Technische Universität Chemnitz

Sonderforschungsbereich 393

Numerische Simulation auf massiv parallelen Rechnern

Thomas Apel Nico Düvelmeyer

Transformation of hexahedral finite element meshes into tetrahedral meshes according to quality criteria

Preprint SFB393/03-09

Abstract The paper is concerned with algorithms for transforming hexahedral finite element meshes into tetrahedral meshes without introducing new nodes. Known algorithms use only the topological structure of the hexahedral mesh but no geometry information. The paper provides another algorithm which is then extented such that quality criteria for the splitting of faces are respected.

Key Words finite element mesh, mesh generation, hexahedral mesh, tetrahedral mesh

AMS subject classification 65N50; 51M20, 52C22, 05B45

Preprint-Reihe des Chemnitzer SFB 393

ISSN 1619-7178 (Print) **ISSN 1619-7186** (Internet)

SFB393/03-09 **May 2003**

Contents

Authors' addresses:

Thomas Apel
Nico Düvelmeyer

TU Chemnitz
Fakultät für Mathematik
D-09107 Chemnitz

`na.apel@na-net.ornl.gov`
`http://www.tu-chemnitz.de/~tap`

`Nico.Duevelmeyer@mathematik.tu-chemnitz.de`
`http://www.tu-chemnitz.de/~nduev`

1 Introduction

For three-dimensional finite element calculations both hexahedral and tetrahedral meshes are widely used. The authors have experience with both types of mesh but our adaptive finite element code for tetrahedral elements [1] is better developed than that for hexahedral meshes [2]. There are two reasons why we might want to transform a hexahedral mesh into a tetrahedral one, namely comparisons of both programs for the same problem, and complicated third party hexahedral meshes that we might want to use with our tetrahedral code.

Both the given hexahedral and the desired tetrahedral mesh should satisfy the admissibility conditions in the sense of Ciarlet [3, pages 38, 51], in particular we assume that any face of any element is either a subset of the boundary of the domain or face of another element of the mesh. We do not consider meshes with so-called hanging nodes.

It is simple to subdivide one hexahedron into five or six tetrahedra. At the first glance it seems easy to subdivide each hexahedron of a mesh to get a tetrahedral mesh. But these local subdivisions are not independent from each other: to be an admissible mesh, faces must be split in the same way in both adjacent elements. The next idea might be first to split all faces and then to subdivide the hexahedra according to this face partition. However, it turns out that this is not possible for arbitrary face-partitions, see Section 2.

Nevertheless, the problem *is* easy to solve, see [6, page 16]: Enumerate all vertices globally. Then divide each quadrilateral face by the diagonal that starts from the vertex with the minimal vertex number (of that face). This ensures compatibility of the mesh. Consider an arbitrary hexahedron h and denote by v that vertex of h with minimal number. Three faces of h meet in the vertex v. Observe that the diagonals of these three faces have v in common. This ensures that one can split the hexahedron into six tetrahedra in the way that v is common vertex of all these tetrahedra.

The disadvantage of this method is that geometry information is not used. Since we are given sometimes quite distorted meshes we prefer face-splittings that divide large angles. We will introduce a preference function $a(f)$ that assigns each face f the preferred splitting. Since, as mentioned above, it is not possible to generate tetrahedral meshes for arbitrary face-partitions we have to reject the preference in some cases. In order not to reject arbitrary preferences we use another function $b(f_i, f_j)$ that indicates in cases of conflicting preferences $a(f_i)$ and $a(f_j)$ which of the two preferences is more important.

Tetrahedra have plane triangular faces whereas the shape of a hexahedron is defined in general by the position of its vertices via the iso-parametric mapping [3, pages 224 ff.]. This leads to curved rectangular faces which cannot be split into plane triangles exactly. To avoid technicalities in the definitions below we simply restrict ourselves to convex hexahedra with plane faces; they are the convex hull of their vertices.

Problem 1 *We are given an admissible hexahedral mesh* $M = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{H})$ *defined by a set* $\mathcal{V}$ *of vertices* $v = (x, y, z) \in \mathbf{R}^3$*, a set* $\mathcal{E}$ *of edges* $e = [ab]$ *(*$\{a, b\} \subset \mathcal{V}$*), a set* $\mathcal{F}$ *of faces* $f = \operatorname{conv}\{a, b, c, d\}$ *(*$\{a, b, c, d\} \subset \mathcal{V}$*), and a set* $\mathcal{H}$ *of hexahedra* $h = \operatorname{conv}\{a, b, c, d, e, f, g, h\}$ *(*$\{a, b, c, d, e, f, g, h\} \subset \mathcal{V}$*).*

We look for an admissible tetrahedral mesh $M' = (\mathcal{V}, \mathcal{E}', \mathcal{F}', \mathcal{T})$ *with the same set of vertices* $\mathcal{V}$*, but new sets of edges, faces and elements, denoted by* $\mathcal{E}'$*,* $\mathcal{F}'$*, and* $\mathcal{T}$*, respectively. Each hexahedron has to be the union of (5 or 6) tetrahedra.*

	Number of faces of				
Type	type 1	type 2	type 3	Volume	ϕ_{max}
A	3	0	1	$1/6$	$90°$
B	2	2	0	$1/6$	$90°$
C	1	2	1	$1/6$	$\approx 125.26°$
D	0	0	4	$1/3$	$\approx 70.53°$

Table 1: Characterization of the types of proper tetrahedra

Whenever possible the faces should be split as indicated by the preference function $a(.)$. If the algorithm cannot satisfy both preferences for two faces f_i and f_j then the preference for face $b(f_i, f_j)$ is more important.

As mentioned above there are face-partitions of a hexahedron without a corresponding tetrahedral subdivision of this hexahedron. In Section 2, we will derive a necessary and sufficient property of the face-partition such that a tetrahedral subdivision exists. Section 3 is then devoted to the construction of face-partitions with this property. Hence Problem 1 is solved.

Related work is done by Hacon and Tomei [5]. These authors investigate conditions such that certain special partitioning procedures work. This is a different goal.

2 Local theory

Let us consider a mesh that consists only of one hexahedron h, $\mathcal{H} = \{h\}$. For the ease of description in this section, the considered element is the unit cube, $h = [0,1]^3$. All results of this section can be shown also for more general cases unless modifications are indicated.

We number the vertices of the element as illustrated in Figure 1 and get consequently $\mathcal{V} = \{v_1, v_2, \ldots, v_8\}$, $\mathcal{E} = \{[v_1v_2], [v_2v_3], [v_3v_4], [v_4v_1], [v_5v_6], [v_6v_7], [v_7v_8], [v_8v_5], [v_1v_5], [v_2v_6], [v_3v_7], [v_4v_8]\}$, $\mathcal{F} = \{\operatorname{conv}\{v_1, v_2, v_3, v_4\}, \operatorname{conv}\{v_1, v_2, v_6, v_5\}, \operatorname{conv}\{v_1, v_4, v_8, v_5\}, \operatorname{conv}\{v_2, v_3, v_7, v_6\}, \operatorname{conv}\{v_5, v_6, v_7, v_8\}, \operatorname{conv}\{v_4, v_3, v_7, v_8\}\}$, $h = \operatorname{conv}\{v_1, v_2, \ldots, v_8\}$.

Edges in $\mathcal{E}'$ can only be edges $e \in \mathcal{E}$ (such as $[v_1v_2]$), diagonals of faces $f \in \mathcal{F}$ (like $[v_1v_3]$) and spatial diagonals of h (like $[v_1v_7]$).

We distinguish three possible types of faces (triangles) in $\mathcal{F}'$, see also Figure 2, type 1: right isosceles as one half of a face of $\mathcal{F}$ (e.g. $\triangle v_1v_2v_6$); type 2: right-angled triangles with one edge, one face diagonal and one spatial diagonal of h as sides (e.g. $\triangle v_2v_4v_6$); and type 3: equilateral triangles bounded by face-diagonals (e.g. $\triangle v_1v_6v_8$).

There are four types of *proper* tetrahedra, see Figure 3. Types A, B, and C have three vertices in a face $f \in \mathcal{F}$, and the fourth in the parallel face. The only tetrahedra without a face within the boundary of the cube are regular and of type D. All six edges of these tetrahedra are face-diagonals of M. For all the types of tetrahedra, Table 1 summarizes the types of faces, the volume and the maximal angles between faces, ϕ_{max}. Figure 3 shows examples for each type of tetrahedron.

Remark 2 *One can consider two further types of sets of four vertices. In our example where the hexahedron is a cube, these points are planar and do not form a proper tetrahedron. For*

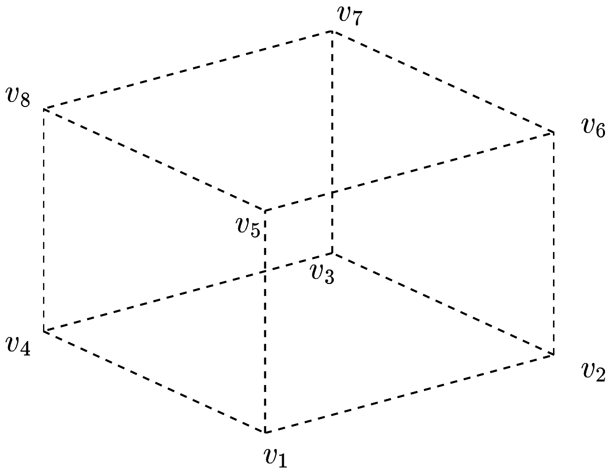
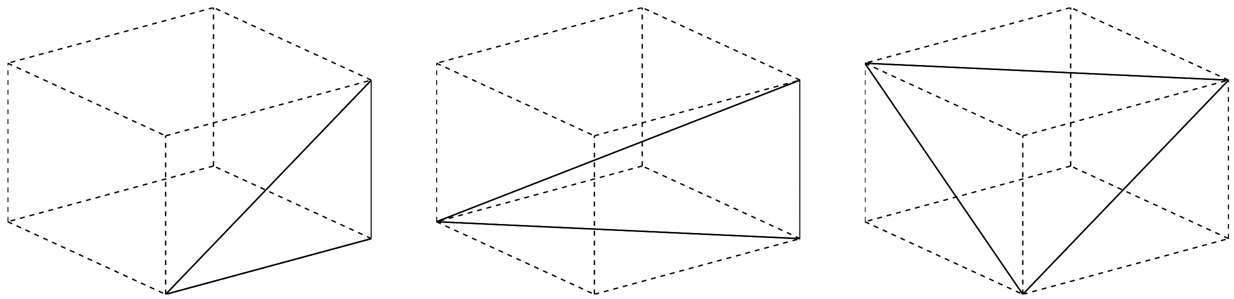
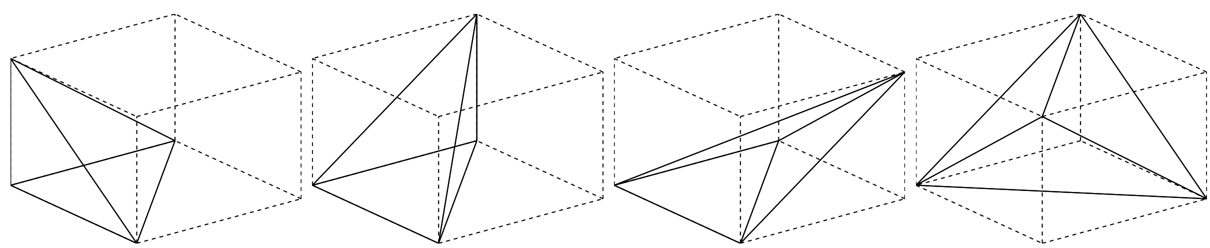


Figure 1: Numbering of vertices in $\mathcal{V}$



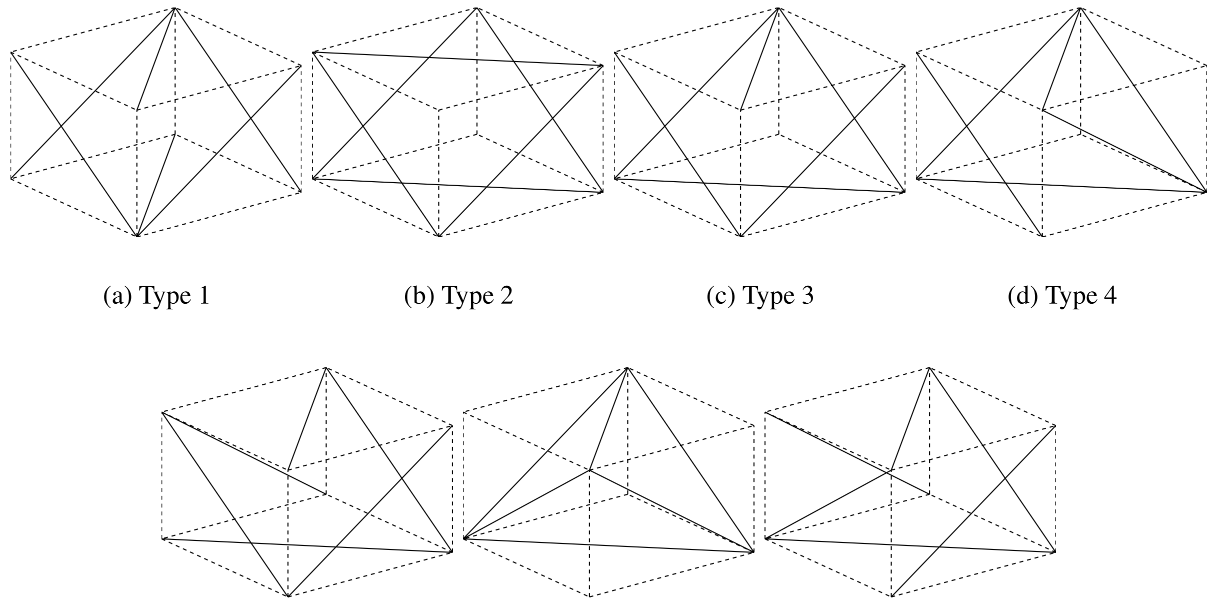
(a) Type 1 (b) Type 2 (c) Type 3

Figure 2: Three types of triangular faces



(a) Type A (b) Type B (c) Type C (d) Type D

Figure 3: Illustration of the types of proper tetrahedra



(a) Type 1 (b) Type 2 (c) Type 3 (d) Type 4

(e) Type 5 (f) Type 6 (g) Type 7

Figure 4: Illustration of the types of face-partitions of a hexahedron

more general hexahedra they may become tetrahedra with $\phi_{max} \approx \pi$ which is not desirable from the numerical point of view. Therefore we do not call them proper, and we postulate that the algorithms do not use them.

A face $f \in \mathcal{F}$ can be split in two ways into two triangles. We can describe it by the vertices of the corresponding diagonal. The $2^6 = 64$ possible face-partitions of the six faces in $\mathcal{F}$ can be grouped into 7 types. Two face-partitions belong to the same type if there is an affine transformation transforming all diagonals from the first to those of the second face-partition. Figure 4 shows examples of all types, Table 2 lists the number of face-partitions of each type and the symbolic representation of the examples. The fact that there are $2^6 = 64 = 4 + 4 + 24 + 12 + 12 + 2 + 6$ different face-partitions shows the completeness of this list.

Now we can state the first result:

Theorem 3 *For a given face-partition of an element h of type 1, 2, 3, 4 or 6 there exists a subdivision of h into a mesh with proper tetrahedra which induces this face-partition. For face-partitions of type 5 and 7 such a subdivision does not exist.*

Proof The first statement can be proven by giving a mesh (list of vertices of the tetrahedra), see Table 3 and Figure 5. For completeness there are given two different possible subdivisions for face-partitions of type 6. Table 4 shows how many subdivisions exist for each type as well as the types of the tetrahedra. Several possibilities are mentioned in one row if they can be transformed into each other by rotation and reflection.

Type	Occurrences	Example (diagonals are given)
1	4	$[v_1v_3],\ [v_5v_7],\ [v_1v_6],\ [v_2v_7],\ [v_4v_7],\ [v_1v_8]$
2	4	$[v_2v_4],\ [v_6v_8],\ [v_1v_6],\ [v_2v_7],\ [v_4v_7],\ [v_1v_8]$
3	24	$[v_2v_4],\ [v_5v_7],\ [v_1v_6],\ [v_2v_7],\ [v_4v_7],\ [v_1v_8]$
4	12	$[v_2v_4],\ [v_5v_7],\ [v_2v_5],\ [v_2v_7],\ [v_4v_7],\ [v_1v_8]$
5	12	$[v_2v_4],\ [v_5v_7],\ [v_1v_6],\ [v_2v_7],\ [v_3v_8],\ [v_1v_8]$
6	2	$[v_2v_4],\ [v_5v_7],\ [v_2v_5],\ [v_2v_7],\ [v_4v_7],\ [v_4v_5]$
7	6	$[v_2v_4],\ [v_5v_7],\ [v_1v_6],\ [v_2v_7],\ [v_3v_8],\ [v_4v_5]$

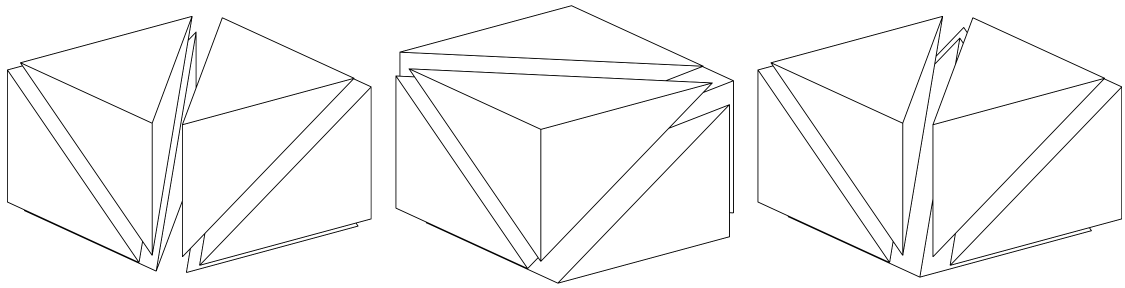
Table 2: Types of face-partitions

Type	t_1	t_2	t_3	t_4	t_5	t_6
1	$\{1,2,3,7\}$	$\{1,3,4,7\}$	$\{1,2,7,6\}$	$\{1,5,6,7\}$	$\{1,4,8,7\}$	$\{1,5,7,8\}$
2	$\{1,2,4,6\}$	$\{2,3,4,7\}$	$\{1,5,6,8\}$	$\{1,4,8,6\}$	$\{2,4,6,7\}$	$\{4,6,7,8\}$
3	$\{1,2,4,7\}$	$\{2,3,4,7\}$	$\{1,2,7,6\}$	$\{1,5,6,7\}$	$\{1,4,8,7\}$	$\{1,5,7,8\}$
4	$\{1,2,4,7\}$	$\{2,3,4,7\}$	$\{1,2,7,5\}$	$\{2,5,6,7\}$	$\{1,4,8,7\}$	$\{1,5,7,8\}$
6	$\{1,2,4,5\}$	$\{2,3,4,5\}$	$\{2,5,6,7\}$	$\{4,5,7,8\}$	$\{2,3,5,7\}$	$\{3,4,5,7\}$
6	$\{1,2,4,5\}$	$\{3,2,7,4\}$	$\{6,5,7,2\}$	$\{8,7,5,4\}$	$\{1,3,8,6\}$	–

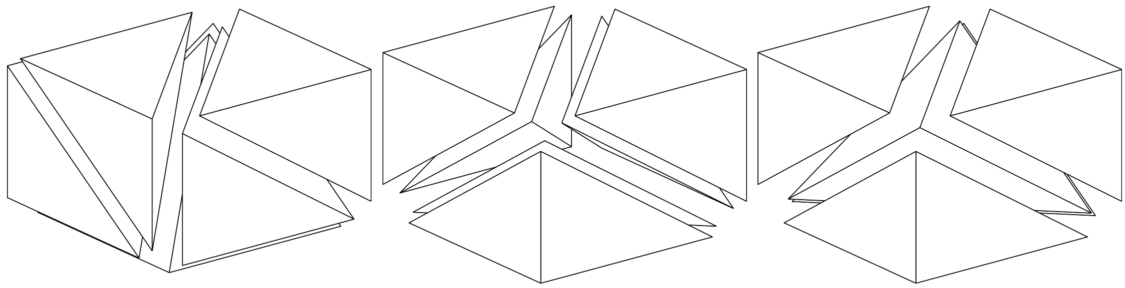
Table 3: Example subdivisions for a given type of face-partition, indicated by the numbers of the vertices

Type of face partition	Number of possibilities	Number of tetrahedra of type A	type B	type C	type D
1	1		6		
2	3	2	2	2	
3	1	1	4	1	
4	2	2	2	2	
6	4	3		3	
6	1	4			1

Table 4: Subdivisions with number and types of tetrahedra



(a) Type 1 (b) Type 2 (c) Type 3



(d) Type 4 (e) Type 6 (1) (f) Type 6 (2)

Figure 5: Example subdivisions for a given type of face-partition

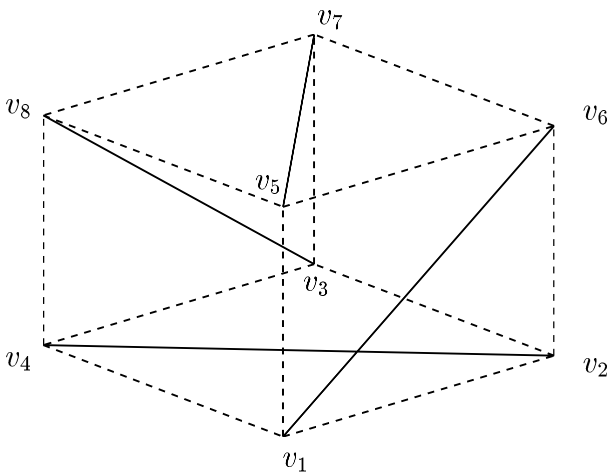


Figure 6: Conflicting partitions of four faces

For face-partitions of type 5 and 7, one can choose four face-diagonals that do not have a common point (e.g. the diagonals $[v_2v_4], [v_5v_7], [v_1v_6]$ and $[v_3v_8]$, see Figure 6). Now we assume that a subdivision of h into proper tetrahedra would partition the corresponding sides in this way and argument by contradiction. The triangle $\triangle v_1v_2v_6$ is the face of a tetrahedron with vertices v_1, v_2, v_6 and $v_i \in \{v_3, v_4, v_7, v_8\}$. The choice $i = 3$ contradicts the partition of the face $\text{conv}\{v_1, v_2, v_3, v_4\}$. The other three cases imply that one of the inner diagonals $[v_4v_6]$, $[v_1v_7]$ or $[v_2v_8]$ belong to $\mathcal{E}'$. Since there can only be one inner diagonal in $\mathcal{E}'$, we get $[v_3v_5] \notin \mathcal{E}'$.

Repeating the argument with triangles $\triangle v_5v_6v_7$, $\triangle v_3v_7v_8$, and $\triangle v_2v_3v_4$ excludes the other spatial diagonals $[v_2v_8]$, $[v_4v_6]$, and $[v_1v_7]$, respectively. Since this contrasts the above statement, Theorem 3 is proved. $\square$

Now we introduce some notation that is useful in the context of a finite element mesh with more than one element. For a hexahedron h a *face-pair* p consists of two non-adjacent faces of h. In the special case of a cube, these are pairs of parallel faces. There are exactly three face-pairs per hexahedron. With a given face-partition there are two *partitioning diagonals* $[v_iv_j]$ and $[v_kv_l]$ for a given face-pair p. The pair p is called *inversive partitioned*, if $\text{conv}\{v_i, v_j, v_k, v_l\}$ is a tetrahedron of type D, otherwise it is called *parallel partitioned*. In the first case the tetrahedron $\text{conv}\{v_i, v_j, v_k, v_l\}$ is called the *orientation tetrahedron* of p. Note that there are only two possibilities for an orientation tetrahedron. With this notation we can reformulate Theorem 3 if we consider all types of face-partitions of h.

Corollary 4 *For a given face-partition, there exists a subdivision of a hexahedron h into proper tetrahedra (which induces this face-partition) if and only if all inversive partitioned face-pairs produce the same orientation tetrahedron.*

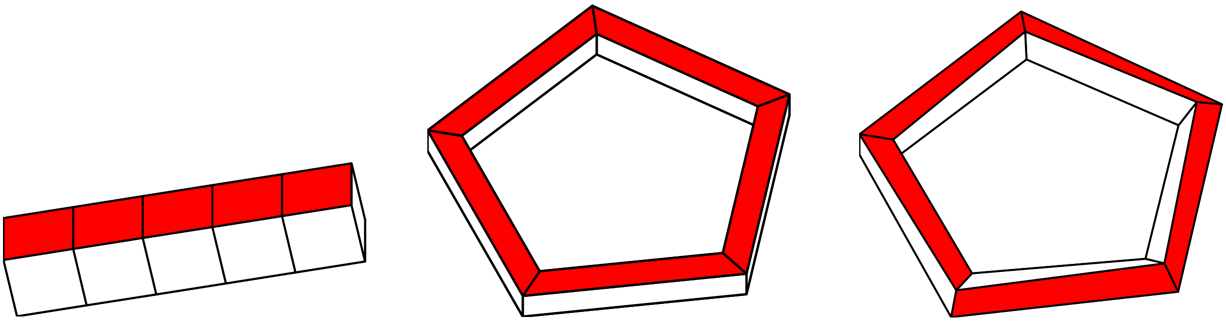
3 Global theory

Let us now introduce a binary relation $\mathbf{P}$ in the set $\mathcal{F}$ of faces:

$$f_i\mathbf{P}f_j \iff \begin{array}{l} i = j \quad \text{or} \quad \exists n \in \mathbf{N}, f_{i_0}, \dots, f_{i_n} \in \mathcal{F}: \\ i_0 = i, i_n = j, \{f_{i_{k-1}}, f_{i_k}\} \text{ forms a face-pair } (\forall k = 1, \dots, n). \end{array}$$

That means, two faces f_i and f_j are in relation, $f_i\mathbf{P}f_j$, if they are equal or if there exists a sequence $f_{i_0}, \dots, f_{i_n}$ of faces such that successive faces form face-pairs in the sense of Section 2.

The relation $\mathbf{P}$ is by construction a congruence relation. Because no face belongs to more than two face-pairs, each congruence class $[f]$ has a linear structure: $[f] = \{f_{i_0}, f_{i_1}, \dots, f_{i_n}\}$ where $\{f_{i_{k-1}}, f_{i_k}\}$ forms a face-pair $(k = 1, \dots, n)$. The faces f_{i_0} and f_{i_n} are either boundary



(a) Chain (b) Parallel ring (c) Twisted ring

Figure 7: Illustration of the different types of equivalence classes; the coloring of faces emphasize the difference between (b) and (c), the faces of the equivalence class are hidden in general

faces or form another face-pair. In the first case, the partition of all faces in $[f]$ can be chosen such that all face-pairs are parallel partitioned (simply by starting with f_{i_0} and continuing with f_{i_k}, $k = 1, \ldots, n$). In this case we call $[f]$ a *chain*, otherwise $[f]$ is called *ring*. We distinguish two sorts of rings. If a partition of the faces exists such that $\{f_{i_{k-1}}, f_{i_k}\}$, $k = 1, \ldots, n$, and also $\{f_{i_0}, f_{i_n}\}$ are parallel partitioned, we call $[f]$ a *parallel ring*. Otherwise $[f]$ is called *twisted ring*, see Figure 7. We note that parallel rings are simple to treat and twisted rings require more care. Note further that twisted rings can appear even in simply connected domains, see [5, Figure 7].

Algorithm 5 *Given a mesh $M = (\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{H})$ this algorithm produces an admissible mesh $M' = (\mathcal{V}, \mathcal{E}', \mathcal{F}', \mathcal{T})$.*

1. *For every element $h \in \mathcal{H}$ choose an orientation tetrahedron (type D) arbitrarily.*
2. *For every congruence class $[f]$ do:*
   - *If $[f]$ is a chain or a parallel ring choose a partition of all faces, such that all corresponding face-pairs are parallel partitioned.*
   - *If $[f]$ is a twisted ring choose a face-pair as f_{i_0}, f_{i_n}. Now choose the partition of these two faces such that this face-pair is inversive partitioned and the orientation tetrahedron equals the one chosen in Step 1. The other face-partitions of this ring are chosen such that all remaining face-pairs are parallel partitioned.*
3. *Every element $h \in \mathcal{H}$ is subdivided with the already chosen face-partition (in the preferred fashion).*

Corollary 4 shows the feasibility of Algorithm 5.

The partition of a face divides two angles and leaves the other two. For approximation reasons it is advantageous to avoid large angles, therefore (and may be also due to other optimality criteria) we can have a preference which face-splitting should be performed. Note however, that the angles can be of nearly equal size such that we might have no preference. We formulate this in mathematical terms by introducing a preference function of face-partitions $a : \mathcal{F} \to \mathcal{P}_2(\mathcal{V}) \cup \{\textit{no\_preference}\}$ ($\mathcal{P}_2(\mathcal{V})$ denotes the set of subsets of $\mathcal{V}$ with two elements), that means, for each face $f = \mathrm{conv}\{v_i, v_j, v_k, v_l\} \in \mathcal{F}$ we have

$$a(\mathrm{conv}\{v_i, v_j, v_k, v_l\}) = \begin{cases} \{v_i, v_k\} & \text{if } [v_i v_k] \text{ is preferred,} \\ \{v_j, v_l\} & \text{if } [v_j v_l] \text{ is preferred,} \\ \textit{no\_preference} & \text{if either diagonal is suited for the splitting.} \end{cases} \tag{1}$$

Since there might not exist a subdivision of the hexahedral mesh that satisfies all preferences let further $b : \mathcal{P}_2(\mathcal{F}) \to \mathcal{F}$ be another preference function defined in the sense that if we cannot use both preferred face-partitions for the faces f_i, f_j, we want to use the preferred face-partition of the face f_i iff $b(f_i, f_j) = f_i$.

Remark 6 *We could have used only one instead of two preference functions. For example we could define $a : \mathcal{F} \to \mathbf{R}$ where positive and negative function values distinguish the preference of the two diagonals and the zero value corresponds to the* no_preference *case. Instead of the preference function $b(f_i, f_j)$ we could exploit the absolute values $|a(f_i)|$ and $|a(f_j)|$. We remark, however, that our description with two functions restricts to the minimum information necessary. Note that $a(f)$ can have only three values, see (1). The function $b(f_i, f_j)$ can adopt only two possible values.*

Another definition seems to be useful. Assume we are given a vector $\underline{c} = [c_m]_{m=1}^{\#\mathcal{F}}$. The length of the vector is equal to the number $\#\mathcal{F}$ of faces in $\mathcal{F}$. Each index m corresponds to a face $f_m = \mathrm{conv}\{v_i, v_j, v_k, v_l\} \in \mathcal{F}$ and the vector entries satisfy $c_m \in \big\{\{v_i, v_k\}, \{v_j, v_l\}, \textit{no\_preference}\big\}$. The vector might be a representation of the function $a(.)$ but it can also be modified, see Algorithm 8 below. With respect to this vector $\underline{c}$, a congruence class $[f] = \{f_{i_0}, f_{i_1}, \ldots, f_{i_n}\}$ is split into *subchains* $(f_{i_j}, \ldots, f_{i_k})$. Here, a subchain $(f_{i_j}, \ldots, f_{i_k})$ is a maximal sequence of consecutive faces with $c_i = \textit{no\_preference}$, $i = i_{j+1}, \ldots, i_{k-1}$. The term *maximal* has a three-fold meaning. Generally it means that the first and the last faces of a subchain (f_{i_j} and f_{i_k}) have a preferred face-partition, $c_{i_j} \neq \textit{no\_preference}$ and $c_{i_k} \neq \textit{no\_preference}$ (otherwise we could enlarge the subchain) but all intermediate face have not. Alternatively, the first or the last face could be a boundary face of a chain (f_{i_0} or f_{i_n}). The third meaning occurs in a ring where all faces have no preferred face-partition. In this case the subchain coincides with $[f]$ (this is the only case where the faces f_{i_j}, f_{i_k} of the representation of the subchain are not unique).

Remark 7 *Note that in a ring we choose arbitrarily a face number to be i_0. In the case $c_{i_0} =$ no_preference there is also a subchain of the form $(f_{i_j}, \ldots, f_{i_n}, f_{i_0}, \ldots, f_{i_k})$ where $j \leq n$, $k > 0$.*

With this notation of a subchain we can formulate the following algorithm. The discussion is given afterwards.

Algorithm 8 *Given a mesh M as well as the preference functions a and b in the above sense, this algorithm produces a mesh $M' = (\mathcal{V}, \mathcal{E}', \mathcal{F}', \mathcal{T})$ as described in Problem 1.*

1. *For every element $h \in \mathcal{H}$ choose an orientation tetrahedron (type D) arbitrarily.*
   Initialize a vector $\underline{c} = [c_i]_{i=1}^{\#\mathcal{F}}$ with $c_i = a(f_i)$.
2. *For every congruence class $[f] = \{f_{i_0}, f_{i_1}, \ldots, f_{i_n}\}$ perform Steps 2a, 2b.*
   - *(a) Repeat until the the vector $\underline{c}$ is not modified any more:*
     For each subchain $(f_{i_j}, \ldots, f_{i_k})$ of $[f]$ do:
     - *i. Check all faces f_i, $i = i_j, \ldots, i_k$, with $c_i =$ no_preference for the possibility to set $c_i \neq$ no_preference such that each face-pair of the subchain is*
       - *either parallel partitioned*
       - *or inversive partitioned and the orientation tetrahedron equals the one chosen in Step 1.*
     - *ii. If Step 2(a)i is not successful (it is not possible to set all c_i, $i = i_j, \ldots, i_k$, to a value distinct from no_preference) set c_{i_j} or c_{i_k} to no_preference according to $b(\{f_{i_j}, f_{i_k}\})$ (in case $f_{i_j} = f_{i_k}$ there is no choice).*
   - *(b) Set c_i for all unpartitioned faces $f_i \in [f]$ according to the condition tested in Step 2(a)i.*
3. *Every element h in H is subdivided with the face-partition $\underline{c}$ (in the preferred fashion).*

The idea behind this algorithm is first to partition all faces in the preferred way. Next we check if it is possible to satisfy all preferences. For complexity arguments we perform this check only in conjunction with an additional requirement: compatibility with given (fixed) orientation tetrahedra. Thus we need a good heuristic for Step 1 in Algorithm 8, see Remark 9 below. What we get is an independence of the partitions of faces not belonging to the same equivalence class. If we notice (in Step 2a) that our preferred face-partitions are incompatible, we have to give up preferences.

The check in Step 2a is very easy: if $c_{i_j} =$ *no_preference*, or $c_{i_k} =$ *no_preference*, or the parallel continuation of c_{i_j} to the face f_{i_k} yields the partition c_{i_k}, then the check is passed (parallel continuation will only produce parallel partitioned face-pairs, unless the subchain coincides with $[f]$ which is completely unpartitioned in which case we can proceed as in Algorithm 5). Otherwise both c_{i_j} and c_{i_k} are predefined ($\neq$ *no_preference*) and parallel continuation from f_{i_j} to f_{i_k} is impossible. In this case we check all pairs $\{f_{i_l}, f_{i_{l+1}}\}$ $(l = j, \ldots, k-1)$ if their partition obtained by parallel continuation from f_{i_j} and f_{i_k}, respectively, induces the chosen orientation tetrahedron. If there is such a pair then the face-partitions are compatible, otherwise not.

In Step 2(a)ii we must change the value of either c_{i_j} or c_{i_k}. This choice is a heuristic decision. If we changed the face-partition to the other diagonal, we could run into new problems with other faces. So we mark them better as *no_preference*. Of course, the order of corrections influences the result. But we try to minimize this influence by only correcting subchains of faces.

As soon as all preferences are made compatible, we continue with Step 2b. Note that this setting is in general not uniquely defined. Any setting is acceptable. Possibly one can add here another optimization step.

Remark 9 *Another interesting question is how to choose orientation tetrahedra in Step 1. Let us discuss the case that the edge-graph of our mesh is bipartite. This means that the set $\mathcal{V}$ of vertices can be divided into two disjoint subsets, say blue and red vertices, such that each edge $e \in \mathcal{E}$ connects vertices of different color. In this case, we can choose for every hexahedron this orientation tetrahedron whose vertices are red. This ensures that every subchain $(f_{i_j}, f_{i_{j+1}}, f_{i_{j+2}})$ (and also longer subchains) passes our check: Either we can choose $c_{i_{j+1}}$ such that both $\{f_{i_j}, f_{i_{j+1}}\}$ and $\{f_{i_{j+1}}, f_{i_{j+2}}\}$ form parallel pairs. Or, when this is not possible, one diagonal of the two diagonals indicated by c_{i_j} and $c_{i_{j+2}}$ has red vertices. In the latter case the inversive partition of the corresponding pair is admissible.*

Note that in most cases the edge-graph is bipartite and we can color the vertices. As we will prove in the Appendix this is for example true when the closure of our domain is simply connected, e.g. homeomorphic to the unit ball. If the edge-graph is not bipartite we could refine the mesh (split every hexahedron into 8 hexahedra, using new vertices in the center of each edge, face and hexahedron) and the resulting mesh then has a bipartite edge-graph.

This coloring provides also an algorithm that produces a mesh with tetrahedra of types A and D only: choose the face-partitions by connecting only red vertices. Then each hexahedron has a face-partition of type 6. This type of subdivision was previously investigated in [5].

Finally we remark that there should be no problem if our elements are slightly distorted and do not have plane faces since the algorithm considers only the topology of the mesh and the preference functions. Problems arise if the orientation of the tetrahedra changes (i.e. the signed volume becomes zero or negative). This must not happen for tetrahedra of type A, since then the isoparametric mapping of the hexahedron is not invertible [7]. For tetrahedra of types B or D we also forbid such an orientation change since this case seems not to be of interest in practice (and would require much more care).

If it happens for type C tetrahedra and if there is at most one degenerated tetrahedron $t^* = t^*(h)$ of type C for every hexahedron $h \in \mathcal{H}$ we can still use a modification of the algorithm: Consider such a hexahedron h with one degenerated tetrahedron t^*. The question is how to avoid that t^* is used in the subdivision. A closer look at the possibilities mentioned in Table 4 shows that the only problematic case is a particular face partition P of h that is of type 3. It can be subdivided in only one specific manner and this includes the degenerated tetrahedron t^*. On the other hand, the partition P contains exactly one inversive partitioned face-pair, and this one can be excluded from use by choosing the appropriate orientation tetrahedron of h. So this problematic face partition is avoided and Algorithm 8 (that respects the chosen orientation tetrahedra) yields an admissible tetrahedral mesh.

Appendix

In this appendix we prove the conjecture stated in Remark 9.

Lemma 10 *Consider a hexahedral mesh of a domain* Ω *and assume that the closure of* Ω *is simply connected. Then the corresponding edge-graph is bipartite.*

Proof On the one hand, the closure of our domain is simply connected, i.e. every loop is homotopic to the trivial loop.

On the other hand the closure of our domain is the union of hexahedral elements and by the result of [6] (mentioned in the introduction) it is also the union of tetrahedral elements (defining a simplicial complex K) where edges are only edges of the hexahedral mesh, diagonals of the quadrilateral faces or space diagonals of the hexahedra. By using Theorem 3.4.15 of [4, page 160] we can conclude that also the simplicial complex K is simply connected in the sense of this book. This means that every edge-loop $v_0 v_1 \dots v_{k-1} v_0$ can be transformed to the trivial loop v_0 in finitely many steps of reduction and expansion (a series $v_i v_{i+1} v_{i+2}$ may be substituted by $v_i v_{i+2}$ if these three vertices all belong to one triangle of K, and vice versa). Now we set the *length* $l([vw])$ of an edge $[vw]$ of K to the number of corresponding hexahedral edges: $l([vw]) = 1$ if $[vw]$ is an hexahedral edge, $l([vw]) = 2$ for face-diagonals and $l([vw]) = 3$ for spatial diagonals. We realize that the parity of the total sum of lengths for an edge-loop is an invariant for reduction and expansion steps (we only need to consider the three types of faces of $\mathcal{F}'$ as introduced in Section 2 since they correspond to all possibilities for triangles of K). For the trivial loop this total sum is zero, so every edge-loop of hexahedral edges must have an even number of edges. This is a sufficient condition for the edge-graph to be bipartite. $\square$

References

[1] Th. Apel and U. Reichel. *SPC-PM Po 3D* v3.3 — User's Manual. Preprint SFB393/99-6, TU Chemnitz, 1999.

[2] S. Beuchler and A. Meyer. *SPC-PM 3AdH* v1.0 — Programmer's Manual. Preprint SFB393/01-08, TU Chemnitz, 2001.

[3] P. G. Ciarlet. *The finite element method for elliptic problems*. SIAM, Philadelphia, 2002.

[4] R. Engelking and K. Sieklucki. *Topology, a Geometric Approach*, volume 4 of *Sigma series in pure mathematics*. Heldermann, Berlin, 1992.

[5] D. Hacon and C. Tomei. Tetrahedral decompositions of hexahedral meshes. *European J. Combin.*, 10:435–443, 1989.

[6] C. P. Rourke and B. J. Sanderson. *Introduction to piecewise-linear topology*. Springer, Berlin, 1982.

[7] O. V. Ushakova. Conditions of nondegeneracy of three-dimensional cells. A formula of a volume of cells. *SIAM J. Sci. Comput.*, 23:1274–1290, 2001.

Other titles in the SFB393 series:

02-01 M. Pester. Bibliotheken zur Entwicklung paralleler Algorithmen - Basisroutinen für Kommunikation und Grafik. Januar 2002.

02-02 M. Pester. Visualization Tools for 2D and 3D Finite Element Programs - User's Manual. January 2002.

02-03 H. Harbrecht, M. Konik, R. Schneider. Fully Discrete Wavelet Galerkin Schemes. January 2002.

02-04 G. Kunert. A posteriori error estimation for convection dominated problems on anisotropic meshes. March 2002.

02-05 H. Harbrecht, R. Schneider. Wavelet Galerkin Schemes for 3D-BEM. February 2002.

02-06 W. Dahmen, H. Harbrecht, R. Schneider. Compression Techniques for Boundary Integral Equations - Optimal Complexity Estimates. April 2002.

02-07 S. Grosman. Robust local problem error estimation for a singularly perturbed reaction-diffusion problem on anisotropic finite element meshes. May 2002.

02-08 M. Springmann, M. Kuna. Identifikation schädigungsmechanischer Materialparameter mit Hilfe nichtlinearer Optimierungsverfahren am Beispiel des Rousselier Modells. Mai 2002.

02-09 S. Beuchler, R. Schneider, C. Schwab. Multiresolution weighted norm equivalences and applications. July 2002.

02-10 Ph. Cain, R. A. Römer, M. E. Raikh. Renormalization group approach to energy level statistics at the integer quantum Hall transition. July 2002.

02-11 A. Eilmes, R. A. Römer, M. Schreiber. Localization properties of two interacting particles in a quasiperiodic potential with a metal-insulator transition. July 2002.

02-12 M. L. Ndawana, R. A. Römer, M. Schreiber. Scaling of the Level Compressibility at the Anderson Metal-Insulator Transition. September 2002.

02-13 Ph. Cain, R. A. Römer, M. E. Raikh. Real-space renormalization group approach to the quantum Hall transition. September 2002.

02-14 A. Jellal, E. H. Saidi, H. B. Geyer, R. A. Römer. A Matrix Model for $\nu_{k_1 k_2} = \frac{k_1+k_2}{k_1 k_2}$ Fractional Quantum Hall States. September 2002.

02-15 M. Randrianarivony, G. Brunnett. Parallel implementation of curve reconstruction from noisy samples. August 2002.

02-16 M. Randrianarivony, G. Brunnett. Parallel implementation of surface reconstruction from noisy samples. September 2002.

02-17 M. Morgenstern, J. Klijn, Chr. Meyer, R. A. Römer, R. Wiesendanger. Comparing measured and calculated local density of states in a disordered two-dimensional electron system. September 2002.

02-18 J. Hippold, G. Rünger. Task Pool Teams for Implementing Irregular Algorithms on Clusters of SMPs. October 2002.

02-19 H. Harbrecht, R. Schneider. Wavelets for the fast solution of boundary integral equations. October 2002.

02-20 H. Harbrecht, R. Schneider. Adaptive Wavelet Galerkin BEM. October 2002.

02-21 H. Harbrecht, R. Schneider. Wavelet Galerkin Schemes for Boundary Integral Equations - Implementation and Quadrature. October 2002.

03-01 E. Creusé, G. Kunert, S. Nicaise. A posteriory error estimation for the Stokes problem: Anisotropic and isotropic discretizations. January 2003.

03-02 S. I. Solov'ëv. Existence of the guided modes of an optical fiber. January 2003.

03-03 S. Beuchler. Wavelet preconditioners for the p-version of the FEM. February 2003.

03-04 S. Beuchler. Fast solvers for degenerated problems. February 2003.

03-05 A. Meyer. Stable calculation of the Jacobians for curved triangles. February 2003.

03-06 S. I. Solov'ëv. Eigenvibrations of a plate with elastically attached load. February 2003.

03-07 H. Harbrecht, R. Schneider. Wavelet based fast solution of boundary integral equations. February 2003.

03-08 S. I. Solov'ëv. Preconditioned iterative methods for monotone nonlinear eigenvalue problems. March 2003.

The complete list of current and former preprints is available via
`http://www.tu-chemnitz.de/sfb393/preprints.html`.