldsymbol{c}$ and $\boldsymbol{b}$ are the respective coefficients of the Jacobian determinant.

$$
\boldsymbol{T} = \left(\begin{array}{cccccccc|ccccccccccccccccccc|c}
\multicolumn{8}{c|}{\mathbb{I}_{8\times8}} & \multicolumn{19}{c|}{\mathbb{O}_{8\times19}} \\
\hline
-1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & \multicolumn{19}{c|}{\multirow{12}{*}{$2\,\mathbb{I}_{12\times12}$}} & \multicolumn{1}{c|}{\multirow{12}{*}{$\mathbb{O}_{12\times7}$}} \\
0 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
-1/2 & 0 & 0 & -1/2 & 0 & 0 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
-1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & -1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
0 & 0 & 0 & 0 & -1/2 & 0 & 0 & -1/2 & \multicolumn{19}{c|}{} & \multicolumn{1}{c|}{} \\
\hline
1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \multicolumn{1}{c|}{\multirow{6}{*}{$4\,\mathbb{I}_{6\times6}$}} & 0 \\
1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & \multicolumn{1}{c|}{} & 0 \\
0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & \multicolumn{1}{c|}{} & 0 \\
0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & \multicolumn{1}{c|}{} & 0 \\
1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & \multicolumn{1}{c|}{} & 0 \\
0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & \multicolumn{1}{c|}{} & 0 \\
\hline
\multicolumn{8}{c|}{$-1/8\,\mathbb{1}_{1\times8}$} & \multicolumn{19}{c|}{$1/2\,\mathbb{1}_{1\times12}$} & \multicolumn{1}{c|}{$-2\,\mathbb{1}_{1\times6}$} & 8
\end{array}\right)
$$

Table 1. Transformation matrix that computes the Bézier coefficients from the sampling of the Jacobian determinant. Submatrix $\mathbb{I}_{m\times m}$ designate the identity matrix of dimension m, submatrix $\mathbb{O}_{m\times n}$ is a m by n matrix with only 0 and $\mathbb{1}_{m\times n}$ is a m by n matrix containing only 1.

Observation 1. *The following non-trivial high-order derivatives of the Jacobian determinant are equal to zero:* $J_{,\xi\xi\eta\eta} = J_{,\xi\xi\zeta\zeta} = J_{,\eta\eta\zeta\zeta} = 0$.

Let us consider the monomial basis $\{M_{ijk}\}_{(0\leq i,j,k\leq 2)}$, where $M_{ijk} = M_{ijk}(\xi,\eta,\zeta) = \xi^i\eta^j\zeta^k$ and let us expand the Jacobian determinant into this basis. Let m_{ijk} be the coefficients of this expansion. Observation 1 admits the following corollary:

Corollary 2. *7 monomial coefficients of the Jacobian determinant are always equal to zero:* $m_{220} = m_{202} = m_{022} = m_{221} = m_{212} = m_{122} = m_{222} = 0$.

Corollary 2 implies that the Jacobian determinant space is of dimension 20 and that it is possible to obtain 7 linear relations between the 27 Bézier/Lagrange coefficients. We will obtain them by writing the expression of the monomimial coefficients in function of the Bézier coefficients. Let $a_{\alpha\beta}$ be the coefficient of monomial α in the expression of the Bézier function β (whose definition is given at equation (4)). Mathematically, we have $B_\beta(\boldsymbol{\xi}) = \sum_{\alpha=1}^{27} a_{\alpha\beta}\, M_\alpha(\boldsymbol{\xi})$. We can thus write:

$$
J(\boldsymbol{\xi}) = \sum_{\beta=1}^{27} b_\beta\, B_\beta(\boldsymbol{\xi}) = \sum_{\alpha=1}^{27} \underbrace{\left[\sum_{\beta=1}^{27} b_\beta\, a_{\alpha\beta}\right]}_{m_\alpha} M_\alpha(\boldsymbol{\xi})
$$

The linear relations between the Bézier coefficients are found by considering the equations $m_\alpha = \sum_{\beta=1}^{27} a_{\alpha\beta}\, b_\beta$ for the 7 monomial coefficients of Corollary 2. This leads to the matrix given in Table 2 that computes the last 7 Bézier coefficients in function of the first ones. Let us write $\boldsymbol{D}$ the matrix that computes the 27 Bézier coefficients from the first 20 Bézier coefficients. Matrix $\boldsymbol{D}$ is constructed by extending the matrix given in Table 2 with an identity matrix of size 20. We have:

$$
\boldsymbol{b} = \boldsymbol{D}\,\boldsymbol{b}_{20}
$$

where $\boldsymbol{b}_{20}$ is the vector containing the first 20 components of $\boldsymbol{b}$.

$$\boldsymbol{b}_{21\to 27} = \left(\begin{array}{cccc|cccc|cccc|cccc} 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & -1/2 & 0 & 0 & 0 \\ 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & -1/2 & 0 & 0 \\ 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & -1/2 & 0 \\ 1/4 & 0 & 0 & 1/4 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 \\ 0 & 0 & 0 & 0 & 1/4 & 1/4 & 1/4 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & -1/2 & -1/2 \\ \hline \multicolumn{8}{c|}{1/4\,\mathbb{1}_{1\times 8}} & \multicolumn{12}{c}{-1/4\,\mathbb{1}_{1\times 12}} \end{array}\right) \cdot \boldsymbol{b}_{20}$$

Table 2. Computation of the last 7 Bézier coefficients in function of the 20 first.

$$\boldsymbol{Q} = \left(\begin{array}{cccc|cccc|cccc|cccc} \multicolumn{8}{c|}{\mathbb{I}_{8\times 8}} & \multicolumn{12}{c}{\mathbb{O}_{8\times 12}} \\ -1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & 0 & & & & & & & & \\ 0 & 0 & -1/2 & -1/2 & 0 & 0 & 0 & 0 & & & & & & & & \\ -1/2 & 0 & 0 & -1/2 & 0 & 0 & 0 & 0 & & & & & & & & \\ -1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & & & & & & & & \\ 0 & -1/2 & 0 & 0 & 0 & -1/2 & 0 & 0 & & & & & & & & \\ 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & 0 & \multicolumn{12}{c}{2\,\mathbb{I}_{12\times 12}} \\ 0 & 0 & 0 & -1/2 & 0 & 0 & 0 & -1/2 & & & & & & & & \\ 0 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & 0 & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & -1/2 & -1/2 & & & & & & & & \\ 0 & 0 & 0 & 0 & -1/2 & 0 & 0 & -1/2 & & & & & & & & \\ \hline -3/4 & -3/4 & -3/4 & -3/4 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -3/4 & -3/4 & 0 & 0 & -3/4 & -3/4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -3/4 & -3/4 & 0 & 0 & -3/4 & -3/4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -3/4 & -3/4 & 0 & 0 & -3/4 & -3/4 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ -3/4 & 0 & 0 & -3/4 & -3/4 & 0 & 0 & -3/4 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -3/4 & -3/4 & -3/4 & -3/4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ \hline \multicolumn{8}{c|}{-5/8\,\mathbb{1}_{1\times 8}} & \multicolumn{12}{c}{1/2\,\mathbb{1}_{1\times 12}} \end{array}\right)$$

Table 3. Transformation matrix that allows to compute the Bézier coefficients from 20 samplings of the Jacobian determinant. matrix of dimension m, submatrix $\mathbb{O}_{m\times n}$ is a m by n matrix with only 0 and $\mathbb{1}_{m\times n}$ is a m by n matrix containing only 1.

Constructing the matrix that computes the 27 Bézier coefficients in function of 20 Lagrange coefficients is now straightforward. Matrix $\boldsymbol{T}$ (see Table 1) is such that the first 20 Bézier coefficients depends only on the first 20 Lagrange coefficients. Let $\boldsymbol{T}_{20\times 20}$ be the 20×20 upper left submatrix of $\boldsymbol{T}$ and $\boldsymbol{c}_{20}$ the first 20 components of $\boldsymbol{c}$. We have that:

$$\boldsymbol{b}_{20} = \boldsymbol{T}_{20\times 20}\,\boldsymbol{c}_{20} \quad \Leftrightarrow \quad \boldsymbol{b} = \boldsymbol{D}\,\boldsymbol{T}_{20\times 20}\boldsymbol{c}_{20} = \boldsymbol{Q}\boldsymbol{c}_{20}$$

where $\boldsymbol{Q}$, the matrix that computes all the Bézier coefficients from the first 20 Lagrange coefficients, is given in Table 3.

5. Expression of the 20 Lagrange coefficients in function of 20 tetrahedral volumes

In this section we show that the 20 Lagrange coefficients that has to be computed are equal to the volume of tetrahedra.

Recalling that $\boldsymbol{x}$ is the column vector $(x, y, z)^{\mathrm{T}}$, the Jacobian matrix can be written as:

$$\boldsymbol{J}(\xi, \eta, \zeta) = \left(\left[\boldsymbol{x}_{,\xi}\right]\left[\boldsymbol{x}_{,\eta}\right]\left[\boldsymbol{x}_{,\zeta}\right]\right),$$

Let us recall that $\boldsymbol{v}_{\alpha\beta}$ denotes the difference $(\boldsymbol{n}_\beta - \boldsymbol{n}_\alpha)$. We can express the derivatives of $\boldsymbol{x}$ from the definition of the mapping (1) and the Lagrange functions given in Appendix A:

$$\begin{cases} \boldsymbol{x}_{,\xi} = \boldsymbol{v}_{12}\,(1-\eta)\,(1-\zeta) + \boldsymbol{v}_{43}\,\eta\,(1-\zeta) + \boldsymbol{v}_{56}\,(1-\eta)\,\zeta + \boldsymbol{v}_{87}\,\eta\,\zeta \\ \boldsymbol{x}_{,\eta} = \boldsymbol{v}_{14}\,(1-\xi)\,(1-\zeta) + \boldsymbol{v}_{23}\,\xi\,(1-\zeta) + \boldsymbol{v}_{58}\,(1-\xi)\,\zeta + \boldsymbol{v}_{67}\,\xi\,\zeta \\ \boldsymbol{x}_{,\zeta} = \boldsymbol{v}_{15}\,(1-\xi)\,(1-\eta) + \boldsymbol{v}_{26}\,\xi\,(1-\eta) + \boldsymbol{v}_{48}\,(1-\xi)\,\eta + \boldsymbol{v}_{37}\,\xi\,\eta. \end{cases}$$

In the following, $\det(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ will denote the determinant of the matrix made up of columns $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$. Note that $\det(\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c})$ equals $(\boldsymbol{a} \times \boldsymbol{b}) \cdot \boldsymbol{c}$ and is a trilinear function. Moreover, if the three vectors have the same origin, then the determinant is also 6 times the volume of the tetrahedron that the vectors define. Lastly, if the three vectors are not linearly independent, then the determinant is zero.

There are two types of Lagrange coefficients we are interested in: the coefficients that correspond to the corners and the coefficients that correspond to the edges of the hexahedron. By symmetry of the problem, there must be also two types of tetrahedra to identify. It is already well-known that the Jacobian determinant computed at a corner corresponds to 6 times the volume of the tetrahedron constructed from the 3 edges of the corner. Let us formulate it mathematically for the first corner:

$$J_1 = \det(\boldsymbol{J}(0,0,0)) = \det(\boldsymbol{v}_{12},\, \boldsymbol{v}_{14},\, \boldsymbol{v}_{15}) = 6\,\mathrm{vol}(\boldsymbol{n}_1,\, \boldsymbol{n}_2,\, \boldsymbol{n}_4,\, \boldsymbol{n}_5)$$

where vol(·) refer to the volume of the tetrahedron defined by the four nodes.

In a similar manner, we can express the 9th value of the Jacobian determinant as the volume of a tetrahedron:

$$\begin{aligned} J_9 = \det(\boldsymbol{J}(1/2,0,0)) &= \det\left(\boldsymbol{v}_{12},\ \frac{\boldsymbol{v}_{14}+\boldsymbol{v}_{23}}{2},\ \frac{\boldsymbol{v}_{15}+\boldsymbol{v}_{26}}{2}\right) \\ &= \det\left(\boldsymbol{v}_{12},\ \frac{\boldsymbol{n}_4+\boldsymbol{n}_3}{2} - \frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{2},\ \frac{\boldsymbol{n}_5+\boldsymbol{n}_6}{2} - \frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{2}\right) \\ &= \det\left(\boldsymbol{v}_{12},\ \left[\frac{\boldsymbol{n}_4+\boldsymbol{n}_3}{2} - \boldsymbol{n}_1\right] + \left[\boldsymbol{n}_1 - \frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{2}\right],\ \left[\frac{\boldsymbol{n}_5+\boldsymbol{n}_6}{2} - \boldsymbol{n}_1\right] + \left[\boldsymbol{n}_1 - \frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{2}\right]\right) \end{aligned}$$

where the terms $\left[\boldsymbol{n}_1 - \frac{\boldsymbol{n}_1+\boldsymbol{n}_2}{2}\right]$ are equal to $-\frac{\boldsymbol{v}_{12}}{2}$. By trilinearity of the determinant and dependency with respect to the first vector $(\boldsymbol{v}_{12})$, the terms $-\frac{\boldsymbol{v}_{12}}{2}$ vanish and we obtain:

$$J_9 = 6\,\mathrm{vol}\left(\boldsymbol{n}_1,\ \boldsymbol{n}_2,\ \frac{\boldsymbol{n}_4+\boldsymbol{n}_3}{2},\ \frac{\boldsymbol{n}_5+\boldsymbol{n}_6}{2}\right)$$

Figure 4 shows the tetrahedra that correspond to four value of the Jacobian determinant.

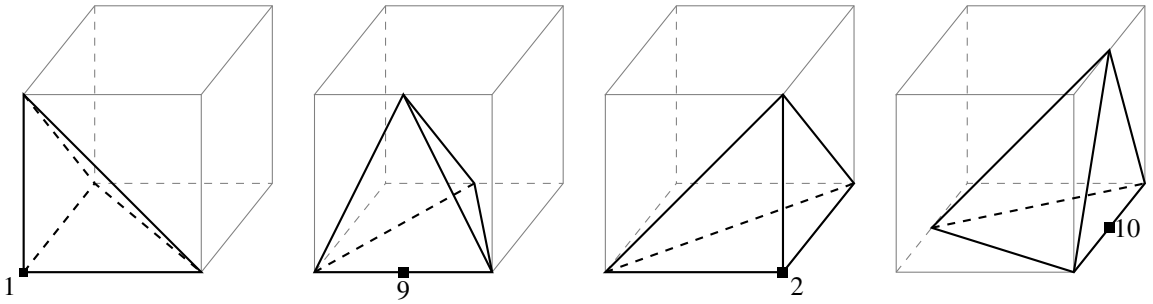

Fig. 4. Different tetrahedra whose volume corresponds to the value of the respective coefficients J_1, J_9, J_2 and J_{10} divided by 6.

6. The algorithm

The algorithm that computes the validity of a linear hexahedron takes as input the 8 nodes coordinates of the element. It returns true if the element is valid and return false if the element is invalid. The execution is the following:

1. Compute the 20 volumes of Section 5 and put them in vector $\boldsymbol{v}$ (ordering them as in Figure 3).
2. If at least one volume is negative, return False.
3. Compute the Bézier coefficients $\boldsymbol{b} = \boldsymbol{Q}\boldsymbol{v}$ where $\boldsymbol{Q}$ is the matrix given in Table 3.
4. If all the Bézier coefficients in $\boldsymbol{b}_{9\to 27}$ are positive, return True.
5. Return `recursive_subdivision`($\boldsymbol{b}$).

In Step 4, the 8 first Bézier coefficients are equal to the volume of the corner tetrahedra and must be positive otherwise the algorithm would have stop at Step 2.

The subdivision algorithm, `recursive_subdivision`($\boldsymbol{b}$), is identical to the subdivision algorithm presented in paper [1] (although implemented in a more efficient manner in our new implementation). It takes a vector of 27 Bézier coefficients as input and return true if the Jacobian determinant is strictly positive on the subdomain, otherwise it returns false. The algorithm is:

1. Subdivide: Compute the subcoefficients $\boldsymbol{b}^i,\ i = 1, \ldots, 8$ as described in paper [1].
2. For each $\boldsymbol{b}^i$:
3. If at least one of the coefficients in $\boldsymbol{b}^i_{1\to 8}$ is negative, return False.
4. If all the coefficients in $\boldsymbol{b}^i_{9\to 27}$ are positive, continue the loop.
5. If `recursive_subdivision`($\boldsymbol{b}^i$) is false, return False.
6. Return True.

In Step 3 of this algorithm, it is checked if the 8 first Bézier coefficients are not negative since they are actual values of the Jacobian determinant. In Step 4, the positivity of the 19 other coefficients ensures that the Jacobian determinant is positive on the corresponding subdomain in which case the algorithm skip Step 5 and continue the loop. While there is no negative real value of the Jacobian determinant but at least one negative Bézier coefficients, the algorithm subdivide (Step 5).

7. Results

We begin the results with unitary tests. The Jacobian determinant of the hexahedron defined in Figure 5 is positive at the 8 corners, the center of the edges, the center of the faces and the center of the volume. Moreover, the hexahedron passes the Ushakova's [15] test6 that requires the computation of 24 tetrahedral volumes. Our algorithm detects that this hexahedron is invalid. Figure 6 presents a hexahedron that does not pass Ushakova's [15] test6, despite the fact that

i	x	y	z
0	0	0	0
1	1	0	0
2	1.7615170641459	0.594764272968121	0.15552188663289
3	0.438888411629833	1.53098020041072	0.185631029838277
4	1.3859049651391	0.0755794018509022	1.77483024073906
5	1.22129676447071	0.271876165350328	0.630922158503566
6	1.77365642274365	1.25103990471942	1.83300604452892
7	0.0769922201302364	0.940424880836765	1.45521546591891

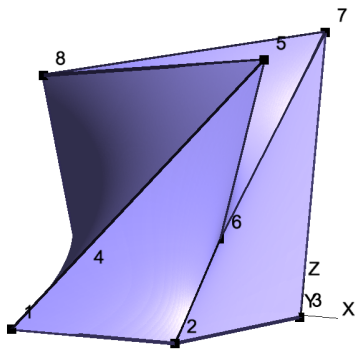

Fig. 5. Invalid hexahedron for which the Jacobian determinant is positive at the 27 nodes of the second-order hexahedron and for which the 24 tetrahedral volumes of Ushakova's [15] test6 are all positive.

the element is valid. In hexahedral mesh community, it is common to measure the quality of hexahedra by computing the minimum of the "scaled Jacobian" on the 8 corners [23,24]. For the hexahedron of Figure 7, this quality measure

i	x	y	z
0	0	0	0
1	1	0	0
2	1.539	0.704696	1.84011
3	0.166589	1.08208	0.162539
4	0.0501127	1.96347	1.56559
5	0.422336	0.00419138	1.43038
6	0.509917	0.0214216	1.55322
7	0.40783	1.73452	1.93234

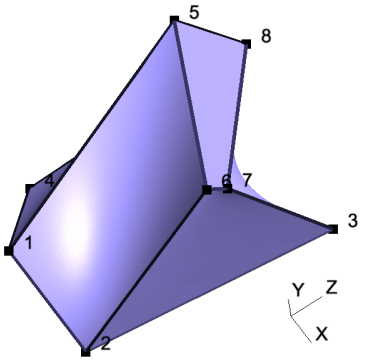

Fig. 6. Valid hexahedron that does not pass Ushakova's [15] test6.

is equal to 0.64 although the element is invalid. This demonstrates that even invalid hexahedra can have a good quality at the corners.

i	x	y	z
1	0.464949491866817	0.358989226966155	0.0133365886410108
2	0.481795709097567	0.358745078890347	0.0163884395886105
3	0.482406079287087	0.351664784691916	0.0235297708059938
4	0.466719565416425	0.339945677053133	0.0278023621326335
5	0.465498825037385	0.320291756950591	-0.00277718436231578
6	0.465987121189001	0.321085238196966	-0.0042420728171636
7	0.501998962370677	0.322367015594958	-0.0116275521103549
8	0.487166966765343	0.308816797387616	0.0115054780724508

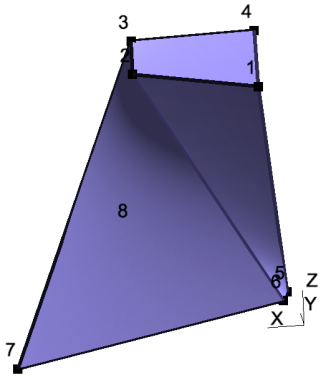

Fig. 7. Invalid hexahedron for which the minimum of the scaled Jacobian computed at the corners is equal to 0.64.

For the next experimentation, we compare our method with some previous methods on different datasets. The results are given in Table 4. The datasets have been generated by the algorithm described in [8] which takes a tetrahedral mesh as input and computes hexahedra that can be created by combining tetrahedra. This algorithm can generate a large amount of hexahedra of different qualtity. We have considered two models. The first one, "Fusee_1", contains 71,947 vertices and 349,893 tetrahedra. The second one is "FT47" and contains 370,401 vertices and 2,085,394 tetrahedra. Both of them are available on the website `www.hextreme.eu`. We have disabled the validity check during the hexahedra creation and, for each model, we have generated three datasets of hexahedra by varying the desired minimal quality q (computed at the corners). Datasets that correspond to $q = -1$ contain a large proportion of invalid hexahedra while datasets that correspond to $q = 0.5$ contain only valid hexahedra. We have compared our new implementation with the previous one [1], as well as the 5 first validity tests presented in [15]. These tests consist in computing the volume of respectively 8, 10, 24, 32 and 58 tetrahedra and returning False as soon as a negative volume is found or returning True if no negative volume is obtained. For each algorithm we store the execution time as well as the number of false valid (the number of invalid hexahedra that pass the test) and the number of false invalid (the number of valid hexahedra that do not pass the test). The experimentation has been conducted in serial on a MacBook Pro 2016 @ 2.9 GHz.

Our new implementation detects the same invalid hexahedra than our previous implementation. We have taken this result as the reference for computing the false invalid and false valid elements of the methods from [15]. Test 1

Table 4. Comparison of our new implementation with some previous methods. The datasets differ in the number of hexahedra and proportion of invalid element amongst them. For each method, the computation time, the number of false valid and the number of false invalid are given. Numbers are given with 3 significant digits.

Dataset	# hex	# invalid		Ours	Johnen et al.[1]	Test 1 [15]	Test 2 [15]	Test 3 [15]	Test 4 [15]	Test 5 [15]
Fusee_1, $q = 0.5$	334,000	0	# false valid	0	0	0	0	0	0	0
			# false invalid	0	0	0	529	10,000	10,000	111,000
			time [s]	0.0565	0.812	0.0180	0.0168	0.0349	0.0549	0.0866
Fusee_1, $q = 0$	2,040,000	79,900	# false valid	0	0	79,900	73,900	0	0	0
			# false invalid	0	0	0	10,200	814,000	814,000	1,590,000
			time [s]	0.339	6.60	0.0945	0.0977	0.161	0.220	0.358
Fusee_1, $q = -1$	6,060,000	4,110,000	# false valid	0	0	80,000	74,000	48,400	0	0
			# false invalid	0	0	0	10,200	814,000	814,000	1,590,000
			time [s]	0.488	15.5	0.202	0.212	0.301	0.347	0.418
FT47, $q = 0.5$	3,000,000	0	# false valid	0	0	0	0	0	0	0
			# false invalid	0	0	0	1,890	115,000	115,000	1,060,000
			time [s]	0.459	6.93	0.181	0.203	0.341	0.463	0.725
FT47, $q = 0$	14,700,000	366,000	# false valid	0	0	366,000	342,000	7	7	0
			# false invalid	0	0	0	38,900	4,880,000	4,880,000	11,100,000
			time [s]	2.43	42.5	0.712	0.872	1.30	1.73	2.33
FT47, $q = -1$	40,500,000	26,100,000	# false valid	0	0	370,000	346,000	247,000	7	0
			# false invalid	0	0	0	38,900	4,880,000	4,880,000	11,100,000
			time [s]	3.17	102	1.55	1.54	2.08	2.49	3.09

computes the volume of corner tetrahedra, which corresponds to a necessary condition. As expected, Test 1 misses invalid elements but never finds false invalid. Test 2 to Test 4 are neither sufficient nor necessary. Test 4 misses very few invalid hexahedra, however. Test 5 corresponds to a sufficient condition and can miss as much as 80% of valid elements (see dataset Fusee_1, $q = -1$).

Our new implementation is about 15 to 30 time faster than the algorithm designed for curvilinear elements and runs at similar speed than Test 5 of [15] which consists in computing 58 tetrahedral volumes. Our new algorithm can check the validity of hexahedra at a rate of between 6 million and 12 million hexahedra per second on a single core. The speed is higher when there is a large proportion of invalid hexahedra since the algorithm can stop at an early stage if a negative Jacobian determinant is obtained.

8. Conclusion

Our implementation is able to check the validity of linear hexahedral elements in a very efficient manner. The algorithm benefit from the robustness of the previous method for checking the validity of curvilinear elements [1] on which it is based. The novelty consists of two improvements: (1) a reduced number of quantities to be computed at the beginning of the algorithm and (2) the computation of those quantities as tetrahedral volumes instead of the Jacobian determinant. The particularization to hexahedra also permits a fine-tuned implementation. Our new code runs more than 15 time faster than the previous code for curvilinear elements and runs at similar speed than the sufficient but not necessary method presented in [15]. More than 6 million hexahedra per second can be analyzed on a single core of a personal computer. The algorithm is simple and can readily be implemented from the information given in this paper. The C++ code will be available in Gmsh (`www.gmsh.info`).

Acknowledgements

This research project was funded by the European Research Council (project HEXTREME, ERC-2015-AdG-694020) and the TILDA project. The TILDA (Towards Industrial LES/DNS in Aeronautics - Paving the Way for Future Accurate CFD) project has received funding from the European Unions Horizon 2020 research and innovation program under grant agreement No 635962. The project is a collaboration between NUMECA, DLR, ONERA, DASSAULT, SAFRAN, CERFACS, CENAERO, UCL, UNIBG, ICL and TsAGI.

Appendix A. Lagrange shape functions of the linear hexahedron

In this paper, we consider the following Lagrange shape functions for the linear hexahedron:

$$\begin{cases} L_1(\xi,\eta,\zeta) = (1-\xi)\ (1-\eta)\ (1-\zeta) \\ L_2(\xi,\eta,\zeta) = \quad \xi \quad (1-\eta)\ (1-\zeta) \\ L_3(\xi,\eta,\zeta) = \quad \xi \quad\ \ \eta \quad (1-\zeta) \\ L_4(\xi,\eta,\zeta) = (1-\xi) \quad \eta \quad (1-\zeta) \\ L_5(\xi,\eta,\zeta) = (1-\xi)\ (1-\eta) \quad \zeta \\ L_6(\xi,\eta,\zeta) = \quad \xi \quad (1-\eta) \quad \zeta \\ L_7(\xi,\eta,\zeta) = \quad \xi \quad\ \ \eta \quad\ \ \zeta \\ L_8(\xi,\eta,\zeta) = (1-\xi) \quad \eta \quad\ \ \zeta. \end{cases}$$

References

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

[2] J.-F. Remacle, R. Gandham, T. Warburton, Gpu accelerated spectral finite elements on all-hex meshes, Journal of Computational Physics 324 (2016) 246–257.

[3] E. Wang, T. Nelson, R. Rauch, Back to elementstetrahedra vs. hexahedra, in: Proceedings of the 2004 International ANSYS Conference, ANSYS Pennsylvania, 2004.

[4] S. E. Benzley, E. Perry, K. Merkley, B. Clark, G. Sjaardama, A comparison of all-hexagonal and all-tetrahedral finite element meshes for elastic and elasto-plastic analysis, in: Proceedings of the 4th International Meshing Roundtable, volume 17, Sandia National Laboratories Albuquerque, NM, 1995, pp. 179–191.

[5] T. C. Baudouin, J.-F. Remacle, E. Marchandise, F. Henrotte, C. Geuzaine, A frontal approach to hex-dominant mesh generation, Advanced Modeling and Simulation in Engineering Sciences 1 (2014) 1–30.

[6] A. Botella, B. Lévy, G. Caumon, Indirect unstructured hex-dominant mesh generation using tetrahedra recombination, Computational Geosciences 20 (2016) 437–451.

[7] D. Sokolov, N. Ray, L. Untereiner, B. Lévy, Hexahedral-dominant meshing, ACM Transactions on Graphics (TOG) 35 (2016) 157.

[8] J. Pellerin, A. Johnen, J.-F. Remacle, Identifying combinations of tetrahedra into hexahedra: a vertex based strategy, in: Proceedings of the 26th International Meshing Roundtable, 2017.

[9] P. M. Knupp, On the invertibility of the isoparametric map, Computer Methods in Applied Mechanics and Engineering 78 (1990) 313–329.

[10] S. A. Ivanenko, Harmonic mappings, in: Handbook of grid generation, CRC Press Boca Raton, Fl, 1999.

[11] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, Journal of Computational Physics 148 (1999) 433–466.

[12] O. V. Ushakova, Conditions of nondegeneracy of three-dimensional cells. a formula of a volume of cells, SIAM Journal on Scientific Computing 23 (2001) 1274–1290.

[13] S. Vavasis, A bernstein-bezier sufficient condition for invertibility of polynomial mapping functions, 2003. Draft.

[14] Z. Shangyou, Subtetrabedral test for the positive jacobian of hexaherdral elements, 2005. Unpublished.

[15] O. V. Ushakova, Nondegeneracy tests for hexahedral cells, Computer Methods in Applied Mechanics and Engineering 200 (2011) 1649–1658.

[16] P. Knabner, S. Korotov, G. Summ, Conditions for the invertibility of the isoparametric mapping for hexahedral finite elements, Finite elements in analysis and design 40 (2003) 159–172.

[17] C. Geuzaine, J.-F. Remacle, Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities, International Journal for Numerical Methods in Engineering 79 (2009) 1309–1331.

[18] A. E. Frey, C. A. Hall, T. A. Porsching, Some results on the global inversion of bilinear and quadratic isoparametric finite element transformations, Mathematics of Computation 32 (1978) 725–749.

[19] S. Zhang, Invertible jacobian for hexahedral finite elements. part 1. bijectivity, `http://www.math.udel.edu/~szhang/research/p/bijective1.ps`, 2005.

[20] R. Leroy, Certificats de positivité et minimisation polynomiale dans la base de Bernstein multivariée, Ph.D. thesis, Université de Rennes 1, 2008.

[21] R. Leroy, Certificates of positivity in the simplicial bernstein basis, `https://hal.archives-ouvertes.fr/hal-00589945/document`, 2011.

[22] E. Cohen, L. L. Schumaker, Rates of convergence of control polygons, Computer Aided Geometric Design 2 (1985) 229–235.

[23] P. M. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. part II—A framework for volume mesh optimization and the condition number of the jacobian matrix, International Journal for Numerical Methods in Engineering 48 (2000) 1165–1185.

[24] S. Yamakawa, K. Shimada, Fully-automated hex-dominant mesh generation with directionality control via packing rectangular solid cells, International Journal for Numerical Methods in Engineering 57 (2003) 2099–2129.