TOWARDS CONCEPTUAL SIMULATIONS FOR LINEAR ELASTICITY ANALYSIS

SEBASTIAN PEÑA SERNA

Supervisor:

Dipl. Math.-Tech. Holger Graf

FRAUNHOFER INSTITUTE FOR COMPUTER GRAPHICS (IGD)

DEPARTMENT FOR INDUSTRIAL APPLICATIONS (A2)

DARMSTADT

GERMANY

May, 2006

TOWARDS CONCEPTUAL SIMULATIONS FOR LINEAR ELASTICITY ANALYSIS

SEBASTIAN PEÑA SERNA

Final Report presented to obtain the International Certificate in New Media Technology (ICPNM)

Supervisor:

Dipl. Math.-Tech. Holger Graf

FRAUNHOFER INSTITUTE FOR COMPUTER GRAPHICS (IGD)

DEPARTMENT FOR INDUSTRIAL APPLICATIONS (A2)

DARMSTADT

GERMANY

May, 2006

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This project aims at developing a novel feature movement mechanism for Finite Element Analysis (FEA) meshes through a haptic pen (Phantom stylus) within a Virtual Reality (VR)-based environment. The tools developed in this project allow for making real-time feature movements and gesture-based manipulations on surface and volume meshes which lead to “what-if-analysis” on the fly. Focus of the work is the real-time operation in highly resources and computing intensive envision meshes.

The approach here presented might enable the project to head towards conceptual simulations within the engineering domain (targeting at linear static analysis). Methodologies are presented for interactive mesh modifications within linear static problems based on tetrahedral meshes, ‘on-the-fly’ feature movements (such as through holes), the adaptation of the surrounding volume mesh, fast and adaptive mesh refinement techniques, quality measures to ensure the consistency and quality of the newly, ‘on-the-fly’ generated elements and the necessary topological operations for unstructured triangular tetrahedral mesh. Because of the interactive environment, all operations have to comply with the real-time constrain and have to be evaluated for local simplicial elements.

In order to reduce the analysis and computation overhead, a new idea was introduced for local evaluations based on an integrated submodeling procedure. From global to local and coarse to fine, here additional constrains like influence and consistency on/with the overall model have to be considered. The approach take essential boundary conditions into account which constrain the isolated part of the submodel and which are derived from the global solution.

Within the engineering environments, the explicit and implicit methodologies from global to locally detailed analysis (from coarse to refined results) allows the approach to head for fast preliminary design evaluations.

The project has relevance for the computer graphics field because it will offer human interaction with feedbacks for real-time applications, supporting industrial applications. In the same way, the industrial companies (like aircraft and automobile manufacturers) could take benefit of the results, since it supports engineers with tools for conceptual simulations and shortening analysis cycles, reducing cost and time to market.

1. INTRODUCTION

During the last decade main efforts within the scientific and engineering visualization community focused on the interactive display and visualization of (large) data sets which result from resource intensive FE or CFD simulations. However, most of those simulations are done off-line which do not allow for interactive 'what-if' analysis in view of a change of the primary design domain. All simulation results are either kept within large databases for post processing purposes or cannot be processed in a 'user-in-the-loop' resp. interactive system. Design changes, such as the movement or suppression of features, within the underlying CAx/FE models usually imply many manual interferences, repair and efforts (remeshing tasks of the area in question, grid repair, etc.) and are not suitable for highly interactive environments. In order to guarantee elements of high quality during remeshing or mesh refinement tasks, time consuming operations and processing are done in off-line processes and the results of the simulations with the newly generated domains are usually post processed in later stages.

Current virtual reality systems are facing a paradigm shift away from pure visualization and design review systems to more generative environments, in which designer are able to sketch their designs or engineers are able to generate their models with the assistance of advanced interaction possibilities. Within pure engineering tasks, such as the analysis of physical behavior, virtual environments only established as post processing unit, in which simulation results can be interactively displayed and visualized to get more in depth information about the simulation mock up. They do not offer possibilities to change interactively the underlying domain, e.g. by manipulating the volume mesh, with an adjustment of the simulation in real-time. This is due to the bottleneck implied by pre-processing and post-processing stages. However, the computer graphics field offers several methods and techniques related to visualization, simulation and user interaction with 3D meshes. These are used to increase the speed of the visualization or transferring processes, as well as to improve the accuracy of the simulation results and the user perception of the performance of the virtual model. In general, these techniques are based on either geometric transformations, topologic transformations or both.

Some techniques related to those are among others: multi-resolution, optimization, refinement, remeshing, simplification, mesh morphing, and shape editing. Although, some of the proposed techniques are very advanced and give valuable procedures for working with meshes such as, progressive transferring, real-time editing, view-dependent resolution, level-of-detail, subdivision, free shape modeling, etc., there are no methods which allow for real-time mesh manipulation keeping quality elements and consistency for conceptual ‘on-the-fly’ FE simulations.

Therefore, this work describes an approach which might enable to head towards conceptual simulations within the engineering domain (targeting at linear static elasticity analysis). Conceptual simulation is a concept based on highly interactive environment, computational simulation and real-time visualization. Conceptual simulation allows for process decision beyond interactive visualization, this new concept allows for interaction within process decision.

Methodologies are presented for interactive mesh modifications within linear static problems based on simplicial meshes, ‘on-the-fly’ feature movements (such as through holes), the adaptation of the surrounding volume mesh, fast and adaptive mesh refinement techniques, quality measures to ensure the consistency and quality of the newly, ‘on-the-fly’ generated elements (this implies evaluating local simplicial elements in real-time) and the necessary topological operations for unstructured meshes.

The following sketches present the aim of this work. As a first task, the conceptual simulation requires user interaction tools. Because of that, a feature selection needs to be developed (figure 1).

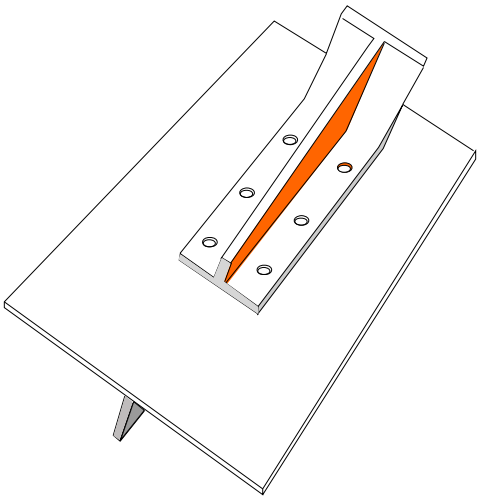


Figure 1: Sketch for feature selection.

Afterwards, the selected feature needs to be moved. Figure 2 shows a sketch of a constrain movement of a through hole. The movement is considered constrain, because the movement of the hole is restricted to the surface where the feature (hole in this sketch) is defined.

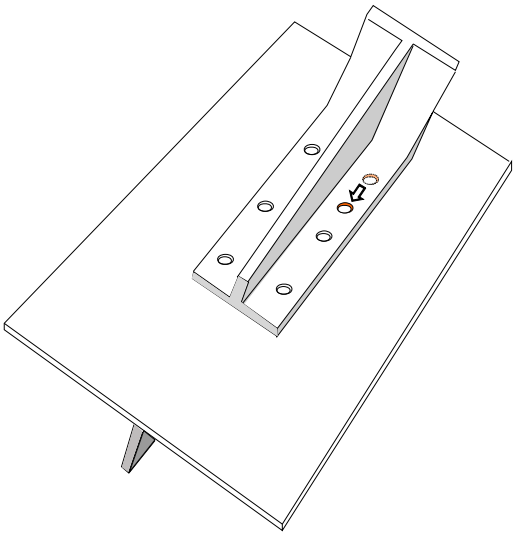


Figure 2: Sketch for constrain feature movement of one through hole.

In addition, several features could be moved during the same cycle (figure 3). In this case the movement applied to one feature must be replied to the other features which are part of the movement cycle.

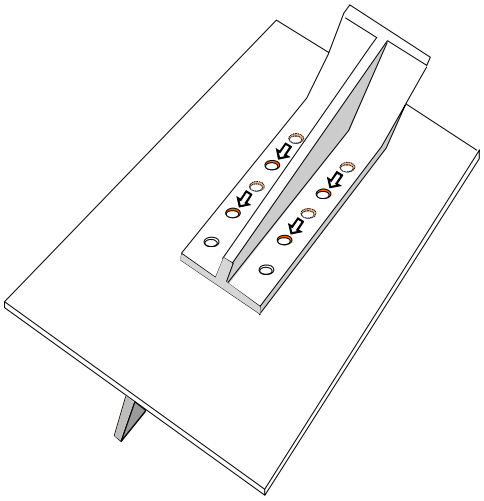


Figure 3: Sketch for constrain feature movement of several simultaneous through holes.

The final sketch (figure 4) represents a free movement which can be assimilated as an extrusion. The idea of this sketch aims at creating the option of increasing or decreasing the dimension of different features of a mesh in order to evaluate the influence on the stress field.

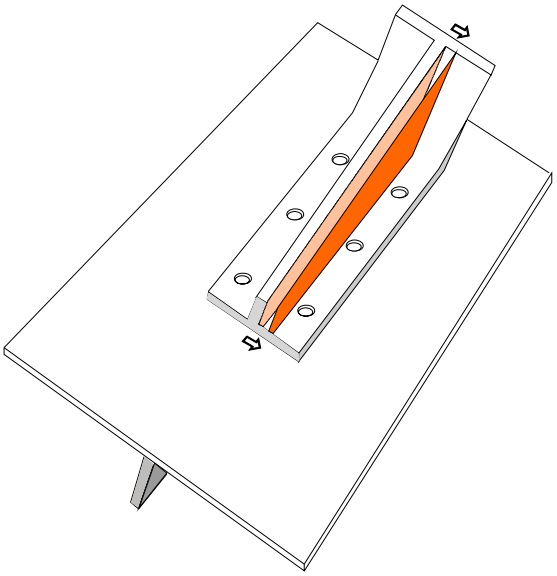


Figure 4: Sketch for free feature movement.

1.1 MOTIVATION

Currently, existing CAD/CAE systems for the analysis and optimization procedure of the digital model are crucial during analysis processes. However, there are several problems as:

a) Redefinition or modifications of boundary conditions for derived simulations.
b) Still lack of efficient algorithms and techniques.
c) Limitations for efficient use (modelling, solver, interaction).
d) Workflow of design and redesign is significantly influenced.

A typical FEM analysis process is composed of several steps. First, the CAD geometry is used as a basis for generating the FE-model (mesh), second the FEM

problem is defined (boundary conditions), third the linear system is built and the solution of the problem is obtained, and finally the results are visualized (figure 5)

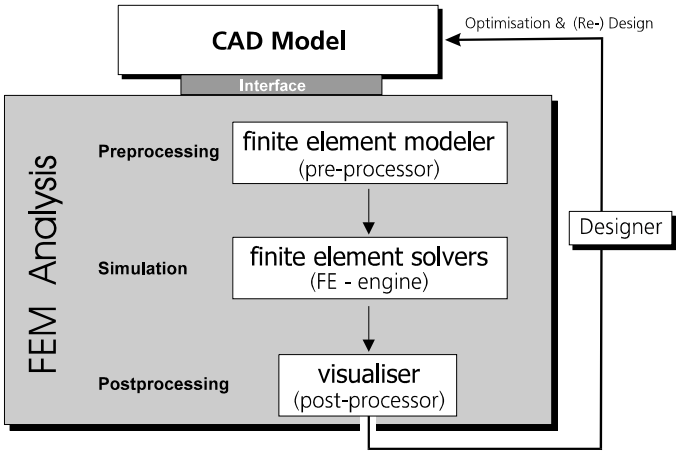


Figure 5: Current FEM Analysis process.

The effort estimated in FEM analysis is:

i) 40% of workload is needed for the CAD geometry.
ii) 30% for mesh generation and refinement including the determination of boundary conditions
iii) 20% for the analysis
iv) 10% for the execution of the simulation.

An additional problem is derived from the separation of the CAD and CAE departments which generates a workflow of re-design and re-analysis time consuming due to lack of integration. Unfortunately, Virtual Reality is only used as visualization tool. Therefore, the aim is to use Virtual Reality as interactive analysis tool in order to bridge the gap between CAD and CAE.

1.2 OBJECTIVES AND METHODS

In order to achieve the aim of this work, the efforts were constrained in view of the following objectives:

i) The user can select and manipulate features without making damage on the boundary or external topology.
ii) The surface mesh must be updated on-the-fly while the user makes the feature manipulation.
iii) The changes made on the surface mesh must be transferred to the volume mesh in order to solve the new model and get the new results.
iv) The new results must be displayed to the user and this one must be able to start the cycle again.
v) Since, the application runs on a client/server architecture, it is required that some tasks take place on the client side and some other on the server side.

The following methods were derived from the previous objectives (see the flow chart in figure 6):

a) Developing a quality measure to evaluate simplicial elements.
b) Developing mesh manipulation techniques (for surface and volume meshes).
c) Developing a submodel technique for computing the solution of the manipulated region.

d) Designing an efficient data structure.
e) Designing a turn around loop integration, in order to integrate the conceptual simulation module with the Visualization tool for Virtual Reality.

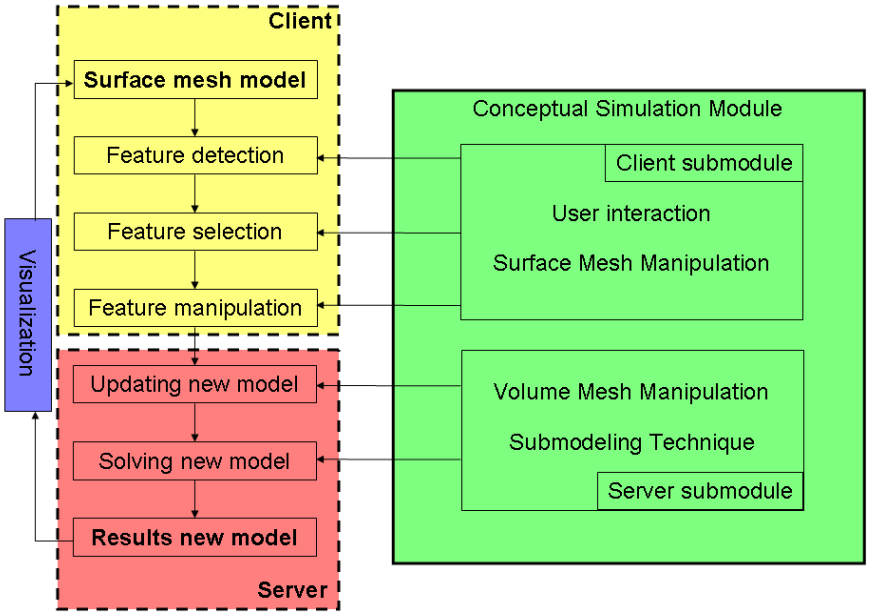


Figure 6: Flow chart of the Feature Manipulation on Surface and Volume Meshes for Conceptual Simulations.

2. RELATED WORK

The presented work is related to several technological aspects. Thus an overview of the related research efforts is presented in this section. As this work aims at interactive, conceptual linear static analysis, the influence of mesh modifications in view of unstructured surface meshes (which suits as the basic representation within a virtual environment) and derived volume meshes (basis of the linear static elasticity simulation) was insvestigated. Although, there are researchers who have been working on topics with a close relation to parts of this project, there are some topics which have not been covered or which are still subject to ongoing research. Given the framework of this project, several areas of investigation have been identified: a) mesh manipulations techniques, b) mesh quality measures and c) efficient data structures suitable for real-time mesh manipulations. The following presents related work categorized on the topics mentioned before.

2.1 MESH MANIPULATION TECHNIQUES

This project makes use of techniques and methodologies from mesh optimization, progressive mesh, and mesh simplification to name few among others, which are interesting topics for computer graphic researchers. Several papers have been published in these fields and it would be impossible to mention all references, however, a sophisticated overview on ongoing work can be found in Bischoff and Kobbelt [Bischoff 04] with a compound of subdivision and multiresolution techniques, Nealen et al. [Nealen 05] with an overview of deforming models, Heckbert [Heckbert 97] with a survey of simplification algorithms, Alliez et al. [Alliez 05] with an overview of surface techniques and Frey and George [Frey 00] with a general mesh generation coverage.

Within this topic, one can mention, Hope et al. [Hoppe 93] with the idea of developing a method for triangular mesh optimization. They sought for reducing the number of vertices of the mesh conserving the topological type of the original mesh based on:

i) an energy function minimization which conserves the fidelity of the data and

ii) local simplicial complex transformation (such as: edge collapse, edge split and edge swap).

This method initially was formulated for surface reconstruction from unorganized points and mesh simplification.

In a further work, Hoppe [Hoppe 96] introduced the Progressive Mesh representation, as a new scheme for storing and transmitting arbitrary triangular meshes. It is a lossless information method which addressed problems like: mesh simplification, level-of-detail approximation, progressive transmission, mesh compression and selective refinement. The method was based on the minimization of an explicit energy metric which measured the accuracy between the simplified mesh and the original, and a sequence of edge collapse transformations which permit a continuous-resolution representation conserving the geometry and the overall appearance of the mesh.

Popovic and Hoppe [Popovic 97] extend the Progressive Mesh (PM) concept to the Progressive Simplicial Complex (PSC), which in addition to the PM, permits changes over the topology of the mesh and therefore, achieves a better fidelity. The PSC can start from an arbitrary triangulation and achieves a base model consisting of a single vertex. Garland and Heckbert [Garland 97] used quadric error matrices to simplify surfaces, maintaining a surface error approximation. They also allowed for contracting vertex pairs which permits topological joining in addition to collapsing edges. Wundrak et al. [Wundrak 2006] extended the original PM algorithm from triangular to quadrilateral elements. Hence, they also extended the edge collapse operation to quads. Furthermore, they presented an algorithm which permits applying quadric error metrics for the simplification of dynamic meshes.

Another interesting group of activities can be grouped around surface editing. Within this topic, Biermann et al. [Biermann 02] develop a set of algorithms based on multiresolution subdivision surfaces which allow for cut-and-paste operations on surface meshes with a wide variety of blending modes. Suzuki et al. [Suzuki 00] proposed an interactive mesh dragging with an adaptive remeshing technique. The presented technique permits dragging a part of a surface by adaptive local remeshing around the part. This method detects geometrical or topological damage and solves those damages based on operations like edge collapse, edge split and edge swap. However, this technique has drawback caused by the limitations presented in the definitions of a minimum and maximum edge length, which force the method to work with homogenous meshes. On the other hand, the algorithm converge when an appropriate ratio between the minimum and maximum length of the edges is chosen, otherwise, the algorithm could go into infinitive loops.

As well as progressive mesh and simplification methods for surfaces, there similar work for tetrahedral meshes. Staadt and Gross [Staadt 98] presented the implementation of progressively tetrahedralization generated through a sequence of edge collapses. They extended the PM scheme from Hoppe, introducing cost functions for tetrahedral meshes and tests which guarantee the mesh consistency. Pajarola et al. [Pajarola 99] introduced the implant sprays procedure combined with the edge collapse error method, providing an efficient mesh encoding of progressive tetrahedralization. Kraus and Ertl [Kraus 00] presented a method for simplifying nonconvex tetrahedral meshes. They avoided the inversion and intersection of the cells because of the edge collapse operation, by a preprocessing step which converts nonconvex meshes to convex ones. Chopra and Mayer [Chopra 02] proposed a different method instead of the edge collapse in order to simplify progressively a tetrahedral mesh; they introduced the tetrahedron fusion concept. The disadvantage with the techniques mentioned above is the real-time issue, since those techniques use minimization procedures which affected the performance of the methods.

2.2 QUALITY MEASURES

In general, the quality of a mesh itself depends on the element size and shape. However, this quality could have different interpretations, depending on the use given to the mesh, for example interpolation, approximation or finite elements methods. Shewchuk [Shewchuk 02] explained the relations between mesh geometry, interpolations errors and stiffness matrix conditions and he also showed some quality measures which permit evaluating the quality of a mesh. Ahlmann [Ahlmann 03] presented quality metrics for tetrahedral meshes and explained the distinctions between the shape, the size and the element number.

In addition, depending on the method used to build the mesh, there are different techniques to get one with a good quality. For example to name few among others, there are different techniques for Delaunay meshes like, weighted Delaunay refinement ([Cheng 03]) or replacing the circumcenters with off-centers ([Uegoer 05]). Munson [Munson 05] proposed to optimize the quality of mesh elements, computing the optimal positions of the vertices for improving the average element quality. Liu and Joe [Liu 96] exposed a local refinement algorithm based on 8-subtetrahedron subdivision, guaranteeing a good quality of the mesh.

In the same way, there are some researchers who have been proposing methods for generating tetrahedral meshes with good quality. Alliez et al. [Alliez 05] proposed a method called Variational Tetrahedral Meshing which meshes a 3D domain with a minimization procedure. This procedure combines 3D Delaunay triangulation and vertex relocation, minimizing a global energy over the domain. Molino et al. [Molino 03] implemented a tetrahedral mesh generation algorithm, producing quality elements and well conditioned for deformable bodies.

2.3 DATA STRUCTURE

A robust data structure is always needed for user interaction and real-time applications. Even more, when the user interaction or the application itself involves dynamically changes over the geometry, topology and simulation results of the model. Chen and Akleman [Chen 03] established a framework for the theory and practice of 2-manifold (surface) modeling. They also presented some concepts, data structures and operations related to mesh modeling.

Some researchers have developed data structures for specific applications. In order to reference some examples: Lévy et al. [Lévy 01] developed a data structure for rendering complex unstructured grids, De Floriani et al. [Floriani 03] presented a survey on data structures for Level-of-Detail models, Hippold and Ruenger [Hippold 04] proposed a data management for adaptive hexahedral FEM and Tobler and Maierhofer [Tobler 06] developed a data structure for rendering and subdivision.

Unfortunately, the available data structures only are efficient for specific tasks, though, an efficient data structure for dynamically increasing and decreasing the number of vertices and elements without sorting and reallocating memory, and keeping the consistency of the mesh with appropriate topological tests, is not available. In addition, data structures with information between components (vertex, edge, element 2D or element 3D), in order to identify in real-time, neighbors of a component and reduce the computation time and the complexity of the used algorithms, are not common.

3. METHODS

In this chapter is presented the methodologies which are used for the different steps important to this work. The first step aimed at developing a quality measure which ensures the consistency and the quality of the mesh in view of interactive manipulations, i.e. a quality measure which can evaluate the newly generated elements ‘on-the-fly’. The second step implements and develops mesh manipulation techniques for surface and volume meshes. The third step solves the manipulated part of the mesh through a submodeling method. The next step looked for designing an effective data structure which supports the requirements of the objectives and the needs on the mesh manipulation. Finally, the client/server loop is closed in order to complete the cycle, feature detection, feature manipulation, submodeling computation and results re-visualization.

3.1 QUALITY MEASURE

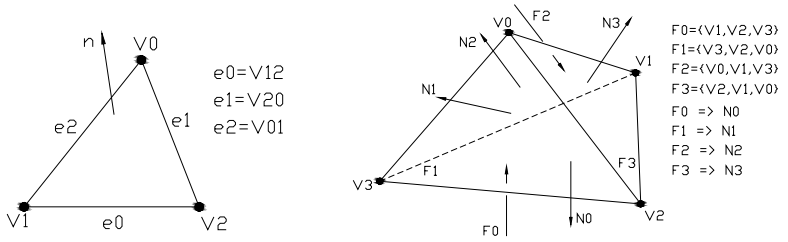
This section presents a new quality measure in view of unstructured grid generation for 'on-the-fly' mesh modifications within highly interactive environments, e.g. virtual reality systems. Aim is to lead real-time/interactive FE mesh modifications, such as feature movements, with adaptive mesh refinements, and at the same time to keep up and ensure a high quality of newly created elements. The measure intends to identify and resolve damages during 'on-the-fly' mesh modifications, preserves the consistency of the original mesh, measures accuracy of the element quality and can be evaluated in real-time. The proposed quality measure is based on computing the ratio between height and median of simplicial elements, aiming at maintaining and improving the conditioning of the stiffness matrix for finite element analysis. Here is showed that it can be successfully used for surface and volume FE-mesh real-time modifications and that it allows for evaluating ‘on-the-fly’, whether topological operations have to be done or not.

An interactive new measure identifies ‘on-the-fly’ ill-shaped simplicial elements (triangle=tri and tetrahedron=tet) which might have negative influence on the later simulation. It is based on the calculation of the *median vector* (***MV***), which points directly to the mid point of the opposite edge (tri) resp. triangle median point of the opposite face (tet). The computation of the quality measure does not require complex mathematical operations. Thus, it allows for evaluating and adjusting the quality of elements within real-time constraints. It can be shown that the elaborated measure complies with the requirements for a scale invariant, well-defined quality measure and preserves the conditioning of the stiffness matrix.

The concept for keeping the consistency and quality of the mesh during remeshing tasks is based on computing and calculating the measure for simplicial elements (*simQ-simplexQuality, triQ* for triangles and *tetQ* for tetrahedrons) in order to analyze the quality of it and determine if the simplex is subject to topological operations. *simQ* can get values within the closed interval [0,1]. E.g., the regular tetrahedron (which has four equilateral triangular faces) gets a *tetQ* = 1 and the equilateral triangle gets a *triQ* = 1. A value equal to zero (0) means that the simplex is degenerated, in other words, the vertices of the simplex are

either collinear (triangle) or coplanar (tetrahedron), or two or more vertices share the same geometric point. Due to its calculation, the presented method has the advantage of being completely independent of the geometric characteristics of the simplex. It can determine from a topological point of view, if a simplicial element is well-shaped or not. That means, the method is independent from how big or small the simplex is: independent from the length of the element edges.

As the computation of *simQ* does not require complex mathematical operations, this is an advantage in view of the real time constraints. Figure 7 shows the topological nature of tris and tets defined by vertices *Vi*, edges *ei*, faces *Fi* and normals *Ni* in order to derive the computation of *simQ*.



a) Triangle a) Tetrahedron

Figure 7: Topologies of investigated simplicial elements.

3.1.1 Derivation of a Quality Measure

The *Vertex Factor* (*VF*) concept is introduced for the derivation of the quality measure, which is given for a specific vertex. This section shows the computation of *VF* for given Vertices (defined by ***Vi***). First is calculated the vectors generated by the edges which are incident to the given vertex. Those vectors have the tail at the given vertex and the head at the opposite vertex of the incident edges. Let *n* be the number of vertices of the given simplex (1).

$$\boldsymbol{Vij} = \boldsymbol{Vj} - \boldsymbol{Vi} \qquad i, j \in \{0,..., n-1\}, i \neq j \tag{1}$$

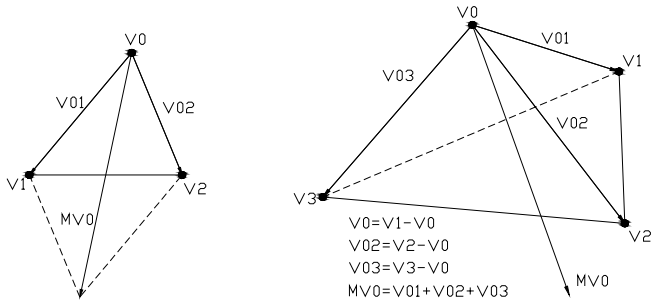
The *Median Vector* (***MV***) is the sum of the vectors generated in (1):

$$\boldsymbol{MVi} = \sum_{j=0, j\neq i}^{n-1} \boldsymbol{Vij} \tag{2}$$

The unit vector of ***MVi*** is given by:

$$\boldsymbol{MVi}_u = \frac{\boldsymbol{MVi}}{|MVi|} \tag{3}$$

As example, figure 8 shows the calculation of ***MV0*** that corresponds to vertex ***V0***.



a) ***MV0*** for a triangle a) ***MV0*** for a tetrahedron

Figure 8: Definition of the Median Vector (***MV0***).

In case of a tetrahedron the direction of ***MV*** to a given vertex, points directly to the triangle median point of the opposite face, see figure 9. The magnitude of the vector is three times the magnitude of the *Cevian* from the vertex ***Vi*** to the opposite face *Fi*. For triangles, ***MV*** points directly to the mid point of the opposite edge and its magnitude is two times the *Cevian* from Vertex ***Vi*** to the opposite edge ***ei***.

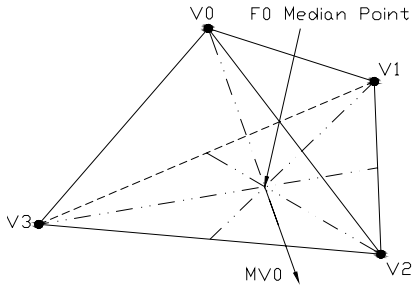


Figure 9: The triangle median point of face *F0* is the intersection between the line defined by the vector ***MV0*** and the vertex ***V0***, and the plane defined by ***V1***, ***V2*** and ***V3***.

Given this characteristic of the ***MV*** vector, *VF* is derived, in order to qualify the shape of a given simplicial element. To calculate *VF*, the dot product between the direction of ***MV*** and a unit normal vector ***ni*** which for tetrahedrons is given by the normal (***Ni***) of the opposite face to ***Vi*** is computed (for triangles is equivalent to the normal vector of the opposite edge (***nei***)). Let $<\boldsymbol{a},\boldsymbol{b}>$ denote a bilinear form:

$$VFi = <\boldsymbol{MVi}_u, \boldsymbol{ni}> \tag{4}$$

In case of triangles, the normal vector of an edge (***nei***) is perpendicular to the corresponding edge, pointing to the outside of it. Let $\boldsymbol{a}\times\boldsymbol{b}$ denote the cross product, equation (5) allows computing the normal of an edge:

$$\boldsymbol{nei} = \boldsymbol{ei}\times\boldsymbol{n}, \qquad \text{where}: \boldsymbol{e0}=\boldsymbol{V12}, \boldsymbol{e1}=\boldsymbol{V20} \wedge \boldsymbol{e2}=\boldsymbol{V01} \tag{5}$$

Normalizing gets:

$$\boldsymbol{nei}_u = \frac{\boldsymbol{nei}}{|nei|} \tag{6}$$

Figure 10 shows the computed normals used in the calculation of *VF* for a triagle.

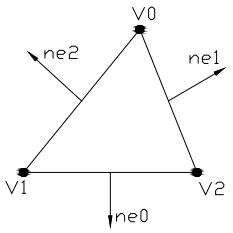
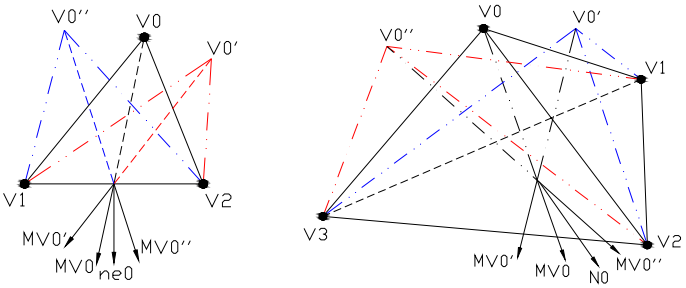


Figure 10: Definition of the normal vector of an edge.

Figure 11 shows the behavior and change of ***MV*** when vertex ***V0*** is moved freely and other vertices of the simplex are fixed. It is possible to see how ***MV*** changes with regard to the unit normal ***ni*** (***nei*** for triangles and ***Ni*** for tetrahedra) which is finally the main ingredient for the quality measure.



a) Triangle performance of ***MV***. b) Tetrahedron performance of ***MV***.

Figure 11: Graphical representation of the behavior of vector ***MV*** when vertex ***V0*** is moved.

The quality measure *simQ* is finally computed by building the product between every *VF* of the simplicial element (7).

$$simQ = \prod_{i=0}^{n-1} VFi \tag{7}$$

Where *n* is the number of vertices of the simplex and *simQ* is named *triQ* for triangles and *tetQ* for tetrahedra.

Shewchuk [Shewchuk 02] proposes seven properties which he recommends to consider when developing a quality measure. It can be shown that these properties are fulfilled for the proposed quality measure. All degenerated element have a quality equal to zero, since either the magnitude of ***MVi*** is equal to zero or the dot product $< \boldsymbol{MVi}_u, \boldsymbol{ni} >$ is null in the case of sliver tets. The quality measure is scale-invariant, i.e. it gets the same value for two different elements with the same shape but different sizes (either area or volume). The quality measure achieves a maximum value of one (see Table 1 and Table 2). All inverted elements have a negative quality, which can be identify by the sign of $< \boldsymbol{MVi}_u, \boldsymbol{ni} >$. Figures 24 and 25 show that the function *simQ(Vi)* is smooth and *quasiconvex* for non-inverted elements and the gradient of *simQ(Vi)* is nonzero for degenerate elements. This is due to the fact that, the gradient which is perpendicular to the level curve can be calculated and the slope is defined at this point on the surface.

3.1.2 Geometrical Interpretation

The properties of the proposed quality measure (*simQ*) can also be shown by geometrical interpretation. As explained before, ***MV*** has the characteristics of having a magnitude of two or three times the *Cevian* from the vertex to the

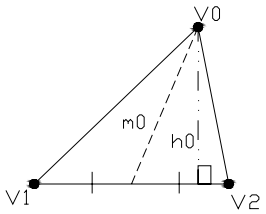
opposite mid point of the edge or median face (geometric centroid or center of mass) depending on triangle or tetrahedron shape. For simplification, it will show the geometrical interpretation on the basis of a triangle.

If it is extended ***MV0*** from ***V0*** to an imaginary vertex and mirror ***V02***, ***V01*** at ***V12*** it is constructed an imaginary parallelogram (figure 3a). Completed to a parallelogram, the two polygonal diagonals (in this case, ***MV0*** and ***V12***) bisect each other, hence, the first confirmation is that ***MV0*** points to the mid point of ***V12*** (or ***e0***). As the imaginary parallelogram is formed by two equal triangles (the triangle *V0V1V2* and its mirrored image) and ***MV0*** goes through the mid point of ***V12***, ***MV0*** automatically classifies as a triangle median. Thus ***MVi*** has a magnitude of two times the triangle median formed by the *Cevian* from a vertex ***Vi*** to the min point of the opposite edge ***ei***.

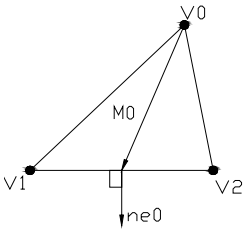
In case of a tetrahedron, the prove is straight forward and similar in construction but would require more details which would blur the report unnecessarily. However, it is noticed that ***MVi*** (for tetrahedra) has a magnitude of three times the tetrahedron median formed from the vertex ***Vi*** to the median point of the opposite triangular face *Fi*.

Since, the factor *VFi* is computed by the dot product of $\boldsymbol{MVi_u}$ (the normalized ***MVi***) and a vector perpendicular to either the opposite edge (tri) or face (tet), it will be shown that $< \boldsymbol{MVi}, \boldsymbol{ni} >$ has a result of two or three time (tri and tet respc.) the height or altitude of the simplex.

By definition, the altitude of a simplex is the *Cevian* which is perpendicular to the edge *ei* or the face *Fi* (tri and tet respc.) and connects the vertex ***Vi*** (see figure 12a).



a) Definition of median (*m0*) and height (*h0*) of a triangle.



b) Definition of a vector ***M0*** with magnitude *m0* and the direction of the median.

Figure 12: Definitions for the geometrical interpretation.

In addition, the bilinear form of a unit vector (***ni***) and an arbitrary vector results in the length of the projection on an extension to the unit vector. Given the projection length, the height *h0* is calculated for the triangle which is equal to:

$$h0 =< \boldsymbol{M0}, \boldsymbol{ne0} > \tag{8}$$

Given (8) and the magnitude of ***MVi***, the *VFi* factor can be redefined as follows:

$$\begin{aligned} \boldsymbol{VFi} &= < \frac{\boldsymbol{MVi}}{|MVi|}, \boldsymbol{ni} > \\ &= \frac{1}{|MVi|} < \boldsymbol{MVi}, \boldsymbol{ni} > \\ &= \frac{1}{|MVi|} \alpha hi = \frac{1}{\alpha mi} \alpha hi = \frac{hi}{mi} \end{aligned} \tag{9}$$

Where *hi* is the height, *mi* the length of the simplex median vector for vertex ***Vi***, and α =2 (triangle), α =3 (tetrahedra). Therefore, (7) is equivalent to:

$$simQ = \prod_{i=0}^{n-1} \frac{hi}{mi} \tag{10}$$

As a special feature for simplicial elements, *hi* and *mi* are equal if the vertices of the simplex are equidistant and if the volume is not zero (tet). In other configuration of vertices, *hi* is always smaller than *mi*. Because of this, it is useful to use this ratio, in order to measure the quality of a simplex given by (10). Another advantage which we can derive by geometric reasoning is that the quality measure can be defined by shape and size of the simplex. This might be possible if ***MVi*** is not normalized. In this case the quality measure would be proportional to $simQ \cong (\alpha \bar{h})^n$, where $\bar{h}$ is the average height of the simplex. Although, the idea was not explored in detail within this work, it could be explored in a future work.

3.1.3 Influence on Stiffness Matrix

Since, the manipulation cause changes on the domain because of the topological operations, which results from the evaluation of the quality measure, the influence of the measure on the stiffness matrix of the linear static system, was evaluated. Thus, within this work was investigated the influence of modified simplicial, onto the stiffness matrix which provides the results for the simulation. From literature is known, that the problem of linear elasticity can be derived from a variational formulation which is similar to the variational formulation of the Poisson equation (11) using a Galerkin method ([Langtangen 03]). Because of that, the Poisson equation is used to explain the influence on the stiffness matrix. Considering $\Omega \to R^n$ the domain and $\partial\Omega = \Gamma$ the boundary, the process aims at finding $u : \Omega \to R$, a scalar field defined over Ω where $f(p)$ is the internal source of the problem and ∇^2 is the Laplacian.

$$-\nabla^2 u = f(p) \tag{11}$$

In order to get the weak form, also called the variational formulation of the problem, (11) is multiplied by a test function $v : \Omega \to R$ and integrated over the domain Ω (12).

$$-\int_{\Omega} \nabla^2 u v d\Omega = \int_{\Omega} f(p) v d\Omega \tag{12}$$

Equation (14) is the weak form of (11) and results from applying the Green Equation (13) to (12) in order to reduce its order. Where $<\boldsymbol{a},\boldsymbol{b}>$ denotes a bilinear form and ∇ the gradient.

$$\int_{\Gamma} v < \nabla \boldsymbol{u}, \boldsymbol{n} > d\Gamma = \int_{\Omega} (v div(\nabla u) + \nabla u \nabla v) d\Omega \tag{13}$$

$$\int_{\Omega} \nabla u \nabla v d\Omega - \int_{\Gamma} v < \nabla \boldsymbol{u}, \boldsymbol{n} > d\Gamma = \int_{\Omega} f(p) v d\Omega \tag{14}$$

In order to obtain the Discrete Problem, the basis functions $\phi_i(p)$ which are Lagrange interpolating polynomials, form a space of functions H^1 (15) with first partial derivate.

$$H^1 = span\{\phi_1(p), \phi_2(p), ..., \phi_n(p)\} \tag{15}$$

Therefore, the vector field u can be interpolated with the basis functions defined in (15) as:

$$u(p) = \sum_i u_i \phi_i(p) \tag{16}$$

where $u_i = u(p_i)$, $p_i \in \Omega$ for $i, j = 0..n$ and n is the number of nodes. In the same way, the test function v is interpolated as follows:

$$v(p) = \sum_j v_j \phi_j(p) \tag{17}$$

Given definitions (16) and (17), the bilinear operator is defined as:

$$\begin{aligned} a(u,v) &= \int_{\Omega} \nabla u \nabla v d\Omega \\ &= a(\sum_i u_i \phi_i(p), \sum_j v_j \phi_j(p)) \\ &= \sum_{ij} u_i v_j a(\phi_i(p) \phi_j(p)) \end{aligned} \tag{18}$$

and the linear operator as:

$$\begin{aligned} l(v) &= \int_{\Omega} f v d\Omega \\ &= \sum_j v_j l(\phi_j(p)) \end{aligned} \tag{19}$$

Finally, the discrete form is given by:

$$\sum_{ij} u_i v_j a(\phi_i(p) \phi_j(p)) = \sum_j v_j l(\phi_j(p)) \tag{20}$$

In order to establish the linear system, the following notation is adopted. If $a_{ij} = a(\phi_i(p)\phi_j(p))$ and $l_j = l(\phi_j(p))$, therefore:

$$\begin{aligned} &< \boldsymbol{Au}, \boldsymbol{v} > = < \boldsymbol{l}, \boldsymbol{v} > \\ &< \boldsymbol{Au}, \boldsymbol{v} > - < \boldsymbol{l}, \boldsymbol{v} > = 0 \\ &< \boldsymbol{Au} - \boldsymbol{l}, \boldsymbol{v} > = 0 \end{aligned} \tag{21}$$

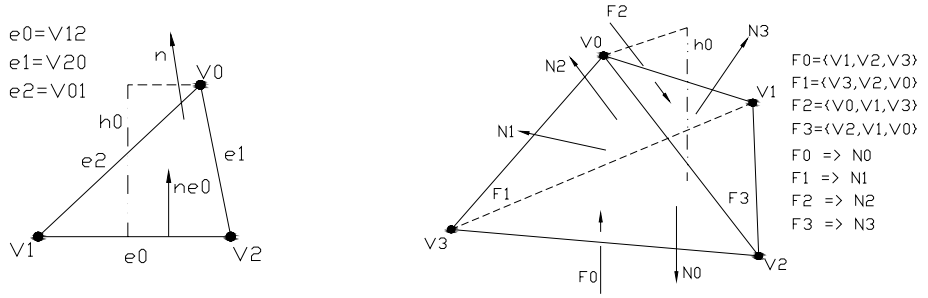
Since, this is valid for every $v \in H^1$ then

$$\boldsymbol{Au} = \boldsymbol{l} \tag{22}$$

where $\boldsymbol{A} = a_{ij} = \int_{\Omega} \nabla\phi_i(p)\nabla\phi_j(p)d\Omega$ and $u = \sum_i u_i\phi_i(p)$. For a domain divided into elements

$$\boldsymbol{A} = \sum_{elms} \int_{\Omega_{elm}} \nabla\phi_i(p)\nabla\phi_j(p)d\Omega_{elm} \tag{23}$$

Figure 13 presents some definitions which are used for calculating the gradient for a triangle and a tetrahedron.



a) Triangle definitions. b) Tetrahedron definitions.

Figure 13: Definitions for calculating the gradient vector.

The following example presents the stiffness matrix for a triangle linear element. The first step aims at calculating the gradient vector ($\nabla\phi_i(p)$) of the basis function for every vertex of the triangle. If $\nabla\phi_i(p) \neq 0$, the gradient is perpendicular to the level curve through (x_0, y_0) (if $z = f(x,y)$), or perpendicular to the surface level (for tetrahedron element) through (x_0, y_0, z_0) (if $F(x,y,z) = 0$). Given the characteristic of the basis functions for first order Lagrange polynomials (P1 Elements), e.g. the basis function ϕ_0 (for vertex ***V0***) gets values: $\phi_0(V0) = 1$ and $\phi_0(V1) = \phi_0(V2) = 0$, the gradient vector is defined by the slope and the direction as follows:

$$|\nabla\phi_i| = \frac{1}{h_i}, \text{ and for triangles } \nabla\phi_i = \boldsymbol{ne_i} = \frac{\boldsymbol{n} \times \boldsymbol{e_i}}{|e_i|} \text{ and tetrahedrons } \nabla\phi_i = \boldsymbol{Ni} \tag{24}$$

With equations (23) and (24), the stiffness matrix for a triangular linear element is:

$$\boldsymbol{K_e} = \begin{bmatrix} \nabla\phi_0 \cdot \nabla\phi_0 & \nabla\phi_0 \cdot \nabla\phi_1 & \nabla\phi_0 \cdot \nabla\phi_2 \\ \nabla\phi_0 \cdot \nabla\phi_1 & \nabla\phi_1 \cdot \nabla\phi_1 & \nabla\phi_1 \cdot \nabla\phi_2 \\ \nabla\phi_0 \cdot \nabla\phi_2 & \nabla\phi_1 \cdot \nabla\phi_2 & \nabla\phi_2 \cdot \nabla\phi_2 \end{bmatrix} = \frac{1}{4A} \begin{bmatrix} |e_0|^2 & \boldsymbol{e_0} \cdot \boldsymbol{e_1} & \boldsymbol{e_0} \cdot \boldsymbol{e_2} \\ \boldsymbol{e_0} \cdot \boldsymbol{e_1} & |e_1|^2 & \boldsymbol{e_1} \cdot \boldsymbol{e_2} \\ \boldsymbol{e_0} \cdot \boldsymbol{e_2} & \boldsymbol{e_1} \cdot \boldsymbol{e_2} & |e_2|^2 \end{bmatrix} \tag{25}$$

As, the stiffness matrix is dependent on the gradient, and this is affected by the slope, which was shown is the inverse of the height; it is possible to derive that a small height will generate a bigger value in the diagonal of the matrix and as a consequence a bigger eigenvalue. This is due to the fact that the sum of the eigenvalues is equal to the trace of the matrix (26) which is the sum of the diagonal elements.

$$\sum_{i=0}^{n-1} \lambda_i = tr(A) \tag{26}$$

According to 3.1.2, the quality measure can be expressed as (10), and therefore, the quality measure uses the height of the simplex in order to compute the quality which directly influences the stiffness matrix. On the other hand, equation 10 shows an elegant ratio, since; this ratio is smaller when the element is ill-shaped, due to the difference between height and median, which is bigger.

3.2 MESH MANIPULATION TECHNIQUES

Based on the above explained quality measure for the identification of ill-shaped elements, this section shows the strategies for the topological operations during (re) meshing tasks. Here, the mesh manipulation techniques which are implemented mainly based on unstructured triangular (display within the virtual environment) and tetrahedral meshes (simulation model). Different mesh manipulation techniques ranging from simple selection mechanism to complex feature detection and modification techniques have been implemented.

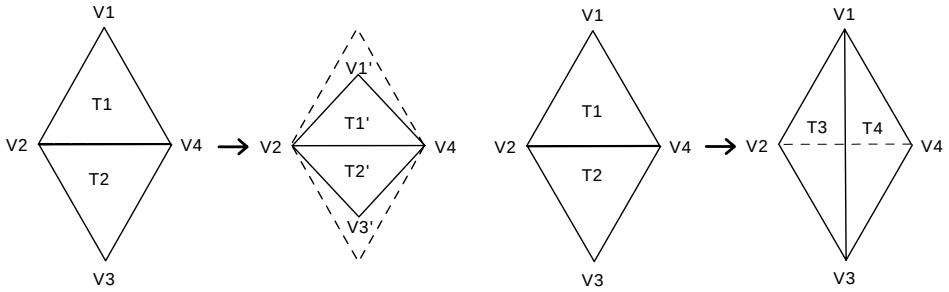
3.2.1 Geometric and Topologic Transformations

The aim targeted at the manipulation and movements of features such as through holes. Therefore, based on a representation of the pure 3D simulation model, it was investigated: feature detection and selection methods, feature movements based on constrained 3D interactive manipulations, movements of features with adaptive refinements on the surface mesh, its instantiation into the volume mesh and topological operations needed to overcome ill-shaped elements. As mentioned above, re-meshing tasks might jeopardize the consistency and quality of the mesh.

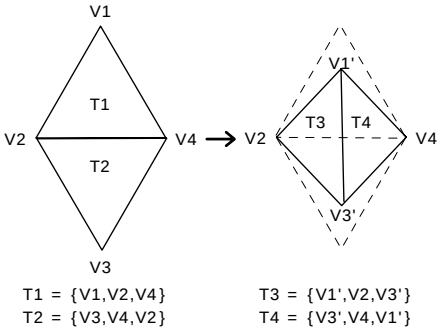
The different processes presented in this work deal with geometric and topologic transformations on surface and volume meshes. Hence, the applied transformations must conserve external topology and the boundary conditions (for the FEM simulation) of the mesh. Due to the nature of topologic operations, the here presented geometric transformations aim at changing geometric information, i.e. they cause a change of vertex information at the same time

trying to keep the connectivity (topology) (e.g. rigid transformations such as pure translations and rotations). On the other hand, topologic transformations create new configurations of connectivity between different vertices, creating or deleting vertices.

Figure 14 presents examples which show three different cases: a) a geometric transformation of vertices V1 and V3, b) a topologic transformation of the triangles T1 and T2 and c) a geometric and topologic transformation of the vertices V1 and V3 and the triangles T1 and T2.



a) Geometric transformation. b) Topologic transformation.



c) Geometric and topologic transformation.

Figure 14: Examples of geometric and topologic transformation.

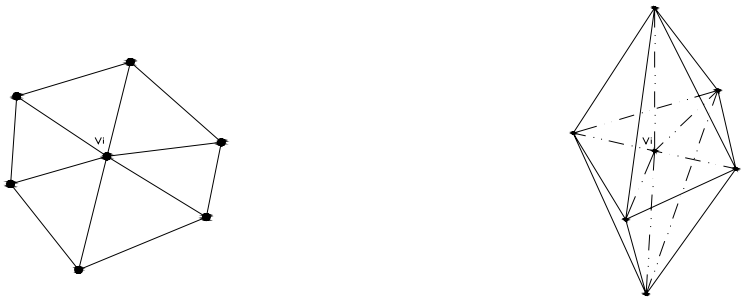
If a geometric transformation is applied, the original elements (in this case, triangles) are kept. As soon as a topologic transformation is applied, new elements are created and the old ones are deleted.

3.2.2 Topological Operations during Mesh Manipulations

The implemented techniques for mesh manipulation of the surface/volume mesh can be classified as feature dragging or surface/volume re-meshing techniques. This process consists in moving vertices of selected simplicial elements causing mesh collapse/split operations in a way that allows for keeping the consistency of the mesh using local topological operations. Those operations deal with adapting ill-shaped areas of the given simplicial mesh to a well-shaped area of it. The idea aims at computing the quality of the affected elements during a user defined movement of a group of vertices and performing topological operations for those elements with a low quality (a quality lower than a given *threshold*). The ill-shaped simplexes are then modified by operations such as edge collapse, edge split, tetrahedron collapse-face swap which are applied appropriately, depending on the damage or degeneracy of the elements in question. With the above mentioned geometric and topologic transformations, the smallest and biggest edges (as well as the sliver tetrahedron) are adapted to maintain a mesh with a

good quality and preserves the consistency for the newly triggered simulation run.

As mentioned above, this approach is based on the computation of the simplicial quality measure (*simQ*) explained in 3.1 for each element. For the sake of simplicity, it is assumed that one vertex is moved, hence, the degeneracies are caused around that vertex. Because of that, the incident simplexes to that vertex (*start(Vi)*) are the candidates for a topological operation (see figure 15). Therefore, *simQ* must be computed for every element which belongs to *start(Vi)*.



a) Incident triangles to Vi. b) Incident tetrahedra to Vi.

Figure 15: Start of a simplexes around vertex Vi (*start(Vi)*).

In calculating *simQ* any ill-shaped simplex belonging to *start(Vi)* and its indicator for a damage is identified, the appropriate topological operation for solving that damage is defined afterwards.

The following process presents a real-time method for adapting ill-shaped areas of simplicial meshes to well-shaped areas: The first step consists in finding the smallest *Vector Factor* (*VFs*) for a minimal *simQ* (27) (which might be lower than a *threshold*). According to (9) all *VFi's* are based on the ratio between the height and the median to a given vertex *Vi*, therefore, a small ratio means a smaller height and/or a bigger median. This is the case if the difference between the smallest edges and the biggest edge incident to *Vi*, is bigger and thus, the damage (degeneracy) of the simplex depends on the smallest or biggest edge.

$$VFs = \min\{VFi\}, \qquad i = 0...n-1 \tag{27}$$

where *n* is the number of vertices of the simplex. However, as a general degeneracy could be caused by smaller edges, bigger edges or in the case of tetrahedral meshes, sliver tetrahedrons, it is needed to identify the different cases. Therefore, the length average of the edges for the simplex is computed (28).

$$d_A = \frac{\sum_{j=0}^{k-1} |ej|}{k}, \qquad j = 0...k-1 \tag{28}$$

where *k* is the number of edges of the simplex, three for triangles and six for tetrahedra. In addition, the shortest (29) and longest (30) edge incident to the vertex *Vi* have to be identified:

$$d_S = \min\{|Vij|\}, \qquad i, j = 0...n-1 \wedge j \neq i \tag{29}$$

$$d_L = \max\{|Vij|\}, \qquad i, j = 0..n-1 \wedge j \neq i \tag{30}$$

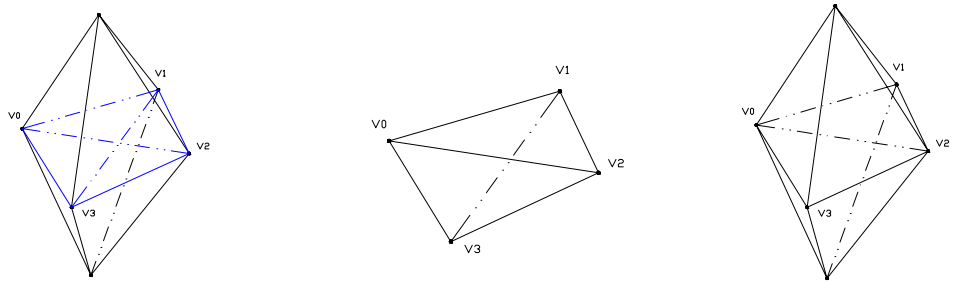
where *n* is the number of vertices of the simplex. In order to define the appropriate topological operation (*edge split* or *edge collapse*) or a combination of topological operations (only for tetrahedra, *tetrahedron collapse-face swap*), the factors *SPL* (31) and *COL* (32) are computed. These factors allow identifying if the simplex has a bad quality because of a shorter, longer or a degeneracy as a sliver tetrahedron which has two large dihedral angles and its four vertices are almost forming a plane. Therefore, if the longest edge (30) is much bigger than the average, the factor SPL should be smaller than COL, hence a split operation must be performed.

$$SPL = \frac{d_A}{d_L} \tag{31}$$

$$COL = \frac{d_S}{d_A} \tag{32}$$

In case of an identification of a sliver tetrahedron the absolute value of the difference between the two factors SPL and COL has to be calculated. If inequality (33) is satisfied, the combined topological operation *tetrahedron collapse-face swap* solves the degeneracy (figure 16). This can be identified due to the difference between the longest and shortest edges which are quite similar (typical for well-shaped elements), but the tetrahedron has a bad quality.

$$|SPL + COL| < 2 - \frac{threshold}{5} \tag{33}$$



a) Ill-shaped tetrahedron, because of a sliver tetrahedron.

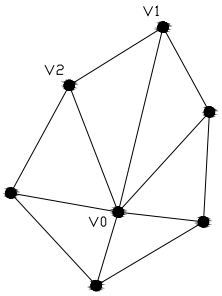
b) Sliver tetrahedron formed by the vertices V0, V1, V2 and V3.

c) Combination of a *tetrahedron collapse* and a *face swap* operation

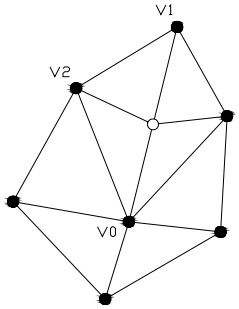
Figure 16: Example of a *tetrahedron collapse-face swap* operation.

The two following excluding inequalities determine if an *edge split* or *edge collapse* operation has to be applied. If inequality (34) is satisfied, an *edge split* operation must be performed (figure 17 presents an example of an *edge split* operation):

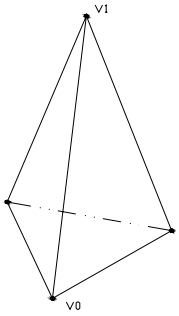
$$SPL < COL \tag{34}$$



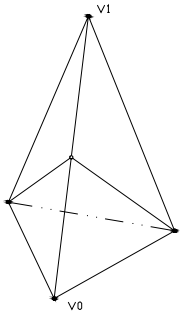
a) Ill-shaped triangle, because of the edge V01. The edge is too long for the conformed *start(V0)*.



b) Well-shaped triangle after an addition of a vertex, because of the *edge split* operation.



c) Ill-shaped tetrahedron, because of a too large edge (**V01**).

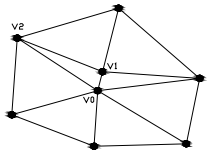


d) *Edge split* operation applied between the vertices V0 and V1.

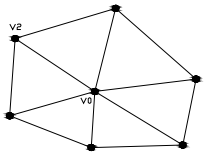
Figure 17: Example of an *edge split* operation.

Otherwise inequality (35) is satisfied and an *edge collapse* is performed. Figure 18 shows an example of an *edge collapse* operation.

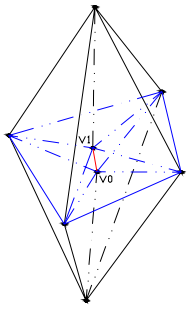
$$COL < SPL \tag{35}$$



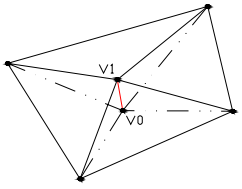
a) Ill-shaped triangles, because of the edge V01. The edge is too short for the conformed *start(V0)*



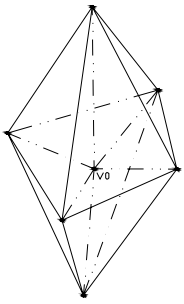
b) Well-shaped triangles after an edge removing, because of the *edge collapse* operation.



c) Ill-shaped tetrahedra, because of a too short edge (**V01**).



d) Tetrahedra around the edge **V01** which are collapsed.



e) Generated tetrahedra after an *edge collapse* operation.

Figure 18: Example of an *edge collapse* operation.

After the application of the topological operation, the new generated simplexes should be processed again as the new elements might also be ill-shaped.

3.3 SUBMODELING

Submodeling is a technique for computing the solution of a region of interest which belongs to a model with a global solution. This technique was developed for linear static elasticity problem and it is totally transparent to the user, who does not detect that the new solution was computed locally. The techniques complies with the real-time requirement and is used for mesh refinement operations in order to compute new solutions from global to local analysis and from coarse to fine meshes.

This section describes an approach to achieve real-time isotropic linear static elasticity computation based on a global solution and performing the computation on an isolated mesh. The isolated mesh corresponds to the part of the mesh which was manipulated. This is possible if the essential boundary conditions of the isolated mesh are derived as the nodal deformation solution given by the global solution. Essential boundary conditions are the boundary conditions of the Finite Element problem which need to be defined in order to achieve a solution of the system, in order to avoid singularities. The essential boundary conditions are defined as displacements over the boundary of the mesh.

3.3.1 Isotropic Linear Static Elasticity Analysis

The theory of Linear Elasticity is based on Solid Continuum Mechanics. The following formulation is adopted from [Langtangen 03]. The analysis is based on the solution of the Navier equation modeling the linear elasticity problem which is the Newton's second law (36).

$$\rho \frac{\partial^2 u_r}{\partial t^2} = \frac{\partial \sigma_{rs}}{\partial x_s} + \rho b_r \qquad where: r,s = 1..3 \tag{36}$$

The term in the left-hand side represents the acceleration of the medium, on the right-hand side the first term represents internal forces and second term represents external or body forces. From the Solid Continuum Mechanics, (36) models a solid domain immerses in a fluid with a defined acceleration, affected by external forces (e.g. gravity) and internal forces in the medium due to stresses. For the linear static elasticity analysis the acceleration of the medium term can be neglected, therefore the Newton's low can be simplify as (37).

$$\frac{\partial \sigma_{rs}}{\partial x_s} = -\rho b_r \tag{37}$$

Equation (37) governs the problem, where the aim is to calculate the *stress tensor* represented by σ_{rs} with 6 unknown values, and the density ρ and the external forces b_r which are the known values. In order to solve the problem, additional information is used to complete the system, such as the material properties. When the linear static problem is solved for elastic materials, a relation between the stresses and the deformations (*u*) is given by the *strain tensor* (38).

$$\varepsilon_{rs} = \frac{1}{2}\left(\frac{\partial u_r}{\partial x_s} + \frac{\partial u_s}{\partial x_r}\right) \tag{38}$$

The Hooke's generalized law (39) highlights the relation between stresses and strain.

$$\sigma_{rs} = \lambda \frac{\partial u_q}{\partial x_q} \delta_{rs} + 2\mu\varepsilon_{rs} \tag{39}$$

where λ and μ are Lame's elasticity constants and δ_{rs} the Kronecker delta, which represents the values of an Identity matrix. Given the previous definitions, the stress tensor can be expressed as (40).

$$\sigma_{rs} = \lambda \frac{\partial u_q}{\partial x_q} \delta_{rs} + \mu(\frac{\partial u_r}{\partial x_s} + \frac{\partial u_s}{\partial x_r}) \tag{40}$$

Finally, the stress tensor definition (40) is integrated within the equation of Newton's law as follows:

$$\frac{\partial}{\partial x_s}\left(\lambda \frac{\partial u_q}{\partial x_q} \delta_{rs} + \mu(\frac{\partial u_r}{\partial x_s} + \frac{\partial u_s}{\partial x_r}) \right) = -\rho b_r \tag{41}$$

The Finite Element Method is used to create the linear system $Ku = c$ and solve the linear static elasticity problem and getting. The solution of the model is the displacement field, which is used to compute the stress and strain tensor at every node. The first step aims at generating an approximation of the displacement field by means of interpolation basis functions N (42). Where *n* is the total number of nodes.

$$\hat{u}_r = \sum_j u_j^r N_j(x_1, x_2, x_3) \qquad where: i, j = 1..n \tag{42}$$

In order to reduce the order of the problem and to obtain a variational formulation, the Galerkin Method is applied to (43). The process is shown first for stress component.

$$\int_\Omega \sigma_{rs} \frac{\partial N_i}{\partial x_s} d\Omega = \int_{\partial\Omega} N_i \sigma_{rs} n_s d\Gamma \tag{43}$$

Equation (44) shows the corresponding formulation for every two joining nodes K_{ij}. This equation is getting by means of replacing the stress tensor by (40) in (43).

$$K_{ij} = \int_\Omega \left(\mu(\sum_k \frac{\partial N_i}{\partial x_k} \frac{\partial N_j}{\partial x_k}) \delta_{rs} + \mu \frac{\partial N_i}{\partial x_s} \frac{\partial N_j}{\partial x_r} + \lambda \frac{\partial N_i}{\partial x_r} \frac{\partial N_j}{\partial x_s} \right) d\Omega \tag{44}$$

The right-hand side of the equation of the Newton's law, which represents the external forces term, is defined as follows (45):

$$c_i = \int_{\partial\Omega n} N_i \sigma_{rs} n_s d\Gamma \tag{45}$$

Given the equations (44) and (45), the linear system can be established. In order to solve the problem, boundary conditions must be defined. Those boundary conditions are divided into essential boundary conditions and natural boundary conditions. The essential boundary conditions are prescribed displacements which are defined at some nodes of the meshed domain. The natural boundary conditions are considered external pressure over the surface of the domain.

3.3.2 Submodeling Solver

As was mentioned before, the part of the mesh where the manipulation was performed is isolated and the nodal global displacement solution is used as essential boundary conditions in the interface between the isolated region and the global model. Figure 19 shows the selection of the part of the mesh of interest in order to apply the submodeling technique.

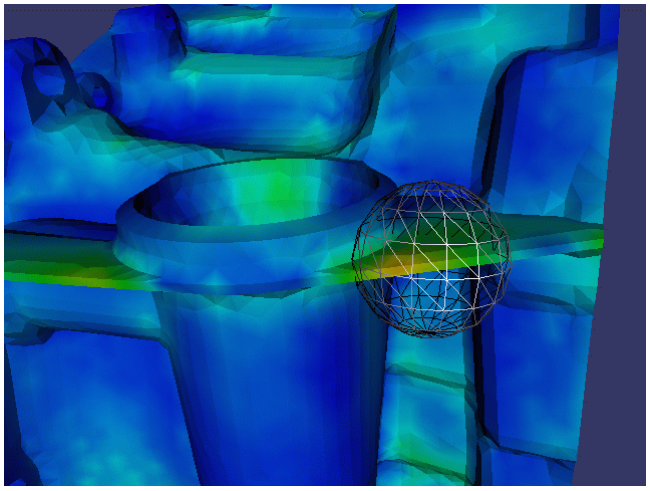


Figure 19: The region of the mesh of interest is selected.

Given that, although the model is isolated and only part of the mesh participate within the new computation, the whole model is considered because the interface between the whole mesh and the submodel is governed by the displacements of the global mesh at the interface which will affect the internal solution of the submodel. Figure 20 presents an isolated region where the submodeling technique is applied.

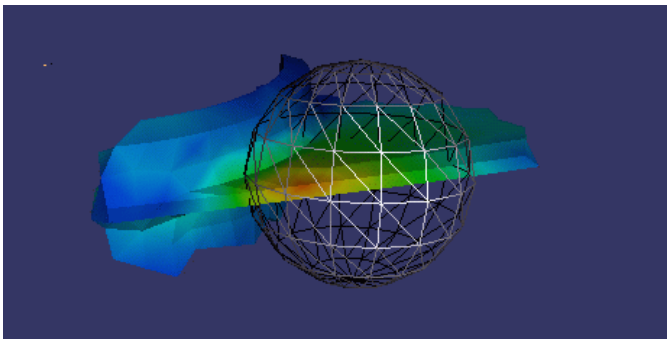


Figure 20: The region of interest is isolated.

3.4 DATA STRUCTURE DESIGN

Given the context of the manipulations, an efficient time computation and consuming, and robust data structure was developed for surface (triangular) and

volume (tetrahedral) meshes. The data structure permits dynamically increasing and decreasing the number of vertices and elements without sorting and reallocating memory, and keeping the consistency of the mesh with appropriate topological tests. Additionally, because of the nature of the topological operations (for the mesh manipulations), whole access from a component (vertex, edge, element 2D or element 3D) to another component was given, in order to identify in real-time, neighbors of a component and reduce the computation time and the complexity of the used algorithms. This direct access permits also adding "intelligence" to pure meshes with very low complexity algorithms which allows obtaining the boundary of surface or volume meshes up-to-date (considering the different changes caused by the feature manipulations) and on-the-fly, as well as algorithms for defining and grouping elements which can heuristically belong to a surface definition.

The data structure was developed in order to comply with the following constrains:

a) Accessing information in real-time, in order to avoid data searching.
b) Connecting information between the different elements (1D, 2D and 3D).
c) Updating information while the surface or volume mesh is changing.
d) Providing a data structure which allows adding and removing information without sorting.
e) Designing a data structure inexpensive (in terms of computing time) in the process of filling and updating.
f) Designing a data structure which supports surface elements (triangles and quadrangles) and volume elements (tetrahedra and hexahedrons).

Figure 21 shows a chart with the designed data structure architecture.

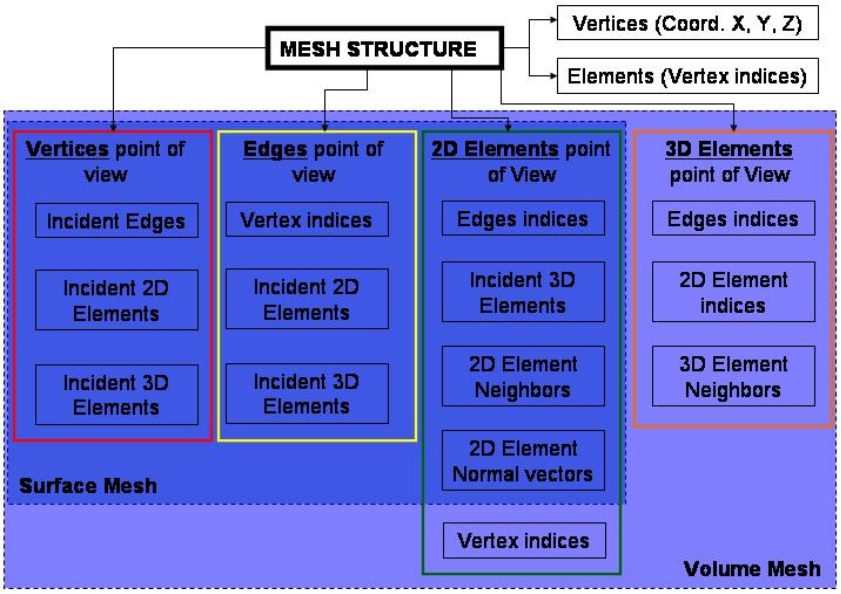


Figure 21: Design of the data structure.

Advantages of the designed data structure:

i) It performs computations and component relations while the information of vertices and elements are stored. In other words, it is not needed an additional time or computing process in order to fill the data structure.
ii) It works with indices (integers). There are only two fields which work with floats, the Vertices and the Normal Vectors.

iii) It permits to create and to delete vertices or elements on-the-fly. Since, it flags the killed elements and removes the links of the killed information (vertex or element). It does not deal with memory reallocation in order to fill a space left by a killed information.
iv) It works with memory buffers which allow adding information without doing memory petitions very often.
v) The additional information (additional to vertices and elements) allows for accessing to every required field on-the-fly, as everything is connected.
vi) Because of v), the data structure is able to make faster computations of: a) getting the surface boundary, b) getting the volume boundary, c) creating a special type of CAD IDs (given a heuristic), among others.

3.5 TURN AROUND LOOP INTEGRATION

The architecture of the Visualization tool used for the Conceptual Simulation process is described in figure 22. This tool is used for interactive exploration of data in Virtual and Augmented Reality. This architecture has the advantage of being module-based, thus, several modules can be added to it in order to increase the services.

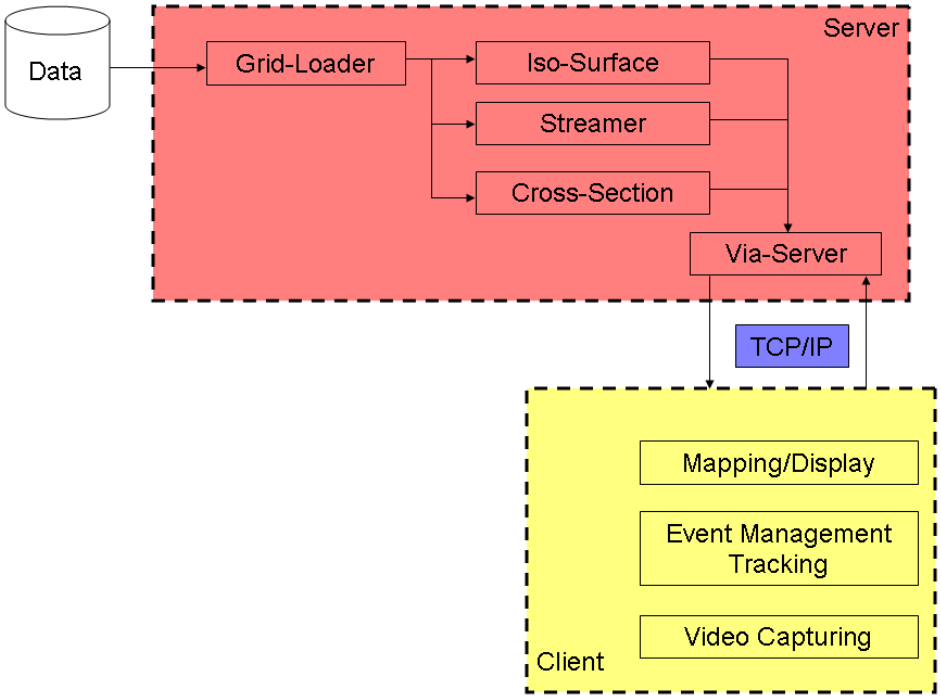


Figure 22: Architecture of the Visualization tool used for the Conceptual Simulation process.

The Conceptual Simulation was created as a module which was dived into submodules, the client and the server submodule. The client submodule deals with the surface mesh manipulation and the user interactions. The server submodule deals with the volume mesh instantiation and the submodeling solver. The turn around loop integration looks for creating a continued process where a user can interact with the model, change the model and retrieve information of the new results from the FEM solver. Figure 23 shows the flow chart of this process.

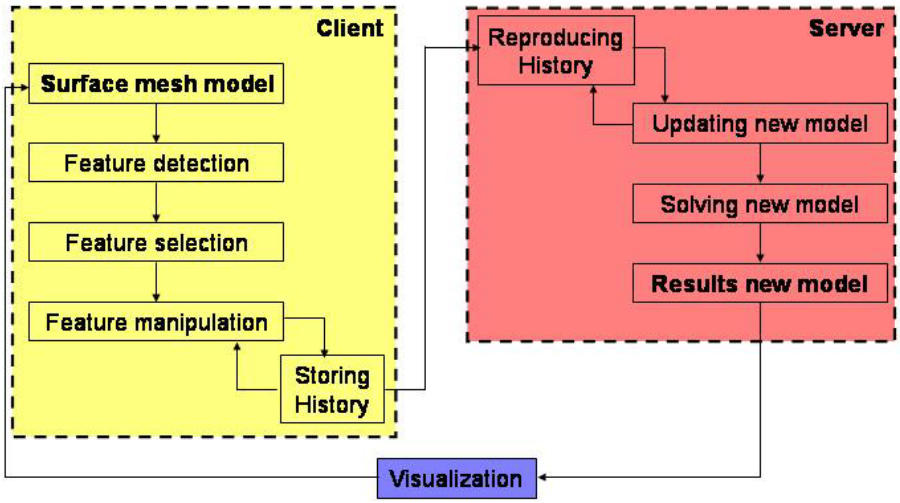


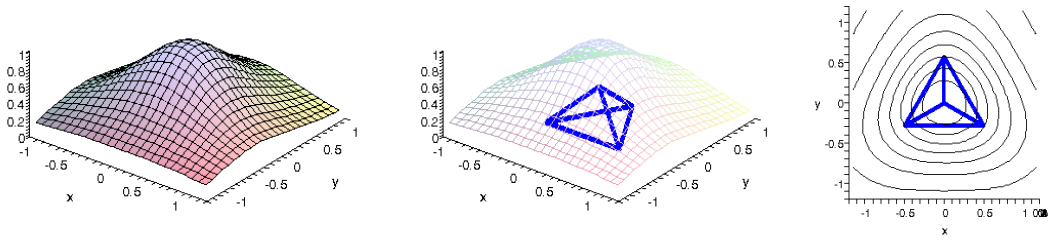
Figure 23. Turn around loop integration diagram.

4. RESULTS

This paragraph presents several results which show the behavior and performance of the proposed quality measure, the performance of the topological operations and the results of the manipulation. For quality measure results, first is shown the functional behavior of the measure in view of moving single vertices and second is presented a benchmark of comparisons of the proposed quality measure with measure available within literature. For the topological operations, statistics from the performed operations during manipulations are shown in order to present the reliability of the process. Last but not least, it is showed the first results of an application for real-time manipulations based on geometric and topologic operations which are supported by the proposed methods.

4.1 QUALITY MEASURE PERFORMANCE

In order to present the global performance of *tetQ*, the following figures (figures 24 and 25) show the behavior of *tetQ(Vi)* for 2 degrees of freedom movements in x-,y-direction of one of the vertices (***Vi***) with the z coordinate fixed at height h. We assume a regular shape of the tetrahedron with the constraint x-/y-movement of the upper vertex.



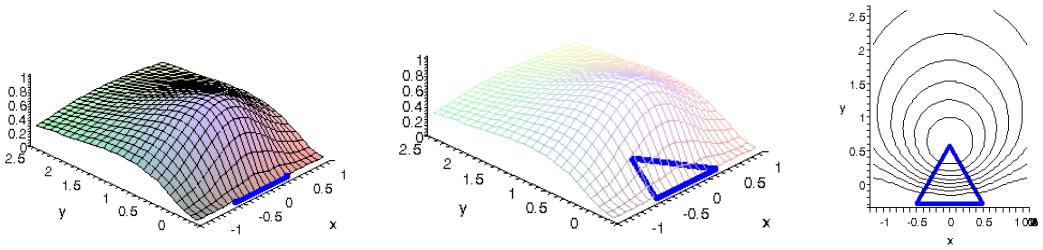
a) Grid representation. b) Wireframe representation. c) Contour representation.

Figure 24: Performance of *tetQ*, with the regular tetrahedron on the background.

Figure 24 shows the behavior of the quality measure as function of different positions caused by the upper vertex of the regular tetrahedron sketched in the diagrams. From figure 24a, it is possible to conclude that in case of a regular tetrahedron *tetQ* gets the maximum value *tetQ*=1.0. The quality gets worse as soon as the vertex leaves the circumsphere of the tet forming an element of low quality.

In order to present the behavior of *triQ*, figure 25 shows a diagram for *triQ(Vi)* in which the equilateral triangle is sketched. From figure 25a,c one can conclude that as soon as the moved vertex (upper vertex with fixed edge at bottom line of diagram) is leaving its position with ideal quality measure value (*triQ(Vi)*=1), the triangle starts degenerating and might be candidate to topological operations such as edge collapse or split (dependent on the user defined threshold of the quality measure). As an additional observation, figure 25c shows that the quality of the triangle decreases in a lower ratio when angles of the triangle tend to be smaller. On the other hand when the angles of the triangle tend to be bigger, the quality decreases faster. This performance is according to the FE-mesh literature,

where several authors have been exposed that the smaller angles are better for simulations (stiffness matrix conditioning) than the bigger angles.



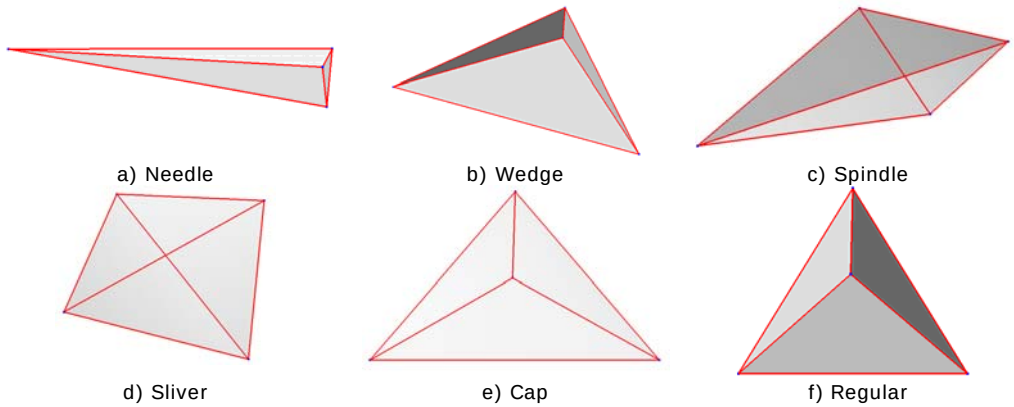
a) Grid representation. b) Wireframe representation. c) Contour representation.

Figure 25: Graphic for the function of *triQ* with the equilateral triangle on the background.

4.2 QUALITY MEASURE BENCHMARKING

In this section is presented a comparison of the results derived from the proposed measure with currently available quality measures in literature [Shewchuk 02], which are summarized in table 1. The proposed quality measure was benchmarked in view of five groups of ill shaped tetrahedra ([Ahlmann 03]).

The group of bad shaped tetrahedra is composed of the so-called tetrahedra: Needle, Wedge, Spindle, Sliver and Cap (figure 26). A tetrahedron Needle is a tetrahedron which has small dihedral angles and some edges are significantly smaller than the other ones (figure 26a). A Wedge tetrahedron is a tetrahedron with one edge much smaller than its others (figure 26b). A tetrahedron which has only one large dihedral angle or one long edge and one small edge is called Spindle (figure 26c). The Sliver tetrahedron has two large dihedral angles and its four vertices are almost forming a plane (figure 26d). A Cap tetrahedron has three large dihedral angles, and therefore one of the vertices is very close to the triangle formed by the other three (figure 26e).



a) Needle b) Wedge c) Spindle

d) Sliver e) Cap f) Regular

Figure 26: Tetrahedra used for the comparison in the table 1.

The results are compared with quality measures described by Shewchuk [Shewchuk 02], which from our point of view, are measures which might be computed on-the-fly. A quality measure that is based on the calculation of eigenvalues of the stiffness matrix causes high overhead during the computation of the solution of the characteristical polynomial. Therefore, it is not useful for on-the-fly evaluations in highly interactive environments with thousands of elements. Quality measures proposed by Shewchunk [Shewchuk 02] need to be pre-processed and post-processed. Those measures have to be adapted in order to

avoid possibly undefined results (pre-processing stage) and the measures have to be scaled afterwards in order to compare its results with a reference tetrahedron of good quality, e.g. the regular tetrahedron. The proposed quality measure is independent of any pre- or post-processing process and does not need any reference tetrahedron/simplex.

The results of the scale-invariant and the size-shape quality measure from Shewchuk can not be interpreted without a comparison. However, the quality measure proposed in this work is bound between 0 and 1, therefore, it is easy to interpret and it allows for working with *heuristics* in order to get a specific quality of a simplicial element.

Given the results of table 1, obtained during benchmarking and comparing these with the results of the different quality measures evaluated for known ill shaped tetrahedra, it is possible to conclude that, the performance of the proposed quality measure behaves towards expectations. As shown in table 1, the quality measure would allow for an earlier identification of ill shaped elements and might prevent inconsistencies in the mesh. E.g. the sliver element has lower values, therefore a lower quality and it will be easily identified in view of topological operations.

Table 1: Benchmarking of the proposed quality measures with quality measure available in the literature for tetrahedral elements.

Tetrahedron Group	**tetQuality (*tetQ*)**	**Scale-invariant Shewchuk**	**Size-Shape Shewchuk**	**Quality Mes. Knupp**	**Quality Mes. Parthasarathy**	**Quality Mes. Cougny**
Needle	**0.0286512**	0.272439 (0.5282)	0.176615 (0.4270)	0.209519	0.0182342	0.0332488
Wedge	**0.1084100**	0.255745 (0.4958)	0.162334 (0.3924)	0.364519	0.2413200	0.0237305
Spindle	**0.0811974**	0.331845 (0.6434)	0.229746 (0.5554)	0.437571	0.0662913	0.0952045
Sliver	**0.0169825**	0.252102 (0.4888)	0.159258 (0.3850)	0.360994	0.1767770	0.0219825
Cap	**0.0286512**	0.168260 (0.3262)	0.092891 (0.2245)	0.209519	0.1000000	0.0025442
Regular	**1.00**	0.515747 (1.0000)	0.413602 (1.0000)	1.00	1.00	1.00

For completion, figure 27 shows a classification of ill-shaped triangles. Those triangles are called, Cap, Needle and a combination of Needle and Cap. A Cap triangle can be identified because the triangle has one angle close to 180° (figure 27a). A Needle triangle has two edges much longer than the other (figure 27c). The Needle-Cap is a combination of the previous two (figure 27b).

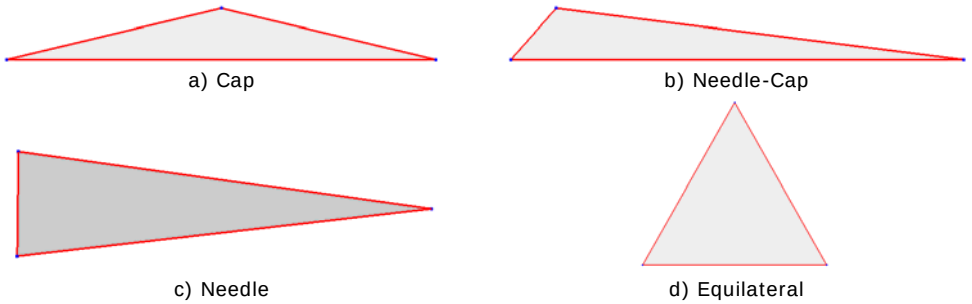


Figure 27: Triangles used for the comparison in the table 2.

The triangles of figure 27 were used to compute the quality measures exposed in the table 2, where the quality measure proposed in this paper is compared with quality measures available in the literature (see above).

Table 2: Benchmarking of the proposed quality measures with quality measure available in the literature for triangular elements.

Tetrahedron Group	triQuality (*triQ*)	Scale-invariant Shewchuk	Scale-invariant smooth Shewchuk	Quality Mes. Bhatia	Quality Mes. Aspect Ratio	Quality Mes. Radius Ratio
Cap	**0.0791896**	0.0674137 (0.4409)	0.224804 (0.3927)	0.246154	0.125000	30.217100
Needle-Cap	**0.0435764**	0.0595504 (0.3894)	0.197970 (0.3458)	0.203424	0.125000	58.313700
Needle	**0.1503380**	0.0832451 (0.5444)	0.280063 (0.4893)	0.342282	0.232877	0.0605143
Equilateral	**1.00**	0.1528980 (1.0000)	0.572357 (1.0000)	1.00	1.00	5.3333300

4.3 MANIPULTION STATISTICS

The following tables (3, 4, 5, and 6) present some statistics results of the topological 3D mesh editing operations, where one can see the evaluation performance in real-time. Since, the topological operations are made locally, the elements and the vertices affected belong only to a part of the mesh (here Surface 1, Surface 2). Table 3 shows the difference between the number of vertices and elements in the initial stage and the final stage having finalized the manipulation of the triangular front/back surfaces (moving a through hole - figure 31) of the domain.

Table 3: Number of affected elements and vertices (surface).

	Num Local Elemts	Num Local Verts
Surface 1		
Initial	252	164
After	276	176
Surface 2		
Initial	738	419
After	752	426

Table 4 shows the number of split and collapse operations performed during the manipulation of the surface mesh. It is not shown the time in this table, because every operation was performed in less than 1 ms.

Table 4: Number of topological operations performed (surface).

Num Split Op	153
Num Coll Op	162

The topological operations applied on tetrahedral meshes have higher complexity because of the multiple topological relations between elements. For example, for a triangular mesh, every edge has a maximum of two incident elements, however for a tetrahedral mesh, an edge has much more. Some of the study examples have up to 16 incident elements to and edge.

This complexity consumes more time in order to find the new topological relation of the elements which will replace the degenerated ones. Though the computation of the quality measure and the appropriate decision (without performing the topological operation) for the topological operation are still performed in less than 1 ms. It was achieve manipulations (quality measure + decision + topological operations) in less than 20 ms.

Table 5 shows primary results for a tetrahedral mesh manipulation.

Table 5: Number of operations and time (volume).

Cycle	Item	Split	Collapse
1	Elements	6	12
	Time	16ms	16ms
2	Elements	3	0
	Time	15ms	0ms
3	Elements	6	12
	Time	16ms	15ms

Table 6 indicates the number of new vertices and elements created during the manipulation.

Table 6: Number affected elements and vertices (volume).

	Num Local Elemts	Num Local Verts
Initial	770	546
After	778	550

4.4 CONCEPTUAL SIMULATION

4.4.1 Feature Detection

The figure 28 shows a model with different colors, the green hole is the highlighting process and the red hole is the selection process.

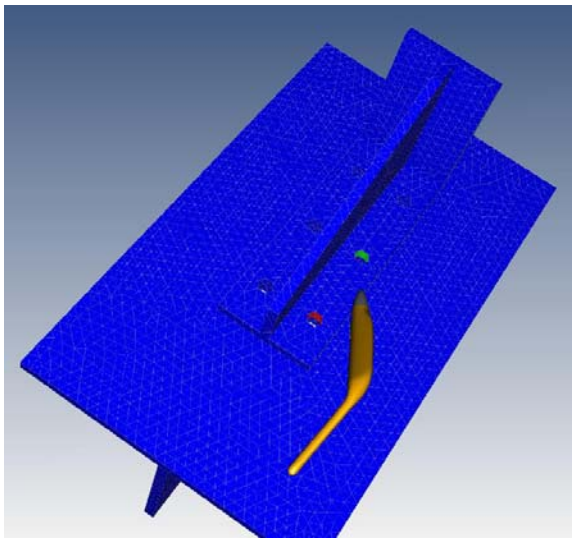


Figure 28: Feature detection.

4.4.2 Face Group Selection

The figure 29 shows an example on how can be selected a group of faces. With this application it is possible to retrieve information of vertex and element labels of the selected group of faces.

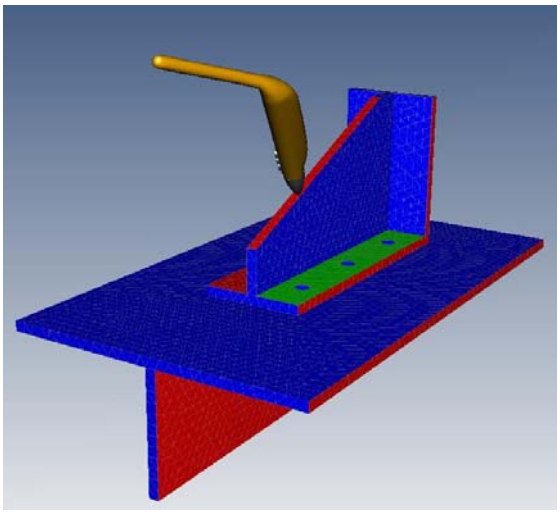


Figure 29: Face group selection.

4.4.3 Feature Movement Application

Here is presented an application which was developed in this context. It aims at interactive, real-time linear static elasticity analysis, in which one can seen the effect of moving features in the domain on the resulting stress field by re-generating a new volume mesh around the area of interest. Figure 30 shows the initial step for the feature movement within a virtual reality system. A through hole is selected and it is started a dragging procedure with a 6DOF interactive input device.

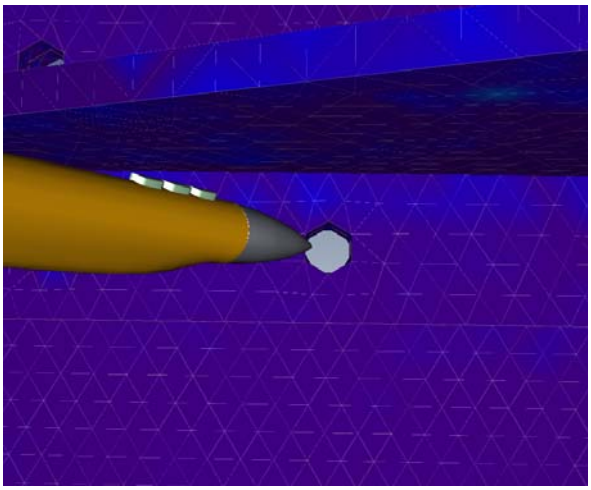


Figure 30: Feature movement, initial step.

Figure 31 shows how the through hole is dragged on the surface and re-meshing starts while dragging.

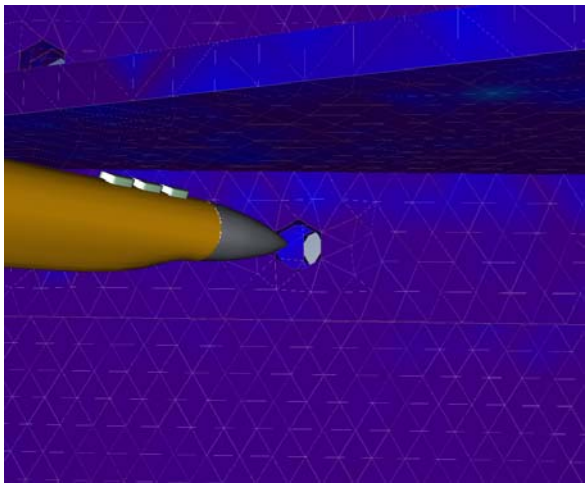


Figure 31: Feature movement, middle step.

Figure 32, the through hole was dragged to the left and the top and the volume mesh has been re-instantiated.

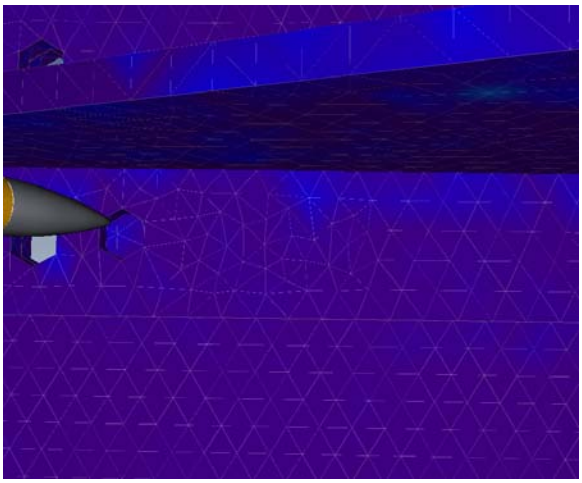


Figure 32: Feature movement, final step.

5. CONCLUSIONS AND FUTURE WORK

Within this work was proposed a new methodology which might help one heading towards conceptual simulations. It aims at establishing a direct link of a virtual environment with a linear static elasticity analysis targeting real-time FE analysis in the context of conceptual simulations. A new quality measure was developed in order to help one in evaluating 3D volume mesh modifications during on-the-fly mesh manipulations such as moving through holes. The proposed quality measure offers the possibility to keep up the quality of the measure while performing topological operations during remeshing tasks. It was shown that the measure has also a good behavior in view of the stiffness matrix which suits as basis for the calculation of the stress field.

The coupling to the simulation was made in order to re-simulate the parts of the model which have been manipulated. The given approach is a submodeling technique which is integrated with the feature manipulation, in order to allow for fast and precise computation of linear static elasticity problems. The technique is totally transparent to the user who can manipulate the mesh and see how the stress field changes in real-time. This approach allows just calculating a subpart of the model avoiding long model preparation and re-simulation running over the whole model.

As a future work, we will study several interaction techniques which can perform mesh modifications in addition to the feature movements. For example, it might interesting to implement a free shape modeling with the aim an adoption of free basis domain. Sketching free form profile surfaces over the domain could allow us to define additional changes to the domain, e.g. rounding off edges, extending beams, etc. We aim at a re-simulation rather than pure animation providing us an immediate feedback on the influence of the modification onto the stress field.

On the other hand, it would be also interesting coupling the feature manipulation with additional physical models like heat transfer or fluid dynamics, a coupling to CAD transformations is a challenge which might offer further space for research activities and which can only be solved by long term research efforts.

6. REFERENCES

[Bischoff 04] Stefan Bischoff. and Leif Kobbelt: Teaching Meshes, Subdivision and Multiresolution Techniques. Computer-Aided Design Vol. 36 (14) pg. 1483-1500. 2004.

[Nealen 05] Andrew Nealen, Matthias Mueller, Richard Keiser, Eddy Boxerman and Mark Carlson: Physically Based Deformable Models in Computer Graphics. State-of-the-Art Report. Eurographics 2005.

[Heckbert 97] Paul S. Heckbert and Michel Garland: Survey of Polygonal Surface Simplification Algorithms. Technical Report, Carnegie-Mellon University, School of Computer Graphics. 1997.

[Alliez 05] Pierre Alliez, Giuliana Ucelli, Craig Gotsman and Marco Attene: Recent Advances in Remeshing of Surfaces. Technical Report, AIM@SHAPE Network of Excellence. 2005.

[Frey 00] Pascal Jean Frey and Paul-Louis George: Mesh Generation. Hermes Science Publishing. ISBN 1-903398-00-2. pg. 553-742. 2000

[Hoppe 93] Hugues Hoppe and Tony DeRose and Tom Duchamp and John McDonald and Werner Stuetzle: Mesh Optimization. Computer Graphics Vol. 27, pg. 19-26. 1993.

[Hoppe 96] Hugues Hoppe: Progressive Mesh. SIGGRAPH '96, pg. 99-108. 1996.

[Popovic 97] Jovan Popovic and Hugues Hoppe: Progressive Simplicial Complexes. SIGGRAPH '97, pg. 217-224. 1997.

[Garland 97] M. Garland and P. Heckbert: Surface Simplification Using Quadric Error Metric. SIGGRAPH '97 Proceedings. 1997.

[Wundrak 06] Wundrak S., Henn T. and Stork A.: Dynamic Progrssive Triangle-Quadrilateral Meshes. WSCG'2006. 2006.

[Biermann 02] H. Biermann and I. Martin and F. Bernardini and D. Zorin: Cut-and-Paste Editing of Multiresolution Surfaces. SIGGRAPH '02 Proceedings, pg 312-321. 2002.

[Suzuki 00] Hiromasa, Suzuki; Yusuke, Sakurai; Takashi, Kanai; Fumihiko, Kimura: Interactive mesh dragging with an adaptive remeshing technique. In: The Visual Computer Publisher: Springer-Verlag GmbH, ISSN: 0178-2789 (Paper) 1432-8726 (Online) Vol. 16, N. 3-4 May 2000. pp: 159 – 176.

[Staadt 98] Staadt O. and Gross M.: Progressive Tetrahedralizations. Proceeding of IEEE Visualization '98. pp. 397-402. 1998.

[Pajarola 99] Renato B. Pajarola and Jarek Rossignac and Andrzej Szymczak: Implant Sprays: Compression of Progressive Tetrahedral Mesh Connectivity. IEEE Visualization '99, pg. 299-306. 1999.

[Kraus 00] Martin Kraus and Thomas Ertl: Simplification of Nonconvex Tetrahedral Meshes. 2000.

[Chopra 02] P. Chopra and J. Meyer: TetFusion: An Algorithm for Rapid Tetrahedral Mesh Simplification. IEEE Visualization '02, pg. 133-140. 2002.

[Shewchuk 02] Shewchuk J.: What is a Good Linear Element? Interpolation, Conditioning and Quality Measures. Proceedings, 11th International Meshing Roundtable , pp.115-126. 2002.

[Ahlmann 03] Alhmann J.: Qualitaets-Metriken und Optimierung von Tetraeder-Netzen. Institut fuer Rechnerentwurf und Fehlertoleranz. Universitaet Karlsruhe. 2003.

[Cheng 03] Cheng S. and Day T.: Quality Meshing with Weighted Delaunay Refinement. SICOMP, Vol 33 Issue 1, pg. 69-93. 2003.

[Uegoer 05] Uegoer A.: Quality Triangulations Made Smaller. EWCG '05. 2005.

[Munson 05] Munson T.: Optimizing the Quality of Mesh Elements. ANL/MSC-P1260-0605. 2005.

[Liu 96] Liu A. and Joe B.: Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. Mathematics of Computation, Vol 65, pg. 1183-1200. 1996.

[Molino 03] Molino N., Bridson R. and Fedkiw R.: Tetrahedral Mesh Generation for Deformable Bodies. SCA '03. 2003.

[Alliez 05] Alliez P., Cohen-Steiner D., Yvinec M. And Desbrun M.: Variational Tetrahedral Meshing. SIGGRAPH '05. 2005.

[Chen 03] Chen J. and Akleman E.: Topologically Robust Mesh Modelling: Concepts, Data Structure and Operations. 2003.

[Lévy 01] Bruno Lévy and Guillaume Caumon and Stéphane Conreaux and Xavier Cavin: Circular Incident Edge List: a Data Structure for Rendering Complex Unstructured Grids. IEEE Visualization '01. 2001.

[De Floriani 03] De Floriani L., Kobbelt L. and Puppo E.: A Survey of Data Structures for Level-Of-Detail Models. 2003.

[Hippold 04] Hippold J. and Ruenger G: A Data Management and Communication Layer for Adaptive, Hexahedral FEM. Lecture Notes in Computer Science, Springer-Verlag GmbH, Vol. 3149, pg. 718. 2004.

[Tobler 06] Tobler R. and Maierhofer S.: A Mesh Data Structure for Rendering and Subdivision. WSCG' 06. 2006.

[Langtangen 03] Hans Peter Langtangen: Computational Partial Differential Equations. Springer-Verlag, Second Edition. 2003. ISBN: 3-540-43416-X.