Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR TP 97241

Discontinous Galerkin finite element method with anisotropic local grid refinement for inviscid compressible flows

J.J.W. van der Vegt and H. van der Ven

DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF.		SECURITY CLASS.
	NLR TP 97241 U		Unclassified

ORIGINATOR
National Aerospace Laboratory NLR, Amsterdam, The Netherlands

TITLE
Discontinous Galerkin finite element method with anisotropic local grid refinement for inviscid compressible flows

PUBLISHED IN
Journal of Computational Physics

AUTHORS	DATE	pp	ref
J.J.W. van der Vegt and H. van der Ven	90425	45	27

DESCRIPTORS

Adaption	Finite element method
Algorithms	Galerkin method
Compressible flow	Hexahedrons
Computational fluid dynamics	Transonic flow
Data structures	Unstructured grids (mathematics)
Euler equations of motion	Wings

ABSTRACT
A new discretization method for the three-dimensional Euler equations of gas dynamics is presented, which is based on the discontinuous Galerkin finite element method. Special attention is paid to an efficient implementation of the discontinuous Galerkin method that minimizes the number of flux calculations, which is generally the most expensive part of the algorithm. In addition a detailed discussion of the truncation error of the presented algorithm is given. The discretization of the Euler equations is combined with anisotropic grid refinement of an unstructured, hexahedron type grid to achieve optimal resolution in areas with shocks, vortices and other localized flow phenomena. The data structure and searching algorithms necessary for efficient calculation on highly irregular grids obtained with local grid refinement are discussed in detail. The method is demonstrated with calculations of transonic flow on the ONERA M6 wing.

Abstract

A new discretization method for the three-dimensional Euler equations of gas dynamics is presented, which is based on the discontinuous Galerkin finite element method. Special attention is paid to an efficient implementation of the discontinuous Galerkin method that minimizes the number of flux calculations, which is generally the most expensive part of the algorithm. In addition a detailed discussion of the truncation error of the presented algorithm is given. The discretization of the Euler equations is combined with anisotropic grid refinement of an unstructured, hexahedron type grid to achieve optimal resolution in areas with shocks, vortices and other localized flow phenomena. The data structure and searching algorithms necessary for efficient calculation on highly irregular grids obtained with local grid refinement are discussed in detail. The method is demonstrated with calculations of the supersonic flow over a 10° ramp and the ONERA M6 wing under transsonic flow conditions.

1 Introduction

The Discontinuous Galerkin (DG) finite element method has some unique features which make it an excellent choice for the solution of the Euler equations of gas dynamics using anisotropic, local grid refinement. Local grid refinement is a very flexible tool to increase grid resolution in regions with complex or non-smooth flow phenomena, but generally results in highly irregular, unstructured grids, which put severe demands on the accuracy and flexibility of the flow solver. The DG finite element method is an extremely local scheme and therefore less sensitive to grid regularity, which makes it a good candidate to be combined with local grid refinement. This paper discusses a new algorithm which extends the discontinuous Galerkin finite element method for the Euler equations of gas dynamics to three dimensions in combination with local grid refinement to improve solution quality. Special emphasis will be put on an efficient implementation and study of discretization error and data structure for the DG finite element method on unstructured grids with hexahedral elements.

The DG finite element method is a mixture of a finite volume and finite element method. It was first proposed by Lesaint and Raviart [13] and extended to hyperbolic conservation laws by Cockburn, Shu et al. [7, 9, 10]. In the DG finite element method the flow field in each element is locally expanded in a polynomial series and equations for the polynomial coefficients are obtained. The DG finite element method therefore not only solves equations for the flow field, but also for the moments of the flow field. No interpolation is necessary to determine the flow state at the element faces in the flux calculation. The information about the flow state at the internal and external element faces can be directly obtained from the polynomial expansion in each element. The only additional information from neighboring elements is the element mean flow state, which is used in the slope limiter. In this way an almost completely local scheme is obtained, which does not lose accuracy on highly irregular grids.

The use of separate equations for the flow gradients in the DG finite element method has as important benefit that it is not necessary to determine the flow gradients from data in neighboring elements. This is commonly done in MUSCL type finite volume methods using Gauss' identity, but this method requires a certain grid regularity which is not required for the DG finite element method. The use of local grid refinement results in hanging nodes, but the DG finite element method does not have any difficulty with hanging nodes because they do not enter the discretization due to the local series expansion of the flow field, which results in a cell based scheme. A significant

benefit of the cell based DG finite element method in comparison with node based finite element methods is that the mass matrix of each element is uncoupled from other elements and it is not necessary to invert a large mass matrix for the complete finite element system. The element based polynomial expansion in the DG finite element method makes it easy to use degenerated hexahedra, such as prisms and tetrahedra. The discontinuous Galerkin method, together with Runge- Kutta time integration, is an excellent candidate for parallel computing due to it's local behavior, as was demonstrated by van der Ven and van der Vegt [24]. A disadvantage of the DG finite element method is that it requires more variables per element, because it is necessary to store several moments of the flow field. The increase in number of variables does not have to be a limitation because grid adaptation will generally reduce the number of elements needed for a given accuracy and therefore reduce the memory requirements significantly.

The DG finite element method has until now primarily been used in two-dimensions. Cockburn and Shu [8] applied the method on triangle based grids, while Lin and Chin [14] and Bey and Oden [5] used quadrilateral elements. The first extension of the DG finite element method to three-dimensional flows was presented by van der Vegt [22] and will be discussed more in detail in this paper. Applications to three-dimensional vortical type flows can be found in van der Vegt and van der Ven [23].

The second topic in this paper is the use of anisotropic grid refinement to improve solution quality. Accurate solutions of three-dimensional flows with highly localized flow phenomena frequently can only be obtained with reasonable efficiency using grid adaptation. Several types of grid adaptation are possible, the most important methods for compressible flow are local grid refinement (h-refinement) and methods which redistribute grid points (r-refinement). One of the main benefits of local grid refinement is that one does not have global constraints on the grid generation. In this paper a new grid adaptation method for the three-dimensional Euler equations of gas dynamics will be discussed.

The numerical method is a combination of local grid refinement of hexahedral elements with the DG finite element method. The grid adaptation is done independently in all three directions to allow for maximum flexibility. Many local flow phenomena, such as shocks and shear layers, are locally pseudo two-dimensional and anisotropic grid refinement is more efficient in these cases than isotropic refinement.

Until now most of the unstructured algorithms for the Euler and Navier-Stokes equations use tetrahedral elements, for a review see [11]. The use of hexahedral, unstructured grids is a more

recent development, e.g. Aftosmis [1]. Hexahedra suffer less from loss of accuracy due to anisotropic refinement than tetrahedra, because the elements do not degenerate after successive refinements in one direction. Hexahedron elements are also more accurate on highly stretched grids which are necessary for applications to viscous flows. In order to deal with complicated geometries, elements such as prisms and tetrahedra are used to deal efficiently with topological degeneracies. An additional benefit of hexahedra is the fact that the initial coarse grid can be provided by standard multi-block grid generators which are widely available.

The data structure for anisotropic h-refinement is more complicated than for unstructured methods without grid refinement. In the present study it is found to be more efficient to replace the commonly used element based octree data structure with a face based data structure. Especially when one does not want to impose restrictions on the number of neighboring elements. The description of this data structure is given special attention in this paper.

The outline of the paper is as follows. First, the Discontinuous Galerkin finite element method will be discussed for the three-dimensional Euler equations of gas dynamics, followed by a study of the discretization error of the DG method presented in this paper. Next, the grid adaptation procedure will be discussed and an overview of the data structure and searching algorithms necessary for anisotropic grid refinement with hexahedral type elements will be given. Finally, the grid adaptation algorithm will be demonstrated with calculations of the supersonic flow about a 10° ramp and with calculations of the ONERA M6 wing under transsonic flow conditions.

2 Governing Equations

The Euler equations for inviscid gas dynamics in conservation form can be expressed as:

$$\frac{\partial}{\partial t}\mathbf{U}(\mathbf{x},t) + \frac{\partial}{\partial x_j}\mathbf{F}_j\left(\mathbf{U}(\mathbf{x},t)\right) = 0, \qquad (\mathbf{x},t) \in \Omega \times (0,T), \tag{1}$$

with initial condition $\mathbf{U}(\mathbf{x},0) = \mathbf{U}_0(\mathbf{x})$, $\mathbf{x} \in \Omega$ and boundary condition $\mathbf{U}(\mathbf{x},t)|_{\partial\Omega} = \mathcal{B}(\mathbf{U},\mathbf{U}_w)$, $(\mathbf{x},t) \in \partial\Omega \times (0,T)$; where $\mathcal{B}$ denotes the boundary operator and $\mathbf{U}_w$ the prescribed boundary data. Here $\Omega \in R^3$ is an open domain with boundary $\partial\Omega \subset \bar{\Omega}$ and $t \in (0,T)$ represents time. The summation convention is used on repeated indices in this paper. The vectors with conserved flow

variables $\mathbf{U} : \bar{\Omega} \times (0,T) \rightarrow R^5$ and fluxes $\mathbf{F}_j$, $j = \{1,2,3\}$; $\mathbf{F}_j : R^5 \rightarrow R^5$, are defined as:

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho E \end{pmatrix}; \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p\delta_{ij} \\ u_j(\rho E + p) \end{pmatrix},$$

where $i = \{1,2,3\}$ and ρ, p and E denote the density, pressure and specific total energy, u_i the velocity component in the Cartesian coordinate directions x_i and δ_{ij} the Kronecker delta symbol. This set of equations is completed with the equation of state: $p = (\gamma - 1)\rho(E - \frac{1}{2}u_i u_i)$, with γ the ratio of specific heats.

3 Discontinuous Galerkin Approximation

The discontinuous Galerkin approximation of the Euler equations is defined by the following steps:

- Suppose the open domain Ω is a polyhedron and denote by $\mathcal{T}_h$ a tessellation of Ω into a disjunct set of polyhedra K_j, $j \in N^+$, such that $\overline{\cup K_j} = \bar{\Omega}$. Each polyhedron K has n faces e_K^i, $i \in N^+$ with $\cup_i e_K^i = \partial K \subset \bar{K}$. Each face e_K^i can connect to multiple faces $e_{K'}^j$. The faces e_K^i are split into sub-faces $s_K^i(K',j) = e_K^i \cap e_{K'}^j$. The faces $s_K^i(K',j)$ therefore always connect to only two neighboring elements in Ω, viz. K and K'. This greatly facilitates the update of the fluxes through element boundaries. The boundary faces $e_K^i \subset \partial\Omega$ are denoted b_K^i. As basic elements hexahedra ($n = 6$) are used, but in order to deal with topologically degenerated cases, hexahedra with degenerated edges, such as prisms and tetrahedra, are allowed when necessary.

- Each of the elements $K_j \in \mathcal{T}_h$ is related to the cubic master element $\hat{K} = [-1,1]^3$, with local coordinates, $\hat{\mathbf{x}} = (\xi,\eta,\zeta)^T$; $\xi,\eta,\zeta \in [-1,1]$, by means of the mapping $F_K : \hat{\mathbf{x}} \in \hat{K} \rightarrow \mathbf{x} \in K$, using the standard linear finite element shape functions:

$$F_K : \mathbf{x}(\xi,\eta,\zeta) = \sum_{i=1}^{m_K} \mathbf{x}_K^i \psi_i(\hat{\mathbf{x}}), \tag{2}$$

  with $\psi_i(\hat{\mathbf{x}})$ trilinear element shape functions and $\mathbf{x}_K^i$ the coordinates of the corner points of the hexahedron K, ($m_K = 8$). More details about the mapping F_K can be found in the appendix.

- Define $P^k(\hat{K})$ as the space of polynomial functions of degree $\leq k$ on the master element $\hat{K}$: $P^k(\hat{K}) = \text{span}\{\hat{\phi}_j, j = 0, \cdots, M\}$. In this paper M is restricted to 3, so the four basis functions $\hat{\phi}_j$ are: $\hat{\phi}_j \in \{1, \xi, \eta, \zeta\}$.

- Define $P^k(K)$ as the space of functions associated to functions in $P^k(\hat{K})$ through the mapping F_K: $P^k(K) = \text{span}\{\phi_j = \hat{\phi}_j \circ F_K^{-1}, j = 0, \cdots, M\}$.

- Define $\mathbf{V}_h^1(K) = \{\mathbf{P}(K) = (p_1, \cdots, p_5)^T \mid p_i \in P^1(K)\}$, then $\mathbf{U}(\mathbf{x}, t)|_K$ can be approximated by $\mathbf{U}_h(\mathbf{x}, t) \in \mathbf{V}_h^1(K) \otimes C^1[0, T]$ as:

$$\mathbf{U}_h(\mathbf{x}, t) \equiv \mathcal{P}\left(\mathbf{U}(\mathbf{x}, t)|_K\right) = \sum_{m=0}^{3} \hat{\mathbf{U}}_m(K, t)\phi_m(\mathbf{x}), \tag{3}$$

  with $\mathcal{P}$ the projection operator to the finite dimensional space $\mathbf{V}_h^1(K)$.

A major difference with standard node based Galerkin finite element methods is that the expansion of $\mathbf{U}(\mathbf{x}, t)$ is local in each element, without any continuity across element boundaries. This has as important benefit that hanging nodes, which frequently appear after h-refinement, do not give any complications because they do not arise in the formulation of the discretization scheme.

A weak formulation of the Euler equations is obtained by multiplying Eq. (1) with $\mathbf{W}_h \in \mathbf{V}_h^1(K)$, integrating over element K using Gauss' identity, and replacing the exact solution $\mathbf{U}$ with its approximation $\mathbf{U}_h \in \mathbf{V}_h^1(K) \otimes C^1[0, T]$:

Find $\mathbf{U}_h \in \mathbf{V}_h^1(K) \otimes C^1[0, T]$, such that $\mathbf{U}_h(\mathbf{x}, 0) = \mathcal{P}\left(\mathbf{U}_0(\mathbf{x})|_K\right) \in \mathbf{V}_h^1(K)$, and for $\forall \mathbf{W}_h \in \mathbf{V}_h^1(K)$:

$$\begin{aligned} \frac{\partial}{\partial t}\int_K \mathbf{W}_h^T(\mathbf{x})\mathbf{U}_h(\mathbf{x}, t)d\Omega = & - \sum_p \int_{s_K^p} \mathbf{W}_h^T(\mathbf{x}) \left(\mathbf{n}^T(\mathbf{x})\mathcal{F}(\mathbf{U}_h)\right) dS \\ & - \sum_p \int_{b_K^p} \mathbf{W}_h^T(\mathbf{x}) \left(\mathbf{n}^T(\mathbf{x})\mathcal{F}(\mathcal{B}(\mathbf{U}_h, \mathbf{U}_w))\right) dS \\ & + \int_K \nabla \mathbf{W}_h^T(\mathbf{x})\mathcal{F}(\mathbf{U}_h)d\Omega, \end{aligned} \tag{4}$$

with $\mathcal{F} = \mathbf{F}_j$, $j = \{1, 2, 3\}$ and $\mathbf{n}$ the unit outward normal vector at the faces s_K^p and b_K^p.

Introducing the polynomial expansions for $\mathbf{U}_h$ and $\mathbf{W}_h$ into the weak formulation of the Euler equations we obtain the following set of equations for the coefficients $\hat{\mathbf{U}}_m$:

$$\begin{aligned}\frac{\partial}{\partial t}\hat{U}_{mi}(K,t)\int_K \phi_n(\mathbf{x})\phi_m(\mathbf{x})d\Omega = & - \sum_p \int_{s_K^p} \phi_n(\mathbf{x})n_j(\mathbf{x})F_{ij}(\mathbf{U}_h)dS \\ & - \sum_p \int_{b_K^p} \phi_n(\mathbf{x})n_j(\mathbf{x})F_{ij}(\mathcal{B}(\mathbf{U}_h,\mathbf{U}_w))dS \\ & + \int_K \frac{\partial \phi_n(\mathbf{x})}{\partial x_j}F_{ij}(\mathbf{U}_h)d\Omega \qquad i\in\{1,\cdots,5\}, \\ & \qquad\qquad\qquad\qquad\qquad\quad n\in\{0,\cdots,3\}, \end{aligned} \tag{5}$$

with F_{ij} the i-th element of flux vector $\mathbf{F}_j$. The integral on the left hand side of Eq. (5) represents the mass matrix $M(K)$ with elements $M_{nm}(K)$, for which an analytic expression is given in the appendix. The relation given by Eq. (5) can be expressed symbolically as:

$$\frac{\partial}{\partial t}\hat{U}_{mi}(K,t) = L_{mi}\left(\mathbf{U}_h\right) \equiv M^{-1}_{nm}R_{ni}(\mathbf{U}_h), \tag{6}$$

where $L_{mi}(\mathbf{U}_h)$ stands symbolically for the spatial nonlinear operator and $R_{ni}(\mathbf{U}_h)$ represents the components of the right hand side of Eq. (5).

3.1 Flux Calculation

Due to the fact that the polynomial basis functions $P^k(K)$ are discontinuous across element boundaries it is necessary to replace the flux at element boundaries with a monotone flux, $\mathbf{H}(\mathbf{U}_h^{int(K)},\mathbf{U}_h^{ext(K)})$, which is consistent, $\mathbf{H}(\mathbf{U},\mathbf{U}) = \mathbf{n}^T\mathcal{F}(\mathbf{U}) \equiv \hat{\mathbf{F}}(\mathbf{U})$, [9]. Here $\mathbf{U}_h^{int(K)}$ and $\mathbf{U}_h^{ext(K)}$ denote the value of $\mathbf{U}_h$ at $\mathbf{x}\in\partial K$ taken as the limit from the interior and exterior of K. The use of a monotone Lipschitz flux $\mathbf{H}$ introduces upwinding into the Galerkin method by solving the (approximate) Riemann problem given by $(\mathbf{U}_h^{int(K)},\mathbf{U}_h^{ext(K)})$. Suitable fluxes are those from Godunov, Roe, Lax-Friedrichs and Osher. In this paper the Osher approximate Riemann solver [16] is used, because of it's good shock capturing capabilities, and the possibility to easily modify the Riemann problem to account for boundary conditions. An important additional reason for the use of the Osher scheme is that it gives an exact solution for a steady contact discontinuity, and therefore has a very low numerical dissipation in boundary layers, [21], which is important for future extension of the algorithm to the Navier-Stokes equations. The Osher approximate Riemann solver is defined as:

$$\mathbf{H}(\mathbf{U}_h^{int(K)},\mathbf{U}_h^{ext(K)}) = \frac{1}{2}\left(\hat{\mathbf{F}}(\mathbf{U}_h^{int(K)}) + \hat{\mathbf{F}}(\mathbf{U}_h^{ext(K)}) - \sum_\alpha \int_{\Gamma_\alpha(\mathbf{U}_h^{int(K)},\mathbf{U}_h^{ext(K)})} |\partial\hat{\mathbf{F}}|d\Gamma\right), \tag{7}$$

where $\cup_\alpha \Gamma_\alpha$ is a path in phase space between $\mathbf{U}_h^{int(K)}$ and $\mathbf{U}_h^{ext(K)}$. Details of the calculation of this path integral in multi-dimensions can be found in [16]. At the boundary faces b_K^p the path Γ_α must be modified to account for boundary conditions $\mathcal{B}(\mathbf{U}, \mathbf{U}_w)$, with $\mathbf{U}_w$ the prescribed boundary data. In this way a Riemann initial-boundary value problem is solved instead of an initial value problem, [16], and a completely unified and consistent treatment of the flux calculations is obtained, both at interior and exterior faces. In the rest of the paper therefore no distinction will be made between flux calculations at internal or boundary faces.

The flux integrals in Eq. (5) can be calculated using Gauss quadrature rules. Cockburn et al. [9] showed that if the quadrature rules for the surface integrals are exact for polynomials of degree $2k+1$ and exact for polynomials of degree $2k$ for the volume integrals then the order of accuracy of the numerical approximation of the flux integrals on the right hand side of Eq. (5) is $k+1$. In order to preserve uniform flow for hexahedral grids with element boundaries which have a twist, it is necessary to use quadrature rules which are exact for polynomials of degree 3. This can be accomplished using four and nine point product Gauss quadrature rules for the element face and volume integrals, respectively. The number of quadrature points can be slightly reduced by using more sophisticated multi-dimensional Gauss quadrature rules, see Stroud [19], but the direct application of the Gauss quadrature rules to the integrals on the right-hand side of Eq. (5) requires a prohibitively large number of flux calculations. This makes the discontinuous Galerkin method unnecessarily expensive when only second order accuracy is required. Recently this problem was also addressed by Atkins and Shu [2], but they restricted themselves to tetrahedral elements. Tetrahedral elements result in significantly easier flux integrals than hexahedral elements, but tetrahedra are not easy to use for anisotropic grid refinement, because successive refinements in one direction create tetrahedra with very small angles between faces resulting in large numerical errors. A second order accurate discontinuous Galerkin discretization can be obtained using the following approximation to the flux integrals at the element boundary face s_K^p:

$$\begin{aligned}\int_{s_K^p} \phi_n(\mathbf{x}) H_i(\mathbf{U}_h^{int(K)}, \mathbf{U}_h^{ext(K)}) dS \;\cong\; & \frac{1}{2}\left(F_{ij}(\bar{\mathbf{U}}_h^{int(K)}) + F_{ij}(\bar{\mathbf{U}}_h^{ext(K)})\right) \int_{s_K^p} \phi_n(\mathbf{x}) n_j(\mathbf{x}) dS - \\ & \frac{1}{2}\left(\sum_\alpha \int_{\Gamma_\alpha(\bar{\mathbf{U}}_h^{int(K)}, \bar{\mathbf{U}}_h^{ext(K)})} |\partial\hat{\mathbf{F}}| d\Gamma\right) \int_{s_K^p} \phi_n(\mathbf{x}) dS, \\ & i \in \{1, \cdots, 5\}, \\ & n \in \{0, \cdots, 3\}, \qquad (8)\end{aligned}$$

with H_i and F_{ij} the elements of the vectors $\mathbf{H}$ and $\mathbf{F}_j$, respectively. The flow states $\bar{\mathbf{U}}_h = \frac{1}{|s_K^p|}\int_{s_K^p} \mathbf{U}_h(\mathbf{x})dS$ in the element face are defined as:

$$\bar{\mathbf{U}}_h^{int(K)} = \frac{1}{|s_K^p|}\sum_{m=0}^{3} \hat{\mathbf{U}}_{m,K}\int_{s_K^p} \phi_{m,K}(\mathbf{x})dS \tag{9}$$

$$\bar{\mathbf{U}}_h^{ext(K)} = \frac{1}{|s_K^p|}\sum_{m=0}^{3} \hat{\mathbf{U}}_{m,K'}\int_{s_K^p} \phi_{m,K'}(\mathbf{x})dS, \tag{10}$$

with K' the index of the element connected to element K at the face s_K^p. The suffices K and K' of $\phi_m(\mathbf{x})$ refer to the limit of $\phi_m(\mathbf{x})$ taken from the interior and exterior of element K at face s_K^p, respectively.

It is important to approximate $\bar{\mathbf{U}}_h$ using the complete series expansion of $\mathbf{U}_h$ given by Eq. (3), because the naive approximation $\bar{\mathbf{U}}_h \cong \mathbf{U}_h(\xi = 0, \eta = 0, \zeta = 0)$ does not result in a second order accurate discretization for elements which are a deformed cube. Simple analytic expressions for the element face moments $\int_{s_K^p} \phi_n(\mathbf{x})n_j(\mathbf{x})dS$ are given in the appendix. The first component ($n = 0$) is the surface area normal vector used in finite volume calculations, whereas the other moments represent cross-products between the element face edges. The integrals $\int_{s_K^p} \phi_n(\mathbf{x})dS$ are calculated using a four point Gauss quadrature rule. With this modification the integration of the fluxes becomes approximately equally expensive as for upwind finite volume schemes using an (approximate) Riemann solver and requires only one flux calculation for each element face.

Another important benefit of using $\bar{\mathbf{U}}_h$ instead of $\mathbf{U}(\xi = 0, \eta = 0, \zeta = 0)$ is that a stronger coupling between the equations for the expansion coefficients is obtained, which significantly improves stability. A detailed discussion of the order of accuracy of the flux discretization is given in the next section.

The volume flux integrals in Eq. (5) can be further evaluated resulting in:

$$\begin{aligned} \int_K \frac{\partial \phi_n(\mathbf{x})}{\partial x_j} F_{ij}(\mathbf{U}_h)d^3x &= 0 && n = 0 \\ &= \int_{\hat{K}} S_j^n(\hat{\mathbf{x}})F_{ij}(\mathbf{U}_h)d^3\hat{x} && n = 1,2,3 \end{aligned} \tag{11}$$

with:

$$\begin{aligned} \mathbf{S}^1(\hat{\mathbf{x}}) &= \mathbf{x}_\eta \times \mathbf{x}_\zeta \\ \mathbf{S}^2(\hat{\mathbf{x}}) &= \mathbf{x}_\zeta \times \mathbf{x}_\xi \\ \mathbf{S}^3(\hat{\mathbf{x}}) &= \mathbf{x}_\xi \times \mathbf{x}_\eta. \end{aligned}$$

here $\mathbf{x}_\xi$, $\mathbf{x}_\eta$ and $\mathbf{x}_\zeta$ denote derivatives of $\mathbf{x}$ with respect to the local coordinates ξ, η and ζ of the master element $\hat{K}$. The volume flux integrals in Eq. (11) are approximated as:

$$\int_{\hat{K}} S_j^n(\hat{\mathbf{x}}) F_{ij}(\mathbf{U}_h) d^3\hat{x} \cong F_{ij}(\bar{\mathbf{U}}_h) \int_{\hat{K}} S_j^n(\hat{\mathbf{x}}) d^3\hat{x}. \tag{12}$$

The geometric contribution $\int_{\hat{K}} S_j^n(\hat{\mathbf{x}}) d^3\hat{x}$ can be calculated analytically and is discussed in the appendix. The flow field $\bar{\mathbf{U}}_h$ for the volume integrals is defined as:

$$\begin{aligned} \bar{\mathbf{U}}_h &= \frac{1}{|K|} \sum_{m=0}^{3} \hat{\mathbf{U}}_m \int_K \phi_m(\mathbf{x}) d^3x \\ &= \frac{1}{M_{0,0}(K)} \sum_{m=0}^{3} \hat{\mathbf{U}}_m M_{m,0}(K), \end{aligned} \tag{13}$$

with $M_{n,m}(K)$ the elements of the mass matrix M_K for element K.

3.2 Slope Limiter

The discretization of the flow field, Eq. (5), does not guarantee a monotone solution without overshoots in areas with discontinuities. Cockburn et al. [9] presented a local projection method for the discontinuous Galerkin discretization of multi-dimensional scalar conservation laws, which makes the algorithm TVB stable and satisfies a maximum principle when combined with a TVD Runge-Kutta time integration method [18]. Cockburn et al. [9] used triangular elements and the extension to quadrilaterals is presented by Bey and Oden [5]. The extension to the Euler equations is usually done with a local characteristic decomposition, but in multiple dimensions this decomposition is only approximate and it is not guaranteed that the limiter satisfies a maximum principle. Therefore a slightly different approach is followed and the multi-dimensional limiter proposed by Barth and Jesperson [4], with modifications due to Venkatakrishnan [25], is used directly on the conservative variables. This limiter saves the considerable expense of computing the local characteristic decomposition.

Define for each component $\bar{U}_{i,K}$, $i = \{1, \cdots, 5\}$, of the element average $\bar{\mathbf{U}}_K = \frac{1}{|K|} \int_K \mathbf{U}_h(\mathbf{x}) d\Omega$:

$$\begin{aligned} U_{i,K}^{\min} &= \min_{\forall K' \in N(K)} (\bar{U}_{i,K}, \bar{U}_{i,K'}) \\ U_{i,K}^{\max} &= \max_{\forall K' \in N(K)} (\bar{U}_{i,K}, \bar{U}_{i,K'}), \end{aligned}$$

with $N(K)$ the set of neighboring elements which satisfy $s_K^p(K', j) \neq \emptyset$, $|K|$ the volume of element K and $\bar{U}_{i,K'}$ the neighboring element averages. In order to maintain monotonicity the approximate

flow field $\mathbf{U}_h$ must satisfy $\mathbf{U}_h(\mathbf{x}) \in [\mathbf{U}_K^{\min}, \mathbf{U}_K^{\max}]$, $\forall \mathbf{x} \in K$, which is accomplished with the limiter functions $\Phi_{i,K}$, $i \in \{1, \cdots, 5\}$:

$$\Phi_{i,K} = \min_{\forall s_K^p \neq \emptyset} \begin{cases} \phi_L\left(\frac{U_{i,K}^{\max} - \bar{U}_{i,K}}{U_{i,K}^* - \bar{U}_{i,K}}\right) & \text{if} \quad U_{i,K}^* - \bar{U}_{i,K} > 0 \\ \phi_L\left(\frac{U_{i,K}^{\min} - \bar{U}_{i,K}}{U_{i,K}^* - \bar{U}_{i,K}}\right) & \text{if} \quad U_{i,K}^* - \bar{U}_{i,K} < 0 \\ 1 & \text{if} \quad U_{i,K}^* - \bar{U}_{i,K} = 0 \end{cases} .$$

Here $U_{i,K}^*$ are the components of $\mathbf{U}_h$ used in the flux calculation at the cell faces $s_K^p(K', j)$. The function $\phi_L(y)$ replaces $\min(1, y)$ in the original Bart and Jesperson limiter and is defined as:

$$\phi_L(y) = \frac{y^2 + 2y}{y^2 + y + 2}.$$

Defining $\triangle = U_{i,K}^* - \bar{U}_{i,K}$, $\triangle_+ = U_{i,K}^{\max} - \bar{U}_K$ and $\triangle_- = U_{i,K}^{\min} - \bar{U}_K$ and replacing $\triangle_\pm^2$ with $\triangle_\pm^2 + \epsilon_m^2$ a smoother limiter, with significantly improved convergence to steady state, is obtained:

$$\Phi_{i,K} = \min_{\forall s_K^p \neq \emptyset} \begin{cases} \frac{\triangle_+^2 + \epsilon_{m,K}^2 + 2\triangle\triangle_+}{\triangle_+^2 + \epsilon_{m,K}^2 + 2\triangle^2 + \triangle\triangle_+} & \text{if} \quad \triangle > 0 \\ \frac{\triangle_-^2 + \epsilon_{m,K}^2 + 2\triangle\triangle_-}{\triangle_-^2 + \epsilon_{m,K}^2 + 2\triangle^2 + \triangle\triangle_-} & \text{if} \quad \triangle < 0 \\ 1 & \text{if} \quad \triangle = 0 \end{cases}$$

The coefficients $\epsilon_{m,K}$ are set equal to $\epsilon_{m,K} = (C\triangle_{m,K})^3$, with $\triangle_{m,K}$ the minimum distance between the element face centers of two opposite faces of element K in the local directions ξ, η or ζ of $\hat{K}$. A close resemblance with the original Barth and Jesperson limiter is obtained if $C = 0$. In this paper $C = 1$ is used, but for cases with strong shocks a slightly smaller value should be used. Large values of C prevent the limiter from being active in smooth parts of the flow field, which improves convergence to steady state and accuracy, but this can result in insufficient limiting in areas with discontinuities. The limiter Φ_K is applied independently to each component of the flow field:

$$\tilde{U}_{mi} = \Phi_{i,K} \hat{U}_{mi} \qquad i = \{1, \cdots, 5\},\ m = \{1, 2, 3\}$$

no summation on i.

The coefficients $\hat{\mathbf{U}}_m, m = \{1, 2, 3\}$ in Eq. (3) represent the gradient of the flow field with respect to the local coordinates in $\hat{K}$. This modification of the local gradient would violate conservation of $\mathbf{U}_h$ in K if the element is not a rectangular cube, which can be corrected by modifying the coefficient $\hat{\mathbf{U}}_0$:

$$\tilde{U}_{0,i} = \hat{U}_{0,i} + \frac{1}{M_{0,0}} \sum_{m=1}^{3} (1 - \Phi_{i,K})\, M_{m,0} \hat{U}_{mi} \qquad i = \{1, \cdots, 5\}$$

no summation on i.

This relation is obtained directly from the condition: $\frac{1}{|K|}\int_K \tilde{\mathbf{U}}_h(\mathbf{x})d\Omega = \bar{\mathbf{U}}_K$. The limiting operation can now be expressed as:

$$\tilde{U}_{mi} = \Pi_{mni}(\mathbf{U}_h)\hat{U}_{ni} \qquad i = \{1,\cdots,5\},\ n,m = \{0,\cdots,3\}$$

no summation on i

with

$$\Pi_{mni}(\mathbf{U}_h) = \begin{pmatrix} 1 & (1-\Phi_i)M_{1,0}/M_{0,0} & (1-\Phi_i)M_{2,0}/M_{0,0} & (1-\Phi_i)M_{3,0}/M_{0,0} \\ 0 & \Phi_i & 0 & 0 \\ 0 & 0 & \Phi_i & 0 \\ 0 & 0 & 0 & \Phi_i \end{pmatrix}$$

The limited flow field $\tilde{\mathbf{U}}_h$ in element K then is equal to:

$$\tilde{\mathbf{U}}_h(\mathbf{x},t) = \sum_{m=0}^{3} \tilde{\mathbf{U}}_m(t)\phi_m(\mathbf{x}). \tag{14}$$

3.3 Time Integration

For each element K a system of ordinary differential equations is now obtained:

$$M_K \frac{\partial}{\partial t}\hat{\mathbf{U}}_K = \mathbf{R}_K(\mathbf{U}_h),$$

with $\hat{\mathbf{U}}_K$ a vector with the moments of the flow field in each element, $\hat{\mathbf{U}}_m, m = \{0,\cdots,3\}$ and $\mathbf{R}_K$ the right-hand side of Eq. (5). The equations for $\frac{\partial}{\partial t}\hat{\mathbf{U}}_K$ are integrated in time using the third order accurate TVD Runge-Kutta scheme from Shu [18] which is directly coupled with the limiting procedure discussed in the previous section:

$$\begin{aligned}
\tilde{U}^{(1)}_{mi}(K) &= \Pi_{mpi}(\mathbf{U}^{(1)}_h)\left(\tilde{U}_{pi}(K,t) + \triangle t(K)M^{-1}_{np}(K)R_n(\tilde{\mathbf{U}}_h(K,t))\right) \\
\tilde{U}^{(2)}_{mi}(K) &= \Pi_{mpi}(\mathbf{U}^{(2)}_h)\left(\frac{3}{4}\tilde{U}_{pi}(K,t) + \frac{1}{4}\tilde{U}^{(1)}_{pi}(K) + \frac{1}{4}\triangle t(K)M^{-1}_{np}(K)R_n(\tilde{\mathbf{U}}^{(1)}_h)\right) \\
\tilde{U}^{(3)}_{mi}(K) &= \Pi_{mpi}(\mathbf{U}^{(3)}_h)\left(\frac{1}{3}\tilde{U}_{pi}(K,t) + \frac{2}{3}\tilde{U}^{(2)}_{pi}(K) + \frac{2}{3}\triangle t(K)M^{-1}_{np}(K)R_n(\tilde{\mathbf{U}}^{(2)}_h)\right)
\end{aligned}$$

$i \in \{1,\cdots,5\}$, no summation on i

$$\tilde{\mathbf{U}}_m(K,t+\triangle t) = \tilde{\mathbf{U}}^{(3)}_m \qquad m,p \in \{0,\cdots,3\}.$$

where the limiting operator Π_{mpi} depends on the unlimited flow field after each Runge-Kutta stage. This Runge-Kutta scheme is stable for CFL numbers less than one, but all calculations are done with a CFL=0.7. The use of TVD Runge-Kutta methods in the time integration is crucial for stability, as was demonstrated by Cockburn et al. [9] and is also experienced during the present calculations. A significant difference of the present cell based finite element discretization in comparison with node based FEM is that the mass matrix M_K of each element is uncoupled from other elements and can be easily inverted because it is only a 4×4 matrix.

For steady state calculations convergence is accelerated using local time stepping. The local time step $\triangle t(K)$ is determined from the relation:

$$\triangle t(K) \leq \frac{-2|K|\mathrm{CFL}}{\sum_{K'=1}^{N(K')} |s_{KK'}| \min_{\alpha}(\hat{u}^{\alpha}_{K'} - |\hat{u}^{\alpha}_{K'}|, \hat{u}^{\alpha}_{K'} \pm c^{\alpha}_{K'} - |\hat{u}^{\alpha}_{K'} \pm c^{\alpha}_{K'}|)}. \tag{15}$$

Here $N(K')$ is the number of element faces $s_{KK'}$ connecting to element K. The symbols $\hat{u}^{\alpha}_{K'}$ and $c^{\alpha}_{K'}$ represent the normal velocity and speed of sound at the end points of each subpath Γ_{α} in phase space, connecting $\bar{\mathbf{U}}_h^{int(K)}$ and $\bar{\mathbf{U}}_h^{ext(K)}$. This information is directly available when calculating the Osher flux at the element faces and does not require any additional work. The use of Eq. (15) to determine the local time step results in a very robust time integration method.

4 Error Estimates for the Flux Approximation

The numerical approximation $\mathcal{L}$ to the nonlinear operator L, defined in Eq. (6), using the approximations to the flux integrals Eqs. (8) and (12), does not satisfy the conditions stated by Cockburn et al. [9] necessary to obtain a second order accurate approximation $\mathcal{L}$ to the operator L. In this section it will be demonstrated that these conditions are overly restrictive and that the numerical approximation $\mathcal{L}$ presented in this paper also results in a second order accurate approximation to L, but with at least four times less flux calculations. In order to obtain an error estimate for $|L - \mathcal{L}|$ the following contributions have to be considered:

- An estimate for the error in the numerical discretization of the surface flux integrals, Eq. (8). This estimate is obtained using a Taylor series expansion with remainder for the flux $\mathbf{F}_j(\mathbf{U}_h(\mathbf{x},t))$ at both sides of S:

$$\left| \int_S F_{ij}(\mathbf{U}_h(\mathbf{x},t))\phi_n(\mathbf{x})n_j(\mathbf{x})dS - F_{ij}(\bar{\mathbf{U}}_h(t)) \int_S \phi_n(\mathbf{x})n_j(\mathbf{x})dS \right| \leq K^1_{ijl}(t) \left| \int_S \triangle U_l(\mathbf{x},t)\phi_n(\mathbf{x})n_j(\mathbf{x})dS \right|,$$

with the constant $K^1_{ijl}(t)$ defined as:

$$K^1_{ijl}(t) = \sup_{\substack{\mathbf{x}\in S \\ D(\theta(\mathbf{U}_h(\mathbf{x},t)))\in(0,1)}} \left| \frac{\partial F_{ij}\left[\bar{\mathbf{U}}_h(t) + \theta(\mathbf{U}_h(\mathbf{x},t))\left(\mathbf{U}_h(\mathbf{x},t) - \bar{\mathbf{U}}_h(t)\right)\right]}{\partial U_l} \right|,$$

$\bar{\mathbf{U}}$ given by Eqs. (9-10) for both sides of the element face and $\triangle U_l(\mathbf{x},t) = U_l(\mathbf{x},t) - \bar{U}_l(\mathbf{x},t)$. The function θ depends on $\mathbf{U}_h$, but has always values in the range $(0,1)$. This error estimate can be further refined using the following relation for $\triangle U_l(\mathbf{x},t)$:

$$\triangle U_l(\mathbf{x},t) = \sum_{m=1}^{3} \hat{U}_{ml}(K,t)\left(\phi_m(\mathbf{x}) - \frac{1}{|S|}\int_S \phi_m(\mathbf{x})dS\right),$$

which is immediately obtained from the series expansion for $\mathbf{U}_h(\mathbf{x},t)$, Eq. (3), the definition of $\bar{\mathbf{U}}(t)$, Eqs. (9-10) and the relation $\phi_0(\mathbf{x}) \equiv 1$, resulting in:

$$\left|\int_S F_{ij}(\mathbf{U}_h(\mathbf{x},t))\phi_n(\mathbf{x})n_j(\mathbf{x})dS - F_{ij}(\bar{\mathbf{U}}_h(t))\int_S \phi_n(\mathbf{x})n_j(\mathbf{x})dS\right| \le K^1_{ijl}(t)\left|\sum_{m=1}^{3}\hat{U}_{ml}(K,t)\right| \cdot \left|\int_S \phi_m(\mathbf{x})\phi_n(\mathbf{x})n_j(\mathbf{x})dS - \frac{1}{|S|}\int_S \phi_m(\mathbf{x})dS\int_S \phi_n(\mathbf{x})n_j(\mathbf{x})dS\right|. \tag{16}$$

The contribution of the surface integrals of the basis functions $\phi_n(\mathbf{x})$ and their product with the normal vector $\mathbf{n}(\mathbf{x})$ can be further evaluated using the following relation:

$$\left|\int_S f(\mathbf{x})dS\right| \le |f(\bar{\mathbf{x}})|\,|S| + \frac{1}{2}\sup_{\substack{\mathbf{x}\in S \\ D(\theta(\mathbf{x}))\in(0,1)}} \left|\frac{\partial^2 f\left[\bar{\mathbf{x}} + \theta(\mathbf{x})\left(\mathbf{x}-\bar{\mathbf{x}}\right)\right]}{\partial x_j \partial x_k}\right| \left|\tilde{M}_{jk}\right|, \tag{17}$$

which is obtained using a Taylor series expansion of $f(\mathbf{x})$ around the center of gravity $\bar{\mathbf{x}}$ of face S. Here $\tilde{M}_{jk}$ and $\bar{\mathbf{x}}$ are defined as:

$$\tilde{M}_{jk} = \int_S x_j x_k dS - \frac{1}{|S|}\int_S x_j dS \int_S x_k dS, \tag{18}$$

$$\bar{\mathbf{x}} = \frac{1}{|S|}\int_S \mathbf{x}dS. \tag{19}$$

The integrals $\tilde{M}_{jk}$ can be estimated using the following assumption:

Assumption 3.1: Each element K satisfies the condition $\left|\hat{\mathbf{x}}^i\right| \le h > 0$, $i \in \{1,\cdots,8\}$.

The coefficients $\hat{\mathbf{x}}^i$ are linear combinations of the position vectors $\mathbf{x}^i$ of the element vertices and are discussed together with the estimates for $\tilde{M}_{jk}$ in the appendix. This assumption

implies that each element can be contained in a cube with maximum dimensions h for all sides.

The error in the numerical approximation of the surface flux integrals can now be estimated as:

$$\left| \int_S F_{ij}(\mathbf{U}_h(\mathbf{x},t))\phi_n(\mathbf{x})n_j(\mathbf{x})dS - F_{ij}(\bar{\mathbf{U}}_h(t)) \int_S \phi_n(\mathbf{x})n_j(\mathbf{x})dS \right|$$

$$\leq \sup_{j\in\{1,\cdots,5\}} K^1_{ijl}(t) \left| \sum_{m=1}^{3} \hat{U}_{ml}(K,t) \right| \left(C_1(\bar{\mathbf{x}})h^4 + C_2(\bar{\mathbf{x}})h^6 \right) \quad (20)$$

where the coefficients C_1 and C_2 only depend on derivatives of ϕ_m and n_j at $\bar{\mathbf{x}}$, but not on $\mathbf{x}$.

- The error estimate for the complete flux integrals of element K is obtained by considering the total flux through ∂K:

$$\int_{\partial K} F_{ij}(\mathbf{U}_h(\mathbf{x},t))\phi_n(\mathbf{x})n_j(\mathbf{x})dS = \sum_{p=1}^{6} \int_{e_K^p} F_{ij}(\mathbf{U}_h(\mathbf{x},t))\phi_n(\mathbf{x})n_j(\mathbf{x})dS,$$

with e_K^p, $p \in \{1,\cdots,6\}$ one of the six faces of a hexahedral element K. The faces are numbered such that e_K^p is opposite to face e_K^{p+1}, see Fig. 1. The normal vector $\mathbf{n}(\mathbf{x})$ at faces e_K^1 and e_K^2 is defined as:

$$\mathbf{n}(\mathbf{x}) = \frac{\mathbf{x}_\eta \times \mathbf{x}_\zeta}{|\mathbf{x}_\eta \times \mathbf{x}_\zeta|}.$$

With similar relations at the other faces. This relation results in an inward pointing normal vector at faces with $p = 1$, 3 or 5, so $\mathbf{n}(\mathbf{x})$ at these faces is replaced with $-\mathbf{n}(\mathbf{x})$ and we obtain the following estimate for the total flux through ∂K:

$$\left| \int_{\partial K} \mathbf{F}_{ij}(\mathbf{U}_h)\phi_n(\mathbf{x})n_j(\mathbf{x})dS - \right.$$

$$\left. \sum_{p=1}^{3} \left(\mathbf{F}_{ij}(\bar{\mathbf{U}}_{h,2p}) \int_{e_K^{2p}} \phi_n(\mathbf{x})n_j(\mathbf{x})dS - \mathbf{F}_{ij}(\bar{\mathbf{U}}_{h,2p-1}) \int_{e_K^{2p-1}} \phi_n(\mathbf{x})n_j(\mathbf{x})dS \right) \right|$$

$$\leq \sup_{j\in\{1,\cdots,5\}} K^1_{ijl}(t) \left| \sum_{m=1}^{3} \hat{U}_{ml}(K,t) \right| \sum_{p=1}^{3} \left(\left| C_1(\bar{\mathbf{x}}_{2p}) - C_1(\bar{\mathbf{x}}_{2p-1}) \right| h^4 + \left| C_2(\bar{\mathbf{x}}_{2p}) - C_2(\bar{\mathbf{x}}_{2p-1}) \right| h^6 \right)$$

$$\leq \sup_{j\in\{1,\cdots,5\}} K^1_{ijl}(t) \left| \sum_{m=1}^{3} \hat{U}_{ml}(K,t) \right| \left(C'h^5 + C''h^7 \right), \quad (21)$$

where the supremum in K^1_{ijl} is taken over all $\mathbf{x} \in \partial K$ and the suffix p refers to the face index. In addition the fact is used that the functions $C_1(\bar{\mathbf{x}})$ and $C_2(\bar{\mathbf{x}})$ are Lipschitz continuous and $|\bar{\mathbf{x}}_{2p} - \bar{\mathbf{x}}_{2p-1}| \leq h$.

- The Osher flux contribution in Eq. (7) can be expressed as:

$$\sum_{\alpha=1}^{5}\int_{\Gamma_\alpha\left(\mathbf{U}_h^{int(K)}(\mathbf{x},t),\mathbf{U}_h^{ext(K)}(\mathbf{x},t)\right)}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma=\sum_{\alpha=1}^{5}\left(\hat{\mathbf{F}}\left(\mathbf{U}_h^{\alpha,2}(\mathbf{x},t)\right)-\hat{\mathbf{F}}\left(\mathbf{U}_h^{\alpha,1}(\mathbf{x},t)\right)\right), \tag{22}$$

with $\mathbf{U}_h^{\alpha,n}(\mathbf{x},t)\in[\mathbf{U}_h^{int(K)}(\mathbf{x},t),\mathbf{U}_h^{ext(K)}(\mathbf{x},t)];n=\{1,2\}$, because the intermediate states $\mathbf{U}_h^{\alpha,n}(\mathbf{x},t)$ are defined using Riemann invariants along the paths Γ_α in phase space. For a detailed definition of the intermediate states in the Osher flux in three dimensions, see [16].

In the smooth part of the flow field the difference between $\mathbf{U}_h^{int(K)}(\mathbf{x},t)$ and $\mathbf{U}_h^{ext(K)}(\mathbf{x},t)$, $\mathbf{x}\in\partial K$ is $O(h^2)$. This follows immediately from the polynomial expansion of $\mathbf{U}_h$, Eq. (3), which gives the following estimate for the intermediate states:

$$\left|\mathbf{U}_h^{\alpha,2}(\mathbf{x},t)-\mathbf{U}_h^{\alpha,1}(\mathbf{x},t)\right|\leq Ch^2,\qquad\forall\mathbf{x}\in\partial K. \tag{23}$$

The above relations, Eqs. (22-23), can be used to obtain the following estimate for the error in the approximation of the integrals of the Osher flux over the element faces in Eq. (8):

$$\left|\int_S\left(\sum_{\alpha=1}^{5}\int_{\Gamma_\alpha}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma\right)\phi_n(\mathbf{x})dS-\left(\sum_{\alpha=1}^{5}\int_{\bar{\Gamma}_\alpha}\left|\partial\hat{\mathbf{F}}(\bar{\mathbf{U}}(t))\right|d\Gamma\right)\int_S\phi_n(\mathbf{x})dS\right|$$
$$\leq\sup_{\mathbf{x}\in S}\sum_{\alpha=1}^{5}\left|\int_{\Gamma_\alpha}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma-\int_{\bar{\Gamma}_\alpha}\left|\partial\hat{\mathbf{F}}(\bar{\mathbf{U}}(t))\right|d\Gamma\right|\left|\int_S\phi_n(\mathbf{x})dS\right|,$$

with: $\Gamma_\alpha=\Gamma_\alpha(\mathbf{U}_h^{int(K)}(\mathbf{x},t),\mathbf{U}_h^{ext(K)}(\mathbf{x},t))$ and $\bar{\Gamma}_\alpha=\Gamma_\alpha(\bar{\mathbf{U}}_h^{int(K)}(t),\bar{\mathbf{U}}_h^{ext(K)}(t))$. The contribution with the difference between the Osher fluxes based on the pointwise data $\mathbf{U}_h(\mathbf{x},t)$ in the element face S and the flux based on the element face averaged data $\bar{\mathbf{U}}_h(t)$ can be estimated as:

$$\left|\int_{\Gamma_\alpha}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma-\int_{\bar{\Gamma}_\alpha}\left|\partial\hat{\mathbf{F}}(\bar{\mathbf{U}}(t))\right|d\Gamma\right|\leq K_{il}^2\sup_{\mathbf{x}\in S}\left|U_l^{\alpha,2}(\mathbf{x},t)-U_l^{\alpha,1}(\mathbf{x},t)-\left(\bar{U}_l^{\alpha,2}(t)-\bar{U}_l^{\alpha,1}(t)\right)\right|, \tag{24}$$

with the coefficient K_{il}^2 defined as: $K_{il}^2=n_jK_{ijl}^1$. This relation is obtained using the representation of the Osher flux given by Eq. (22). The right hand side of Eq. (24) is estimated using Eq. (23), which implies that the difference in intermediate states at the interior and exterior part of the element face are expressed as:

$$\mathbf{U}_h^{\alpha,2}(\mathbf{x},t)-\mathbf{U}_h^{\alpha,1}(\mathbf{x},t)=\triangle\mathbf{U}_h^{\alpha}(\mathbf{x},t)h^2,$$

with $\triangle \mathbf{U}_h^\alpha(\mathbf{x},t)$ a Lipschitz continuous function, which yields the final estimate for the Osher fluxes:

$$\begin{aligned}\left|\int_{\Gamma_\alpha}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma-\int_{\bar{\Gamma}_\alpha}\left|\partial\hat{\mathbf{F}}(\bar{\mathbf{U}}(t))\right|d\Gamma\right| &\leq K_{il}^2\sup_{\mathbf{x}\in S}\left|\triangle U_l^\alpha(\mathbf{x},t)-\triangle\bar{U}_l^\alpha(t)\right|h^2\\ &\leq C'h^3.\end{aligned}$$

The following estimate for the error in the numerical approximation of the surface integrals of the Osher flux is subsequently obtained:

$$\left|\int_S\left(\sum_{\alpha=1}^5\int_{\Gamma_\alpha}\left|\partial\hat{\mathbf{F}}(\mathbf{U}_h(\mathbf{x},t))\right|d\Gamma\right)\phi_n(\mathbf{x})dS-\left(\sum_{\alpha=1}^5\int_{\bar{\Gamma}_\alpha}\left|\partial\hat{\mathbf{F}}(\bar{\mathbf{U}}(t))\right|d\Gamma\right)\int_S\phi_n(\mathbf{x})dS\right|\leq C''h^5, \tag{25}$$

where the estimate for the surface integral of the element face moments:

$$\left|\int_S\phi_n(\mathbf{x})dS\right|\leq 48h^2,$$

is used, which is obtained with the relations for the element face Jacobian and the mapping F_K, discussed in the appendix.

- The error in the numerical approximation of the volume integrals can be obtained in a procedure analogously to that for the flux integrals, but with S replaced by V, and the mean flow state $\bar{\mathbf{U}}_h(t)$ defined by Eq. (13):

$$\begin{aligned}&\left|\int_K\frac{\partial\phi_n(\mathbf{x})}{\partial x_j}F_{ij}(\mathbf{U}_h(\mathbf{x},t))d^3x-F_{ij}(\bar{\mathbf{U}}_h(t))\int_K\frac{\partial\phi_n(\mathbf{x})}{\partial x_j}d^3x\right|\\ &\qquad\leq\sup_{j\in\{1,\cdots,5\}}K_{ijl}^3(t)\left|\sum_{m=1}^3\hat{U}_{ml}(K,t)\right|\left(C_3(\bar{\mathbf{x}})h^5+C_4(\bar{\mathbf{x}})h^7\right)\end{aligned} \tag{26}$$

with the constants $K_{ijl}^3(t)$ defined as:

$$K_{ijl}^3(t)=\sup_{\substack{\mathbf{x}\in K\\ D(\theta(\mathbf{U}_h(\mathbf{x},t)))\in(0,1)\\ \{}}\left|\frac{\partial F_{ij}\left[\bar{\mathbf{U}}_h(t)+\theta(\mathbf{U}_h(\mathbf{x},t))\left(\mathbf{U}_h(\mathbf{x},t)-\bar{\mathbf{U}}_h(t)\right)\right]}{\partial U_l}\right|.$$

The coefficients C_3 and C_4 only depend on derivatives of ϕ_m at $\bar{\mathbf{x}}$, but not on $\mathbf{x}$.

The error estimate for the numerical discretization of the nonlinear operator L_{in} in Eq. (6) using the approximations given by Eqs. (8) and (12) is obtained by combining the results of the estimates given by Eqs. (21), (25) and (26), yielding:

$$\begin{aligned}|L_{ni}-\mathcal{L}_{ni}| &\leq \left|M_{nm}^{-1}\right|\left|\sum_{m=1}^3\hat{U}_{ml}(K,t)\right|C'h^5\\ &\leq \left|\sum_{m=1}^3\hat{U}_{ml}(K,t)\right|C''h^2,\end{aligned}$$

where the estimate for the mass matrix $\left|M_{nm}^{-1}\right| \leq C'''/h^3$ is used, which is discussed in the appendix. The error caused by the numerical approximation of the surface and volume integrals and the Osher flux difference scheme is thus $O(h^2)$, which is of the same order as the error in the polynomial approximation of $\mathbf{U}(\mathbf{x},t)$ by $\mathbf{U}_h(\mathbf{x},t)$ in Eq. (3). A second order accurate spatial discretization which would satisfy the conditions required by Cockburn et al. [9] needs Gaussian quadrature rules with at least four quadrature points and would therefore be at least four time more expensive.

It should be noted that the error estimates showing the second order accurate spatial accuracy of the discontinuous Galerkin discretization does not depend on the smoothness of the grid, demonstrating the fact that an extremely local discretization is obtained, which is especially useful for grid adaptation based on local grid refinement, discussed in the next section.

5 Directional Grid Adaptation

The grid adaptation procedure is based on subdividing elements independently in each of their three local coordinate directions, ξ, η or ζ. A coarse initial grid is used, which is generated with a multi-block structured grid generator. This initial, structured multi-block grid is transferred into an unstructured hexahedral grid, and degenerated hexahedra, such as prisms and tetrahedra, are used when topological degeneracies make this necessary. This grid is called root grid. The root grid can also be generated directly, without first using a block-structured grid, but this is not part of the present paper. After calculating the flow field, elements are split in the local ξ-direction if:

$$\frac{R_K^\xi}{\max\limits_{\forall K \in \mathcal{T}_h} R_K^\xi} > \text{tolerance}, \tag{27}$$

with the sensor function R_K^ξ for element K defined as:

$$R_K^\xi = \max_{i \in \{1,\cdots,6\}, \forall K' \in N^\xi(K)} (V_K^i - V_{K'}^i)^2 \triangle \xi_K^2. \tag{28}$$

Here $\triangle \xi_K$ is the length of element K in the local ξ-direction, and $N^\xi(K)$ the indices of the neighboring elements of element K in the ξ-direction. Equivalent expressions are used for the η and ζ directions. The vector $\mathbf{V}$ has a elements: $\mathbf{V} = (\rho, u_1, u_2, u_3, \gamma M_\infty^2 p, p_{t-loss})^T$, with p_{t-loss} the total pressure loss defined as;

$$p_{t-loss} = 1 - \frac{p}{p_\infty}\left(\frac{1+\frac{\gamma-1}{2}M^2}{1+\frac{\gamma-1}{2}M_\infty^2}\right)^{\frac{\gamma}{\gamma-1}}$$

and $M^2 = u_i u_i / c^2$ the local Mach number with $c = \sqrt{\gamma p/\rho}$ the speed of sound. The suffix ∞ refers to free stream values. These variables are used as adaptation sensor, because they represent all

relevant flow phenomena to be captured by the adaptation process without preference for one or two specific phenomena as is frequently done. The total pressure loss is added as a sensor, because this is a conserved quantity outside shocks in inviscid compressible flow and gives a good measure for the numerical error. The sensor presented in this section is based on the equidistribution principle, see for instance Marchant et al. [15]. It's main advantage is that it prevents discontinuities, such as shocks, from dominating the refinement sensor, because at some point the element length in these regions becomes so small that other flow features will start to become important.

Each element is adapted independently in all three directions, by dividing the elements which meet the adaptation criterion into two new elements.

6 Data Structure

The success of an unstructured grid adaptation algorithm strongly depends on the efficiency of the data structure. The data structure for h-type grid adaptation is more complicated than for r-type adaptation, because one element can be connected to multiple neighboring elements. An important criterion in the design of the data structure is that no searching is required in the calculation of the flow field. All the necessary searching to update the data structure is done during the adaptation step. This greatly enhances the efficiency of the code, because all the basic operations then can be vectorized and parallelized using a proper coloring and domain decomposition scheme. Until now, most of the applications with local refinement of hexahedron type elements presented in the literature where restricted to two-dimensional flows, where generally a quadtree data structure is used. In three dimensions this becomes an octree data structure. An octree data structure is, however, more suited for isotropic element refinement, where each element has eight children, but is inefficient for anisotropic grid refinement.

An efficient data structure for the DG finite element method is obtained using the element faces instead of the elements as the basic component. This has several major advantages. The primary loop in a DG finite element method is the calculation of the fluxes, which can be done without any searching using a face based data structure. A second benefit of a face based data structure versus an element based data structure is that each sub-face $s_K^i(K', j)$ can only have two neighboring elements, whereas each element can have an unlimited number of neighbors. A loop over element faces can therefore be done without searching using a face based data structure. The face based data structure has some resemblance with the edge based data structure commonly used

with vertex based unstructured algorithms using tetrahedra.

6.1 Grid Structure

Each element K is related to its master element $\hat{K}$ with the mapping F_K, Eq. (2). The faces and vertices of element $\hat{K}$ are numbered uniquely, see Fig. 1, and the topology of each element K is defined by the coordinates of the vertices and the mapping F_K. The following arrays are used to define the grid structure: Array $ICG(icell, n), (n = 1, \cdots, 8)$ to store the addresses of the vertices of the elements and array $IcTree(icell, n), (n = 1, \cdots, 4)$ to store the element connectivity. The first element of $IcTree$ is the address of the parent element and the second and third element are the addresses of the first and second child. For efficiency reasons also the type of refinement (ξ, η or ζ direction) is stored.

Due to the dynamic behavior of the grid, points are added and deleted, it is important to store the grid points efficiently. This is done using an AVL-tree data structure. For a detailed description of AVL trees see [12] and [26]. The array IG_{AVL} contains the addresses of the x, y and z-coordinates of the grid points. The AVL-tree uses the same key as proposed in [20], viz. $(x_1, y_1, z_1) < (x_2, y_2, z_2)$ if $x_1 < x_2$, or if $x_1 = x_2$ and $y_1 < y_2$ or if $x_1 = x_2$ and $y_1 = y_2$ and $z_1 < z_2$.

Together with vectors for the x, y and z-coordinates of the grid points this information is sufficient to describe the grid. The use of an AVL-tree is very efficient. When a element is divided it is possible to find in $O(log_2(N))$ steps if a grid point already exists in the tree or must be added. Both insertion and deletion of an element in the AVL tree can be done in $O(log_2(N))$ operations, with N the number of grid points.

6.2 Establishing Face to Element Connectivity

The most difficult part of h-type grid adaptation on an unstructured hexahedral mesh is to establish the face to element connectivity $s^i_K(K', j)$. It is impractical to try to determine in advance the large number of possible connections, even if only a limited number of neighboring elements is allowed. The following algorithm can find all possible connections:

At the root grid level all element connections are known, because they can be obtained from the original unadapted grid. At this level there is no local grid refinement.

For all root element faces the addresses and face indices of the two elements which connect to this element face are stored in the array $IfTree$. Next, the tree $IcTree$ is traversed. For each element face which is the connection between the two children elements K' and K'', $(s^i_{K'}(K'', j) =$

$e^i_{K'} \wedge s^i_{K'}(K'', j) = e^j_{K''}$), the addresses and face indices of the left and right children are also stored in array $IfTree$. The set of these faces and the root element faces are called elementary faces.

To find the remaining face to element connections each elementary face is mapped to the domain $[0,1] \times [0,1]$, with local (s,t) coordinates. Then for each side of the elementary face the tree $IcTree$ is traversed to find the local (s,t) coordinates of the four corners and center of the element faces of the children elements which connect to the elementary face. This can be done easily using the type of refinement, (ξ, η or ζ direction), stored in array $IcTree$ and the face index of the elementary element which is the same for all kids. If necessary the local coordinate system (s', t') of element face $e^j_{K'}$ is transformed to the (s,t) coordinate system of element face e^i_k.

The coordinates of the corner points and element face centers at both sides of the elementary face are stored in arrays $FaceKeyL$ and $FaceKeyR$. For both sides of the element face also the addresses of the children are stored in separate binary trees $IfTreeL$ and $IfTreeR$, using the element face center as key. This part of the algorithm has some similarity to that proposed in [20] to find hanging nodes in a node based finite element method. Their problem is a point search problem, but the determination of the face to element connectivity is a geometric searching problem and in this paper the alternating digital tree algorithm is used, [6].

First, for all the elements on the left side of the element face, the tree $IfTreeR$ is traversed to find the element face at the opposite side which has the same corner points or is completely contained in the left element face. This can be done in $O(\log_2(N))$ operations. The same is done for all the elements at the right element face. In order to efficiently eliminate face to element connections which occur twice, it is necessary to store the new face to element connections in a binary tree.

After this search most face to element connections are found, but depending on the refinement strategy it is possible that one element face connects at both sides to more than one element, Fig. 2. If this happens it's face to element connection is not established in the previous search and the element faces for which no connection can be found must be split into two faces on one of the sides of the elementary face, Fig. 3. These faces are called sub-faces. By cyclically splitting the element faces for which no connection can be found on one side in the local s and t directions and restarting the search for those faces for which no connection was established finally all connections will be found. It is easy to test if all element to face connections are found because their area should add up to one on both sides. After the search is completed, redundant sub-faces are merged and all connections are added to the tree $IfTree$.

The alternative to subdivision of element faces would be to further subdivide elements, but this can easily generate new faces which connect to more than one element. This does not occur with subdivision of element faces and the searching algorithm will finish in finite time. The only complication of using sub-faces is that they have to be accounted for in the flux calculation, because now the face e_K^i is subdivided into several faces instead of one. The corrections to the surface integrals of the fluxes are discussed in the appendix. With this algorithm all face to element connections are found and the algorithm can be parallelized completely, because the determination of the subdivision of each elementary face is completely independent from one another.

The calculation of the element face fluxes can be done easily in one loop over the element faces, without any difficulty caused by hanging nodes. This algorithm can be completely vectorized and parallelized using a proper coloring and domain decomposition scheme. For more details, see van der Ven and van der Vegt [24].

7 Discussion and Results

The Discontinuous Galerkin discretization of the Euler equations of gas dynamics and the grid adaptation algorithm have been tested on two cases. The first case is the supersonic flow about a 10° ramp, which serves as a simple two-dimensional example to demonstrate the grid adaptation algorithm. The second case is the transsonic flow about the ONERA M6 wing [3, 27], which is a more complicated three-dimensional flow. The supersonic flow field about a 10° ramp generates an oblique shock with a 39.314° angle with respect to the flow direction. A nice feature of this problem is that it can be easily compared with the exact solution for an oblique shock using the Rankine Hugoniot relations. The problem is also a good test case for the grid adaptation algorithm, because the shock is not lined up with a grid line. The initial grid is uniform and consists of 600 elements and during each adaptation step, first the 5% of elements with the lowest values of the sensor function are deleted if they are not a root element, and subsequently the 20% of elements with the highest values of the sensor function are refined, independently in each direction. Table I gives an overview of the number of elements and grid points after each adaptation step. A detailed view of the final adapted grid is presented in Figure 4, which shows that the grid is well adapted to the shock. An interesting feature is that there is no adaptation ahead of the ramp because the flow field is uniformly supersonic. The pressure field over the ramp is shown in Figure 5, which shows that the adaptation significantly improves the capturing of the shock and produces a nearly

monotone shock profile. The value of the pressure behind the shock, viz. $p_2 = 0.304$ compares well with the exact value $p_2 = 0.304746$. Here the pressure is made dimensionless as $p = p^*/(\gamma M_\infty^2)$, with γ the ratio of specific heats ($\gamma = 1.4$) and M_∞ the free stream Mach number. Figure 5 also shows the grid points along the ramp in the final adapted grid.

The second test case is the ONERA M6 wing which has a trapezoidal planform with 30° leading edge sweep, and a taper ratio of 0.56. The wing sections are based on the symmetrical ONERA-D profile with 5% thickness/chord ratio. The wing tip is rounded by rotating the tip section around its symmetry axis. The free stream Mach number is 0.84 and angle of attack is 3.06°.

The grid adaptation was started by first calculating a steady solution on the initial grid, which consists of 131072 elements and 137425 grid points. The grid is subsequently adapted three times, independently in all three directions and the final grid consists of 339226 elements and 398356 grid points. See Table II for more details. This adaptation process is completely controlled by the adaptation sensor. The only user interaction is the specification of the increase in number of elements during each adaptation step, which is done before the simulation started.

All calculations are done with a local CFL number of 0.7. Fig. 6 shows the convergence history of the L_2 residual. The spikes indicate the various instances when the grid is adapted. It can be seen that convergence is relatively slow, because local time stepping is the only technique used to accelerate convergence. The implementation of a multigrid algoritm to speed up convergence is currently in progress. One of the main factors influencing convergence is the activity of the slope limiter in the far field, for an analysis of this problem see [25]. The Venkatakrishnan modifications to the Bart and Jesperson limiter significantly improve convergence, but can still be improved upon. Grid adaptation generally has a positive influence on convergence as can be seen in Fig. 6.

The time history of the lift force C_L is presented in Fig. 7. The final values $C_L = 0.290$ and $C_D = 0.0136$ are very close to the results obtained in literature, e.g. [27].

The use of the sensor functions R_K, Eqs. (27) and (28), which approximate the gradient of the primitive flow variables in all three directions, is effective in capturing the relevant flow features. Generally the most dominant feature for adaptation is the stagnation region, especially on the initial coarse grid, but shocks and shear layers are being captured well after refinement. An important feature of the sensor function is that it is weighted with the local grid distance, which prevents one aspect of the flow to constantly dominate the adaptation process. This is strongly influenced by the power of $\triangle\xi_K$ in Eq. (28).

Fig. 8 shows the final adapted grid which clearly shows the lambda shock structure. The

mesh adapts to regions with large flow activity and significantly improves resolution in the shock regions and around the tip. Fig. 8 shows that the two shocks merge at 87% span and separate at approximately 94% span. The shock structure compares well with the results obtained by Rausch et al. [17]. For efficient adaptation it proved to very important to be able to both add and delete elements, because initially the grid is primarily refined in the stagnation and rear shock regions which tend to become overresolved in the initial adaptation stages. The position of the shocks also significantly changes during the adaptation process when the flow field becomes better resolved. The shock sensor is, however, qualitative and further improvements in sensor functions based on some estimate of the numerical error will contribute to improved efficiency in the grid adaptation process.

The pressure coefficient C_P for the initial grid and the three adapted solutions in cross-sections at $y = 0.20S$, $0.44S$, $0.65S$, $0.80S$ and $0.90S$, with S the wing span, are presented in Figs. 9 to 13. Also the experimental data from [3] are presented. The pressure coefficient is defined as $C_P = \frac{p-p_\infty}{\frac{1}{2}\rho V_\infty^2}$, with V_∞ the free stream velocity. The correlation with the experiments is good, especially considering the fact that the calculations are inviscid. The improvements due to the adaptation are very clear, especially in resolving the inviscid shock structure, and the adaptation process clearly converges to a final solution.

The calculations are done on the NEC SX-4/16 computer at NLR and required approximately 5 hours for the ONERA M6 wing. The flow solution part of the program runs approximately at a speed of 4.4 Gflops on seven processors, which is 31% of the peak speed with seven processors. More details about the performance and parallelization strategy will be presented elsewhere.

8 Concluding Remarks

The extension of the discontinuous Galerkin method using hexahedron type elements to three dimensional inviscid, compressible flow has been successfully demonstrated. An efficient technique for the flux calculations is presented and it is shown that the DG finite element method can be nicely combined with anisotropic grid adaptation, which significantly improved accuracy. A new algorithm to establish face to element connectivity is presented which works well with h-refinement of hexahedral elements and the DG finite element method. Results of supersonic flow about a 10° ramp and transsonic flow about the ONERA M6 wing are presented, which demonstrate the efficiency of the adaptation algorithm in capturing the lambda shock wave and resolving localized

flow phenomena. The DG finite element method is a very local scheme which works well on highly irregular grids and reaches a high efficiency on a parallel vector computer. Future work will especially concentrate on improving convergence using a multigrid technique.

9 Acknowledgement

The authors would like to thank Dr. B. Oskam for his advice and continued support during the course of this project. Thanks are also due to G.J.D. Zondervan and W.J. Piers for their assistance in developing the post-processing software.

References

[1] M.J. Aftosmis, Upwind method for simulation of viscous flow on adaptively refined meshes, *AIAA J.* **32**, 268 (1994).

[2] H.L. Atkins and C.-W. Shu, Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations, ICASE Report, No. 96-51, 1996 (unpublished).

[3] J. Barche, (Ed.), Experimental data base for computer program assessment, AGARD AR-138 (1979).

[4] T.J. Barth and D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA Paper 89-0366, 1989 (unpublished).

[5] K.S. Bey and J.T. Oden, A Runge-Kutta discontinuous finite element method for high speed flows, AIAA Paper 91-1575-CP, 1991 (unpublished).

[6] J. Bonet and J. Peraire, An alternating digital tree (ADT) algorithm for 3D geometric searching and intersection problems, *Int. J. for Num. Meth. in Eng.* **31**, 1 (1991).

[7] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comput.* **52**, 411 (1989).

[8] B. Cockburn and C.-W. Shu, The P^1-RKDG Method for two-dimensional Euler equations of gas dynamics, ICASE Report No. 91-32, 1991 (unpublished).

[9] B. Cockburn, S. Hou and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, *Math. Comput.* **54**, 545 (1990).

[10] B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, *J. Comput. Phys.* **84**, 90 (1989).

[11] H. Deconinck and T.J. Barth, Special course on unstructured grid methods for advection dominated flows, AGARD Report 787 (1992).

[12] D.E. Knuth, *The Art of Computer Programming, Volume 3, Sorting and Searching* (Addison-Wesley, 1973).

[13] P. Lesaint and P.A. Raviart, On a finite element method for solving the neutron transport problem, in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, edited by C. de Boor (Academic Press, 1974), p. 89.

[14] S.-Y. Lin and Y.-S. Chin, Discontinuous Galerkin finite element method for Euler and Navier-Stokes equations, *AIAA J.* **31**, 2016 (1993).

[15] M.J. Marchant and N.P. Weatherhill, Adaptivity techniques for compressible inviscid flows, *Comp. Meth. in Appl. Mech. and Eng.* **106**, 83 (1993).

[16] S. Osher and S. Chakravarthy, Upwind schemes and boundary conditions with applications to Euler equations in general geometries, *J. Comput. Phys.* **50**, 447 (1983).

[17] R.D. Rausch, J.T. Batina and H.T.Y. Yang, Spatial adaptation procedures on tetrahedral meshes for unsteady aerodynamic flow calculations, AIAA Paper 93-0670, 1993 (unpublished).

[18] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* **77**, 439 (1988).

[19] A.H. Stroud, *Approximate Calculation of Multiple Integrals* (Prentice-Hall, 1971).

[20] Z. Tan and P.L. Varghese, Directionally adaptive finite element method for multidimensional Euler and Navier-Stokes equations, in *Proc. 11th AIAA CFD Conference, Orlando, Florida, 1993*, AIAA Paper 93-3320-CP.

[21] J.J.W. van der Vegt, Higher-order accurate Osher schemes with application to compressible boundary layer stability, AIAA Paper 93-3051, 1993 (unpublished).

[22] J.J.W. van der Vegt, Anisotropic grid refinement using an unstructured discontinuous Galerkin method for the tree-dimensional Euler equations of gas dynamics, in *Proc. 12th AIAA CFD Conference, San Diego, California, 1995*, AIAA Paper 95-1657.

[23] J.J.W. van der Vegt and H. van der Ven, Hexahedron based grid adaptation for future large eddy simulation, in *Proc. Progress and Challenges in CFD Methods and Algorithms, Seville, Spain, 1995*, AGARD CP-578, p. 22-1.

[24] H. van der Ven and J.J.W. van der Vegt, Experiences with advanced CFD algorithms on NEC SX-4, in, *Proc. 2nd International Meeting on Vector and Parallel Processing, Porto, Portugal, 1996.*

[25] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, *J. Comput. Phys.* **118**, 120 (1995).

[26] N. Wirth, *Algorithms & Data Structures* (Prentice-Hall, 1986).

[27] H. Yoshihara and P. Sacher, (Eds.), Test cases for inviscid flow field methods, AGARD AR-211 (1985).

A Appendix: Analytic Expressions for Metrical Coefficients

The calculation of the geometric integrals which appear in the discontinuous Galerkin finite element discretization can be done numerically with a Gauss quadrature rule of sufficient order or analytically. The use of Gauss quadrature rules is straightforward, but computationally expensive. In this appendix analytic expressions are given which require significantly less computational work than the use of quadrature rules. The calculation of the integrals in the discontinuous Galerkin finite element discretization is greatly simplified by expressing the mapping F_K for hexahedral elements, Eq. (2), as:

$$F_K : \mathbf{x}(\xi,\eta,\zeta) = \hat{\mathbf{x}}_K^1 + \hat{\mathbf{x}}_K^2\xi + \hat{\mathbf{x}}_K^3\eta + \hat{\mathbf{x}}_K^4\zeta + \hat{\mathbf{x}}_K^5\xi\eta + \hat{\mathbf{x}}_K^6\xi\zeta + \hat{\mathbf{x}}_K^7\eta\zeta + \hat{\mathbf{x}}_K^8\xi\eta\zeta. \tag{29}$$

The position of the element vertices $\mathbf{x}_K^n$ is indicated in Fig. 1. The coefficients $\hat{\mathbf{x}}_K^n = (\hat{x}_K^n, \hat{y}_K^n, \hat{z}_K^n)^T$ are obtained from the relation:

$$(\hat{x}_K^1, \cdots, \hat{x}_K^8)^T = A(x_K^1, \cdots, x_K^8)^T \tag{30}$$

with the matrix A defined as:

$$A = \frac{1}{8}\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & 1 & -1 & 1 & -1 & -1 & 1 \end{pmatrix}$$

with identical relations for $\hat{y}_K^n$ and $\hat{z}_K^n$, with x in Eq. (30) replaced by y and z, respectively.

A.1 Mass Matrix

An important component in both the calculation of the mass matrix and the volume integrals in Eq. (5) is the Jacobian J_{F_K} of the mapping F_K. The Jacobian J_{F_K} for hexahedral elements can be expressed as:

$$J_{F_K} \equiv Det\left|\frac{\partial(x,y,z)}{\partial(\xi,\eta,\zeta)}\right| = \sum_{k=0}^{2}\sum_{j=0}^{2}\sum_{i=0}^{2} b_{ijk}\xi^i\eta^j\zeta^k \tag{31}$$

Here Det denotes the determinant of a matrix. The non-zero coefficients b_{ijk} are defined as:

$$\begin{array}{lll}
b_{0,0,0} = D_{2,3,4} & b_{1,2,0} = D_{5,3,8} & b_{0,2,1} = D_{8,3,7} \\
b_{1,0,0} = D_{2,3,6} + D_{2,5,4} & b_{0,0,1} = D_{2,7,4} + D_{6,3,4} & b_{1,2,1} = D_{8,5,7} \\
b_{2,0,0} = D_{2,5,6} & b_{1,0,1} = D_{2,8,4} + D_{2,7,6} + D_{6,5,4} & b_{0,0,2} = D_{6,7,4} \\
b_{0,1,0} = D_{2,3,7} + D_{5,3,4} & b_{2,0,1} = D_{2,8,6} & b_{1,0,2} = D_{6,8,4} \\
b_{1,1,0} = D_{2,3,8} + D_{2,5,7} + D_{5,3,6} & b_{0,1,1} = D_{5,7,4} + D_{6,3,7} + D_{8,3,4} & b_{0,1,2} = D_{8,7,4} \\
b_{2,1,0} = D_{2,5,8} & b_{1,1,1} = 2D_{7,6,5} & b_{1,1,2} = D_{8,7,6} \\
b_{0,2,0} = D_{5,3,7} & b_{2,1,1} = D_{5,8,6} &
\end{array} \tag{32}$$

with:

$$D_{ijk} = Det(\hat{\mathbf{x}}^i, \hat{\mathbf{x}}^j, \hat{\mathbf{x}}^k) \tag{33}$$

The mass matrix $M_{nm}(K)$ is now equal to:

$$\begin{aligned}
M_{nm}(K) &= \int_K \phi_n(\mathbf{x})\phi_m(\mathbf{x})d^3x \\
&= \int_{\hat{K}} \hat{\phi}_n(\hat{\mathbf{x}})\hat{\phi}_m(\hat{\mathbf{x}})J_{F_K}(\hat{\mathbf{x}})d^3\hat{x} \\
&= N_{\alpha_n+\alpha_m,\beta_n+\beta_m,\gamma_n+\gamma_m} \qquad\qquad n,m \in \{0,\cdots,3\}
\end{aligned} \tag{34}$$

with $\alpha_n = \{0,1,0,0\},\ \beta_n = \{0,0,1,0\},\ \gamma_n = \{0,0,0,1\}$. The coefficients N_{nml} are defined as:

$$N_{nml} = \sum_{k=0}^{2}\sum_{j=0}^{2}\sum_{i=0}^{2} b_{ijk} Q_{(k+l+1)} Q_{(j+m+1)} Q_{(i+n+1)}$$

with the coefficients b_{ijk} given by Eq. (32) and Q_j defined as:

$$Q_j = \frac{1}{j}(1-(-1)^j)$$

A.2 Element Face Moments

The element face moment integrals can be calculated analytically using the mapping F_K, Eq. (29):

- Face with index 1:

$$\begin{aligned}
\int_{e_K^1} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \sigma_0 \left(\mathbf{x}_K^1 - \mathbf{x}_K^7\right) \times \left(\mathbf{x}_K^3 - \mathbf{x}_K^5\right) && n=0 \quad (35) \\
&= \sigma_1 \left(\mathbf{x}_K^1 - \mathbf{x}_K^7\right) \times \left(\mathbf{x}_K^3 - \mathbf{x}_K^5\right) && n=1 \quad (36) \\
&= \sigma_2 \left(\mathbf{x}_K^5 - \mathbf{x}_K^7\right) \times \left(\mathbf{x}_K^3 - \mathbf{x}_K^1\right) && n=2 \quad (37) \\
&= \sigma_3 \left(\mathbf{x}_K^1 - \mathbf{x}_K^5\right) \times \left(\mathbf{x}_K^7 - \mathbf{x}_K^3\right) && n=3 \quad (38)
\end{aligned}$$

with $\sigma_n = \{\frac{1}{2}, -\frac{1}{2}, \frac{1}{6}, \frac{1}{6}\}$. The integrals $\int_{e_K^2} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS$ for a face with index 2 can be obtained by a simple permutation of the vertices $\mathbf{x}_K^n$ in Eqs. (35-38): $1 \to 2$, $7 \to 8$, $3 \to 4$, $5 \to 6$ and using $\sigma_n = \{\frac{1}{2}, \frac{1}{2}, \frac{1}{6}, \frac{1}{6}\}$.

- Face with index 3:

$$\begin{aligned}
\int_{e_K^3} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \sigma_0 \left(\mathbf{x}_K^1 - \mathbf{x}_K^6\right) \times \left(\mathbf{x}_K^5 - \mathbf{x}_K^2\right) && n = 0 \quad (39)\\
&= \sigma_1 \left(\mathbf{x}_K^1 - \mathbf{x}_K^2\right) \times \left(\mathbf{x}_K^6 - \mathbf{x}_K^5\right) && n = 1 \quad (40)\\
&= \sigma_2 \left(\mathbf{x}_K^1 - \mathbf{x}_K^6\right) \times \left(\mathbf{x}_K^5 - \mathbf{x}_K^2\right) && n = 2 \quad (41)\\
&= \sigma_3 \left(\mathbf{x}_K^1 - \mathbf{x}_K^5\right) \times \left(\mathbf{x}_K^2 - \mathbf{x}_K^6\right) && n = 3 \quad (42)
\end{aligned}$$

with $\sigma_n = \{\frac{1}{2}, \frac{1}{6}, -\frac{1}{2}, \frac{1}{6}\}$. The integrals $\int_{e_K^4} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS$ for a face with index 4 can be obtained by a simple permutation of the vertices $\mathbf{x}_K^n$ in Eqs. (39-42): $1 \to 3$, $2 \to 4$, $5 \to 7$, $6 \to 8$ and using $\sigma_n = \{\frac{1}{2}, \frac{1}{6}, \frac{1}{2}, \frac{1}{6}\}$.

- Face with index 5:

$$\begin{aligned}
\int_{e_K^5} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \sigma_0 \left(\mathbf{x}_K^1 - \mathbf{x}_K^4\right) \times \left(\mathbf{x}_K^2 - \mathbf{x}_K^3\right) && n = 0 \quad (43)\\
&= \sigma_1 \left(\mathbf{x}_K^1 - \mathbf{x}_K^2\right) \times \left(\mathbf{x}_K^3 - \mathbf{x}_K^4\right) && n = 1 \quad (44)\\
&= \sigma_2 \left(\mathbf{x}_K^1 - \mathbf{x}_K^3\right) \times \left(\mathbf{x}_K^4 - \mathbf{x}_K^2\right) && n = 2 \quad (45)\\
&= \sigma_3 \left(\mathbf{x}_K^1 - \mathbf{x}_K^4\right) \times \left(\mathbf{x}_K^2 - \mathbf{x}_K^3\right) && n = 3 \quad (46)
\end{aligned}$$

with $\sigma_n = \{\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, -\frac{1}{2}\}$. The integrals $\int_{e_K^6} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS$ for a face with index 6 can be obtained by a simple permutation of the vertices $\mathbf{x}_K^n$ in Eqs. (43-46): $1 \to 5$, $2 \to 6$, $3 \to 7$, $4 \to 8$ and using $\sigma_n = \{\frac{1}{2}, \frac{1}{6}, \frac{1}{6}, \frac{1}{2}\}$.

A.2.1 Sub-Face Corrections

Subfaces are defined as a rectangular subdomain $[p_1, p_2] \times [q_1, q_2] \subset \partial\hat{K} = [-1, 1] \times [-1, 1]$

- Faces with index $i = 1$ or 2:

$$\begin{aligned}
\int_{s_K^i} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \frac{1}{4}(p_2 - p_1)(q_2 - q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 0 \\
&= \frac{1}{4}(p_2 - p_1)(q_2 - q_1)\int_{e_K^i} \phi_1(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 1 \\
&= \frac{1}{8}(p_2 - p_1)(q_2 - q_1)^2\int_{e_K^i} \phi_2(\mathbf{x})\mathbf{n}(\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2 - p_1)(q_2^2 - q_1^2)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 2 \\
&= \frac{1}{8}(p_2 - p_1)^2(q_2 - q_1)\int_{e_K^i} \phi_3(\mathbf{x})\mathbf{n}(\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2^2 - p_1^2)(q_2 - q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 3
\end{aligned}$$

- Faces with index $i = 3$ or 4:

$$\begin{aligned}
\int_{s_K^i} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \frac{1}{4}(p_2 - p_1)(q_2 - q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 0 \\
&= \frac{1}{8}(p_2 - p_1)(q_2 - q_1)^2\int_{e_K^i} \phi_1(\mathbf{x})\mathbf{n}(\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2 - p_1)(q_2^2 - q_1^2)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 1 \\
&= \frac{1}{4}(p_2 - p_1)(q_2 - q_1)\int_{e_K^i} \phi_2(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 2 \\
&= \frac{1}{8}(p_2 - p_1)^2(q_2 - q_1)\int_{e_K^i} \phi_3(\mathbf{x})\mathbf{n}(\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2^2 - p_1^2)(q_2 - q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n = 3
\end{aligned}$$

- Faces with index $i = 5$ or 6:

$$\begin{aligned}
\int_{s_K^i} \phi_n(\mathbf{x})\mathbf{n}(\mathbf{x})dS &= \frac{1}{4}(p_2-p_1)(q_2-q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n=0 \\
&= \frac{1}{8}(p_2-p_1)(q_2-q_1)^2\int_{e_K^i} \phi_1(\mathbf{x})\mathbf{n}\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2-p_1)(q_2^2-q_1^2)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n=1 \\
&= \frac{1}{8}(p_2-p_1)^2(q_2-q_1)\int_{e_K^i} \phi_2(\mathbf{x})\mathbf{n}(\mathbf{x})dS + \\
&\quad \frac{1}{8}(p_2^2-p_1^2)(q_2-q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n=2 \\
&= \frac{1}{4}(p_2-p_1)(q_2-q_1)\int_{e_K^i} \phi_0(\mathbf{x})\mathbf{n}(\mathbf{x})dS && n=3
\end{aligned}$$

A.3 Volume Moments

$$\begin{aligned}
\int_{\hat{K}} \mathbf{S}^1(\hat{\mathbf{x}})d^3\hat{x} &= \frac{1}{4}(\mathbf{x}_K^3+\mathbf{x}_K^4-\mathbf{x}_K^5-\mathbf{x}_K^6)\times(\mathbf{x}_K^7+\mathbf{x}_K^8-\mathbf{x}_K^1-\mathbf{x}_K^2)+ \\
&\quad \frac{1}{12}(\mathbf{x}_K^1-\mathbf{x}_K^2-\mathbf{x}_K^7+\mathbf{x}_K^8)\times(\mathbf{x}_K^3-\mathbf{x}_K^4-\mathbf{x}_K^5+\mathbf{x}_K^6) \\
\int_{\hat{K}} \mathbf{S}^2(\hat{\mathbf{x}})d^3\hat{x} &= \frac{1}{4}(\mathbf{x}_K^6+\mathbf{x}_K^8-\mathbf{x}_K^1-\mathbf{x}_K^3)\times(\mathbf{x}_K^2+\mathbf{x}_K^4-\mathbf{x}_K^5-\mathbf{x}_K^7)+ \\
&\quad \frac{1}{12}(\mathbf{x}_K^2-\mathbf{x}_K^4-\mathbf{x}_K^5+\mathbf{x}_K^7)\times(\mathbf{x}_K^1-\mathbf{x}_K^3-\mathbf{x}_K^6+\mathbf{x}_K^8) \\
\int_{\hat{K}} \mathbf{S}^3(\hat{\mathbf{x}})d^3\hat{x} &= \frac{1}{4}(\mathbf{x}_K^2+\mathbf{x}_K^6-\mathbf{x}_K^3-\mathbf{x}_K^7)\times(\mathbf{x}_K^4+\mathbf{x}_K^8-\mathbf{x}_K^1-\mathbf{x}_K^5)+ \\
&\quad \frac{1}{12}(\mathbf{x}_K^1-\mathbf{x}_K^4-\mathbf{x}_K^5+\mathbf{x}_K^8)\times(\mathbf{x}_K^2-\mathbf{x}_K^3-\mathbf{x}_K^6+\mathbf{x}_K^7)
\end{aligned}$$

A.4 Estimates for Geometrical Quantities

The element face Jacobian at a surface $\xi = 1$ of a hexahedral element is defined as:

$$J_\xi = |\mathbf{x}_\eta \times \mathbf{x}_\zeta|$$

and can be estimated using Assumption 3.1 and Eq. (29) as:

$$\begin{aligned}
J_\xi &\leq |\hat{\mathbf{x}}^3+\hat{\mathbf{x}}^5|\,|\hat{\mathbf{x}}^4+\hat{\mathbf{x}}^6| + |\hat{\mathbf{x}}^3+\hat{\mathbf{x}}^5|\,|\hat{\mathbf{x}}^7+\hat{\mathbf{x}}^8|\,|\eta| + |\hat{\mathbf{x}}^7+\hat{\mathbf{x}}^8|\,|\hat{\mathbf{x}}^4+\hat{\mathbf{x}}^6|\,|\zeta| \qquad \eta,\zeta \in [-1,1] \\
&\leq 12h^2
\end{aligned}$$

This estimate can be used to obtain the following estimates for $|\mathbf{x}|$ a vector $\mathbf{x} \in S$, the surface area $|S|$ and the integrals $\tilde{M}_{jk}$, defined in Eq. (18):

$$\begin{aligned} |\mathbf{x}| &\leq 8h \\ |S| &\leq 48h^2 \\ |\tilde{M}_{jk}| &\leq 6144h^4 \end{aligned}$$

Identical results are obtained for other faces of a hexahedral element.

Estimates for the volume Jacobian, defined in Eq. (31), and the mass matrix, Eq. (34), are also obtained using Assumption 3.1 and Eq. (29):

$$\begin{aligned} J &\leq 384h^3 \\ \left|M^{-1}_{nm}\right| &\leq C/h^3 \end{aligned}$$

Adaptation Step	Elements	Grid Points
0	600	1302
1	862	1902
2	1132	2588
3	1513	3604
4	2049	5032
5	2789	7006
6	3799	9458

Table I Number of grid points and elements after each adaptation step for supersonic flow about a 10° ramp.

Adaptation Step	Elements	Grid Points
0	131072	137425
1	199342	215499
2	259965	293471
3	339226	398356

Table II Number of grid points and elements after each adaptation step for transsonic flow about the ONERA M6 wing.

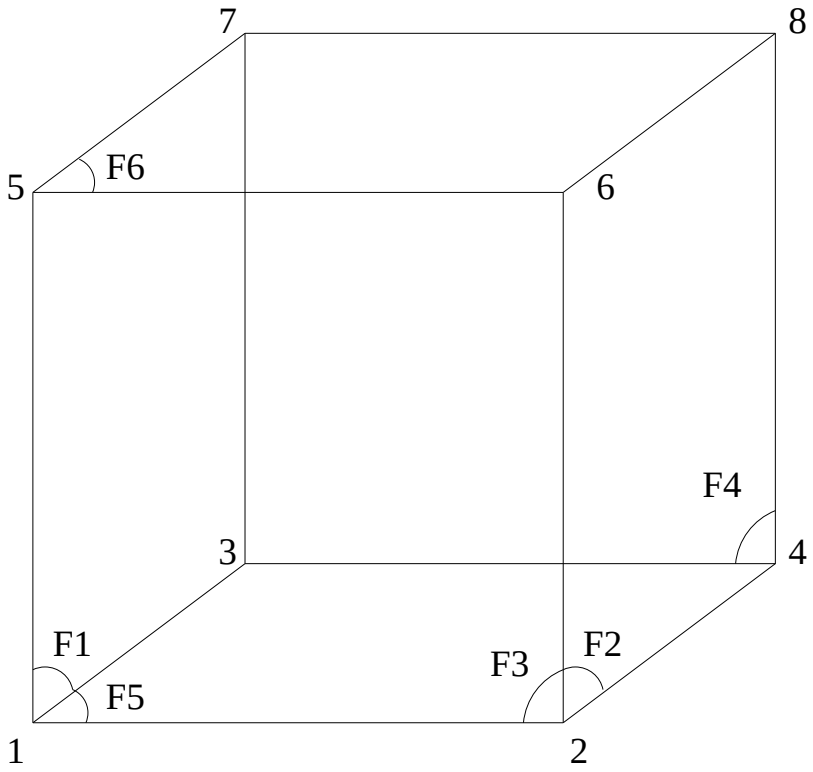

Figure 1: Face and vertex definition of master element $\hat{K}$.

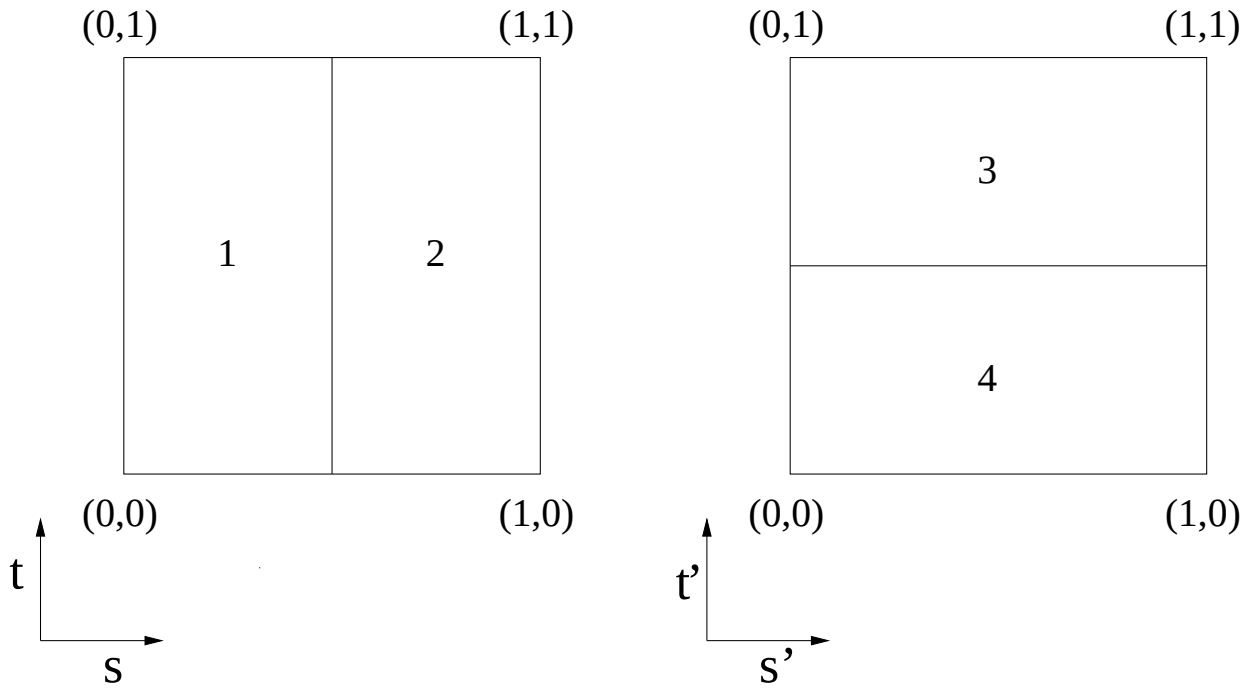

Figure 2: Element refinement at left and right side of elementary face. Elements 1 and 2 can not be connected to elements 3 and 4 with one face.

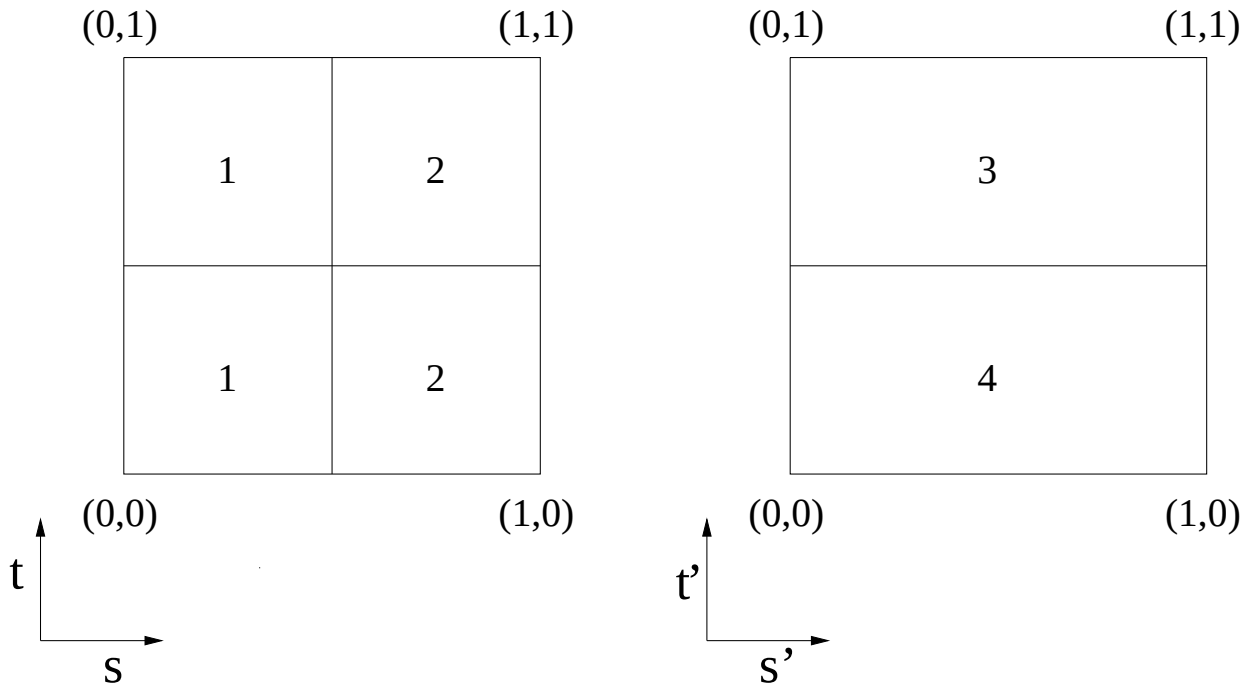

Figure 3: Faces of elements 1 and 2 are split into sub-faces such that each sub-face connects to one element at each side.

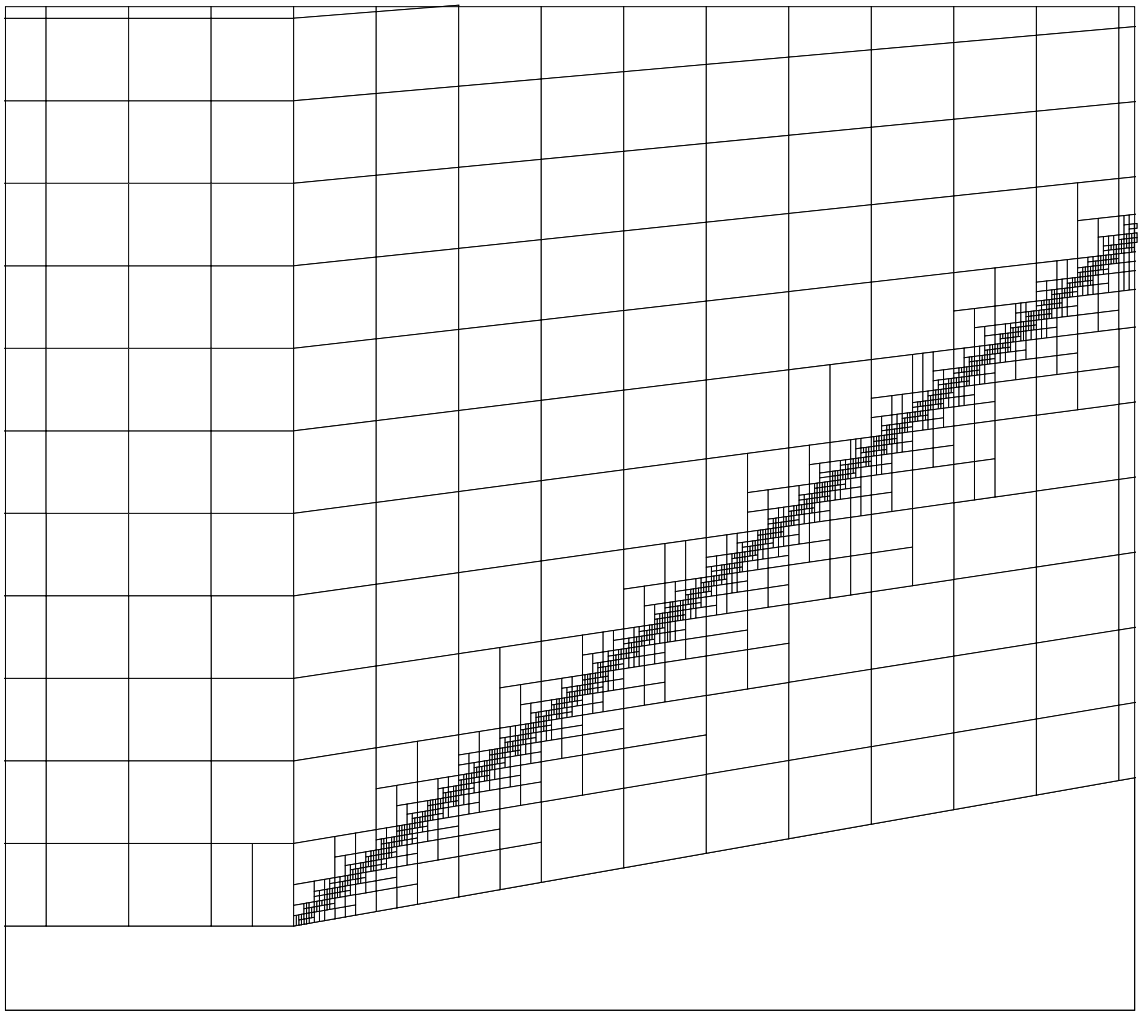

Figure 4: Detail of grid for supersonic flow over a 10° ramp after six adaptations.

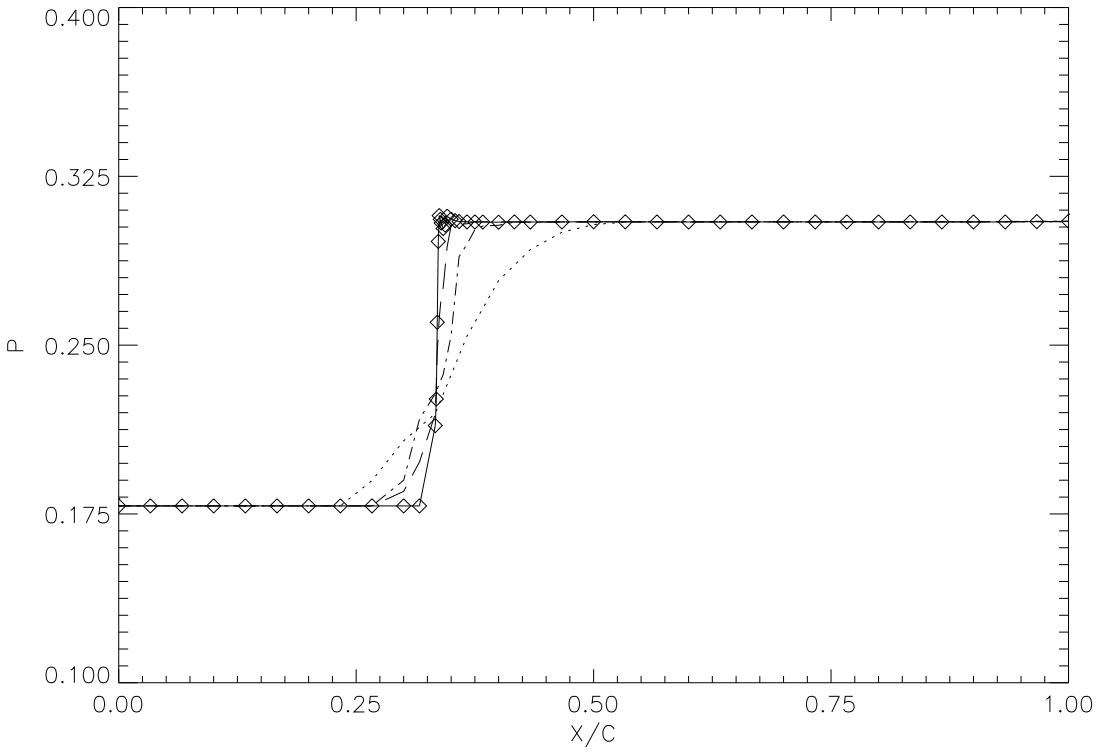

Figure 5: Pressure distribution along a 10° ramp for supersonic flow, $M_\infty = 2.0$ ($\cdots$ original grid, — - — two adaptations, — — four adaptations, —— six adaptations, $\diamond\ \diamond\ \diamond$ final adapted grid).

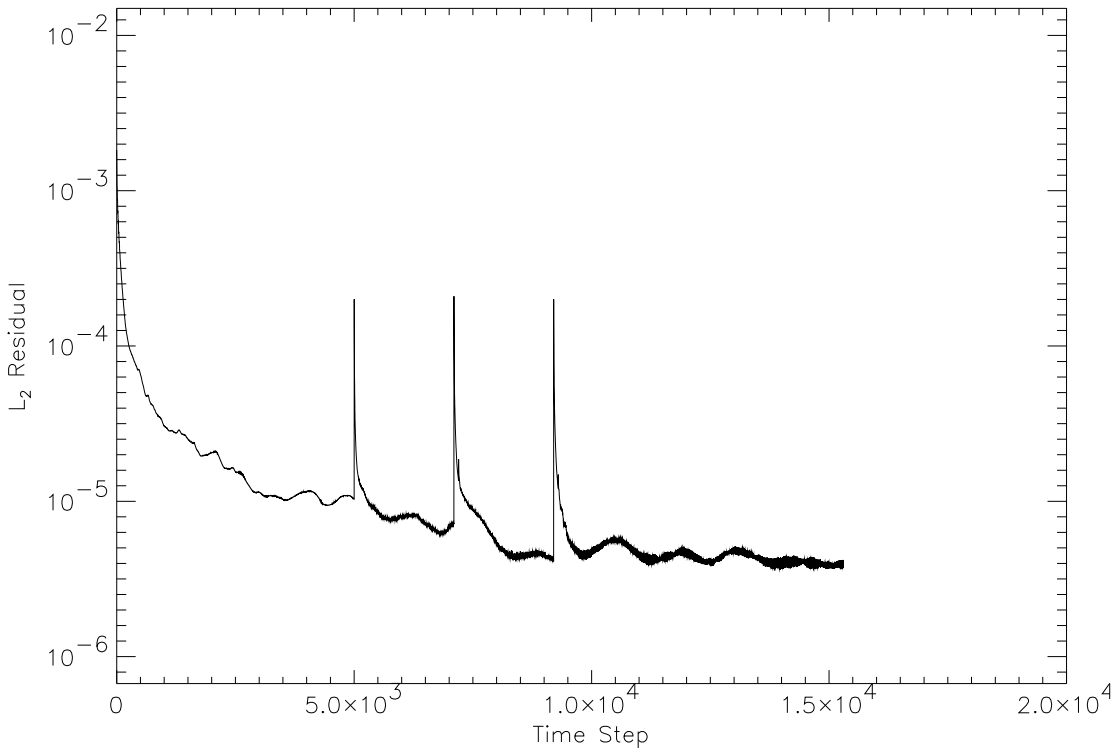

Figure 6: Convergence history of L_2 residual for flow field about ONERA M6 wing.

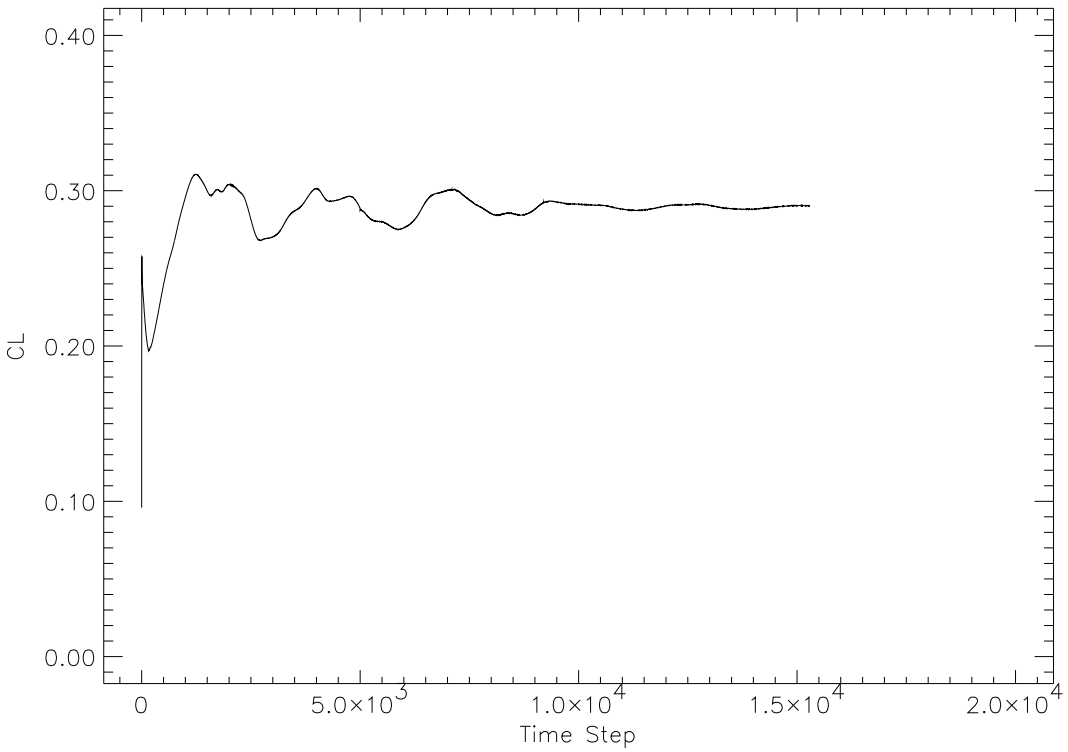

Figure 7: Convergence history of lift force C_L on ONERA M6 wing.

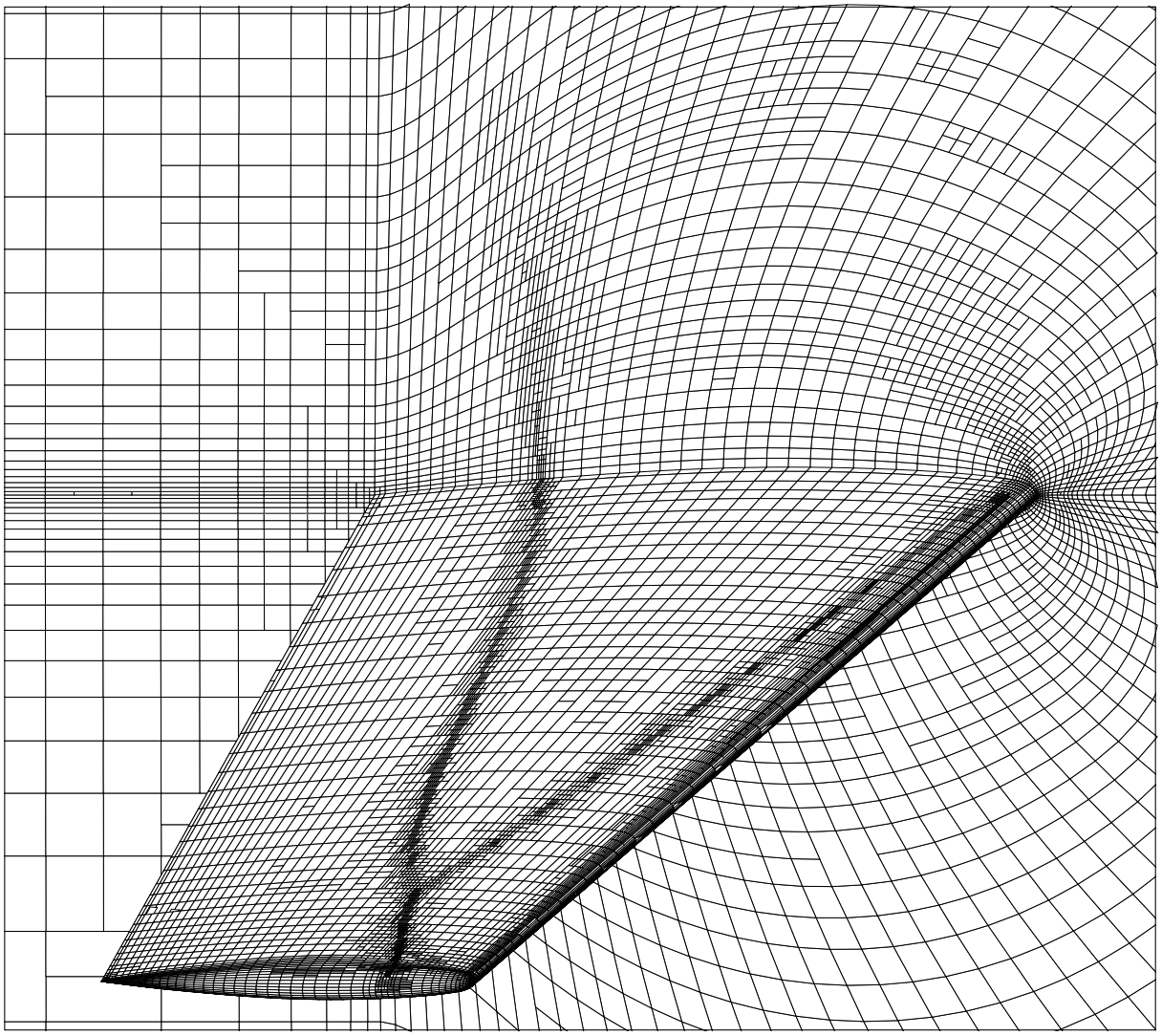

Figure 8: Final adapted grid on ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$.

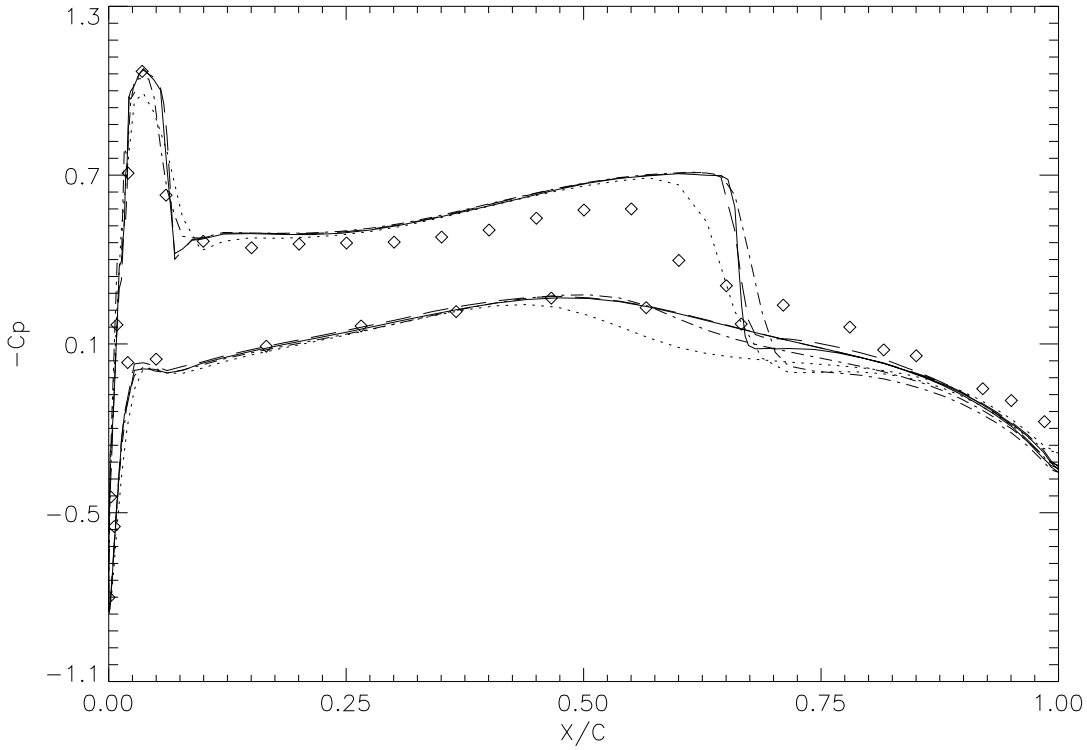

Figure 9: Pressure coefficient C_p at cross-section $y = 0.20S$ of ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$ ($\cdots$ original grid, — - — one adaptation, — — two adaptations, —— three adaptations, $\diamond$ $\diamond$ $\diamond$ experiment).

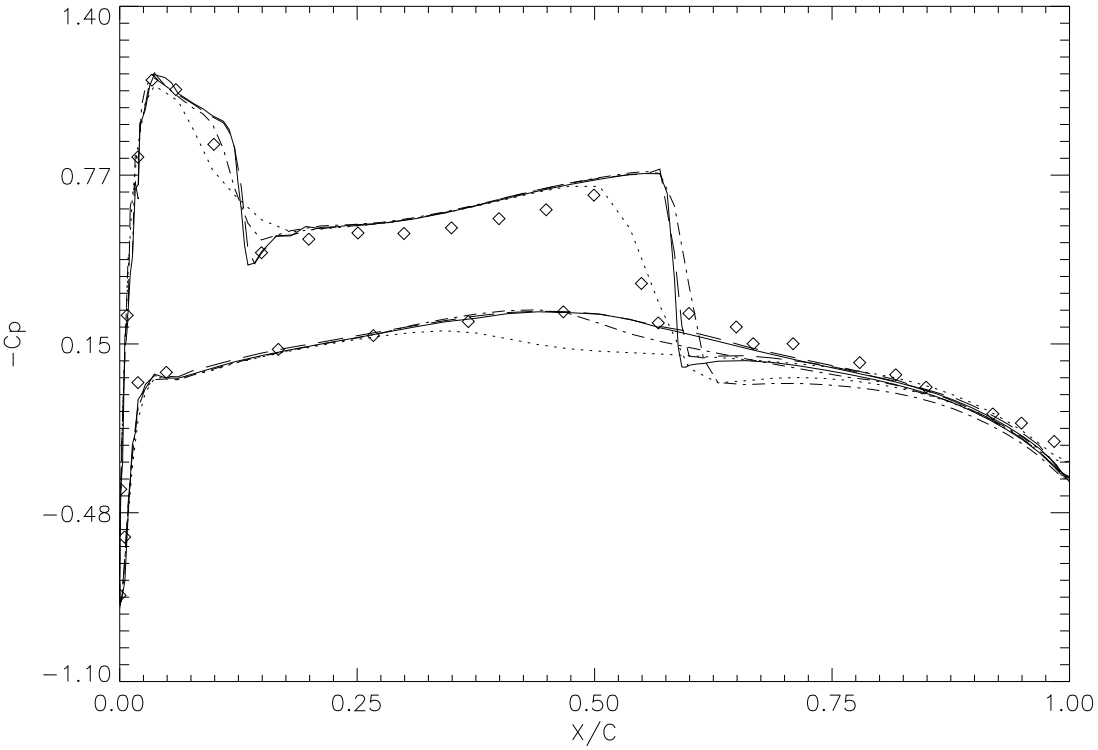

Figure 10: Pressure coefficient C_p at cross-section $y = 0.44S$ of ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$ ($\cdots$ original grid, — - — one adaptation, — — two adaptations, —— three adaptations, $\diamond$ $\diamond$ $\diamond$ experiment).

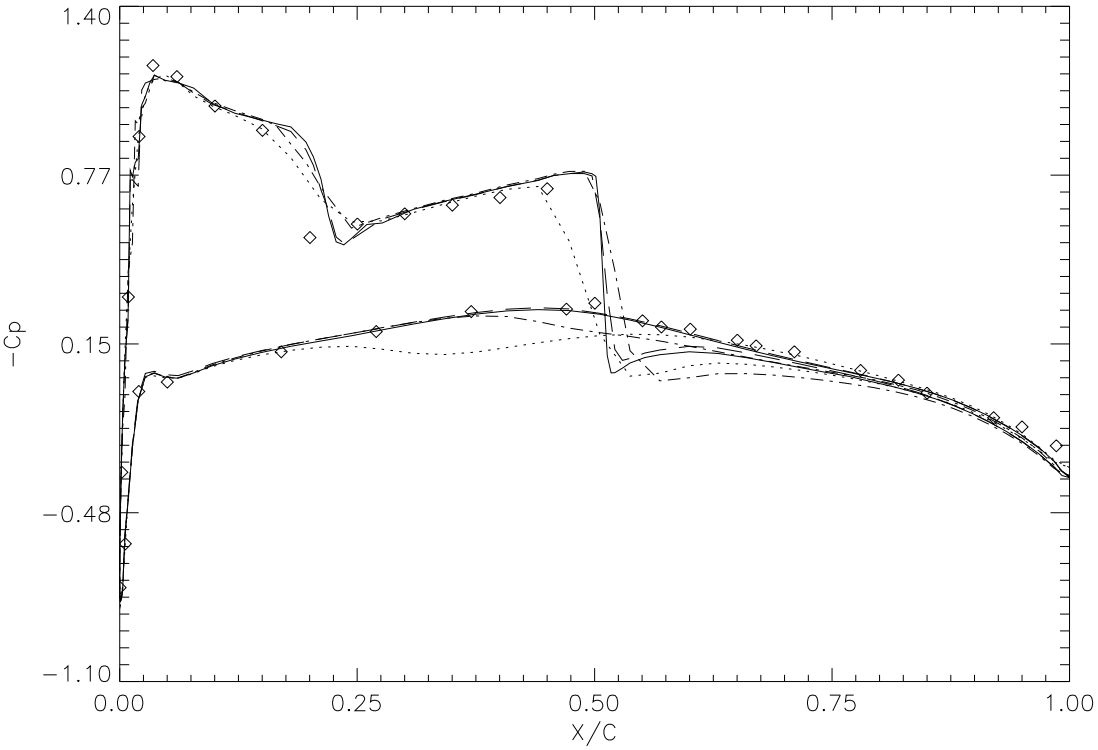

Figure 11: Pressure coefficient C_p at cross-section $y = 0.65S$ of ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$ ($\cdots$ original grid, — - — one adaptation, — — two adaptations, —— three adaptations, $\diamond$ $\diamond$ $\diamond$ experiment).

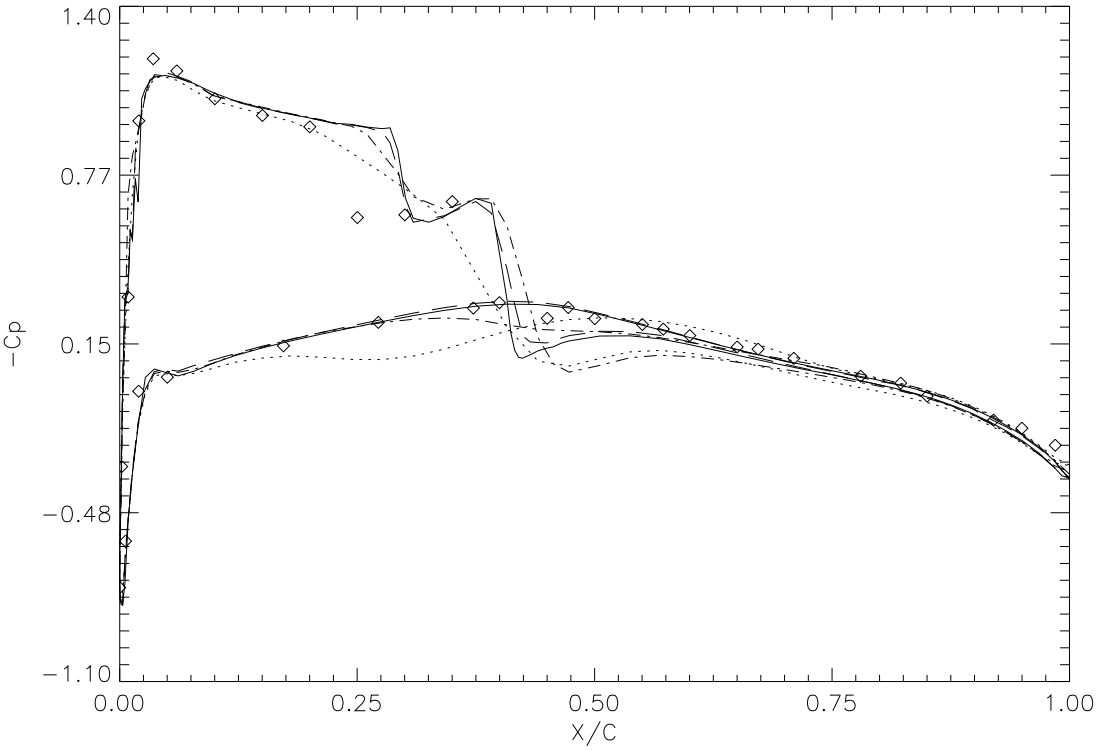

Figure 12: Pressure coefficient C_p at cross-section $y = 0.80S$ of ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$ ($\cdots$ original grid, — - — one adaptation, — — two adaptations, —— three adaptations, $\diamond$ $\diamond$ $\diamond$ experiment).

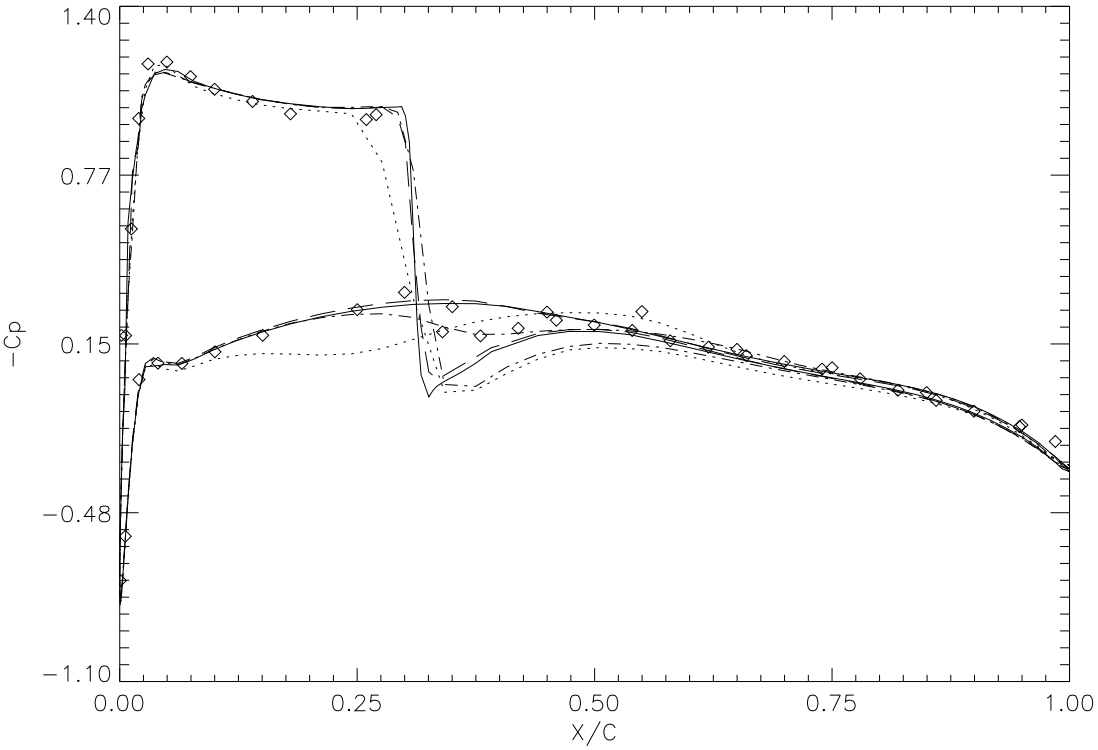

Figure 13: Pressure coefficient C_p at cross-section $y = 0.90S$ of ONERA M6 wing, $M_\infty = 0.84, \alpha = 3.06^\circ$ ($\cdots$ original grid, — - — one adaptation, — — two adaptations, —— three adaptations, $\diamond$ $\diamond$ $\diamond$ experiment).