ICAS-92-4.5.1

A FLUX-DIFFERENCE FINITE VOLUME METHOD FOR STEADY EULER EQUATIONS ON ADAPTIVE UNSTRUCTURED GRIDS

Kris Riemslagh
Labo voor Machines en Machinebouw
Universiteit Gent
Sint-Pietersnieuwstraat 41
9000 GENT, BELGIUM

Erik Dick
Institut de mécanique
Université de Liège
Rue E. Solvay 21
4000 LIEGE, BELGIUM

Abstract

On an unstructured grid a flux-difference splitting type algorithm is formulated for the time dependent Euler equations. The polynomial flux-difference splitting technique is used. This splitting is of Roe-type. The grids considered are composed of triangles. A vertex-centred finite volume method is employed. This means that control volumes are constructed around each vertex of the grid by connecting the centres of surrounding triangles. On each face of a control volume, the first order flux is defined taking into account the variables in the two vertices on both sides of the face. The second order correction results from a limited combination of the positive and negative parts of the first order flux-difference, with corresponding parts obtained through extrapolation. The extrapolation is based on gradients of conserved variables in the vertices. As limiter, the simple minmod-limiter is used.

The time dependent system is integrated explicitely. This is done with a Runge-Kutta type multi-stage time stepping technique, with local time steps.

The domain is discretized using a Delaunay triangular mesh generator, which is able to handle non-convex geometries. To generate the grid, a description of the domain boundary has to be given. Then points are introduced on the boundaries, based on local curvature and local grid spacing information. With these points an initial grid is constructed, conform with the boundary. Then, criteria based on area and aspect ratio of the triangles are used to refine this inital mesh. On this first mesh, which is also a Delaunay triangulation, a flow solution is calculated. As soon as the residual drops below a threshold value, an adaption cycle is started. Using the same procedure as for the first grid, criteria are used to refine the mesh. Instead of being based on mesh properties they are now based on flow properties like pressure and entropy differences.

Results are shown for the transonic flow over a NACA-0012 airfoil.

Introduction

The polynomial flux-difference splitting introduced by the second author [1], is used here to construct a discretization of the time dependent Euler equations on unstructured grids. The flux-difference splitting technique is of Roe-type, i.e. it satisfies the primary requirements formulated by Roe [2]. It is however simpler. Its simplicity follows from dropping the secondary requirement of having a unique definition of average flow variables. This secondary requirement defines the original Roe-splitting within the class of methods allowed by the primary requirements. The secondary requirement is however not necessary.

The vertex-centred finite volume method is employed. On each face of a control volume, the first order flux is defined taking into account the two vertices on both sides of the face. The second order correction is constructed based on the flux-extrapolation concept introduced by Chakravarthy and Osher [3].

Unstructured triangular grids are very attractive for two-dimensional flow calculations. One reason is that geometries of arbitrary complexity can be meshed. The other is that mesh adaption by addition of points is very simple. The generation of unstructured triangular grids can be done in several ways. One of the techniques is based on the well known Delaunay triangulation algorithm formulated by Bowyer [4]. This algorithm, which is rather popular nowadays [5,6,7] is used here. Compared with other grid generation techniques, this algorithm allows a strategy for placing points which is independent of the mechanism to connect the points. The algorithm iteratively generates a grid, by bringing point after point into the grid. Every time, a small portion of the grid is deleted and reconnected to include the new point. These features of the algorithm leed to easy grid adaption through refinement.

The space discretization

Vertex-centred finite volume discretization

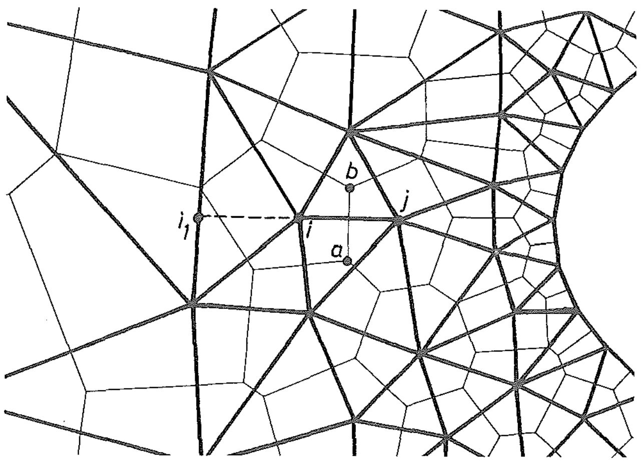

Figure 1: Vertex-centred discretization.

Figure 1 shows part of an unstructured triangular grid. In the vertex-centred finite volume method nodes are located at the vertices of the grid. Every node has a controle volume, constructed by connecting the *centres* of the cells surrounding the node. The typical choice for the *centre* of a cell is the centre of gravity. Here we choose to take the circumcentre as *centre* for the triangles with no obtuse angle. For the triangles with an obtuse angle, the circumcentre would lie outside the triangle. To correct this the *centre* is then taken as the midpoint of the edge opposite to the obtuse angle, figure 1. To close the control volumes on the boundary, the midpoints of the boundary edges are chosen as vertices of the control volumes, figure 1. The advantage of taking the circumcentres is that the faces of the control volume are as much as possible perpendicular to the cell edges. With this choice of the *centre* of triangles, the control volumes coincide with the Voronoi regions, except where obtuse angles occur. The Voronoi regions are defined in the section on mesh generation.

Flux-difference splitting

To define the flux through a side of a control volume, use is made of the flux-difference splitting principle. Euler equations in two dimensions take the form

$$\frac{\partial U}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0, \tag{1}$$

where U stands for the vector of conserved variables, f and g the Cartesian flux vectors, given by

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, g = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vH \end{pmatrix}, \tag{2}$$

with ρ density, u and v Cartesian velocity components, p pressure, $E = p/(\gamma-1)\rho + u^2/2 + v^2/2$ total energy,
$H = \gamma p/(\gamma-1)\rho + u^2/2 + v^2/2$ total enthalpy and γ the adiabatic constant.

For the side ab of the control volume of node i and node j, figure 1, the flux-difference can be written as

$$\Delta F_{i,j} = (n_x \Delta f_{i,j} + n_y \Delta g_{i,j}) \Delta s_{i,j}, \tag{3}$$

where $\Delta f_{i,j}$ and $\Delta g_{i,j}$ denote the differences of the Cartesian flux vectors, n_x and n_y are the components of the unit normal to the side ab in the sense i to j, and where $\Delta s_{i,j}$ is the length of the side ab.

The differences of the Cartesian flux vectors are

$$\Delta f_{i,j} = f_j - f_i \ , \quad \Delta g_{i,j} = g_j - g_i, \tag{4}$$

where f_i and f_j are the flux vectors calculated with the flow variables in node i and node j respectively.

The flux-difference defined by (3) can be written as

$$\begin{aligned} \Delta F_{i,j} &= A_{i,j}(U_j - U_i)\Delta s_{i,j} \\ &= A_{i,j}\Delta U_{i,j}\Delta s_{i,j}. \end{aligned}$$

To define the discrete Jacobian A, we use here the polynomial flux-difference splitting. Full details of this splitting are given in [8,9]. For completeness, we derive here the most essential expressions.

Since the components of the flux vectors form polynomials with respect to the primitive variables ρ, u, v and p, components of the flux-differences can be written as follows

$$\begin{aligned} \Delta\rho u &= \overline{u}\,\Delta\rho + \overline{\rho}\,\Delta u \\ \Delta(\rho u^2 + p) &= \overline{\rho u}\Delta u + \overline{u}\,\Delta\rho u + \Delta p \\ &= \overline{u}^{\,2}\Delta\rho + (\overline{\rho u} + \overline{\rho}\,\overline{u}\,)\Delta u + \Delta p \\ \Delta\rho H u &= \overline{\rho u}(\frac{1}{2}\Delta u^2 + \frac{1}{2}\Delta v^2) + \frac{1}{2}(\overline{u^2} + \overline{v^2})\Delta\rho u \\ &\quad + \frac{\gamma}{\gamma-1}\Delta pu \\ &= \frac{1}{2}(\overline{u^2} + \overline{v^2})\overline{u}\,\Delta\rho + \frac{1}{2}(\overline{u^2} + \overline{v^2})\overline{\rho}\,\Delta u + \overline{u}\,\overline{\rho u}\Delta u \\ &\quad + \frac{\gamma}{\gamma-1}\overline{p}\,\Delta u + \overline{v}\,\overline{\rho u}\Delta v + \frac{\gamma}{\gamma-1}\overline{u}\,\Delta p, \end{aligned}$$

etc., where the bar denotes the mean value. With the definition of $\overline{q}$ and $\overline{\overline{q}}$,

$$\overline{q} = \frac{1}{2}(\overline{u^2} + \overline{v^2}), \quad \overline{\overline{q}} = \overline{u}^{\,2} + \overline{v}^{\,2} - \overline{q}\,,$$

the flux-difference can be written as

$$\Delta f = \begin{pmatrix} \overline{u} & \overline{\rho} & 0 & 0 \\ \overline{u}^{\,2} & \overline{\rho}\,\overline{u} + \overline{\rho u} & 0 & 1 \\ \overline{u}\,\overline{v} & \overline{\rho}\,\overline{v} & \overline{\rho u} & 0 \\ \overline{q}\,\overline{u} & \overline{q}\,\overline{\rho} + \overline{u}\,\overline{\rho u} + \frac{\gamma}{\gamma-1}\overline{p} & \overline{v}\,\overline{\rho u} & \frac{\gamma}{\gamma-1}\overline{u} \end{pmatrix} \begin{pmatrix} \Delta\rho \\ \Delta u \\ \Delta v \\ \Delta p \end{pmatrix}.$$

With the definition of $\overline{\overline{u}} = \overline{\rho u}/\overline{\rho}$, and of the vector $W = (\rho,\ u,\ v,\ p)^T$ the flux-difference Δf is given by

$$\Delta f = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \overline{u} & \overline{\rho} & 0 & 0 \\ \overline{v} & 0 & \overline{\rho} & 0 \\ \overline{q} & \overline{\rho}\,\overline{u} & \overline{\rho}\,\overline{v} & \frac{1}{\gamma-1} \end{pmatrix} \begin{pmatrix} \overline{u} & \overline{\rho} & 0 & 0 \\ 0 & \overline{\overline{u}} & 0 & 1/\overline{\rho} \\ 0 & 0 & \overline{\overline{u}} & 0 \\ 0 & \gamma\overline{p} & 0 & \overline{u} \end{pmatrix} \Delta W.$$

By denoting the first matrix of this equation as T, it is easily seen that the flux-difference Δg can be written in a similar way as

$$\Delta g = T \begin{pmatrix} \overline{v} & 0 & \overline{\rho} & 0 \\ 0 & \overline{\overline{v}} & 0 & 0 \\ 0 & 0 & \overline{\overline{v}} & 1/\overline{\rho} \\ 0 & 0 & \gamma\overline{p} & \overline{v} \end{pmatrix} \Delta W,$$

with $\overline{\rho}\,\overline{\overline{v}} = \overline{\rho v}$.

So any flux-difference can be written as

$$\begin{aligned} \Delta F &= (n_x \Delta f + n_y \Delta g)\Delta s \\ &= T\tilde{A}\Delta W \Delta s. \end{aligned} \qquad (5)$$

The matrix T is the transformation matrix between differences of conserved variables and differences of primitive variables, i.e. $\Delta U = T\Delta W$ or $\Delta W = T^{-1}\Delta U$. So the flux-difference ΔF can be written as

$$\begin{aligned} \Delta F &= T\tilde{A}T^{-1}\Delta U \Delta s \\ &= A\Delta U \Delta s, \end{aligned} \qquad (6)$$

with T^{-1} given by

$$T^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\overline{u}/\overline{\rho} & 1/\overline{\rho} & 0 & 0 \\ -\overline{v}/\overline{\rho} & 0 & 1/\overline{\rho} & 0 \\ (\gamma-1)\overline{\overline{q}} & -(\gamma-1)\overline{u} & -(\gamma-1)\overline{v} & (\gamma-1) \end{pmatrix}.$$

In the expression (5) $\tilde{A}$ takes the form

$$\tilde{A} = \begin{pmatrix} \overline{w} & n_x\overline{\rho} & n_y\overline{\rho} & 0 \\ 0 & \overline{\overline{w}} & 0 & n_x/\overline{\rho} \\ 0 & 0 & \overline{\overline{w}} & n_y/\overline{\rho} \\ 0 & n_x\gamma\overline{p} & n_y\gamma\overline{p} & \overline{w} \end{pmatrix},$$

with $\overline{w} = n_x\overline{u} + n_y\overline{v}$ and $\overline{\overline{w}} = n_x\overline{\overline{u}} + n_y\overline{\overline{v}}$.

It is easy to verify that this matrix $\tilde{A}$ has real eigenvalues and a complete set of eigenvectors.

$$\tilde{A} = R\Lambda L$$

For $n_x^2 + n_y^2 = 1$, the eigenvalue matrix is given by

$$\Lambda = \begin{pmatrix} \lambda^1 & 0 & 0 & 0 \\ 0 & \lambda^2 & 0 & 0 \\ 0 & 0 & \lambda^3 & 0 \\ 0 & 0 & 0 & \lambda^4 \end{pmatrix} = \begin{pmatrix} \overline{w} & 0 & 0 & 0 \\ 0 & \overline{\overline{w}} & 0 & 0 \\ 0 & 0 & \tilde{w}+\overline{c} & 0 \\ 0 & 0 & 0 & \tilde{w}-\overline{c} \end{pmatrix},$$

where $\tilde{w} = (\overline{w}+\overline{\overline{w}})/2$ and $\overline{c}^{\,2} = \gamma\overline{p}/\overline{\rho} + (\overline{w}-\overline{\overline{w}})^2/4$. R and L denote the right and left eigenvector matrices, in orthonormal form. With $\delta = (\overline{w} - \overline{\overline{w}})/2\overline{c}$ they are

$$R = \begin{pmatrix} 1 & 0 & \overline{\rho}/2\overline{c} & -\overline{\rho}/2\overline{c} \\ 0 & n_y & n_x(1-\delta)/2 & n_x(1+\delta)/2 \\ 0 & -n_x & n_y(1-\delta)/2 & n_y(1+\delta)/2 \\ 0 & 0 & \gamma\overline{p}/2\overline{c} & -\gamma\overline{p}/2\overline{c} \end{pmatrix},$$

$$L = \begin{pmatrix} 1 & 0 & 0 & -\overline{\rho}/\gamma\overline{p} \\ 0 & n_y & -n_x & 0 \\ 0 & n_x & n_y & (1+\delta)\overline{c}/\gamma\overline{p} \\ 0 & n_x & n_y & -(1-\delta)\overline{c}/\gamma\overline{p} \end{pmatrix}.$$

The matrix $\tilde{A}$ can be split into positive and negative parts by

$$\tilde{A}^+ = R\Lambda^+ L\ , \quad \tilde{A}^- = R\Lambda^- L\ , \qquad (7)$$

with

$$\Lambda^\pm = \begin{pmatrix} \lambda^{1\pm} & 0 & 0 & 0 \\ 0 & \lambda^{2\pm} & 0 & 0 \\ 0 & 0 & \lambda^{3\pm} & 0 \\ 0 & 0 & 0 & \lambda^{4\pm} \end{pmatrix},$$

where

$$\lambda^{i+} = max(\lambda^i, 0), \quad \lambda^{i-} = min(\lambda^i, 0).$$

With positive and negative matrices, matrices with non-positive and non-negative eigenvalues are meant.

Combination of (6) and (7) gives then the flux-difference

$$\begin{aligned} \Delta F &= TR\Lambda^+ LT^{-1}\Delta U\Delta s + TR\Lambda^- LT^{-1}\Delta U\Delta s \\ &= A^+\Delta U\Delta s + A^-\Delta U\Delta s. \end{aligned} \qquad (8)$$

This allows the definition of the absolute value of the flux-difference by

$$\begin{aligned} |\Delta F| &= |A|\Delta U\Delta s \\ &= A^+\Delta U\Delta s - A^-\Delta U\Delta s. \end{aligned} \qquad (9)$$

Upwind flux definition

For the side ab of the control volume of node i and node j, figure 1, the first order upwind flux is defined by

$$\begin{aligned} F^1_{i,j} &= \frac{1}{2}(F_i + F_j) - \frac{1}{2}|\Delta F_{i,j}| \\ &= \frac{1}{2}(F_i + F_j) - \frac{1}{2}(A^+_{i,j} - A^-_{i,j})\Delta U_{i,j}\Delta s_{i,j}, \end{aligned} \qquad (10)$$

where

$$\begin{aligned} F_i &= (n_x\Delta f_i + n_y\Delta g_i)\Delta s_{i,j}, \\ F_j &= (n_x\Delta f_j + n_y\Delta g_j)\Delta s_{i,j}. \end{aligned}$$

Using (8) the expression (10) can be written as

$$F^1_{i,j} = F_i + A^-_{i,j}\,\Delta U_{i,j}\,\Delta s_{i,j}. \qquad (11)$$

. In order to define a second order flux, the second part in the right hand side of (10), which contains the positive and negative parts of the flux-difference, is decomposed into components along the eigenvectors of the Jacobian, according to

$$\begin{aligned} \Delta F^\pm_{i,j} &= A^\pm_{i,j}\,\Delta U_{i,j}\,\Delta s_{i,j} \\ &= \sum_n r^n_{i,j}\,\lambda^{n\pm}_{i,j}\,l^n_{i,j}\,\Delta U_{i,j}\,\Delta s_{i,j}, \end{aligned} \qquad (12)$$

where r^n and l^n are right and left eigenvectors of A associated to the eigenvalue λ^n. r^n and l^n are components of TR and LT^{-1}. By denoting the projection of $\Delta U_{i,j}$ on the n^{th} eigenvector by

$$\sigma^n_{i,j} = l^n_{i,j}\,\Delta U_{i,j}, \qquad (13)$$

the flux-difference becomes

$$\begin{aligned} \Delta F^\pm_{i,j} &= \sum_n r^n_{i,j}\,\lambda^{n\pm}_{i,j}\,\sigma^n_{i,j}\,\Delta s_{i,j} \\ &= \sum_n \Delta F^{n\pm}_{i,j}. \end{aligned} \qquad (14)$$

The second order flux is then defined by

$$\begin{aligned}F^2_{i,j} &= \frac{1}{2}(F_i + F_j) \\ &-\frac{1}{2}\sum_n \Delta F^{n+}_{i,j} + \frac{1}{2}\sum_n \Delta F^{n-}_{i,j} \\ &+\frac{1}{2}\sum_n \Delta \tilde{\tilde{F}}^{n+}_{i,j} - \frac{1}{2}\sum_n \Delta \tilde{\tilde{F}}^{n-}_{i,j}, \end{aligned} \quad (15)$$

where $\Delta \tilde{\tilde{F}}^{n\pm}$ are limited combinations of flux-difference components $\Delta F^{n\pm}$ and shifted differences $\Delta \tilde{F}^{n\pm}$, in the sense of i for positive components and in the sense of j for negative components. The limiter used here is the minmod-limiter. The shifted flux-difference components are constructed based on shifted differences of conserved variables. These are obtained by elonging the line segment i, j into the adjacent triangles and constructing the intersections with the opposing sides (point i_1 for the shifting in i sense in figure 1). The shifted flux-differences are defined by

$$\begin{aligned}\Delta \tilde{F}^{n\pm}_{i,j} &= r^n_{i,j}\ \lambda^{n\pm}_{i,j}\ \tilde{\sigma}^{n\pm}_{i,j}\ \Delta s_{i,j} \\ &= r^n_{i,j}\ \lambda^{n\pm}_{i,j}\ l^n_{i,j}\ \Delta \tilde{U}^{n\pm}_{i,j}\ \Delta s_{i,j}.\end{aligned} \quad (16)$$

The foregoing technique to define the second order flux commonly is called the flux-extrapolation technique [3].

The time stepping

For the time-dependent Euler equations, the discrete set of equations associated to the node i reads

$$Vol_i \frac{dU_i}{dt} + \sum_k A^-_{i,k}\Delta U_{i,k}\Delta s_{i,k} + S.O. = 0, \quad (17)$$

where the index k loops over the faces of the control volume and the surrounding nodes, where the term $S.O.$ denotes symbolically the second order correction, and where Vol_i is the area of the two dimensional control volume. A single stage time stepping method with local time step on (17) gives

$$\left(\frac{Vol_i}{\Delta t_i}\right)(U^{n+1}_i - U^n_i) + \sum_k A^{-n}_{i,k}\Delta U^n_{i,k}\Delta s_{i,k} + S.O.^n = 0, \quad (18)$$

with the superscript n denoting the time level.

The maximum allowable time step is based on the monotonicity condition of a single step first order time stepping. This monotomicity condition reads

$$\mathcal{R}\left(\frac{Vol_i}{\Delta t_i} I + \sum_k A^-_{i,k}\Delta s_{i,k}\right) \geq 0, \quad (19)$$

where $\mathcal{R}$ means spectral radius. A sufficient condition to satisfy (19) is given by

$$\frac{Vol_i}{\Delta t_i} \geq \sum_k \mathcal{R}(-A^-_{i,k})\Delta s_{i,k} = \sum_k max(0, \bar{c}_{i,k} - \bar{w}_{i,k})\Delta s_{i,k},$$

where $\bar{c}$ is the velocity of sound and $\bar{w}$ is the normal outgoing velocity component on a side.

The three-stage modified Runge-Kutta time stepping is used here.

$$\begin{aligned} U_o &= U^n_i \\ U_1 &= U_0 + \alpha_1\ CFL\ \frac{\Delta t_i}{Vol_i}\ R_0 \\ U_2 &= U_0 + \alpha_2\ CFL\ \frac{\Delta t_i}{Vol_i}\ R_1 \\ U_3 &= U_0 + \alpha_3\ CFL\ \frac{\Delta t_i}{Vol_i}\ R_2 \\ U^{n+1}_i &= U_3, \end{aligned}$$

with

$$R_l = -\sum_k A^{-l}_{i,k}\ \Delta U^l_{i,k}\ \Delta s_{i,k} - S.O.^l.$$

The parameters are chosen according to Van Leer et al. [10]: $\alpha_1 = 0.2884; \alpha_2 = 0.5010; \alpha_3 = 1.0; CFL = 1.325$.

Boundary conditions

To calculate the flow around an airfoil there are two kinds of boundaries, a solid wall on the airfoil itself, and a far-field boundary due to the finite extension of the computational domain.

At solid boundaries, impermiability is imposed by setting the convective part of the flux equal to zero. Thus a special flux definition is used for the boundary edges of the control volumes of boundary nodes. These nodes are updated with the normal procedure for internal nodes. It is then easy to see that the local time step definition is to be modified in the sense that in the sum over the neighbouring edges no contributions from the boundary edges enter.

The farfield boundary is a square aligned with the flow (figure 2). This boundary is interpreted as a channel. Therefore the walls of the channel have to be far enough away from the airfoil in order not to disturb the flow. In the example the farfield lies 100 chords away from the airfoil. For this channel the upper and lower boundaries are treated as solid. For inlet and outlet boundaries flow variables are extrapolated. For a point on an inlet boundary a line is drawn in the direction of the flow. The intersection of this line and the nearest triangle edge that lies completely within the flow field, is determined. The flow variables needed at the boundary are interpolated at the intersection point. At subsonic inlet, the Mach number is extrapolated, while stagnation conditions and flow direction are imposed. At subsonic outflow the reverse is done.

The mesh generation

Delaunay triangulation

The automatic triangulation of an arbitrary set of points can be achieved using the Delaunay triangulation. Robust algorithms to construct this triangulation are nowadays fully developed. The triangulation is based upon the concept of the in-circle criterion. The triangulation is the geometrical dual of the Voronoi diagram. This diagram is the construction of tiles in which a region is associated with every point, in such a way that each region is closer to a point than to any other point in the field. The bounding line segments form the Voronoi diagram. If points with common line segments are connected then the Delaunay triangulation is formed. The vertices of the Voronoi diagram are at the centres of the circles, each passing through three points which form an individual triangle in the triangulation. One of the properties of such a construction is that no point lies inside the circumcircle of a triangle. It is this feature that is used for the construction of the triangulation. The Delaunay triangulation of an arbitrary set of points is the most equiangular, or smoothest triangulation that can be formed with that set.

The basic building block of the grid generator is a routine which allows to add a point to an existing Delaunay triangulation. The result of this operation is again a Delaunay triangulation. This routine can be split into several simple steps [4]. The first step is a search over the triangles to find all of the triangles inside whose circumcircle the new point lies. These triangles are deleted from the triangulation creating a hole or insertion polygon. The new point is then added to the mesh by connecting every point of the insertion polygon with the new point. One can prove that the resulting triangulation is again a Delaunay triangulation.

First grid

The construction of the first grid can be decomposed into several steps. First a description of the domain boundary has to be given. For the NACA-0012 airfoil a analytic equation is known. The farfield boundary is taken to be a rectangle. The second step is to introduce points on the boundaries. This is done based on local curvature and on local grid spacing information. A maximum edgelength L_{max} is defined on every boundary. On a polygon boundary the points are equally distributed with all edges smaller than L_{max}. On an airfoil points are placed so that $\rho L_{local} = constant$. The maximum value of L_{local} is L_{max} and the minimum value is $L_{max}/5$.

The third step is to create a Delaunay triangulation of the set of boundary points. The grid that results typically is atrocious, but this is of no concern as the subsequent insertion of interior points into the mesh dramatically improves the mesh quality. To construct this grid a small procedure is followed. Four points defining a rectangle containing all the boundary points are used to construct two triangles. Now all the boundary points are added to this triangulation using the earlier described algorithm. At this stage the whole rectangle is triangulated, the inside of the domain, as well as the outside. A visual check is done to make sure all the boundary edges appear in the triangulation. Now the triangles that lie outside the domain are removed from the grid. To detect these triangles extra points are added outside the domain, but inside the rectangle. Triangles that are formed with at least one of these points are removed. The extra points give the notion of an outer normal to the domain.

What we have now, is a Delaunay triangulation of the domain defined by boundary points only. In the fourth step the grid is improved by adding points into the interior. Indeed, triangles of which the area is too big might appear in the grid. For this, points are added, and placed in the centre of the circumcircle of the triangle that is selected by the area criterion. Again the previous described algorithm is used to include this new point. Points are added until all triangles have an area that is less than a certain value.

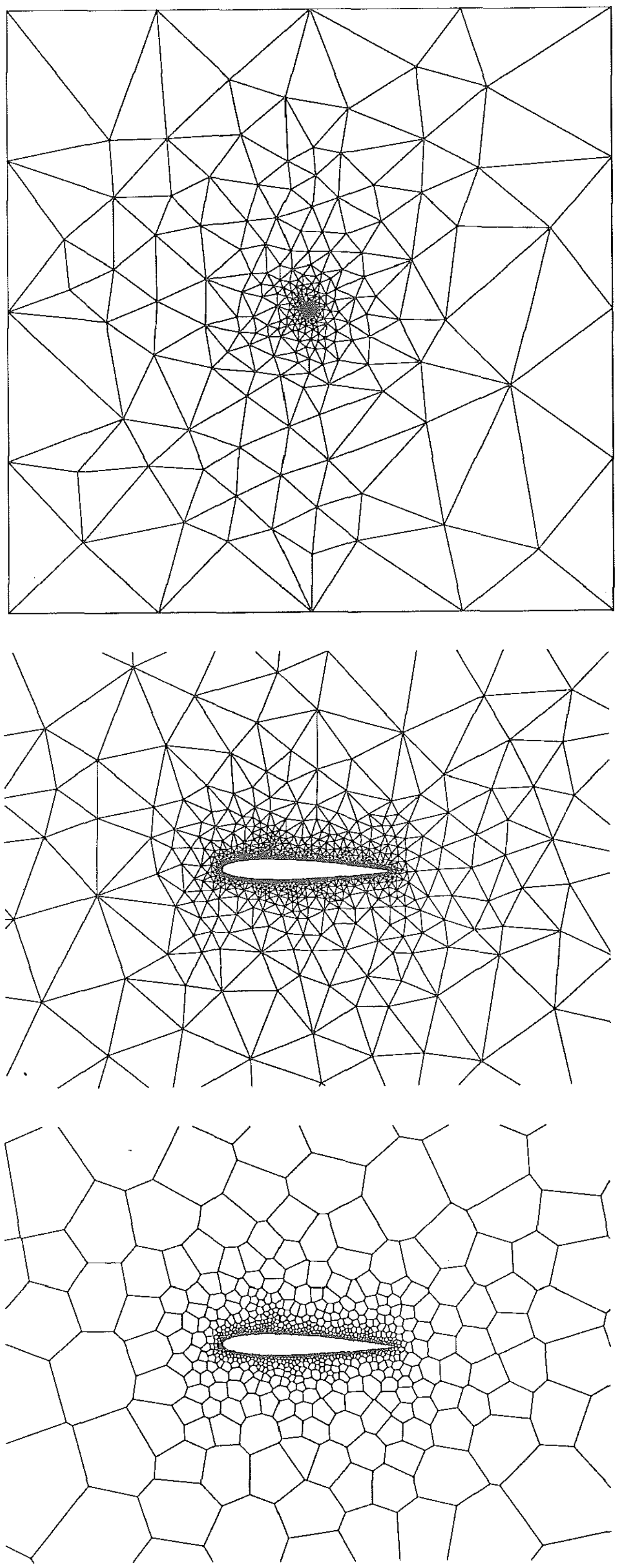
Figure 2: First grid generated based on geometrical criteria. Top: global view; Middle: blow-up of the airfoil region; Bottom: control volumes in the airfoil region.

During the fifth step, points are added, again in the centre of the circumcircle of a triangle. These triangles are now selected by the aspect-ratio criterion. As a measure of the aspect-ratio, the ratio of the circumcircle radius over twice the incircle radius is used. This gives a measure of the skinniness of the triangle. Triangles are refined until every triangle has an aspect-ratio less than 1.6 or has an area less than a minimum value, or until the triangle cannot be further refined. Figure 2 shows the first grid. The side of the square is 200 times the chord of the airfoil.

Adaptivity - mesh refinement

One of the most important advantages of unstructured grids is the possibility to refine locally the mesh during the computation. Successive mesh concentrations in critical zones may be performed, without knowing them at the mesh creation time. The optimal way is to start the computation on a coarse and smooth grid which is not at all specific to a hypothetical solution, and refine during the run, according to criteria specific for the particular physical solution. In the physical problem treated here, transonic flow over an airfoil, interesting regions are of course the shockwaves, which have to be precisely located and also well quantified, stagnation regions, and the trailing edge zone where a contact discontinuity may exist. Criteria can be defined characterizing such flow features. The criteria used here are of local nature. This means that they are calculated with local flow parameters.

The first criterion is based on the pressure difference over an edge. If

$$|p_i - p_j|L_{i,j} \geq \frac{\sqrt{2S_{min}}\;|p_{max} - p_{min}|}{C_p},$$

then the edge i, j will be refined by placing a new point into the centre of the edge. In this formulation p_i is the pressure at node i, $L_{i,j}$ is the length of the edge. The variables in the right hand side are minimum or maximum values taken over all the nodes. S_{min} is the area of the smallest triangle, and C_p is the sensitivity constant for the pressure criterion. This criterion triggers shockwaves and stagnation regions to be refined. The other criterion is based on the entropy difference over an edge, similar to the first criterion:

$$|s_i - s_j|L_{i,j} \geq \frac{\sqrt{2S_{min}}\;|s_{max} - s_{min}|}{C_s},$$

where s is the entropy. This criterion triggers shock waves and tangential discontinuities. To include the new points in the mesh, the earlier described procedure is used. In the refinement criteria, the right hand side of the inequalities are constructed in such a way that the same sensitivity constant can be used on all meshes. The refinement procedure is illustrated in the following figures.

Results

Starting from the grid shown in figure 2, three succesive stages of refinement were done using the foregoing criteria. Figures 3,4 and 5 show respectively the second, third and fourth grid together with the obtained solution in the form of iso-Mach lines. The NACA-0012 profile has an angle of attack of 1.25 degrees and the Mach number of the incoming flow is 0.80.

The number of points in the succesive grids are: 1278, 2159, 4157, 8860. The sensitivity constants are $C_p = 1.0$ and $C_s = 2.2$.

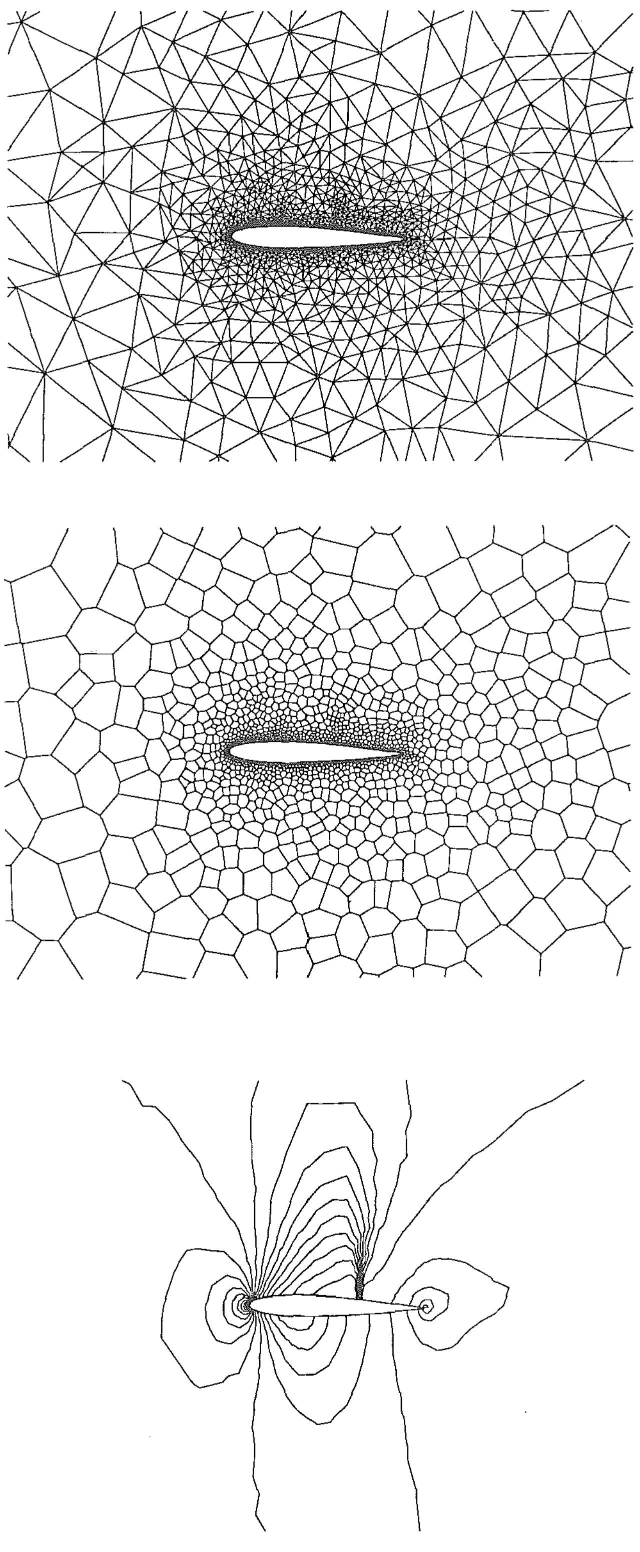

Figure 3: Second grid, top: triangular mesh; middle: control volumes; bottom: iso-Mach lines per 0.05.

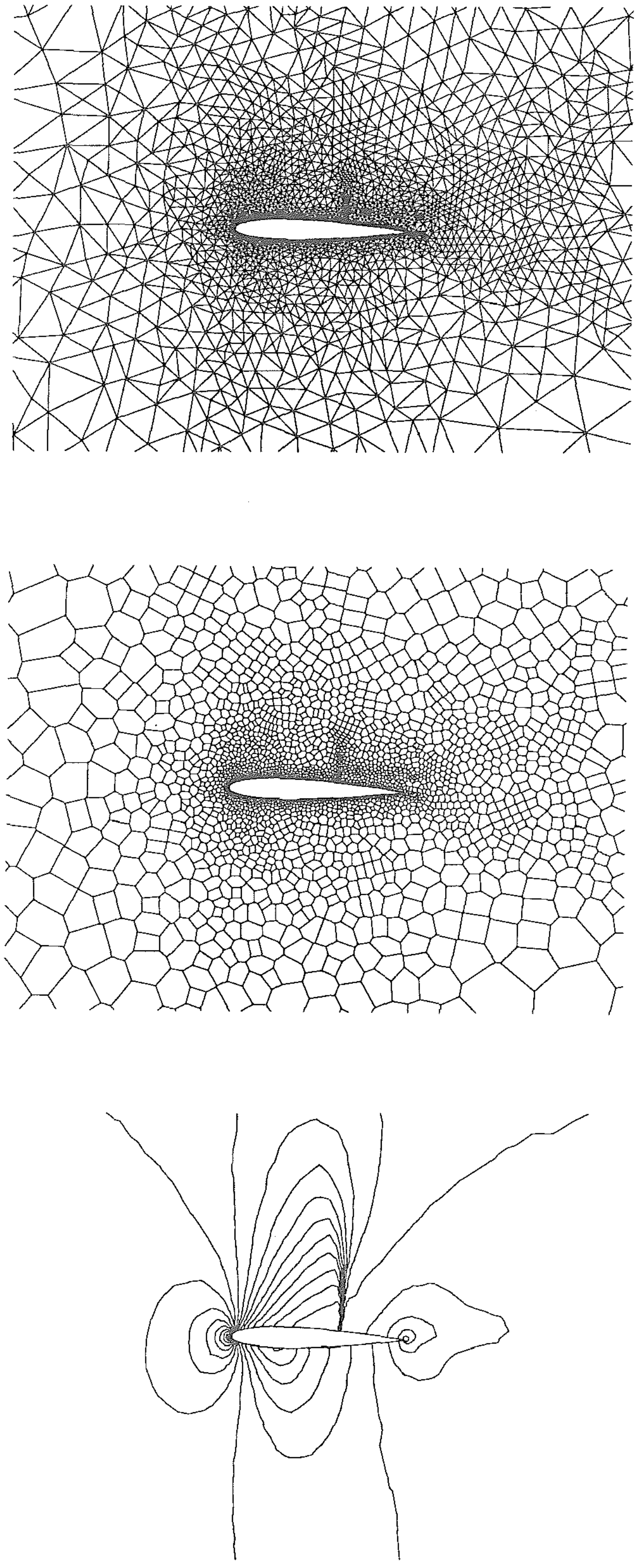
Figure 4: Third grid, top: triangular mesh; middle: control volumes; bottom: iso-Mach lines per 0.05.

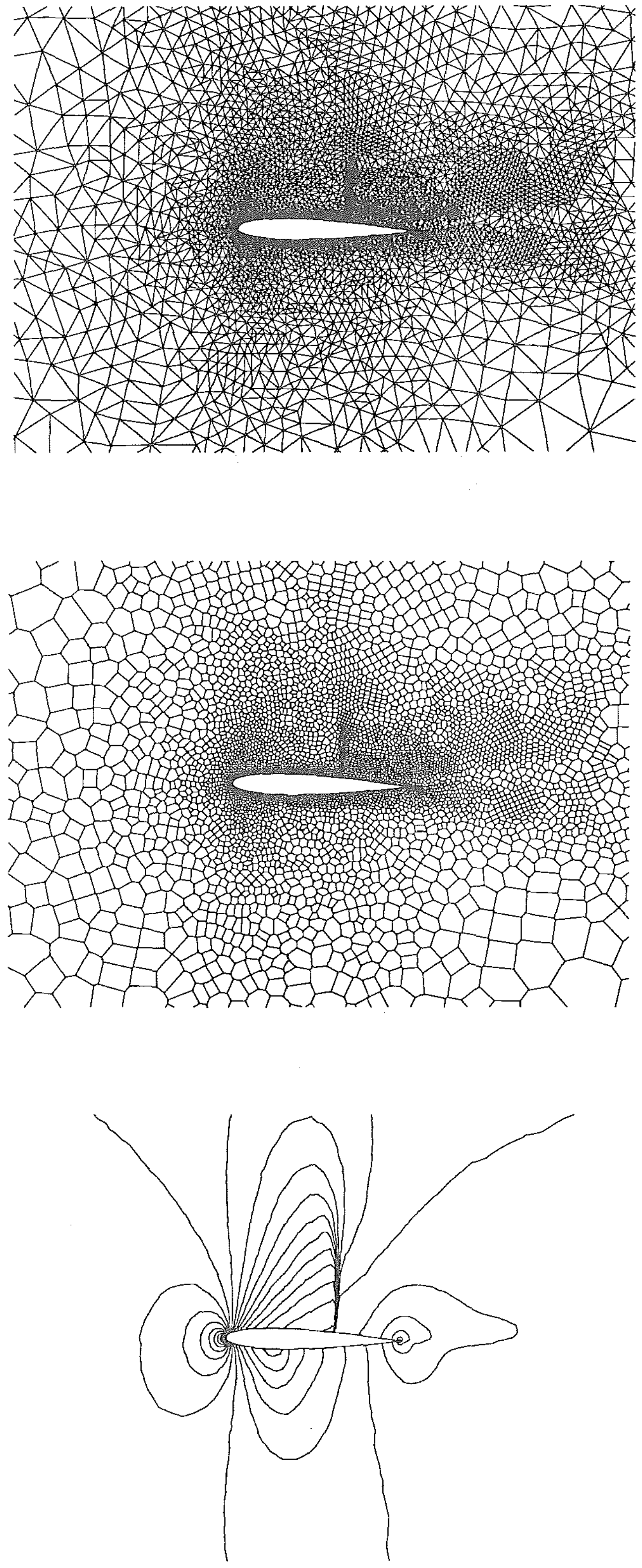
Figure 5: Fourth grid, top: triangular mesh; middle: control volumes; bottom: iso-Mach lines per 0.05.

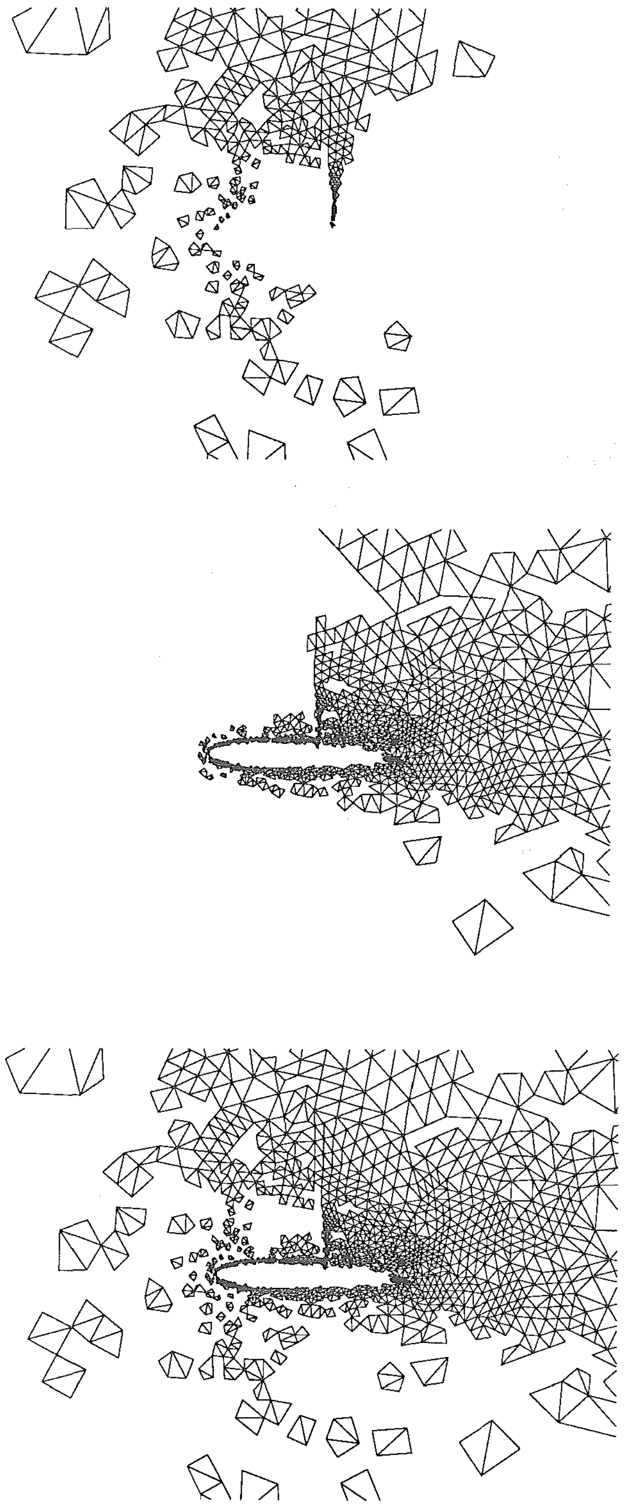

Figure 6: Refinement from third to fourth grid: top: triangles selected for refinement due to the pressure criterion; middle : idem due to the entropy criterion; bottom : idem due to both criteria.

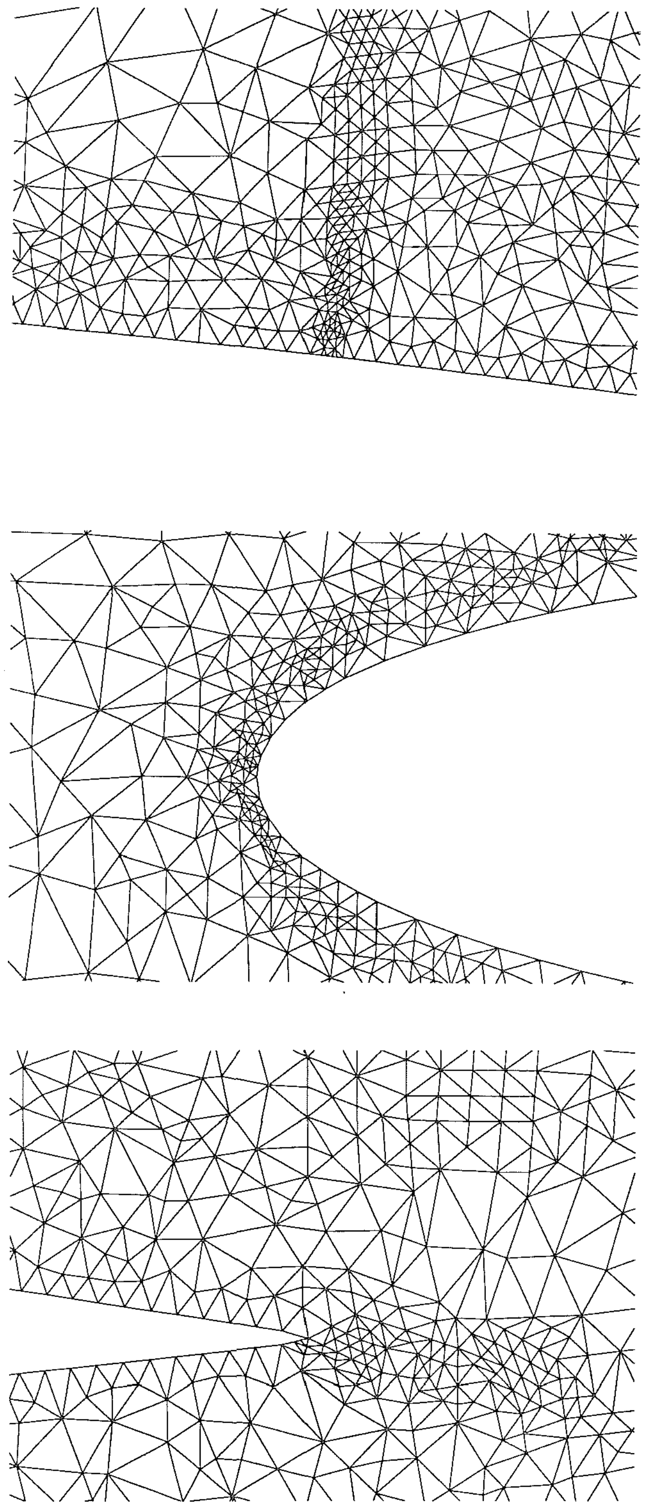

Figure 7: Exploded views of the final grid: top: shock region on suction side; middle : leading edge region; bottom : trailing edge region.

Figure 6 shows the refinement on the third grid. Only the triangles selected for refinement are shown.

Figure 7 shows the final grid structure in the shock region on the suction side, in the leading edge and the trailing edge regions. The shock is sharply resolved. At the leading edge the refinement is mainly due to the entropy criterion (figure 6). This means that some false entropy is created by the algorithm. The formation of an artifical entropy layer can be seen in the grid structure.

Figure 8 shows the convergence history. The logarithm of the maximum residual of all equations over all control volumes is shown as function of work units. The work unit is the computational effort for one stage in the three-stage stepping for 1000 points.

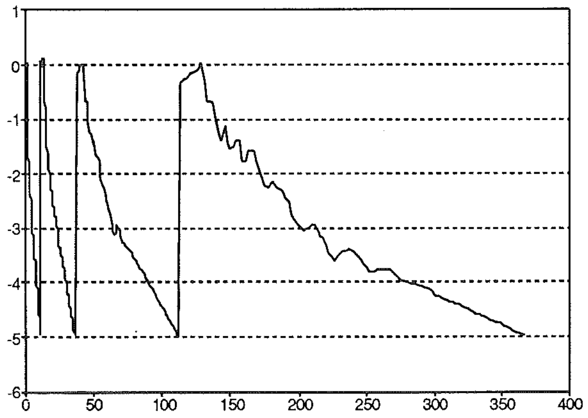

Figure 8: Convergence history: log of residual as function of work units.

Conclusion

By the combination of an adaptive unstructured grid generation technique, suitable refinement criteria and a high resolution flow solver, high quality solutions for Euler equations can be obtained.

Acknowledgement

The research reported here was granted under contract IT/SC/13, as part of the Belgian National Programme for Large Scale Scientific Computing and under contract IUAP/17 as part of the Belgian National Programme on Interuniversity Poles of Attraction, both initiated by the Belgian State, Prime Minister's Office, Science Policy Programming.

References

(1) Dick, E. (1988). A flux-difference splitting method for steady Euler equations. *J. Comp. Phys.*, *76*, 19-32.

(2) Roe, P. L. (1981). Approximate Riemann solvers, parameter vectors and difference schemes. *J. Comp. Phys.*, *43*, 357-372.

(3) Chakravarthy, R. S., Osher, S. (1985). A new class of high accurate TVD schemes for hyperbolic conservation laws. *AIAA paper*, *85-0363*.

(4) Bowyer, A. (1981). Computing Dirichlet tessellations. *Computer Journal*, *24*, 162-166.

(5) Mitty, T. J., Baker, T. J., Jameson, A. (1991). Generation and adaptive alternation of unstructured three-dimensional meshes. *Proc 3rd. Int. Conf. on Numerical Grid Generation in CFD and related Fields*, Barcelona, North-Holland, Amsterdam, 3-12.

(6) Mavriplis, D. J. (1991). Unstructured and adaptive mesh generation for high Reynolds number viscous flows. *Proc 3rd. Int. Conf. on Numerical Grid Generation in CFD and related Fields*, Barcelona, North-Holland, Amsterdam, 79-92.

(7) Weatherill, N. P., Soni, B. K. (1991). Grid adaption and refinement in structured and unstructured algorithms. *Proc 3rd. Int. Conf. on Numerical Grid Generation in CFD and related Fields*, Barcelona, North-Holland, Amsterdam, 143-157.

(8) Dick, E. (1990). Multigrid formulation of polynomial flux-difference splitting for steady Euler equations. *J. Comp. Phys.*, *91*, 161-173.

(9) Dick, E. (1991). Multigrid methods for steady Euler and Navier-Stokes equations based on polynomial flux-difference splitting. *Multigrid Methods III*, Int. Series on Numerical Mathematics, *98*, Birkhauser Verlag Basel, 1-20.

(10) Van Leer, B., Tai, C. H. and Powell, K. G. (1989). Design of optimally smoothing multi-stage schemes for the Euler equations. *AIAA-paper,89-1983*.