Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-343

Tetrahedral grid optimisation: towards a structured tetrahedral grid

J.W. van der Burg

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-343

Tetrahedral grid optimisation: towards a structured tetrahedral grid

J.W. van der Burg

This work has been carried out in NLR's basic research programme, Workplan Number A.1.B.2

This report is based on a presentation held on the 7th International Conference on Numerical Grid Generation in Computational Field Simulations, Whistler, Canada, September 2000.

Division: Fluid Dynamics
Issued: July 2000
Classification of title: Unclassified

Summary

In the paper algorithms for optimising a tetrahedral grid are proposed. In order to obtain an optimised tetrahedral grid a new concept is introduced, namely: a structured tetrahedral grid. This grid type fits naturally into the classification of grid types as made in (Ref. 3). The tetrahedral optimisation algorithm consists of two components:

1. An algorithm to optimise the grid connectivity
2. An equi-distribution algorithm to optimise the cell size of the tetrahedral elements.

Contents

(16 pages in total)

1 Introduction

The generation of a tetrahedral grid that is optimal in terms of grid structure and element size forms a challenging task. It is known that state-of-the-art Delaunay-based tetrahedral grid generation algorithms produce an irregular tetrahedral grid that possesses sliver elements and non-uniform grid connectivities.

Several tetrahedral grid optimisation algorithms have been proposed to prevent the occurrence of grid irregularities which arise due to the tetrahedral grid generation algorithm. Among these methods is Rebay's method which is an algorithm that in fact optimises during generation time (Refs. 1, 2, 4). Another algorithm is based on a chrystallic approach that has been proposed in (Ref. 6). A seed element is utilised to achieve a suitable grid structure. In this chrystallic approach the grid structure near boundaries is difficult to maintain.

A common approach adopted in tetrahedral grid generation is to employ a post-processing step in order to optimise the tetrahedral grid. In this post-processing step grid transformations and smoothing iterations are carried out. Requirement during the transformation and smoothing iterations is that the volumes of the elements remain positive. This approach is pursued in this paper. Before addressing the issue how to achieve an optimal tetrahedral grid one firstly should answer the question how an optimal tetrahedral grid is defined. To this purpose in this paper the concept of a structured tetrahedral grid is introduced.

2 A structured tetrahedral grid

In viscous flow-solver algorithms gradient and diffusion operators are usually approximated by means of a central discretisation based on Gaussian integration. As already shown for a prismatic grid (Ref. 3) the principle of central symmetry plays a crucial role for achieving accuracy. For a tetrahedral grid this property implies that for a central node each surrounding node has one associated node that is located opposite and reflected to the central node. In such a way the first-order error terms due to Taylor expansion of the gradient and/or diffusion operator vanish. If one requires in addition that central symmetry should also hold for the all interior nodes in the grid it can be concluded that the most optimal tetrahedral grid is a uniform tetrahedral grid with equi-sided tetrahedral elements with di-hedral angles of 60 degrees and with edge length h. This is in fact a **structured tetrahedral grid** which fits well in the classification of grid types as presented in (Ref. [3]). For a structured tetrahedral grid each interior node is surrounded by 24 equi-lateral tetrahedral elements with edge sizes h. In addition each interior edge in the grid shares 6 equi-sided tetrahedral elements. An example for such an edge is shown in Figure 1. For a uniform grid and a smooth solution on the grid the gradient and diffusion operator are then approximated with second order accuracy.

In general a structured tetrahedral grid cannot be uniform in cell size since a range of gradients of the flow solution have to be resolved. Refinement of a structured tetrahedral grid can be obtained by controlling the size of tetrahedral elements using an externally specified distribution function. Requirement for second-order accuracy is then that the nodes associated to the central node should be located $1 + O(h^2)$ away from their central symmetric position. Another aspect is that the topology of the structured tetrahedral grid possibly cannot be maintained in parts of the flow domain where the distribution function is non-uniform.

3 Transformations to improve the grid connectivity

To obtain a structured tetrahedral grid firstly an overview of possible transformations that improve the grid connectivity is made. In order to explain these transformations extra notation is introduced. Signify e_n as an interior edge of the tetrahedral grid which has n neighbouring tetrahedral elements. In this chapter transformations of edges e_n and e_m occurring in one tetrahedral element are described.

Well-known transformations are the edge-swaps $\mathbf{S}(e_3)$, $\mathbf{S}(e_4)$ and $\mathbf{S}(e_5)$. In Figure 2 an example of the edge-swap $\mathbf{S}(e_3)$ is shown. Three tetrahedral elements are transformed into two tetrahedral elements where the edge e_3 is deleted. The transformation $\mathbf{S}(e_4)$ leads to a new edge e_4 as shown in Figure 3. The edge-swap $S(e_5)$ yields two edges e_4. These transformations can be represented as

$$\begin{aligned} \mathbf{S}(e_3) &\iff \text{2 elements} \\ \mathbf{S}(e_4) &\iff e_4 \\ \mathbf{S}(e_5) &\iff 2e_4 \end{aligned} \tag{1}$$

Consider two non-connected edges e_n and e_m in one tetrahedral element as shown in Figure 4. In a tetrahedral element three combinations of these two edges e_n and e_m exist. Denote $\mathbf{T}(e_n, e_m)$ as the transformation of the tetrahedral element where the first edge is transformed. The following transformations are possible (disregarding transformations with edges having more than 6 neighbouring elements):

$$\begin{aligned} \mathbf{T}(e_3, e_3) &\iff e_4 \\ \mathbf{T}(e_3, e_4) &\iff e_5 \\ \mathbf{T}(e_3, e_5) &\iff e_6 \\ \mathbf{T}(e_4, e_3) &\iff e_5 \\ \mathbf{T}(e_4, e_4) &\iff e_4 + e_5 \\ \mathbf{T}(e_4, e_5) &\iff e_4 + e_6 \\ \mathbf{T}(e_5, e_3) &\iff 3e_4 \end{aligned} \tag{2}$$

The transformations listed are invertible. The listed transformations show that edges e_3 can be removed from the tetrahedral grid without introducing new edges e_3. In contrast for edges e_4 this property does not hold. The transformation of a tetrahedral element having the combination (e_3, e_5) into the construction e_6 is shown in Figure 5. Since the transformations e_4 do not remove edges e_4 from the grid additional transformations are needed.

To describe these transformations consider now two edges e_n and e_m in one tetrahedral element that are connected (both edges share same node). Transformations are:

$$\begin{aligned}
&\mathbf{R}(e_3, e_3) && \text{does not exist} \\
&\mathbf{R}(e_3, e_4) &\Longleftrightarrow\;& e_3 + 2 \text{ elements} \\
&\mathbf{R}(e_4, e_4) &\Longleftrightarrow\;& e_5 \\
&\mathbf{R}(e_4, e_5) &\Longleftrightarrow\;& e_5 + 2 \text{ elements}
\end{aligned} \tag{3}$$

From the transformations (3) it can be observed that the edges e_4 can be transformed into edges e_3 and e_5. By combining these transformations (3) with the transformation $\mathbf{T}(e_3, e_5)$ edges e_6 can be obtained. This means that the grid can be optimised towards a structured tetrahedral grid.
To illustrate the latter transformation consider the tetrahedral construction shown in Figure 6 that contains three edges e_4 that are connected. By using the transformations $\mathbf{R}(e_4, e_4)$ and $\mathbf{T}(e_3, e_5)$ the optimal construction can be obtained. A requirement that complicates the optimisation of the tetrahedral grid is that during a transformation the minimum volume of a tetrahedral element should be maximised. This requirement makes it sometimes not possible to perform a transformation. Therefore an additional algorithm is needed to optimise the cell size of the tetrahedral elements.

4 Improvement of the distribution of tetrahedral elements

To improve the cell size of the elements in the tetrahedral grid, the tetrahedral elements are redistributed by means of a redistribution algorithm. The grid connectivity is kept fixed.

The algorithm that improves the distribution of the tetrahedral elements is based on a minimisation of the function

$$\min \sum_{l} \frac{V_l^2}{w(\underline{x})} \tag{4}$$

with V_l the volume of a tetrahedral element l in the tetrahedral grid and where a weighting function $w = w(\underline{x})$ is taken into account. Typically, the weighting function is taken as the desired spacing at the point $\underline{x}$ to the power six. The solution of the minimisation is an equi-lateral tetrahedral grid that meets the desired spacing function.
The function that is minimised is quadratic in terms of the physical coordinates of each node in the grid, since the volume of a tetrahedral element l can be defined as (see also Figure 7):

$$V_l = \frac{1}{6}(\underline{n}_l, \underline{x} - \underline{x}_1).$$

By differentiating the function (4) for each node in the grid a 3×3 system for the physical coordinates $\underline{x}$ of the node is obtained

$$\sum_{T_l} B_l \, (\underline{x} - \underline{x}_1) = 0, \tag{5}$$

with the symmetric matrix for tetrahedral element T_l

$$B_l = \left[\underline{n}_l \; \underline{n}_l^T\right]$$

For each node in the interior of the tetrahedral grid (hence disregarding boundary nodes) system (5) is solved by adopting a time-integration algorithm

$$\frac{d\underline{x}}{dt} + \sum_{T_l} B_l \left(\underline{x} - \underline{x}_1\right) = 0, \tag{6}$$

Euler-forward time-integration is taken to approximate the time-derivative. For redistributing a tetrahedral grid typically a few number of time steps are employed.

To ensure that the volume elements remain positive during time stepping an upperbound for the explicit time-step is needed. For the volume of a tetrahedral element V_m at the time level $j+1$ the requirement

$$V_m^{j+1} = \frac{1}{6}(\underline{n}_m, \underline{x}^{j+1} - \underline{x}_{m1}) \geq 0$$

should hold. Substitution of (5) with Euler-forward time integration yields

$$V_m^{j+1} = V_m^j - \Delta t \sum_{T_l} V_l^j (\underline{n}_m^j, \underline{n}_l^j) \geq 0$$

which provides the upperbound for the time step:

$$\Delta t \leq \frac{V_m^j}{\sum_{T_l} V_l^j (\underline{n}_m^j, \underline{n}_l^j)}$$

Since this upperbound is expensive to compute (due to the computation of all the inner products) the time step is determined as:

$$\Delta t = \frac{\min_m V_m^j}{\sum_{T_l} V_l^j}$$

The redistribution algorithm based on time-stepping is fast since the time step only depends on the volumes of the tetrahedral elements of the previous time-level, so that they only have to be computed once every time step. As in the grid very badly shaped tetrahedral elements may be present the volume calculation is based on the algorithm described in (Ref. 5).

5 Conclusions

In the paper the concept of a **structured tetrahedral grid** has been introduced. This grid type fits naturally into the classification of grid types as made in (Ref. 3). A structured type of tetrahedral grid combines the advantages of a high level of automation of the grid generation with a high accuracy of numerical discretisation stencils. This type of grid would for instance be beneficial for time-accurate flow calculations for complex geometric configurations.

In the paper algorithms have been proposed that optimise a tetrahedral grid. The connectivity of the tetrahedral grid can be improved by adopting grid transformations. The element size distribution is improved by means of a redistribution algorithm based on a local time stepping method. The redistribution algorithm is fast compared to a standard Laplacian smoothing algorithm since the positiveness of the tetrahedral elements does not have to be inspected. Application of the algorithms described in the paper clearly lead to an optimised tetrahedral grid both in terms of grid structure as well as conforming the desired distribution function. More work is needed to define a strategy using these algorithms for constructing a structured tetrahedral grid.

In addition the algorithms proposed in this paper provide as well a framework for hybrid (prismatic/tetrahedral) grid optimisation and for unstructured moving grids (see for instance Ref. 7) as needed for time-accurate flow calculations.

6 References

1. S. Rebay, ”Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm”, J. of Comp. Phys., Vol. 106, pp. 125-138, 1993.
2. T.J. Baker, ”Point placement and control of triangle quality for inviscid and viscous mesh generation”, In proceedings of conference on Numerical Grid Generation in Computational Fluid Dynamics and related Fields, Pineridge Press, 1994.
3. J.W. van der Burg, K.M.J. de Cock, J.E.J. Maseland, ”Development of a fully automated CFD system for three-dimensional flow simulations based on hybrid prismatic-tetrahedral grids”, in proceedings of the 5th Int. Conference on Numerical Grid Generation in Computational Fluid Dynamics, Starkville, Mississippi, April, 1996.
4. T.J. Baker, J.C. Vassberg, ”Tetrahedral mesh generation and optimisation”, in proceedings of the 6th International Conference on Numerical Grid Generation and Computational Field Simulations, Greenwich, July, 1998.
5. J.W. van der Burg, ”An accurate and robust algorithm for the in-sphere criterion for automated Delaunay-based tetrahedral grid generation”, in proceedings of the 6th International Conference on Numerical Grid Generation and Computational Field Simulations, Greenwich, July, 1998.
6. H. Hakula, ”Voronoi-segment meshes in CFD: The ELMER experience”, in proceedings of the 6th International Conference on Numerical Grid Generation and Computational Field Simulations, Greenwich, July, 1998.
7. K. Riemslagh, J. Vierendeels, E. Dick, ”Two-dimensional incompressible Navier-Stokes calculations in complex-shaped moving domains”, J. of Eng. Math. 34, pp. 57-73, 1998. In: ”Floating, Flowing, Flying: Pieter J. Zandbergen, Life as Innovator, Inspirator and Instigator in Numerical Fluid Dynamics”, Kluwer Academic Publishers.

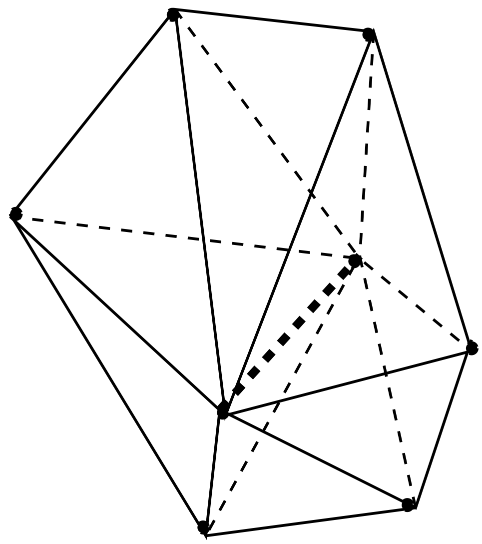

Fig. 1 Example of an edge e_6 (in bold) which has six tetrahedral elements as neighbour

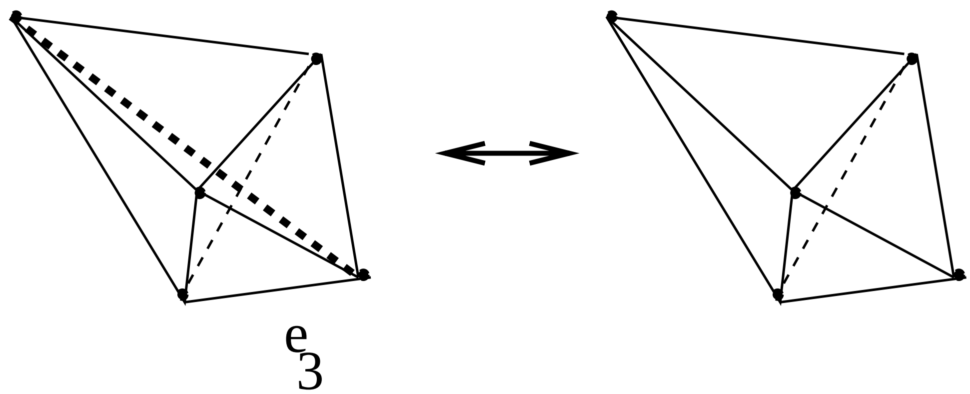

Fig. 2 Example of the edge-swap $\mathbf{S}(e_3)$; edge e_3 (in bold) has three neighbouring elements and is deleted; As a result of the edge-swap two tetrahedral elements remain.

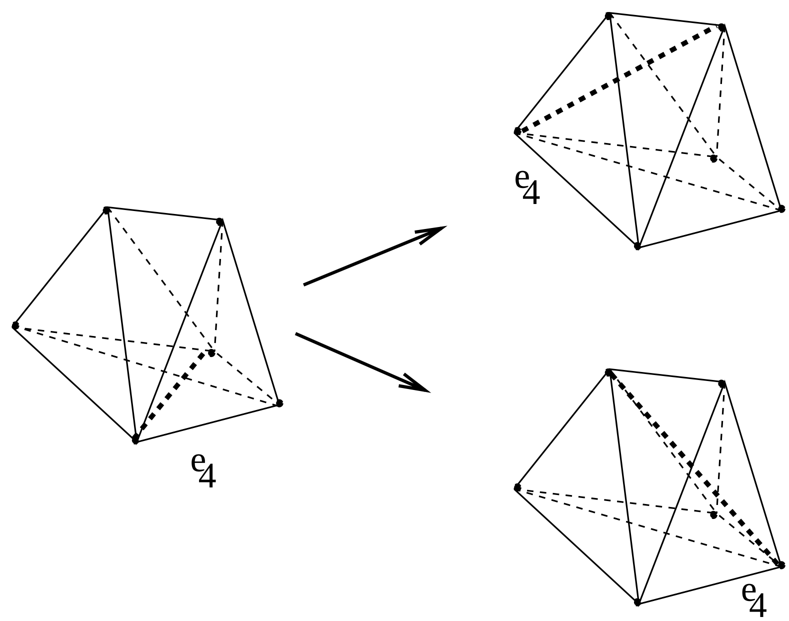

Fig. 3 Examples of the edge-swap $S(e_4)$ into another edge e_4; two possibilities exist

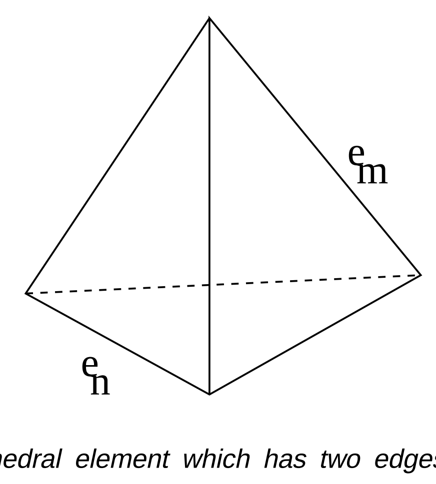

Fig. 4 Example of a tetrahedral element which has two edges e_n and e_m which are non-connected

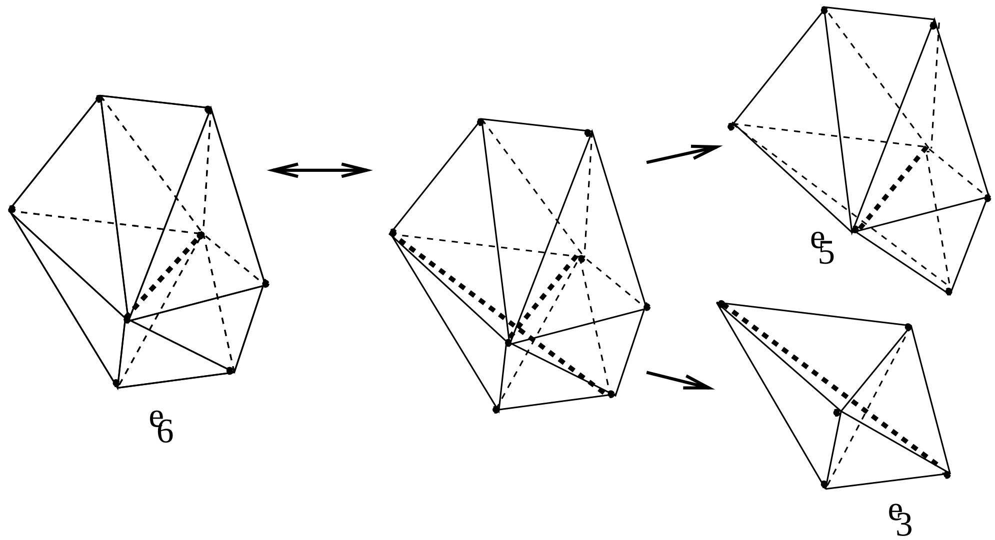

Fig. 5 *Transformation of an edge* e_6 *into two edges edges* e_3 *and* e_5

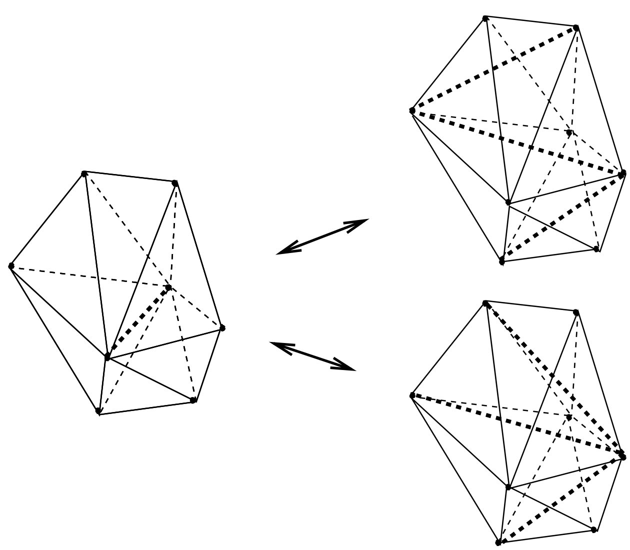

Fig. 6 *Transformation of one edge* e_6 *into three edges* e_4 *each corresponding with four tetrahedral elements; Two possibilities exist.*

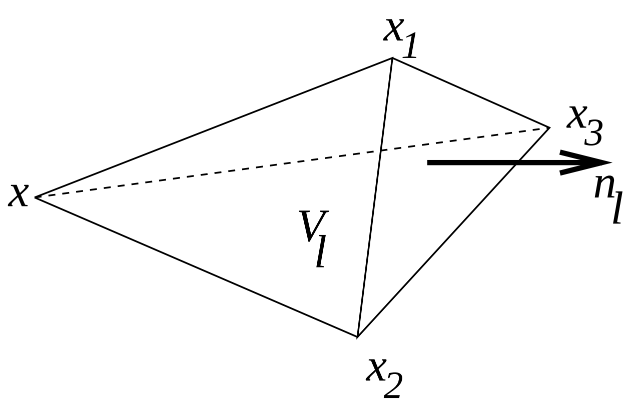

Fig. 7 Definition of the nodes $\underline{x}$, $\underline{x}_1$, $\underline{x}_2$ and $\underline{x}_3$, the normal vector $\underline{n}_l$ of the tetrahedral element l