GEMINI - General Missile Navier-Stokes Integrator

Johan Sowa
SAAB Military Aircraft
Linköping, Sweden

Abstract

A new integrated tool for flow analysis of supersonic vehicles, primarily missiles, has been developed. The system takes a geometry description on standard format as input, automatically generates computational grids and performs space marching Euler and Navier-Stokes calculations in an efficient manner. The geometry format allows very general configurations. Application examples are presented here to validate and exemplify the quality and different use of CFD results obtained with GEMINI. The Euler equation are solved for three closely related configurations for M=2 and angles of attack up to 30 degrees. The lift and moment curves reveal that delta effects as well as absolute levels can be accurately computed with the Euler equations. The Navier-Stokes equations are applied to three different configurations: Heat transfer rate are computed on an ogive-cylinder at M=6, laminar separation on a yawed cone at M=8 and the flow field around a missile with a highly swept delta wing at M=2.

Introduction

CFD is gradually becoming an engineering tool in aerospace industry. This is especially true for supersonic flow analysis where space marching techniques take advantage of the downstream propagation of all information, which reduces computational times considerably.

Space marching Navier-Stokes is confined to flow attached in the streamwise direction. However, cross flow separation, such as body and leading edge vortices are accurately captured. The Baldwin-Lomax turbulence model with the Degani-Schiff modification constitutes a satisfactory model for such flows.

Another aspect which is crucial for engineering applications is the time for generating a computational grid.In GEMINI great effort has been put into the grid generator which automatically generates high quality structured, multizonal computational grids around a wide class of missile and aircraft configurations. The user can interactively modify the grid through a few parameters before launching the computation. The rapid grid generation normally enables us to have computational results within 24 hours after the geometry file is complete.

This paper presents the grid generation and computational method in GEMINI along with results from Euler and Navier-Stokes calculations of several different configuration to show the versatility and accuracy of GEMINI.

Equations

The Navier-Stokes equations

The full Navier-Stokes equations in non-dimensional form read

$$\frac{\partial U}{\partial t}+\frac{\partial F}{\partial x}+\frac{\partial G}{\partial y}+\frac{\partial H}{\partial z} = \frac{M_\infty}{Re}\left(\frac{\partial F_v}{\partial x}+\frac{\partial G_v}{\partial y}+\frac{\partial H_v}{\partial z}\right)$$

where

$$U = \left[\rho \;\; \rho u \;\; \rho v \;\; \rho w \;\; E\right]^T$$

$$F = \left[\rho u \quad \rho u^2+p \quad \rho uv \quad \rho uw \quad u(E+p)\right]^T,$$

$$G = \left[\rho v \quad \rho uv \quad \rho v^2+p \quad \rho vw \quad v(E+p)\right]^T,$$

$$H = \left[\rho w \quad \rho uw \quad \rho vw \quad \rho w^2+p \quad w(E+p)\right]^T,$$

$$F_v = \left[0 \quad \tau_{xx} \quad \tau_{xy} \quad \tau_{xz} \quad u\tau_{xx}+v\tau_{xy}+w\tau_{xz}-q_x\right]^T,$$

$$G_v = \left[0 \quad \tau_{xy} \quad \tau_{yy} \quad \tau_{yz} \quad u\tau_{xy}+v\tau_{yy}+w\tau_{yz}-q_y\right]^T,$$

$$H_v = \left[0 \quad \tau_{xz} \quad \tau_{yz} \quad \tau_{zz} \quad u\tau_{xz}+v\tau_{yz}+w\tau_{zz}-q_z\right]^T$$

The *ideal gas law* states that

$$E = \frac{p}{\gamma-1}+\frac{1}{2}\rho\,(u^2+v^2+w^2)$$

The viscous stress tensor components are

$$\tau_{xx} = 2\mu\frac{\partial u}{\partial x}+\lambda\nabla\bullet(u,v,w) \qquad \tau_{xy} = \tau_{yx} = \mu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)$$

$$\tau_{yy} = 2\mu\frac{\partial v}{\partial y}+\lambda\nabla\bullet(u,v,w) \qquad \tau_{yz} = \tau_{zy} = \mu\left(\frac{\partial w}{\partial y}+\frac{\partial v}{\partial z}\right)$$

$$\tau_{zz} = 2\mu\frac{\partial w}{\partial z}+\lambda\nabla\bullet(u,v,w) \qquad \tau_{xz} = \tau_{zx} = \mu\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)$$

and the heat flux is

$$-\nabla q = \frac{\mu}{Pr\,(\gamma-1)}\nabla\,(c^2)$$

The coefficients of viscosity are related by *Stokes' hypothesis* $3\lambda+2\mu = 0$. *Sutherland's formula* describes how the viscosity depends on temperature

$$\mu = 1.458\times10^{-6}\frac{T^{3/2}}{T+110.4} \qquad \left[\frac{kg}{ms}\right]$$

The non-dimensional quantities are defined as

$$x = \bar{x}/L \qquad u = \bar{u}/\bar{c}_\infty \qquad \rho = \bar{\rho}/\bar{\rho}_\infty$$

$$p = \bar{p}/(\bar{\rho}_\infty \bar{c}_\infty^2) \qquad \mu = \bar{\mu}/\bar{\mu}_\infty \qquad Re = (\bar{\rho}_\infty \bar{u}_\infty L)/\bar{\mu}_\infty$$

where the physical quantities are barred.

The turbulence model

The Baldwin-Lomax algebraic turbulence model[5] is utilized, augmented with the modification by Degani and Schiff[6] which improves predictions when the flow exhibits cross flow separation.

Method 1: Steady space marching

Explicit MacCormack scheme

SAABs original space marching method[1][2][3] is an integral part of GEMINI. This method solves the steady Euler equations

$$\frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0$$

by means of classical space marching, i. e. the steady equations are discretised and explicitly marched downstream with the MacCormack scheme with added artificial dissipation. The explicit scheme poses a stability criterion on the step size in the marching direction which can be very restricting for low Mach numbers and high angles of attack. One other aspect of the scheme is its low ratio between the work done in the equation solver and the work done in the grid generator. This ratio is much higher for schemes with implicit marching of the steady equations or explicit marching with the unsteady equations one plane at a time (which can be viewed as an implicit marching of the steady equations). This low ratio can be very limiting when the body contour requires elliptic grid generation which makes grid generation much more costly.

For a very large class of missiles this scheme offers a very efficient way of obtaining Euler solutions.

Method 2: Pseudo-unsteady space marching

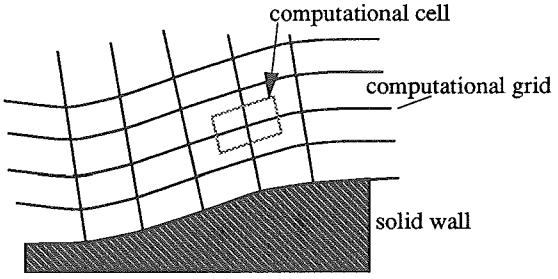
Motivation

The second space marching scheme can be used in Euler or Navier-Stokes (PNS) mode, see also[4]. When Navier-Stokes terms where to be included in GEMINI it was decided to change to the pseudo-unsteady approach to space marching, which means that the unsteady (Euler or Navier-Stokes) equations are iterated towards a steady state in one computational plane at a time. This choice was made of fear of running into severe step size restrictions when marching explicitly. Naturally this will lead to more work in each plane than the steady approach but on the other hand much fewer planes are used. In practise the steady approach gives (Euler) results more rapidly for normal missile configurations. However the ratio of work between equation solving and grid generation is extremely high in the unsteady approach which makes the computational work insensitive to costly grid generation. This, in turn, makes the scheme more efficient for complicated configurations where large amounts of elliptic smoothing of the grid may be necessary.

Another advantage of the pseudo-unsteady approach is the straight-forward extension to treatment of subsonic pockets.

Node centered scheme

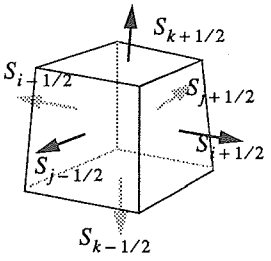
The scheme is node-centered on a body fitted grid, which means that the values are located at the grid points in the grid. A computational cell is constructed around each node point in



which the finite volume scheme is formulated. We call the scheme node-centered since the values are located in the grid nodes but since the node point is located at the center of the computational cell the scheme is also cell-centered in a sense. Dummy points are located outside the computational region to enable slope computations at the boundary points.

Finite volume formulation

Consider the computational cell indexed (i,j,k) whose volume is denoted V_{ijk}. The finite volume discretisation of the Navier-Stokes equations is a balance equation of flow through the cell faces of the computational cell



$$V_{ijk}\frac{\partial}{\partial t}(U_{ijk}) =$$

$$\sum_m \{-[F-F_v, G-G_v, H-H_v]_{m+1/2} \bullet S_{m+1/2}\} +$$

$$\sum_m \{[F-F_v, G-G_v, H-H_v]_{m-1/2} \bullet S_{m-1/2}\}$$

where m loops over i,j and k. The space marching method uses a so called pseudo-unsteady approach which means that the equations are iterated to a steady state in one plane at a time. Before each iteration step the values in the downstream plane i+1 must be assigned a value to close the equation set. Because of the downstream propagation of information we can extrapolate the values from the planes i-1 and i to get the values at i+1

$$u_{i+1} = \alpha u_i + (1-\alpha)\, u_{i-1}$$

where α=2 for Euler computations. For reasons of well posedness[9] we let α=1 in Navier-Stokes computations.

The MUSCL scheme

With W denoting the vector of primitive variables (ρ, u, v, w, p), the inviscid cell face flux is computed as

$$[F, G, H]_{m+1/2} =$$

$$[F(W_{m+1/2}), G(W_{m+1/2}), H(W_{m+1/2})]$$

$W_{m+1/2}$ is the solution of a Riemann problem across the cell face, using Roe's approximate Riemann solver[7]

$$W_{m+1/2} = \Phi(W_{in}, W_{out})$$

W_{in} and W_{out} represent the states just inside and just outside the cell face, respectively.

$$\begin{cases} W_{in} = W_m + f(W_m - W_{m-1}, W_{m+1} - W_m) \\ W_{out} = W_{m+1} - f(W_{m+1} - W_m, W_{m+2} - W_{m+1}) \end{cases}$$

where

$$f(\Delta_-, \Delta_+) = \frac{1}{2}\left(\frac{(\Delta_+^2 + \varepsilon)\Delta_- + (\Delta_-^2 + \varepsilon)\Delta_+}{\Delta_+^2 + \Delta_-^2 + 2\varepsilon}\right)$$

with $\varepsilon = 10^{-40}$.

The 3D Riemann solver

The 3D Riemann problem $\tilde{W} = \Phi(W_L, W_R)$ where L and R denote the left and right sides of the cell face respectively, is reduced to a 1D problem using the cell face normal velocity and the fact that the tangential velocity is passively convected, since the tangential velocity is a Riemann invariant to the two acoustic waves.

The PNS approximation

A space marching procedure is possible if the system of equations is well posed for solution in the marching direction (x). To make the equations parabolic the viscous flux in the marching direction is neglected. In the program we meet this requirement by letting

$$[F_v, G_v, H_v] \bullet S_{i-1/2} = [F_v, G_v, H_v] \bullet S_{i+1/2} = 0$$

Since only the inviscid terms are present in the marching direction space marching is possible if M>1. In the boundary layer where M tends to 0 we apply the Vigneron technique[8] to ensure well posedness for space marching. The inviscid flux in the marching direction is split into

$$[F, G, H] \bullet S_{i+1/2} = \begin{bmatrix} \rho U \\ \rho u U + \omega p S_x \\ \rho v U + \omega p S_y \\ \rho w U + \omega p S_z \\ U(E+p) \end{bmatrix} + \begin{bmatrix} 0 \\ (1-\omega) p S_x \\ (1-\omega) p S_y \\ (1-\omega) p S_z \\ 0 \end{bmatrix}$$

and then the second vector is dropped. In the flux vector

$$U = u S_x + v S_y + w S_z$$

and p is taken from the Riemann solution across the cell face. Well posedness is assured if

$$\omega = min\left(1.0, \frac{\gamma M_i^2}{1 + (\gamma - 1) M_i^2}\right)$$

where M_i is the Mach number normal to the cell face in the marching direction. ω is of course chosen to be 1 if the above expression exceeds 1, which it does if $M_i \geq 1$. It is necessary to apply the Vigneron technique also to the upstream cell face for conservation reasons.

Discretisation of viscous terms

The viscous terms are evaluated with the compact, "staggered" scheme as in [4] For evaluating

$$[F_v, G_v, H_v] \bullet S_{m+1/2} \ .$$

we compute all necessary derivatives located on the cell face

$$(\nabla \rho, \nabla u, \nabla v, \nabla w, \nabla p)_{m+1/2}$$

We introduce an additional cell whose center is the central point of the cell face $m + 1/2$. For m=j the point (i,j+1/2,k) is located approximately half way between the node points (i,j,k) and (i,j+1,k) on the cell face j+1/2. The point is at the center of an auxiliary cell, denoted Ω, with volume $\underline{V}$ and cell face vectors

$$\underline{S}_{i-1/2}, \underline{S}_{i+1/2}, \underline{S}_j, \underline{S}_{j+1}, \underline{S}_{k-1/2}, \underline{S}_{k+1/2} \ .$$

The volume and cell face vectors for this auxiliary cell is computed as the average of the corresponding quantities for cells (i,j,k) and (i,j+1,k). The gradients at the cell face is computed by

$$\begin{aligned} \underline{V}(\nabla u)_{i,j+1/2,k} = {} & u_{i-1/2,j+1/2,k} \cdot \underline{S}_{i-1/2} + \\ & u_{i+1/2,j+1/2,k} \cdot \underline{S}_{i+1/2} + u_{i,j,k} \cdot \underline{S}_j \\ & u_{i,j+1,k} \cdot \underline{S}_{j+1} + \\ & u_{i,j+1/2,k} \cdot \underline{S}_{k-1/2} + u_{i,j+1/2,k+1/2} \cdot \underline{S}_{k+1/2} \end{aligned}$$

The indices on the variables indicate their "location" in space. We note that $u_{i,j,k}$ and $u_{i,j+1,k}$ are node values and can be used directly. The other four values are obtained by averaging, e. g.

$$u_{i-1/2,j+1/2,k} = \frac{u_{i,j,k} + u_{i-1,j,k} + u_{i,j+1,k} + u_{i-1,j+1,k}}{4}$$

Temporal discretisation

Denote the right hand side of the semi-discrete approximation

$$\frac{\partial U_{ijk}}{\partial t} = R_{ijk}(U)$$

An r-stage Runge-Kutta time stepping scheme is used iterating towards a steady-state

$$U_{ijk}^{(m)} = U_{ijk}^{(0)} + \Delta t \cdot \alpha_m \cdot R_{ijk}(U^{(m-1)}) \qquad , m = 1 \ldots r$$

where one time step is defined by

$$U_{ijk}^{(0)} = U_{ijk}^n \qquad U_{ijk}^{n+1} = U_{ijk}^{(r)} \ .$$

The α values can be selected for optimal convergence rate to steady state. The local time step is chosen a

$$\Delta t = \Delta t_{ijk} = min(\Delta t_I, \Delta t_V)$$

where the time step based on the viscous[10] part of the operator is

$$\Delta t_V = \frac{(\gamma - 1) Re Pr \rho}{M_\infty \gamma \mu} \Delta x^2$$

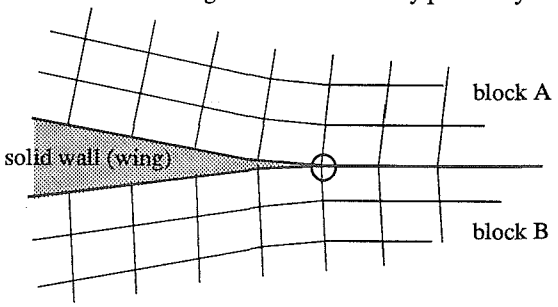
where Δx is the diameter of the biggest sphere that fits into the computational cell. The time step based on the hyperbolic operator is

$$\Delta t_I = min\left(\frac{V_{ijk}}{(u_n + c)\,|S_m|}\right)$$

where m loops over the indices of the cell faces.

A special case

Points located on the boundary between two blocks are doubly defined after an iteration. For most points there is no conflict in this fact since the iteration will yield the same value in both points as long as the dummy point values are properly assigned. There are however cases where two different values will be the result of one iteration such as the situation depicted below where the left neighbor of the boundary point may have



different values in the two blocks. To determine the value which should be shared by the two points a Riemann problem is solved with the two conflicting values as input.

Boundary conditions

Since the scheme is node centered we can apply the non slip condition directly on the solid wall points. The wall temperature is set as fixed or given from an adiabatic wall condition. The pressure gradient normal to the wall is set to zero with a second order accurate formula. For Euler calculations the solid wall conditions uses the formulas for oblique shocks or Prandtl-Meyer expansions.

Automatic grid generation

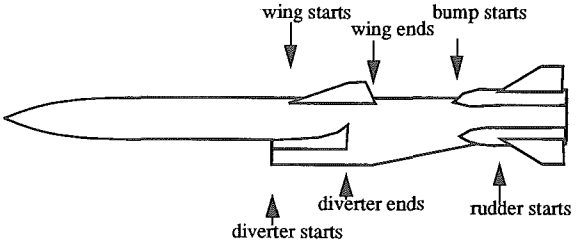
Geometry format

The geometry of the body is given as points on contours of constant x. Guide points help to generate good spline approximations also over sharp edges. A body contour at a specified x is computed by two unidirectional spline interpolations in the geometry data.

Wings are defined as profiles at spanwise locations. The points are given in a coordinate system local to the wing which is later transformed into the missile coordinate system with translations and rotations, defined by a few parameters. Cuttings between wings and body contours are computed by the program and it is therefore only left to the user to specify the deflection angle or the axis location. It is therefore a simple task to e. g. move a wing around on the body or to deflect a control surface. The wings may have an arbitrary plan form and must not even intersect the body. This allows the computation of deflected controls with a gap between the root chord and the body.

Grid sections

The missile is automatically divided into *grid sections* based on the geometry. Throughout each grid section the multi-



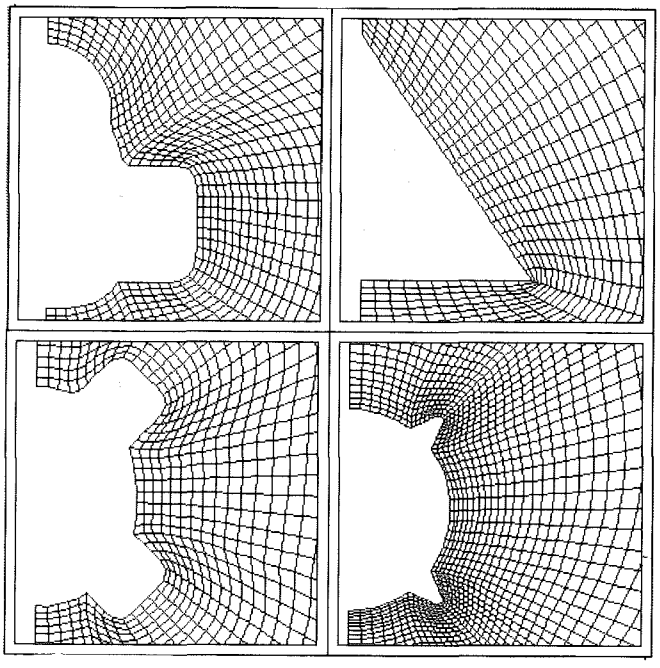
The missile is divided into grid sections

block structure is fixed. The values are transferred to the next grid section by an interpolation, which allows the grid to be completely discontinuous across grid section boundaries.

Cross plane grids

Each cross plane grid is a multizone, structured grid perpendicular to the body axis. The grid generation is automatic, which means that the program recognizes which geometrical objects are present at the current x and then generates a grid according to the current topology. Then, with an interactive graphical pre processor, the user can adjust the grid through a set of grid parameters. These parameters are stored in an indata file and used by the internal grid generator during the computations.

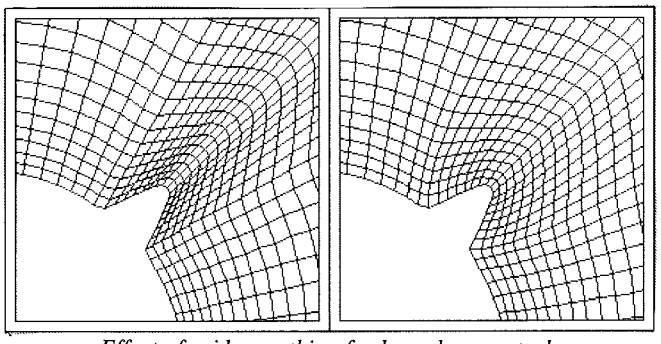
The basic grid is an O-grid wrapped around the body. The body cross section shape is arbitrary and with the boundary orthogonality control, to be described later, a high quality grid can easily be generated around a large class of bodies. Arbitrary body



Arbitrary body shapes are handled by GEMINI

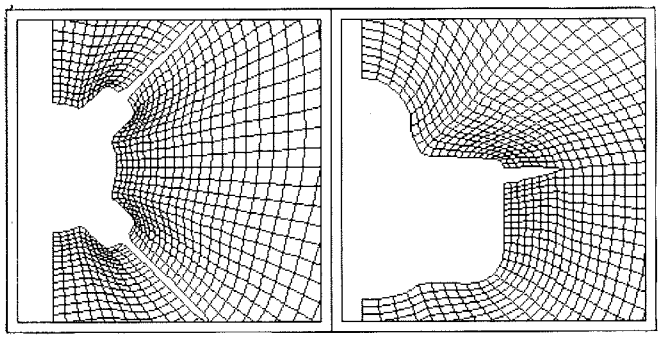
shapes may require elaborate grid generation techniques to obtain a high quality grid minimizing skewness and obtaining or-

thogonality near solid boundaries. A so called grid smoothing technique (described below) fulfills this task automatically.



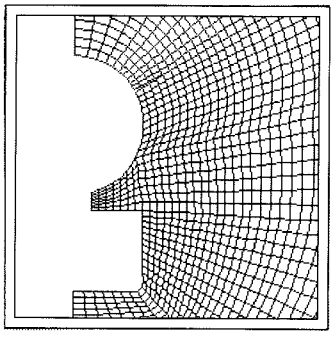
Effect of grid smoothing for boundary control

A thin wing divides the grid into two block. The cutting between the wing and the body is computed internally and need not be specified by the user.



Wings impose multzoning

An additional feature allows the users to add extra blocks or delete a part of the block near the body. In this way can e. g. actuator housings and boundary layer diverters can be modelled.



Additional block for e. g. a boundary layer diverter

Grid generation

After cutting the body, the program computes block boundaries. Grid points are distributed along these boundaries using optional stretching functions and then the interior grid is generated with transfinite interpolation[11] followed by a specified number of smoothing steps with an elliptic operator[11]. One could also view the grid generator as an elliptic one with transfinite interpolation as a means of creating the starting solution for the elliptic solver. The transfinite interpolation can be stated

$$\begin{pmatrix} y \\ z \end{pmatrix}(\xi,\eta) = \bar{r}(\xi,\eta) = [(\Pi_\xi + \Pi_\eta - \Pi_\xi\Pi_\eta)\bar{r}](\xi,\eta)$$

where

$$[\Pi_\xi \bar{r}](\xi,\eta) = \bar{r}(0,\eta)\cdot(1-\varphi_1(\xi)) + \bar{r}(1,\eta)\cdot\varphi_1(\xi)$$

$$[\Pi_\eta \bar{r}](\xi,\eta) = \bar{r}(\xi,0)\cdot(1-\varphi_2(\eta)) + \bar{r}(\xi,1)\cdot\varphi_2(\eta)$$

The functions $\varphi_{1,2}$ contain the stretching. Furthermore they are monotone and satisfy

$$\varphi_{1,2}(0) = 0 \qquad \varphi_{1,2}(1) = 1$$

The elliptic smoother solves the equation

$$g_{22}\bar{r}_{\xi\xi} + g_{11}\bar{r}_{\eta\eta} - 2g_{12}\bar{r}_{\eta\xi} + g(P\bar{r}_\xi + Q\bar{r}_\eta) = 0$$

where

$$g_{11} = x_\xi^2 + y_\xi^2 \qquad g_{22} = x_\eta^2 + y_\eta^2$$

$$g_{12} = x_\xi x_\eta + y_\xi y_\eta \qquad g = (x_\xi x_\eta - y_\xi y_\eta)^2$$

with an SOR scheme. This smoothing procedure has no impact on the cpu-time if the pseudo unsteady equation solver is used since the number of grid planes is relatively small and a large amount of work is done in each plane. On the other hand, if the steady space marching approach is taken the work ratio between equation solving and grid generation is much smaller and thus elliptic smoothing may dominate the cpu-time completely.

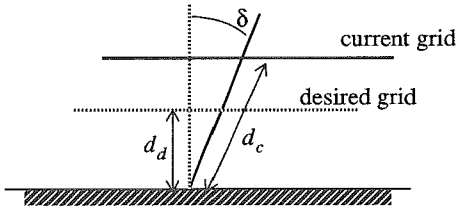
Boundary control

Near solid walls the functions P and Q in the elliptic equation is used to enforce control of orthogonality and cell height. E. g. if j=j1 (the lower limit of j) is a solid wall then

$$P(j,k) = P_1(d_d(j,k) - d_c(j,k))\,e^{-\beta(j-j_1)}$$

$$Q(j,k) = Q_1\delta(j,k)\,e^{-\beta(j-j_1)}$$

where $(d_d - d_c)$ is the difference between the desired cell height and the current one and δ is the deviation from orthogonality in radians



P_1,Q_1 and β are built in constants.

Adaptation

GEMINI performs a simple grid adaptation so that

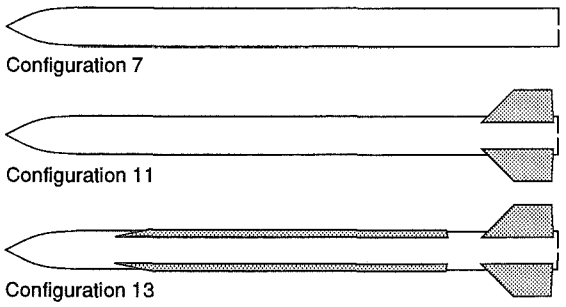
$$y^+ = \frac{\sqrt{\rho_w \tau_w}}{\mu_w} y_1$$

where y_1 is the normal distance out from the solid boundary to the first interior grid point, takes on a specified value in the converged solution in each computational plane. From the users point of view the program takes over the decision about grid stretching in the normal direction. Since y^+ for the first cell is often used as a measure of the resolution of an attached boundary layer, and hence the quality of the solution, this option relieves the user of a trial and error process to get a high quality grid. GEMINI is programmed to check y^+ at certain intervals in the iteration process, regrid the current plane and the downstream plane if required and continue the iteration.

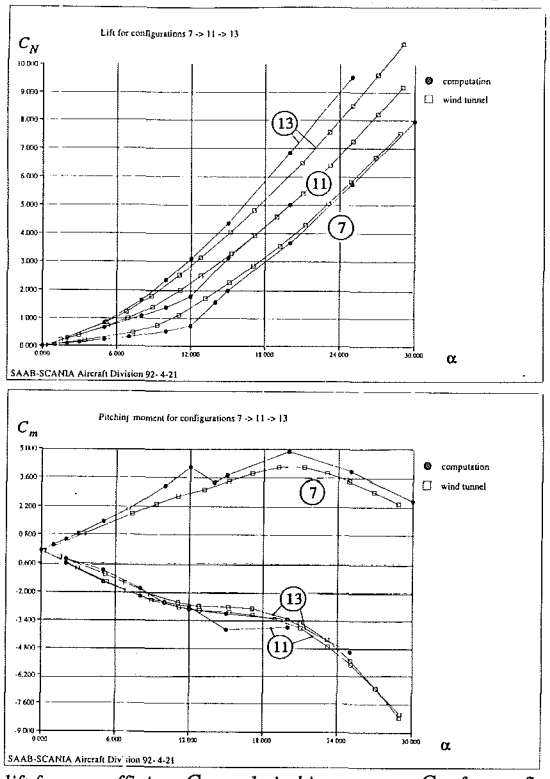
Results

Euler solutions for lift and pitch predictions

A missile consisting of an ogive-cylinder with fins and strakes was subject to a wind tunnel investigation and this wind tunnel data was used for the validation of GEMINIs first Euler solver (the MacCormack scheme)[12] at M=1.93. We present some results as a study where the ogive-cylinder is "dressed on" to asses the quality of the Euler results for different configuration types.



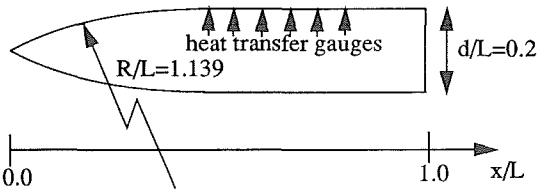
The curves of computed and measured values of lifting force and pitching moment for a "+" configuration are presented below. At a first glance at the curves we see that the delta effects are well predicted as well as the absolute levels of forces and moments. The computed lift and moment on configurations 7 and 11 deviate from measurements for angles of attack around 10 degrees. For angles over 15 degrees the agreement is more satisfying. The deviation is attributed to the viscous separation of body vortices from the smooth cylinder, which can not be predicted by the Euler equations. For higher angles of attack the separation is induced by a cross flow shock and this type of separation can be predicted by the Euler equations. We see that the lift and moment curves for configuration 13 suffers no such deviation since the point of separation is well defined by the strakes. On the other hand the introduction of the strakes seems to overpredict the lifting force for higher angles of attack whereas the moment curve shows a very fine agreement.



The lift force coefficient C_N and pitching moment C_m for configurations 7, 11 and 13 as functions of the angle of attack α (in degrees). Euler solutions.

Heat transfer predictions for laminar and turbulent flow

Navier-Stokes computations were carried out on a circular arc ogive-cylinder for which there are measured data of heat transfer[13]. The study covers a Reynolds number range where



the flow goes from laminar, over transitional to fully turbulent.

The test geometry consists of a circular arc ogive cylinder with a length L=15 inches (=38.1 cm) and a diameter d=3 inches. Coaxial heat transfer gauges are located at x/L=0.467, 0.533, 0.600, 0.667, 0.733 and 0.800. The experiments were conducted under five different test conditions (a test condition is called a T/C)

T/C	M_∞	Re_L	*Type of flow*
1	5.76	$2.0 \cdot 10^6$	*laminar*
2	5.78	$5.8 \cdot 10^6$	*laminar*
3	5.79	$1.3 \cdot 10^7$	*transitional*
4	5.86	$2.6 \cdot 10^7$	*turbulent*
5	5.90	$3.8 \cdot 10^7$	*turbulent*

The temperatures at infinity and at the wall were constant

$$T_\infty = 59.8K \qquad T_w = 290K$$

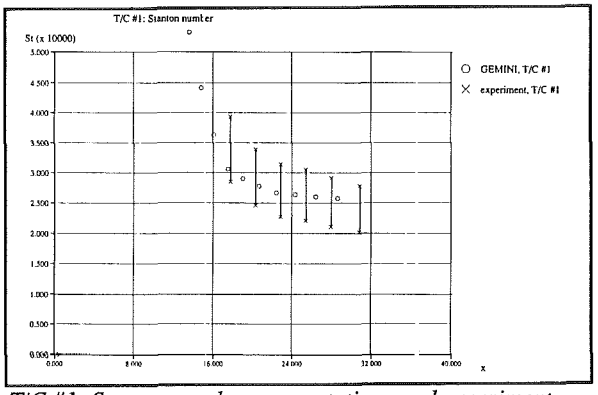
In T/C #3 the transition from laminar to turbulent flow occurred in the interval 0.7<x/L<0.8. In computations the Baldwin-Lomax model was swtched on after x/L=0.74.

The grid adaptation was switched on in all cases except for the transitional one where a constant stretching was held. $y^+ = 1.0$ for the innermost grid point was specified as an adaptation criterion.

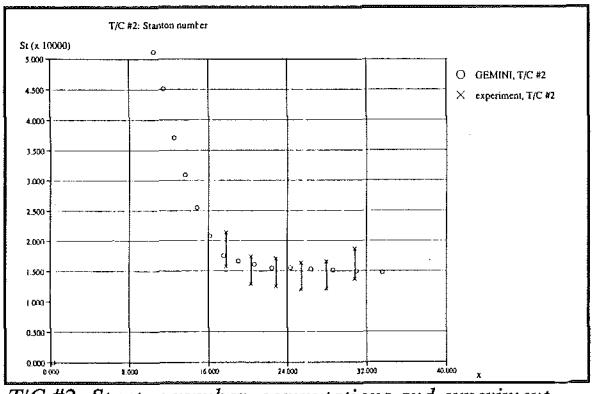
We compare the Stanton number defined by

$$St = \frac{q_w}{\rho_\infty u_\infty c_p (T_{0\infty} - T_w)}$$

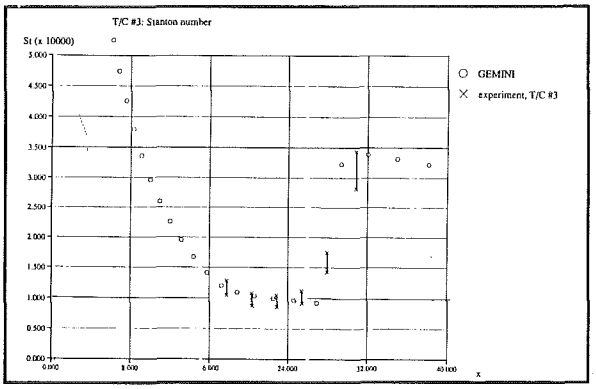
where the heat flux is computed by one-sided differences of second order accuracy. The experimental Stanton numbers[13] uses "edge" values just outside the boundary layer instead of infinity values. We believe that the differences due to the different definitions are very small.



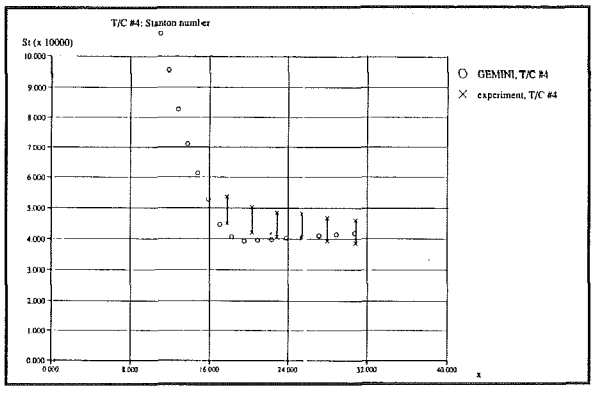
T/C #1: Stanton number, computations and experiment



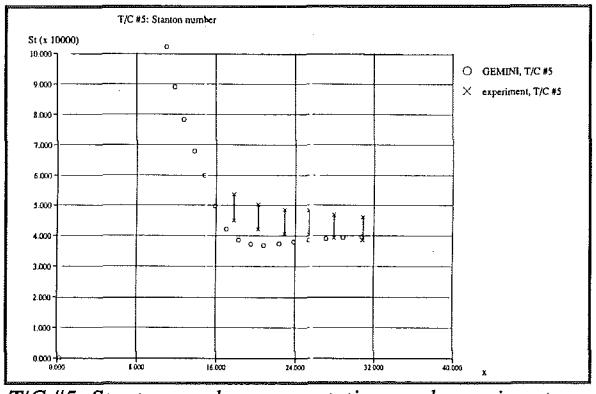
T/C #2: Stanton number, computations and experiment



T/C #3: Stanton number, computations and experiment



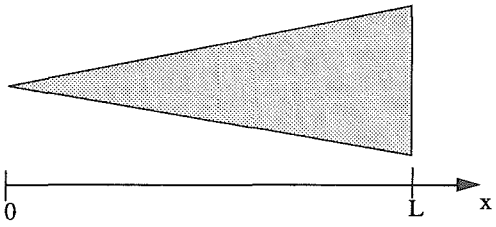
T/C #4: Stanton number, computations and experiment



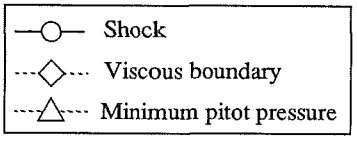
T/C #5: Stanton number, computations and experiment

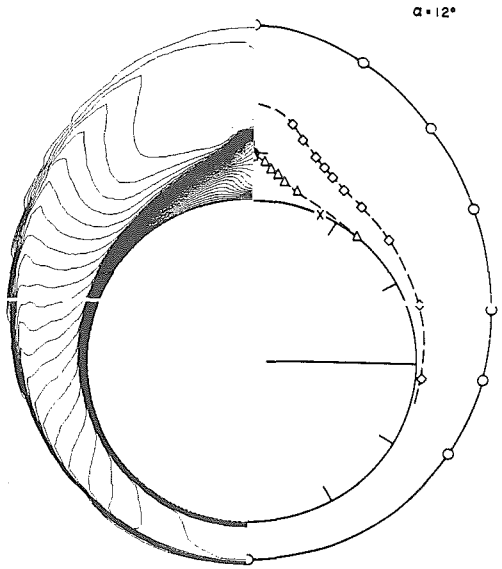
Laminar separation on a yawed cone

The test geometry[12] consists of a circular cone with a 10° semi-apex angle. At x=L a pitot pressure survey of the flow

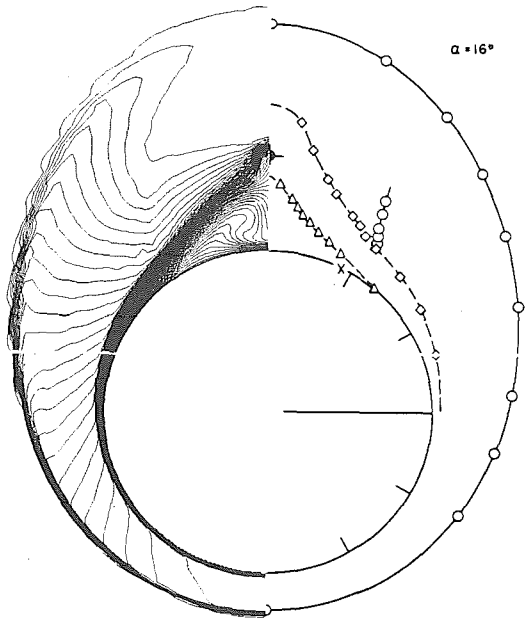


field indicates the positions of the shocks and the viscous boundary. The test conditions are $M_\infty = 7.95$ and $Re_L = 4 \cdot 10^5$ The experimental pitot pressure survey revealed the location of shocks, viscous boundaries and minimum pitot pressure. The following symbols are used to depict the locations.

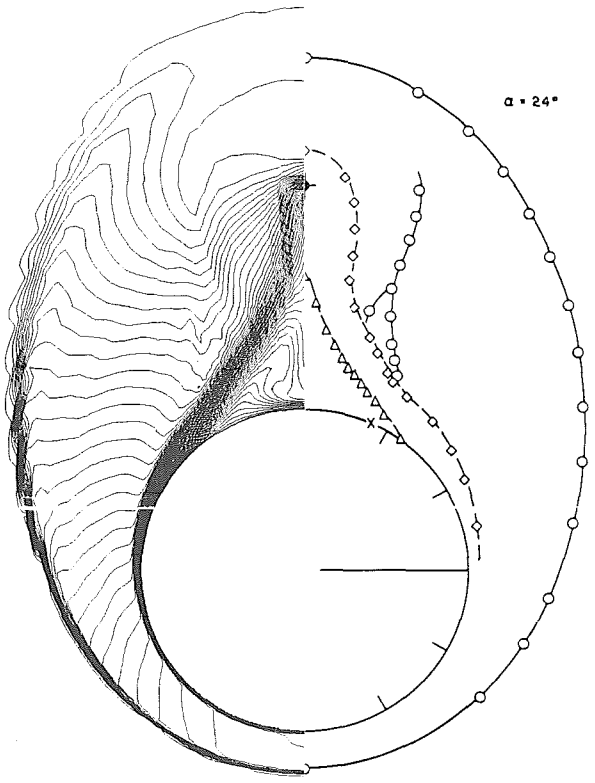




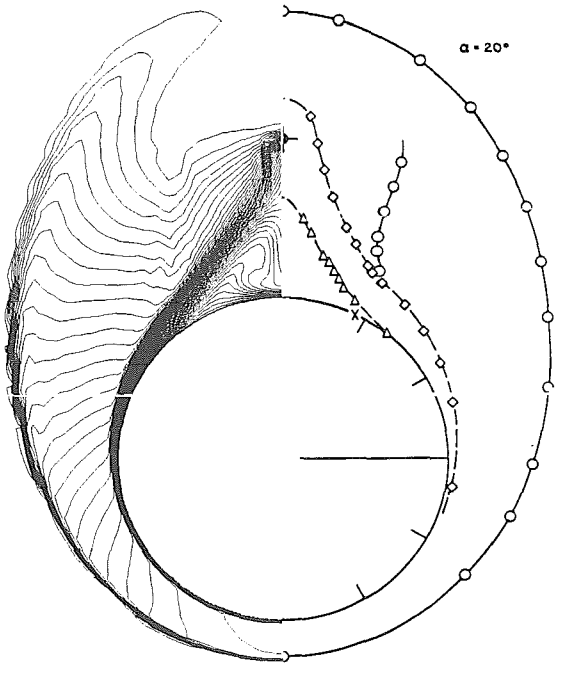
Computed Mach number contours compared with experimental pitot pressure survey for $\alpha = 12°$



Computed Mach number contours compared with experimental pitot pressure survey for $\alpha = 16°$



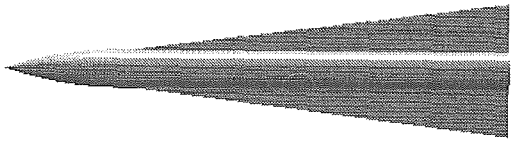
Computed Mach number contours compared with experimental pitot pressure survey for $\alpha = °24$



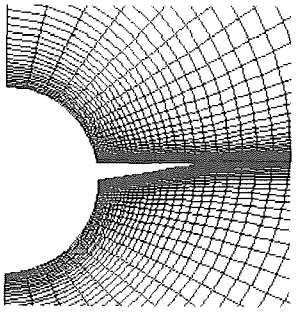
Computed Mach number contours compared with experimental pitot pressure survey for $\alpha = °20$

<u>Navier-Stokes solution on a complete missile</u>

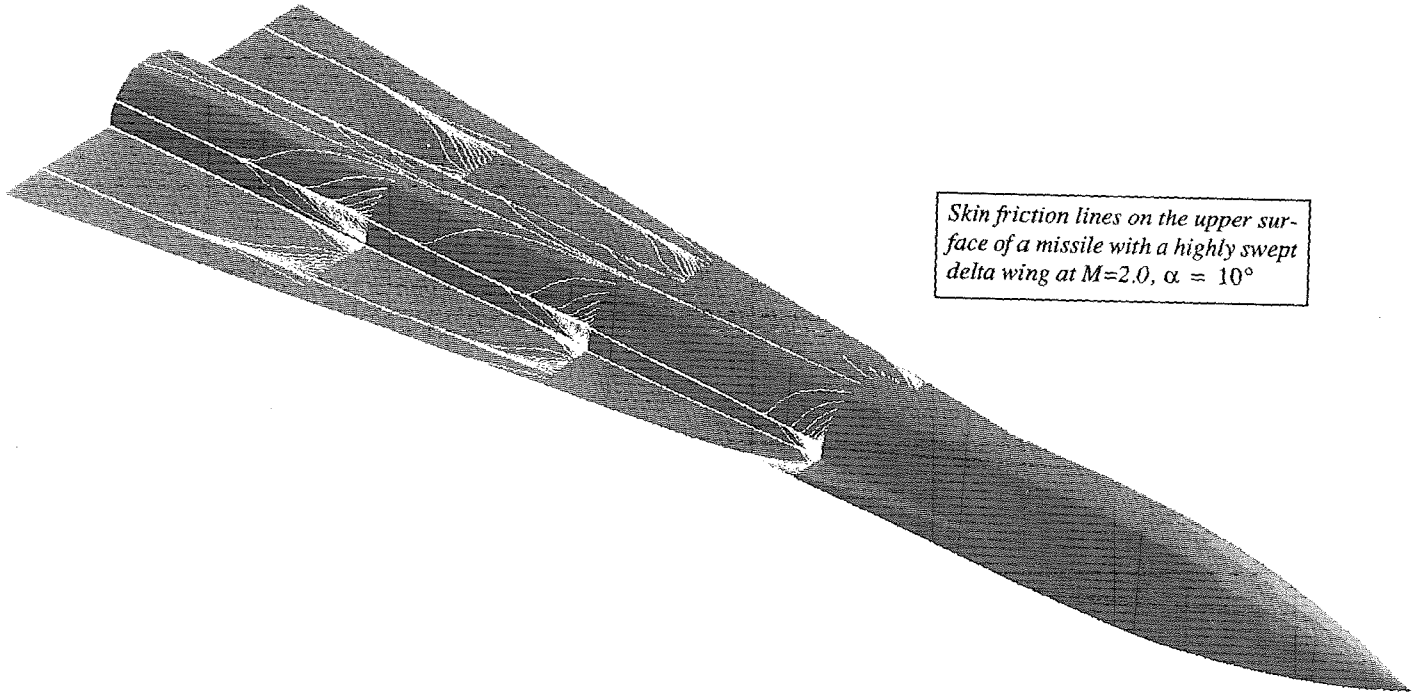
A slender missile with a highly swept wing[13] at M=2.0



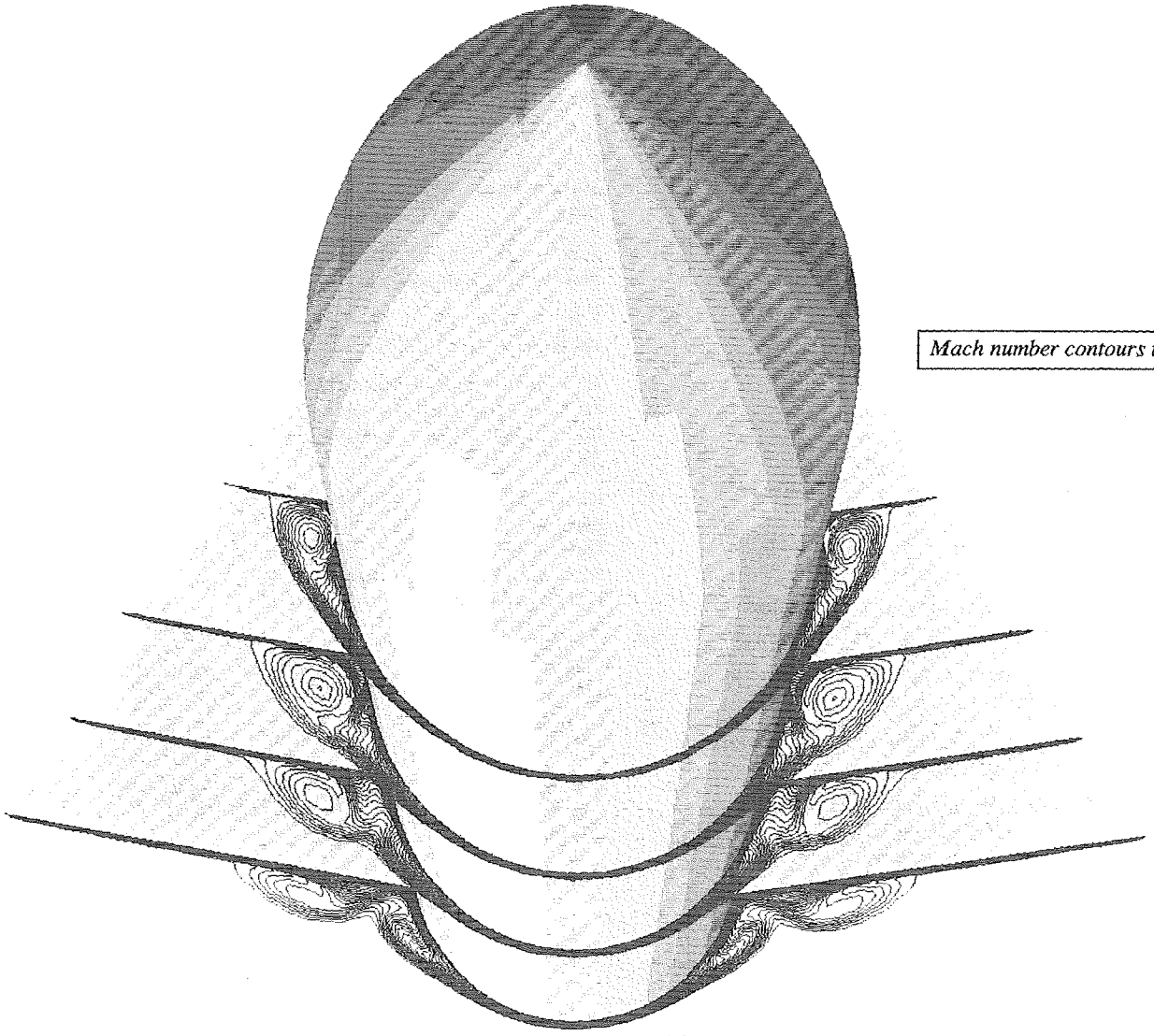
and $\alpha = 10°$ demonstrates a laminar solution to the Navier-Stokes equations on a multiblock configuration. The cross sectional grid comprised 49 points radially and 64 points circumferentially. The results show the skin friction lines on the upper



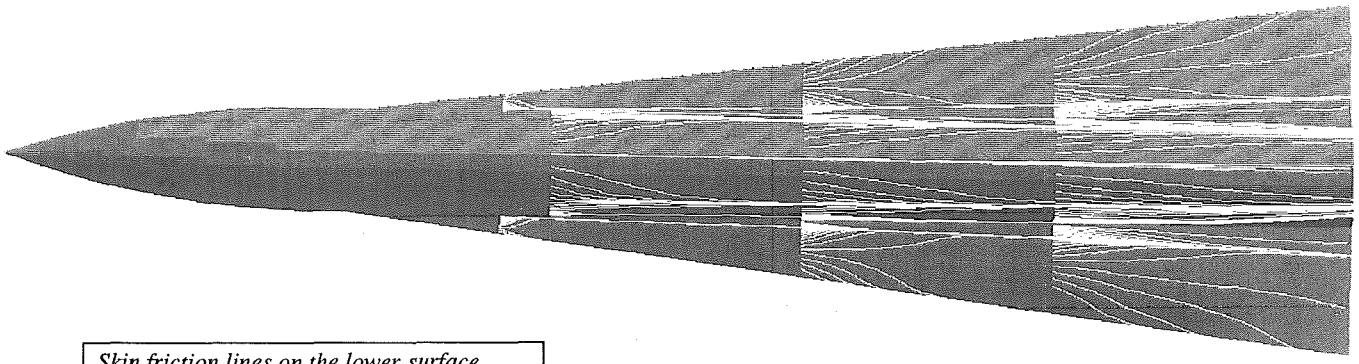
surface of the missile. The Mach contour on the lower side is also shown to reveal the intricate vortex pattern at the body-wing junction.



Skin friction lines on the upper surface of a missile with a highly swept delta wing at M=2.0, $\alpha = 10°$



Mach number contours under the wing.



Skin friction lines on the lower surface.

Conclusions

A Navier-Stokes/Euler codefor supersonic flow has been developed. The method has two Euler solvers and one Navier-Stokes solver. The Navier-Stokes method is based on a pesudo-unsteady approach to space marching with a high resolution upwind scheme for the convective part and a staggered scheme for the viscous terms.

An O-H grid is generated around the vehicle. GEMINI is equipped with an automatic grid generator, with an interface for the user to introduce minor modifications to the grid. By this we practically eliminate the grid generation time and reduce the time from geometry to the first result to in the order of one day. The drawback is, of course, its lack of complete generality. However, a very wide class of missiles (and fighter aircraft) can be captured by the present grid generator.

GEMINI has been applied to several realistic missile configurations with good results. Forces and moments can be predicted with good accuracy by the Euler equations up to high angles of attack. For that purpose it is already in production use at SAAB.

Navier-Stokes computations gives accurate predictions of heat transfer rates and crosswise separation. It will without doubt improve predictions of forces and moments for full configuration, although the computational cost is much higher than for Euler computations.

References

1 B. G. Arlinger, **Computation of Supersonic Flow Around Three-Dimensional Wings**, ICAS-82-6.1.3.

2 B. G. Arlinger, **Computation of Supersonic Flow Around Bodies**, AIAA-84-0259.

3 B. G. Arlinger, **Computation of Supersonic Flow About Complex configurations**, ICAS-84-1.3.2.

4 E. Chaput, F. Dubois, D. Lemaire, G. Moules, J. L. Vaudescal, **FLU3PNS: A Three-Dimensional Thin Layer and Parabolized Navier-Stokes Solver Using the MUSCL Upwind Scheme**, AIAA 91-0728.

5 B. S. Baldwin, H. Lomax, **Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows**, AIAA 78-257.

6 D. Degani, L. B. Schiff, **Computation of Supersonic Viscous Flows Around Pointed Bodies at Large Incidence**, AIAA 83-0034.

7 P. L. Roe, **Approximate Riemann Solvers, Parameter Vectors and Difference Schemes**, Journal of Computational Physics 43, 357-372 (1981).

8 Y. C. Vigneron, J. V. Rakish, J. C. Tannehill, **Calculation of Supersonic Visous Flow over Delta Wings with Sharp Subsonic Leading Edges**, AIAA paper 78-1137.

9 D. S. Thompson, R.J. Matus, **Conservation Errors and Convergence Chracteristics of Iterative Space-Marching Algrithms**, AIAA Journal, Vol.29, No.2, February 1991

10 J. Sowa, **Stability of a Runge-Kutta Method for the Navier-Stokes equations**, BIT 30(1990), 542-560.

11 J. F. Thompson, Z. U. A. Warsi, C. W. Mastin, **Numerical Grid Generation**, North-Holland, 1985

12 H. Åslund, **Validering av MISUMA-beräkningar mot ett vindtunnelprov med modell Rb:m37**, (in Swedish) Report L-0-1 B738, SAAB Military Aircraft, Linköping, Sweden.

13 Richard H. Byers, **Skin Friction and Heat Transfer Predictions for Hypersonic Turbulent Flow Over an Ogive-Cylinder**, Thesis, AFIT/GAE/ENY/90D-4 : AD-A230496, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, 1990.

14 Richard R. Tracy, **Hypersonic Flow over a Yawed Circular Cone**, Thesis, CalTech, Graduate Aeronautical Lab., Memo. 69, 1963.

15 W. Stahl, K. Hartmann und W. Schneider, **Kraft- und Druckverteilungsmessungen an einer Flügel-Rumpf-Kombination mit Flügel kleiner Streckung in kompressibler Strömung** (in German), DLR-FB 72-08, Göttingen