SANDIA REPORT

SAND2015-3830
Unlimited Release
Printed May, 2015

Cutting Tetrahedra by Node Identifiers

Richard M. J. Kramer

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone:	(865) 576-8401
Facsimile:	(865) 576-5728
E-Mail:	reports@adonis.osti.gov
Online ordering:	http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone:	(800) 553-6847
Facsimile:	(703) 605-6900
E-Mail:	orders@ntis.fedworld.gov
Online ordering:	http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2015-3830
Unlimited Release
Printed May, 2015

Cutting Tetrahedra by Node Identifiers

Richard M. J. Kramer
Center for Computing Research
Sandia National Laboratories
P.O. Box 5800, MS 1323
Albuquerque, NM 87185-1323
rmkrame@sandia.gov

Abstract

This report briefly outlines an algorithm for dividing a tetrahedron intersected by a planar interface into conforming sub-tetrahedra. The problem of conformal decomposition of tetrahedral meshes arises in enriched finite element methods; in particular, we are concerned with the Conformal Decomposition Finite Element Method (CDFEM) and variants of the eXtended Finite Element Method (XFEM). The algorithm presented is based on the paper *How to Subdivide Pyramids, Prisms and Hexahedra into Tetrahedra* by Dompierre, Labbé, Vallet, and Camarero (1999), and here is applied and extended to the problem of fully defining and tracking all geometric features of the sub-tetrahedra generated when a tetrahedron is cut by a planar surface.

List of Figures

List of Tables

Conformal Tetrahedron Cutting

1 Introduction

The Conformal Decomposition Finite Element Method (CDFEM) [5, 4] and the eXtended Finite Element Method with Algebraic Constraints (XFEM-AC) [2, 3] are related approaches for resolution of sub-cell interfaces. In either case, a tetrahedral base (or 'parent') mesh supports an interface description, which does not conform to the mesh surfaces. Therefore some tetrahedra are cut by the interface, and those cut elements must be dynamically divided into tetrahedral subelements that conform to the interface.

In general, it is not difficult to enumerate the basic ways in which a tetrahedron can be cut by a plane. Depending on the case, a prism or pyramid can result from the cut that will need to be further subdivided to yield a conforming set of sub-tetrahedra. Effectively, this is a process of choosing secondary cuts to divide quadrilateral sub-faces into triangular sub-faces. Care is required when dividing the three quadrilateral faces of a prism, because of the eight possible cuts, only six result in three valid tetrahedra. The other two cases result in Schönhart polyhedra, which can be tetrahedralized only by the addition of a Steiner vertex.

For the finite element methods of interest, mesh quality is a concern when elements are allowed to be arbitrarily cut. Implicit time-stepping methods can suffer if very small or poorly shaped elements are present in the mesh, as these can severely impact matrix conditioning. In terms of the element subdivision, an obvious approach to improving subelement quality (and analogous to that usually implemented in 2-D) is to choose the secondary cuts on quadrilateral faces such that the largest face angles are divided. This has two significant drawbacks, however: it is dependent on floating point operations, and for prisms, the three face cuts must be carefully coordinated in order to avoid the Schönhart cases.

While not guaranteeing optimal cuts (in the sense of mesh quality) on quadrilateral faces, the algorithm presented in *How to Subdivide Pyramids, Prisms and Hexahedra into Tetrahedra* by Dompierre, Labbé, Vallet, and Camarero (1999) [1] is appealing, in that it guarantees valid sub-tetrahedra by construction, and requires only unique vertex identifiers to cut the mesh. As long as these identifiers are consistent across processors, this algorithm then requires no parallel communication and will deterministically generate consistent tetrahedra.

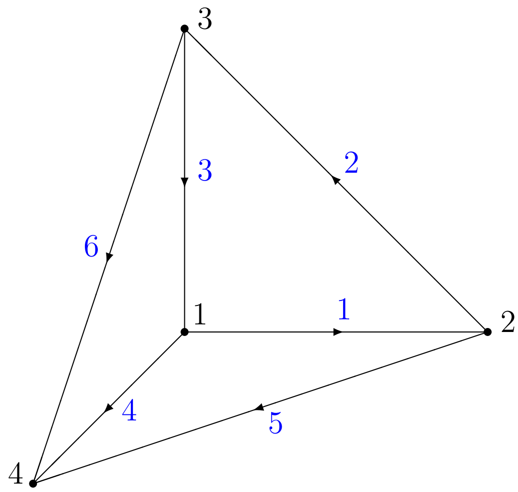
Since we are interested in compatible finite element discretizations in the context of the discrete DeRahm complex, the basic algorithm of [1] needs to be extended to track all of the geometric features of the cut tetrahedron. This report presents the application of the Dompierre, Labbé, Vallet, and Camarero algorithm to the particular problem of generating consistent sub-tetrahedra from a cut parent tetrahedron, while tracking not only nodes, but also the edges and faces of all subelements. This extension also ensures proper orientation of the sub-tetrahedra, and all sub-edges and sub-faces generated by the cut.

For multimaterial or multiple interface applications, it is proposed that multiple cuts of a base tetrahedron could be achieved by completing the tetrahedralization for each cut in turn, i.e., applying the cutting algorithm sequentially based on the logical ordering of the interfaces (which is presumably available). Although potentially generating very small tetrahedra, this approach would retain the generality of the base algorithm, and innumerable special cases of arbitrarily intersecting polyhedra would not need to be considered.

Finally, it is noted that numerical experimentation has shown that the cost of sub-optimality in the ID-based cutting algorithm in terms of matrix conditioning is limited, typically on the order of 10–20 percent. This is arguably a small price to pay for the computational advantages offered by the algorithm. While this observation is based on good quality parent meshes, this is not an unreasonable assumption in the vast majority of cases, particularly because by design, the parent mesh is not required to be body fitted. Also, other strategies are being researched that will far more significantly improve the solvability of the resulting matrix system.

2 Reference Geometries

Critical to the ID-based cutting algorithm is consistent definition of the reference geometry, which includes edge and face definitions. The reference tetrahedron is shown in Figure 1, including its signed edge and face connectivity. Note that the nodal coordinates play no role in the conformal decomposition procedure. For a mesh generated according to this reference, the local edge definitions will be generated in the reference order, and the node ordering of any given tetrahedron will map directly to the node ordering of the reference tetrahedron. Separate edge and face orientation states are then used to indicate the orientation of an arbitrary feature relative to the reference definition.

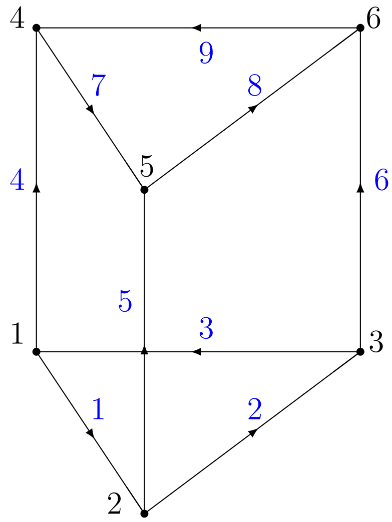


edge	1	2	3	4	5	6
N_1	1	2	3	1	2	3
N_2	2	3	1	4	4	4

face	N_1	N_2	N_3	E_1	E_2	E_3
1	1	2	4	1	5	−4
2	2	3	4	2	6	−5
3	1	4	3	4	−6	3
4	1	3	2	−3	−2	−1

Figure 1: The reference tetrahedron, with node identifiers indicated in black and edge identifiers in blue. Signed edge and face connectivity uses a negative sign to indicate an edge traversed in the opposite direction.

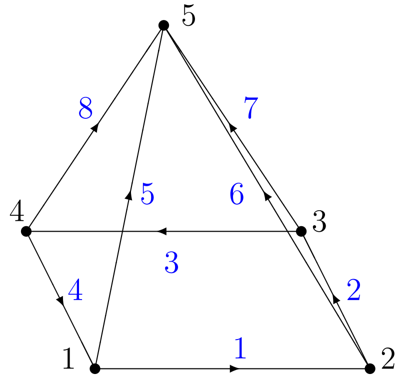
Cutting the tetrahedron can generate a prism or a pyramid, which will then need to be further subdivided to yield the conforming set of tetrahedra. To do this, a reference prism and pyramid must first be defined; these are shown in Figures 2 and 3. Note that these exist only as an intermediate step in the cutting algorithm, so the definition is required only to be consistent within the context of this algorithm.



edge	1	2	3	4	5	6	7	8	9
N_1	1	2	3	1	2	3	4	5	6
N_2	2	3	1	4	5	6	5	6	4

face	N_1	N_2	N_3	N_4	E_1	E_2	E_3	E_4
1	1	3	2		−3	−2	−1	
2	4	5	6		7	8	9	
3	1	2	5	4	1	5	−7	−4
4	1	4	6	3	4	−9	−6	3
5	2	3	6	5	2	6	−8	−5

Figure 2: The reference prism, with node identifiers indicated in black and edge identifiers in blue. Signed edge and face connectivity uses a negative sign to indicate an edge traversed in the opposite direction.



edge	1	2	3	4	5	6	7	8
N_1	1	2	3	4	1	2	3	4
N_2	2	3	4	1	5	5	5	5

face	N_1	N_2	N_3	N_4	E_1	E_2	E_3	E_4
1	1	2	5		1	6	−5	
2	2	3	5		2	7	−6	
3	3	4	5		3	8	−7	
4	4	1	5		4	5	−8	
5	1	4	3	2	−4	−3	−2	−1

Figure 3: The reference pyramid, with node identifiers indicated in black and edge identifiers in blue. Signed edge and face connectivity uses a negative sign to indicate an edge traversed in the opposite direction.

3 The Primary Cut

The starting point for the tetrahedralization of a cut tetrahedron is the initial intersection of the parent element with the interface plane, which generates two polyhedra. It is assumed that the reader has an interface description that will define the intersections of the interface plane with the edges of the tetrahedron, and thus a means to define an initial node list for each polyhedron. For example, a simple cutting procedure is to loop through the edges of the parent tetrahedron and bin nodes to each polyhedron as they are encountered (including intersection nodes). It is highly unlikely, though, that any convenient cutting procedure will result in polyhedra that conform to the reference definitions given in the previous section. Therefore, these "raw" polyhedra must be processed to conform to the reference definitions, but first the fundamental cut types for a tetrahedron need to be identified.

One of four fundamental cut types result when a tetrahedron is intersected by a plane, and depending on the orientation of the base tetrahedron relative to the plane, a total of 25 geometrically distinct cases can be identified. These will be labeled here as:

1. Type 1, the four node/face cuts where a node is separated from its opposite face by a triangular intersection plane to generate a tetrahedron and a prism;

2. Type 2, the three edge/edge cuts where opposite edges are separated by a quadrilateral intersection plane to generate two prisms;

3. Type 3, the twelve node-degenerate cuts where a triangular intersection plane intersects a node of the parent tetrahedron and one face remains uncut to generate a tetrahedron and a pyramid;

4. Type 4, the six edge-degenerate cuts where a triangular intersection plane intersects an edge of the parent tetrahedron to generate two tetrahedra.

Each case may be uniquely identified by the number of nodes lying either on or to one side of the cut plane, and further categorized by cut (or uncut) edge and/or face.

Table 1 shows the reference sub-polyhedra definitions generated by the first step of the cutting algorithm, for each type and case, which are illustrated in Figures 4–7. The front and back polyhedra are identified by the orientation of their shared face, i.e., the face generated by the cut, which is always defined to be consistent with the orientation of the cutting plane. This trivially handles the issue of orientation of the interface relative to the parent tetrahedron and effectively reduces this to a labeling choice.

Because the mapping is dependent on the cutting procedure, it is left to the reader to determine the mapping from their raw polyhedra to the reference definitions in Table 1. The purpose of the reference definitions is to ensure that the sub-elements are well defined (i.e., have positive volume), and for prism or pyramid sub-elements, that they will be consistently cut into valid sub-tetrahedra in the secondary cutting step. Once the polyhedra have been cast into the reference forms based on the cut type, the algorithms for subdividing prisms and pyramids may be applied indiscriminately.

Finally, a convenient feature numbering scheme, which is used in this report, numbers each cut vertex after the edge on which it lies, each cut edge after the face on which it lies, and each cut face after the element in which it lies. Based on a list of each enriched geometric feature, this makes unique identifiers for each cut feature trivial to define. It is noted, however, that any feature numbering scheme devised by the reader may be used, as long as a unique and consistent global value is generated for each feature.

Type		Nodes	Edges	Faces
1/N1	F	1, 5, 6, 7	1, -12, 3, 4, 10, -11	1, -C, 3, 4
	B	7, 5, 6, 4, 2, 3	-10, -12, -11, 9, 7, -8, -5, 2, 6	C, 2, 1, 3, 4
1/N2	F	5, 2, 6, 7	1, 2, -12, -10, 5, 11	1, 2, -C, 4
	B	4, 1, 3, 7, 5, 6	-4, -3, 6, -9, 7, -8, 10, 12, 11	3, C, 1, 2, 4
1/N3	F	6, 5, 3, 7	-12, 2, 3, 11, -10, 6	-C, 2, 3, 4
	B	4, 2, 1, 7, 5, 6	-5, -1, 4, -9, 7, -8, 10, 12, 11	1, C, 2, 3, 4
1/N4	F	5, 6, 7, 4	10, 11, 12, 4, 5, 6	1, 2, 3, -C
	B	6, 5, 7, 2, 1, 3	-10, -12, -11, -8, -7, -9, -1, -3, -2	C, 4, 1, 2, 3
2/E6-1	F	3, 6, 5, 4, 7, 8	3, -14, 2, 6, 3, -12, -4, 11, 5	4, 1, 3, 2, -C
	B	1, 6, 7, 2, 5, 8	-8, 13, -9, 1, -14, 11, 7, -12, -10	3, 2, 4, 1, C
2/E2-4	F	2, 5, 7, 3, 6, 8	-1, -11, -5, 2, 14, -12, 3, 13, -6	1, 3, 4, 2, -C
	B	1, 5, 6, 4, 7, 8	7, 14, 8, 4, -11, 13, -9, -12, 10	4, 2, 1, 3, C
2/E5-3	F	2, 6, 5, 4, 8, 7	2, -14, 1, 5, 12, -11, -6, 13, 4	4, 3, 2, 1, -C
	B	1, 7, 5, 3, 8, 6	9, 11, -7, -3, -13, 14, 10, -12, 8	1, 2, 3, 4, C
3/N1-F1	F	1, 5, 3, 6	-11, 2, 3, 10, -9, 6	-C, 2, 3, 4
	B	2, 4, 6, 5, 1	5, -8, 9, -7, -1, -4, -10, 11	1, 3, C, 4, 2
3/N1-F3	F	1, 2, 5, 6	1, 2, -11, -9, 5, 10	1, 2, -C, 4
	B	6, 4, 3, 5, 1	8, -6, -7, 10, 9, -4, 3, -11	1, 3, 4, C, 2
3/N1-F4	F	1, 5, 6, 4	9, 10, 11, 4, 5, 6	1, 2, 3, -C
	B	2, 5, 6, 3, 1	7, 10, -8, -2, -1, -9, 11, 3	1, C, 3, 4, 2
3/N2-F1	F	5, 2, 3, 6	-11, 2, 3, 10, -9, 6	-C, 2, 3, 4
	B	1, 5, 6, 4, 2	-7, 10, 8, -4, 1, -11, 9, -5	4, C, 2, 1, 3
3/N2-F2	F	1, 2, 5, 6	1, -11, 3, 4, 9, -10	1, -C, 3, 4
	B	6, 5, 3, 4, 2	10, -7, 6, -8, -9, 11, -2, -5	C, 4, 2, 1, 3
3/N2-F4	F	5, 2, 6, 4	9, 10, 11, 4, 5, 6	1, 2, 3, -C
	B	1, 3, 6, 5, 2	-3, 8, 11, -7, 1, -2, -10, 9	4, 2, C, 1, 3
3/N3-F2	F	1, 5, 3, 6	1, -11, 3, 4, 9, -10	1, -C, 3, 4
	B	2, 5, 6, 4, 3	-7, 9, 8, -5, 2, -11, 10, -6	4, C, 3, 2, 1
3/N3-F3	F	5, 2, 3, 6	1, 2, -11, -9, 5, 10	1, 2, -C, 4
	B	1, 4, 6, 5, 3	4, -8, 9, -7, -3, -6, -10, -11	3, 2, C, 4, 1
3/N3-F4	F	5, 6, 3, 4	9, 10, 11, 4, 5, 6	1, 2, 3, -C
	B	1, 5, 6, 2, 3	7, 9, -8, -1, -3, -11, 10, 2	3, C, 2, 4, 1
3/N4-F1	F	6, 5, 3, 4	-11, 2, 3, 10, -9, 6	-C, 2, 3, 4
	B	1, 2, 5, 6, 4	1, 7, 11, 8, 4, 5, -9, 10	1, 2, C, 3, 4

Type		Nodes	Edges	Faces
3/N4-F2	F	1, 5, 6, 4	1, -11, 3, 4, 9, -10	1, -C, 3, 4
	B	5, 2, 3, 6, 4	7, 2, 8, 11, 9, 5, 6, -10	1, 2, 3, C, 4
3/N4-F3	F	5, 2, 6, 4	1, 2, -11, -9, 5, 10	1, 2, -C, 4
	B	1, 5, 6, 3, 4	7, 11, 8, 3, 4, -9, 10, 6	1, C, 2, 3, 4
4/E1	F	1, 2, 5, 4	1, 8, 9, 4, 5, 6	1, 2, 3, -C
	B	1, 2, 3, 5	1, 2, 3, -9, 8, 7	C, 2, 3, 4
4/E2	F	5, 2, 3, 4	8, 2, 9, 4, 5, 6	1, 2, 3, -C
	B	1, 2, 3, 5	1, 2, 3, 7, -8, 9	1, C, 3, 4
4/E3	F	1, 5, 3, 4	8, 9, 3, 4, 5, 6	1, 2, 3, -C
	B	1, 2, 3, 5	1, 7, 9, 4, 5, -8	1, 2, C, 4
4/E4	F	1, 5, 3, 4	-9, 2, 3, 4, -8, 6	-C, 2, 3, 4
	B	1, 2, 5, 4	1, 2, 3, -9, 8, 7	C, 2, 3, 4
4/E5	F	1, 2, 5, 4	1, -9, 3, 4, 5, -8	1, -C, 3, 4
	B	5, 2, 3, 4	9, 2, 7, -8, 5, 6	C, 2, 3, 4
4/E6	F	5, 2, 3, 4	1, 2, -9, -8, 5, 6	1, 2, -C, 4
	B	1, 5, 3, 4	7, 9, 3, 4, -8, 6	1, C, 3, 4

Table 1: Table of sorted nodes, edges and faces of the polyhedra that result from cutting a tetrahedron, for each cut type. The distinction between front and back polyhedra is based on the orientation of the shared face introduced by the cutting plane (identified as ‘C’). Nodes introduced by the cut are numbered sequentially after the parent edge on which they lie. For edges, the part of a cut edge behind the cutting plane receives a new sequential ID based on the parent edge ID, and new edges introduced by the cut on faces of the parent tetrahedron are numbered sequentially after the parent face (see Figures 4–7).

4 Secondary Cuts

The tetrahedralization of a cut element is completed by dividing the prisms and pyramids generated by cut types 1–3 into tetrahedra. Since the reference definitions listed in Table 1 guarantee polyhedra that follow the definitions in Section 2, the cutting procedure simply follows [1] for each case, with the addition of edge and face tracking.

4.1 Cutting a prism

The prism has three quadrilateral faces that must each be divided into two triangular faces. By orienting the prism such that the smallest global node identifier is in the position of node 1 in the reference geometry of Fig. 2, the algorithm reduces the number of possible cuts to just two, depending on the direction in which face 5 is cut. Table 2 from [1] defines an "indirection" table that provides a mapping from an arbitrary prism to one oriented as in Figure 2, which is reproduced here in Table 2. This is extended in Table 3 to also track edges and faces under each transformation.

$\mathrm{V}_{\min}$	I_1	I_2	I_3	I_4	I_5	I_6
V_1	1	2	3	4	5	6
V_2	2	3	1	5	6	4
V_3	3	1	2	6	4	5
V_4	4	6	5	1	3	2
V_5	5	4	6	2	1	3
V_6	6	5	4	3	2	1

Table 2: Indirection table that reorders the vertices of a prism such that the first vertex has the smallest global identifier. I_1–I_6 are the indices of the reordered prism. Reproduced from Table 2 of [1].

$\mathrm{V}_{\min}$	E_1	E_2	E_3	E_4	E_5	E_6	E_7	E_8	E_9	F_1	F_2	F_3	F_4	F_5
V_1	1	2	3	4	5	6	7	8	9	1	2	3	4	5
V_2	2	3	1	5	6	4	8	9	7	1	2	5	3	4
V_3	3	1	2	6	4	5	9	7	8	1	2	4	5	3
V_4	−9	−8	−7	−4	−6	−5	−3	−2	−1	2	1	4	3	5
V_5	−7	−9	−8	−5	−4	−6	−1	−3	−2	2	1	3	5	4
V_6	−8	−7	−9	−6	−5	−4	−2	−1	−3	2	1	5	4	3

Table 3: Reordering table for edges and faces of a prism after rotation such that the vertex with the smallest global identifier is ordered first (complementing Table 2).

Once correctly oriented, faces 3 and 4 of the prism are cut with new edges originating at node 1, which ensures consistency with any adjoining polyhedron. Face 5 is cut such that the new edge is connected to the vertex with the minimum index of the two diagonals.

Since we track only triangular faces, the division of each quadrilateral face generates three additional faces, which are numbered sequentially. Two internal faces are also generated, here numbered 9 and 10, which are arbitrarily chosen to be positively oriented for the first tetrahedron that encounters each one. The three tetrahedra that result from either choice for the edge cutting face 5 are defined in Table 4.

	$(V_2, V_6) < (V_3, V_5)$						$(V_3, V_5) < (V_2, V_6)$					
T_1-N	1	2	3	6			1	2	3	5		
T_1-E	1	2	3	11	12	6	1	2	3	10	5	12
T_1-F	9	5	4	1			3	5	9	1		
T_2-N	1	2	6	5			1	5	3	6		
T_2-E	1	12	-11	10	5	-8	10	-12	3	11	8	6
T_2-F	3	8	10	-9			10	8	4	-9		
T_3-N	1	5	6	4			1	5	6	4		
T_3-E	10	8	-11	4	-7	9	10	8	-11	4	-7	9
T_3-F	6	2	7	-10			6	2	7	-10		

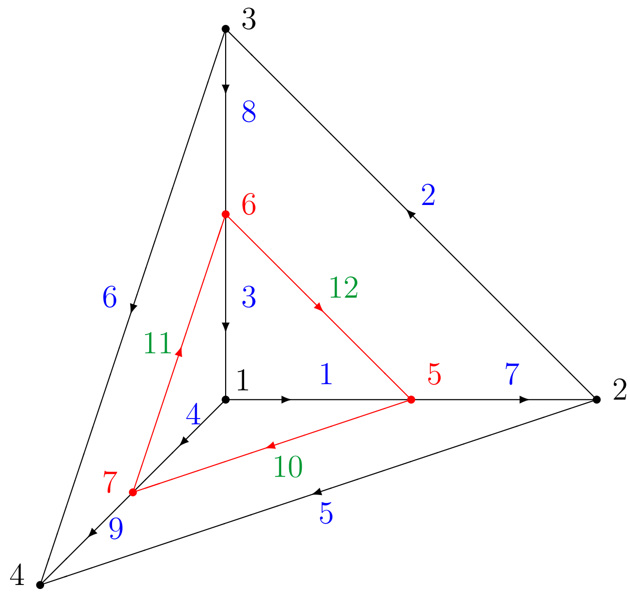
Table 4: Nodes, edges and faces of the three tetrahedra that result from cutting a prism, depending on which vertices edge 12 connects. The quadrilateral faces of the prism are divided such that face 3 is split into triangular faces 3 and 6 by edge 10, face 4 is split into triangular faces 4 and 7 by edge 11, and face 5 is split into triangular faces 5 and 8 by edge 12. Two additional internal faces, 9 and 10, are also defined.

4.2 Cutting a pyramid

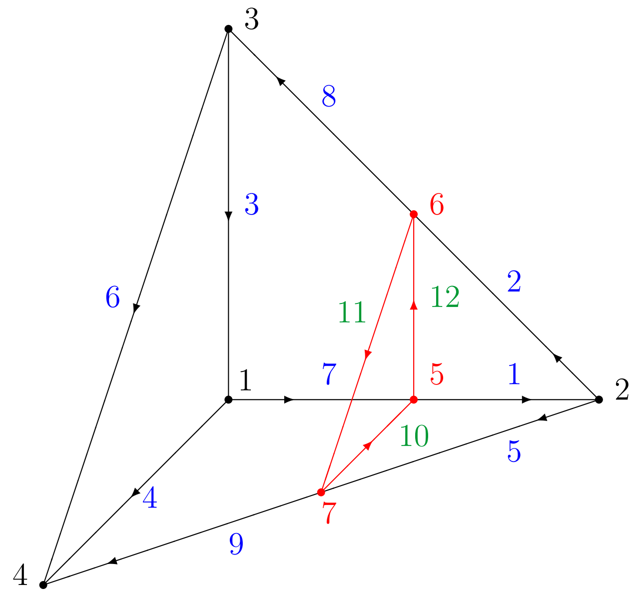
The pyramid has a single quadrilateral face that must be divided into two triangles, and the edge that divides the face is the only additional edge required to define the two tetrahedra (and is numbered 9 here). Following [1], the diagonal chosen is the edge with the minimum index of the two possibilities (and therefore no rotation of the pyramid is required). Since we track only triangular faces, the division of face 5 of the pyramid introduces another face, 6, and the internal face that separates the two tetrahedra is numbered 7. It is arbitrarily chosen to be positively oriented for the second tetrahedron of the pair. Table 5 lists the nodes, edges and faces of the two tetrahedra that result from cutting the reference pyramid for either choice of diagonal.

	$(V_1, V_3) < (V_2, V_4)$						$(V_2, V_4) < (V_1, V_3)$					
T_1-N	1	2	3	5			2	3	4	5		
T_1-E	1	2	-9	5	6	7	2	3	-9	6	7	8
T_1-F	1	2	-7	5			2	3	-7	5		
T_2-N	1	3	4	5			2	4	1	5		
T_2-E	9	3	4	5	7	8	9	4	1	6	8	5
T_2-F	7	3	4	6			7	4	1	6		

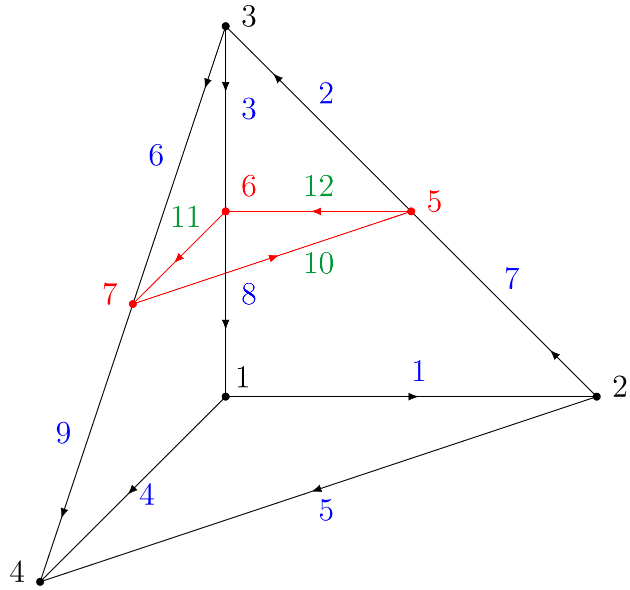
Table 5: Nodes, edges and faces of the two tetrahedra that result from cutting a pyramid, depending on which vertices edge 9 connects. Faces 5 and 6 are the two triangular faces that result from cutting the quadrilateral face of the pyramid, and face 7 is the new internal shared face.



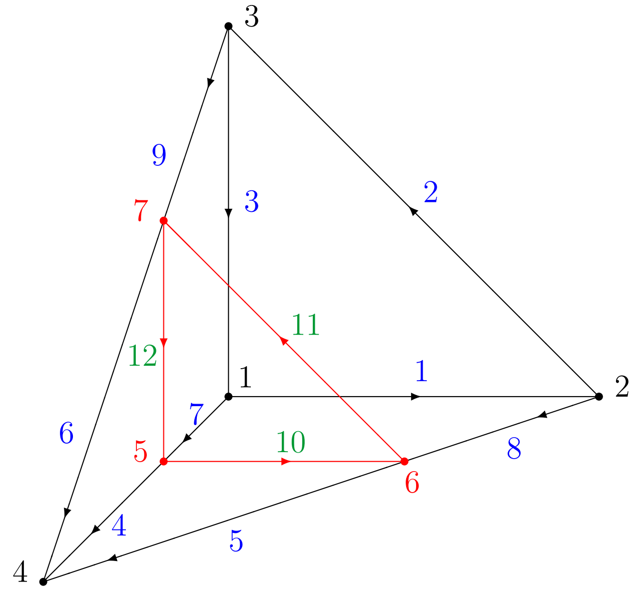
(a) N1



(b) N2

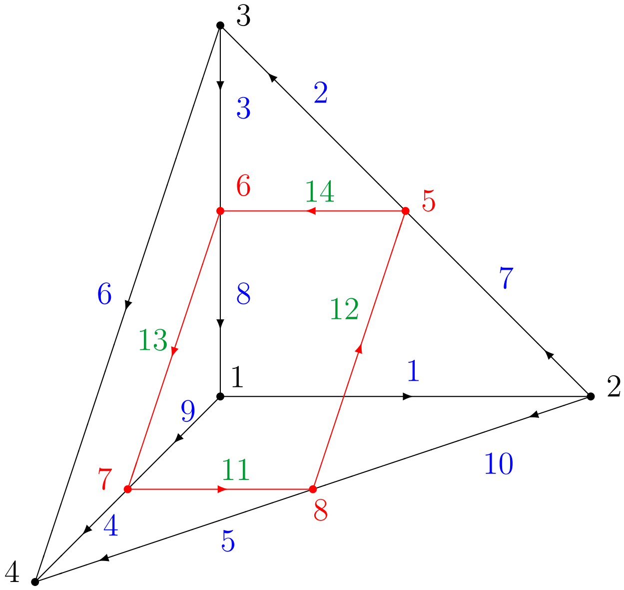


(c) N3

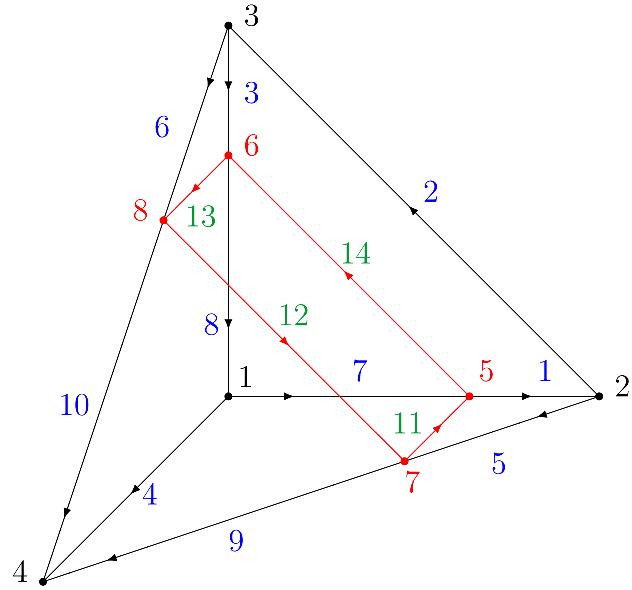


(d) N4

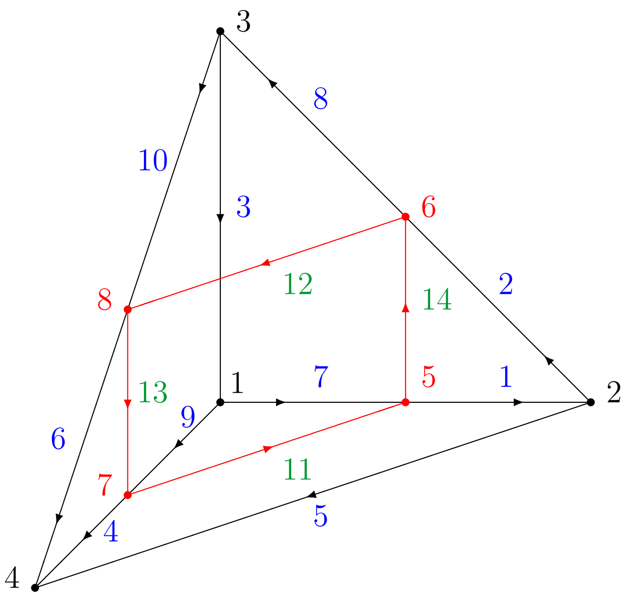
Figure 4: Type 1 cuts. Nodes and edges of the cut plane are shown in red, and identifiers of the new edges in green.



(a) E6/E1

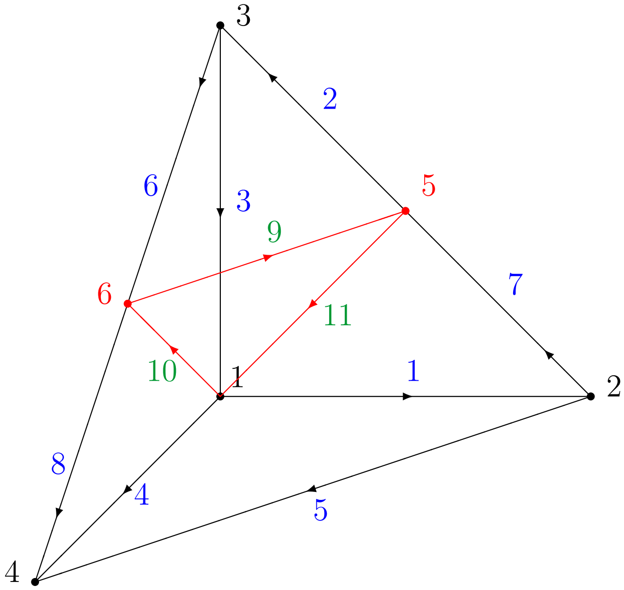


(b) E2/E4

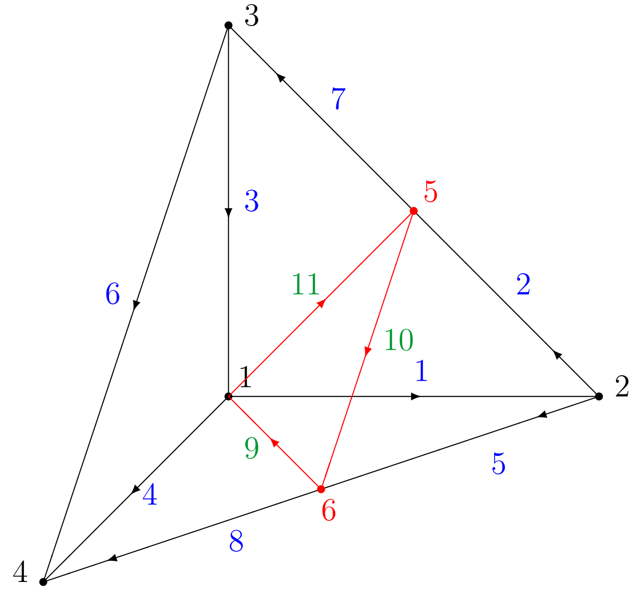


(c) E5/E3

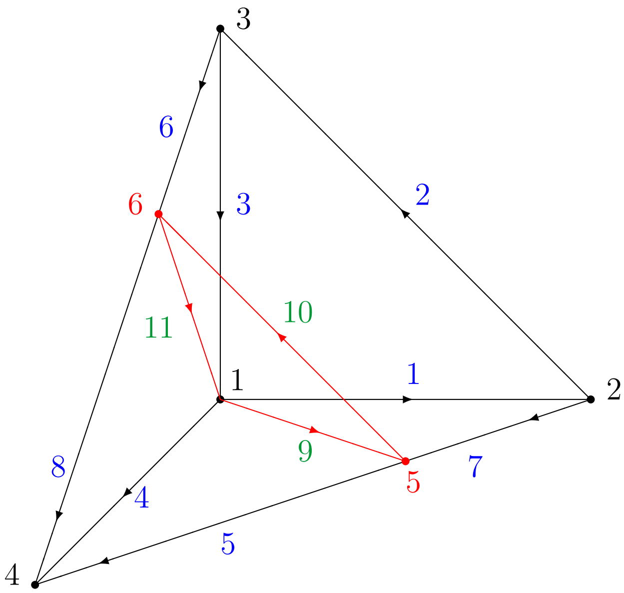
Figure 5: Type 2 cuts. Nodes and edges of the cut plane are shown in red, and identifiers of the new edges in green.



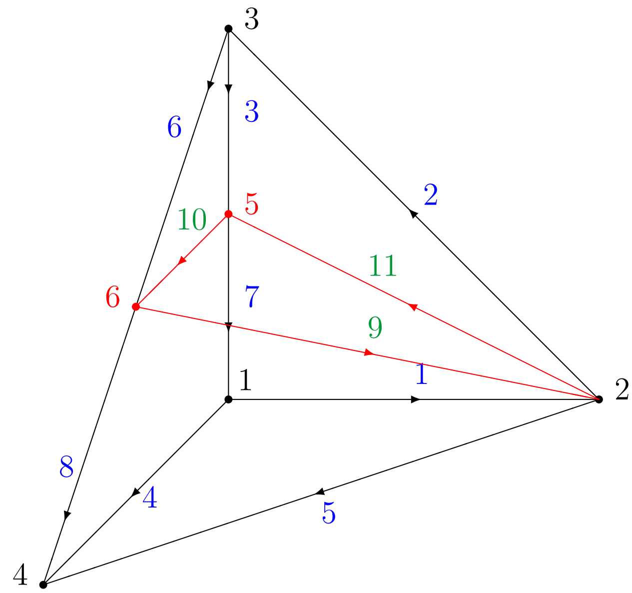
(a) N1-F1



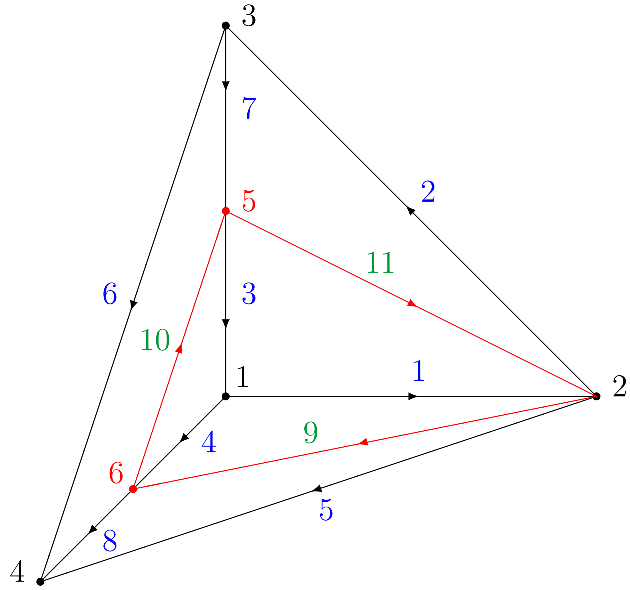
(b) N1-F3



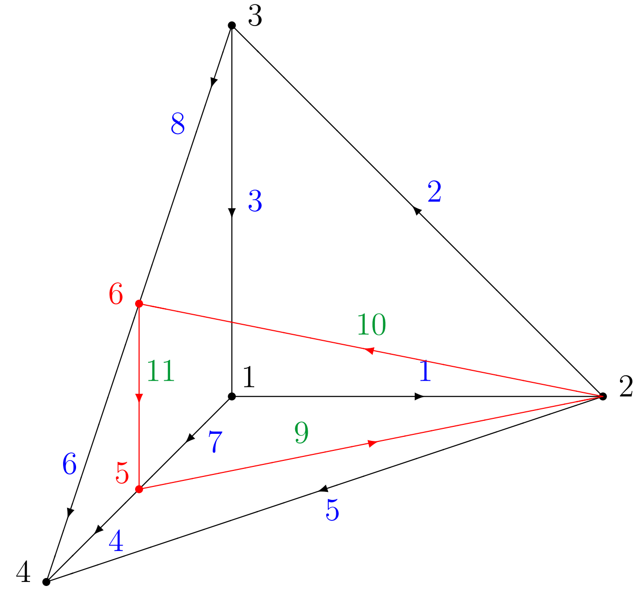
(c) N1-F4



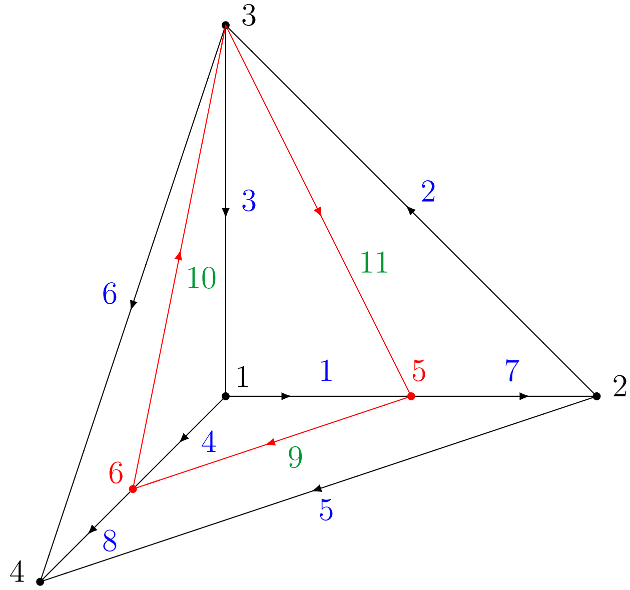
(d) N2-F1



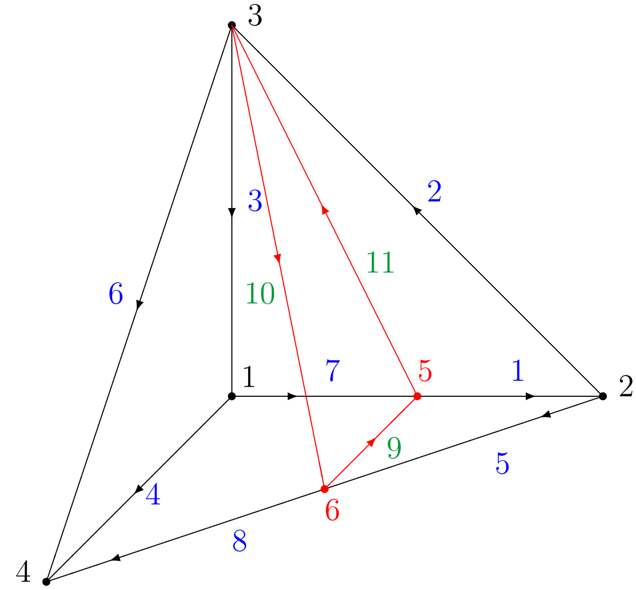
(e) N2-F2



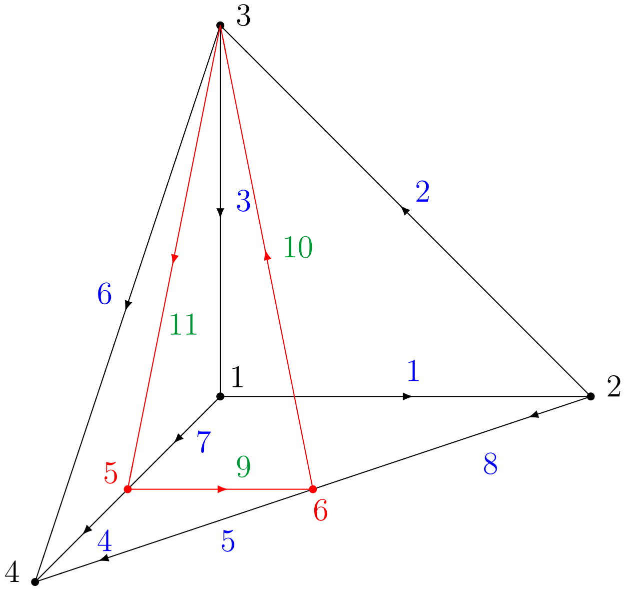
(f) N2-F4



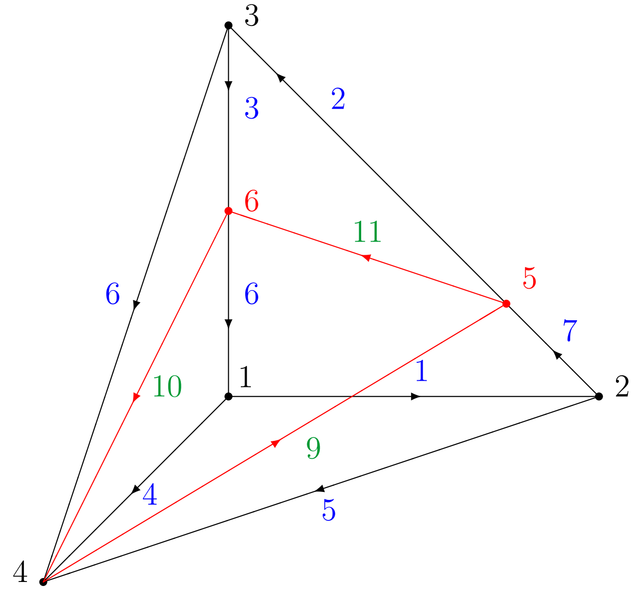
(g) N3-F2



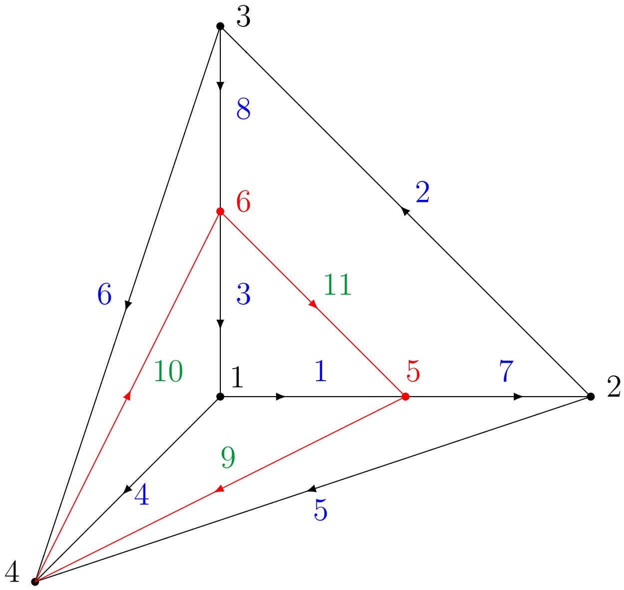
(h) N3-F3



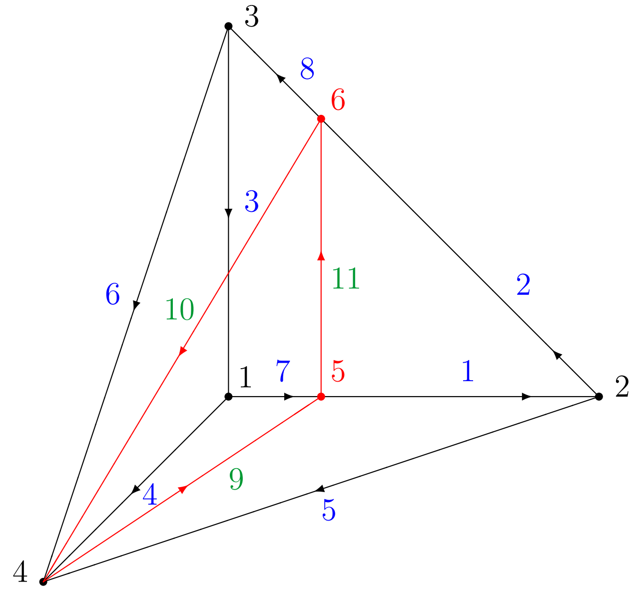
(i) N3-F4



(j) N4-F1

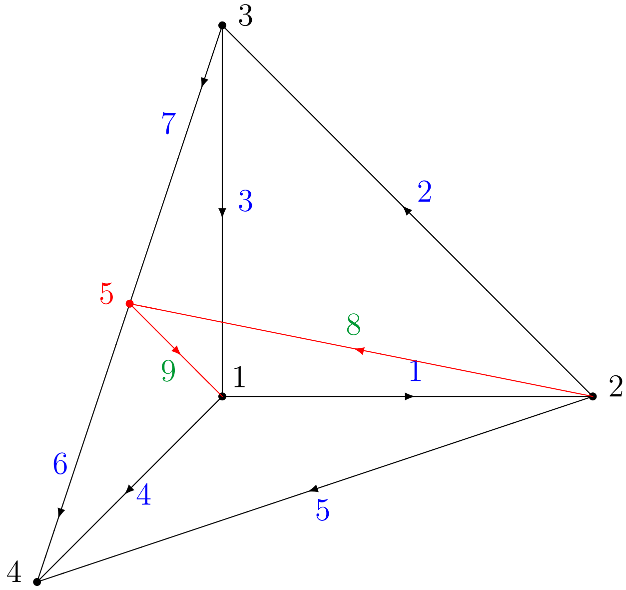


(k) N4-F2

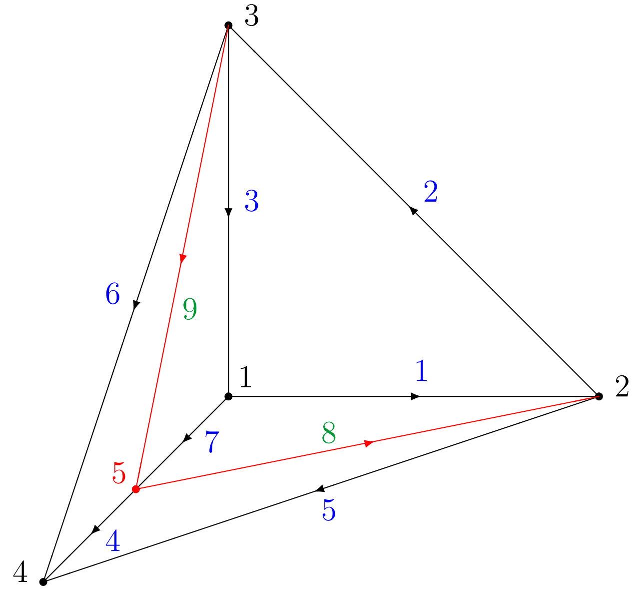


(l) N4-F3

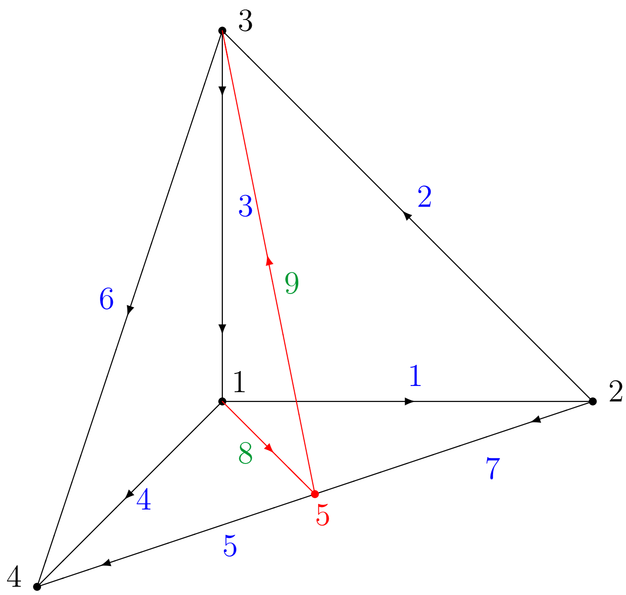
Figure 6: Type 3 cuts. Nodes and edges of the cut plane are shown in red, and identifiers of the new edges in green.



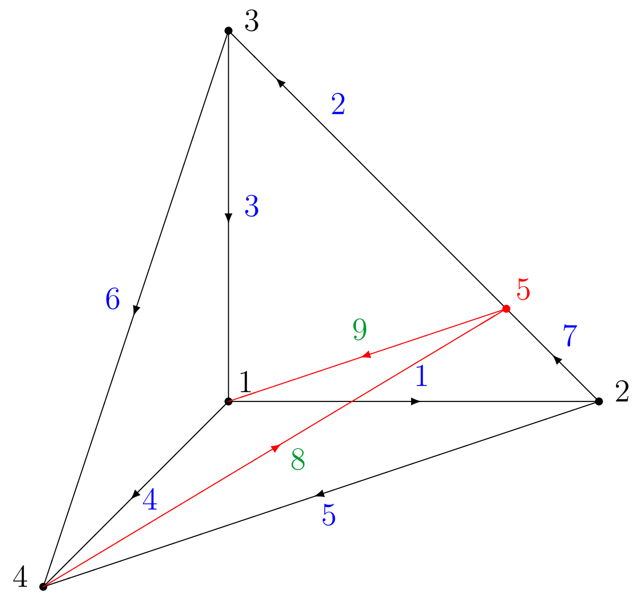
(a) E1



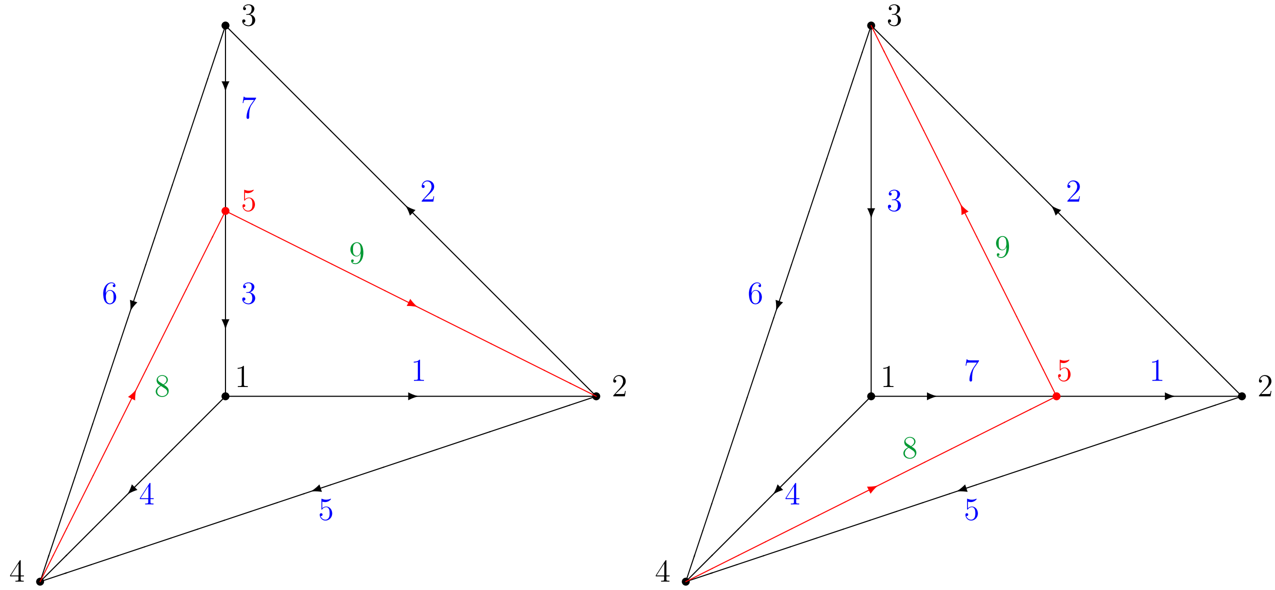
(b) E2



(c) E3

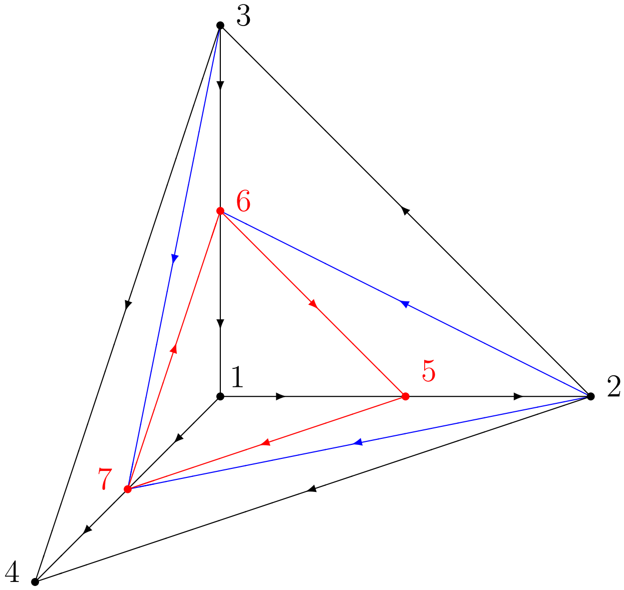


(d) E4

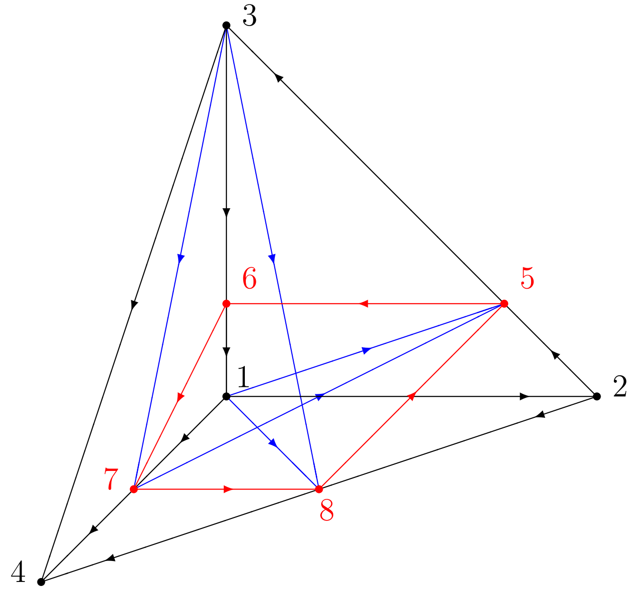


(e) E5 (f) E6

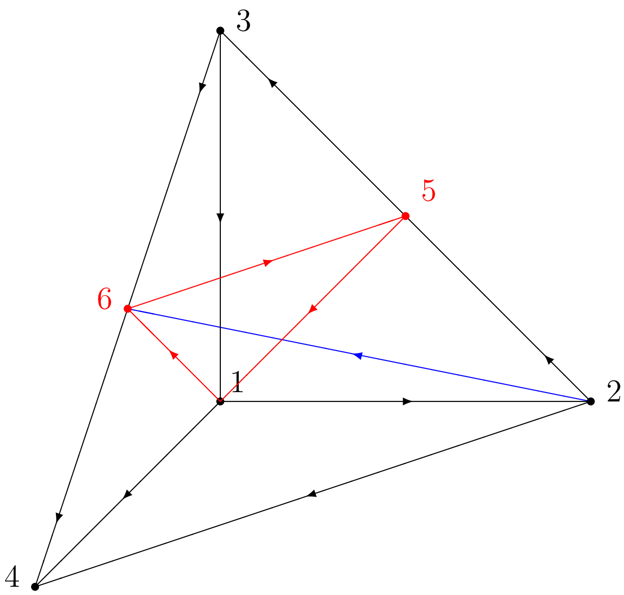
Figure 7: Type 4 cuts. Nodes and edges of the cut plane are shown in red, and identifiers of the new edges in green.



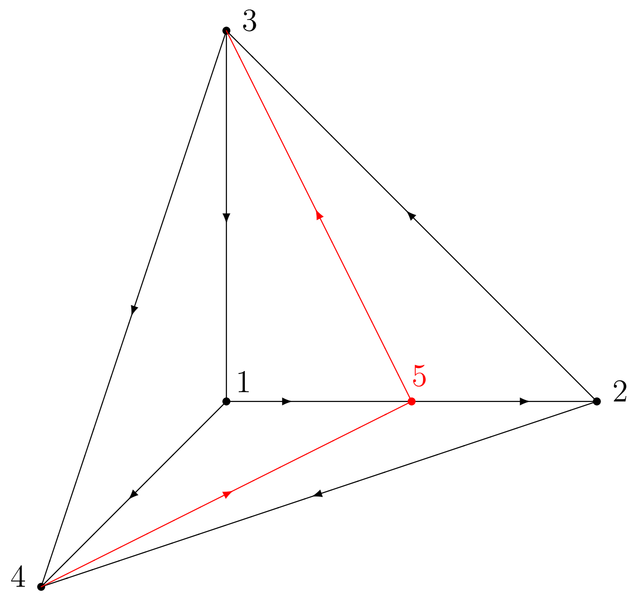
(a) Type 1



(b) Type 2



(c) Type 3



(d) Type 4

Figure 8: An example of each cut type, including secondary cuts as generated based on the shown vertex numbering. Nodes and edges of the primary cut plane are shown in red, secondary cuts are shown in blue. Edge identifiers have been excluded in the interests of clarity. Note that Type 4 generates two tetrahedra so no secondary cut is required.

References

[1] J. Dompierre, P. Labbé, M.-G. Vallet, and R. Camarero. How to subdivide pyramids, prisms, and hexahedra into tetrahedra. In Proceedings of the 8th International Meshing Roundtable, pages 195–204, October 1999.

[2] R. Kramer, P. Bochev, C. Siefert, and T. Voth. An extended finite element method with algebraic constraints (XFEM-AC) for problems with weak discontinuities. Comput. Methods Appl. Mech. Engrg., 266:70–80, 2013.

[3] R. Kramer, P. Bochev, C. Siefert, and T. Voth. Algebraically constrained extended edge element method (**e**XFEM-AC) for resolution of multi-material cells. J. Comp. Phys., 266:596–612, 2014.

[4] R. Kramer and D. R. Noble. A conformal decomposition finite element method for arbitrary discontinuities on moving interfaces. Int. J. Numer. Meth. Engng., 100(2):87–110, 2014.

[5] D. R. Noble, E. P. Newren, and J. B. Lechman. A conformal decomposition finite element method for modeling stationary fluid interface problems. Int. J. Numer. Meth. Fluids, 63:725–742, 2010.

Sandia National Laboratories