CHAPTER 24

# Multi-Block Euler Method Using Patched Grids*

Torsten Berglind†

**Abstract.** A general purpose Multi-block Euler Solver for Compressible flows MESC1 is developed. The Euler solver is of finite-volume type and uses explicit multi-stage Runge-Kutta time stepping. The basic concept is to express all boundary conditions in terms of image values of the flow variables. An arbitrary topology can then be specified by assigning boundary conditions on all outer surfaces of the computational blocks. A windtunnel model consisting of a 65 deg round leading edge delta wing with a sting is used as a test case. A patched $C^1$-continuous multi-block grid around the wing-sting configuration is constructed by transfinite interpolation. The wing is enveloped by an O-O type of grid with parabolic singularities at the wing tip, and the sting is enveloped by an O-O type of grid with polar singularities at the nose and the tail , with a slit for the wing grid. This grid is divided into eight computational blocks, so that each side of a computational block contains precisely one type of boundary condition. The flow at 10 deg angle of attack and at a free stream Mach number of 0.85, serves as a test case. The computation of the flow around a wing-fuselage shows substantially better agreement with measured data than the corresponding computation around a wing alone. This comparison serves as a first validation of the overall computational procedure.

**1. Introduction.** In recent years, an increasing interest has been shown in flow computations on complex flow regions such as a complete aircraft [9-12]. These flow regions are too complex to be mapped onto a single-block grid, with reasonable skewness and grid smoothness restrictions. Numerical methods using structured data, based on single-block mapping, have to be extended to multi-block transformations. The advantage is not only that grid smoothness properties are improved, but also that grid points can be concentrated in regions of interest more efficiently. The flow region is therefore broken into a set of blocks where each block is a hexahedron in the computational space.The decomposition of the

*This work was sponsored by the Defense Materiel Administration of Sweden.

†FFA, The Aeronautical Research Institute of Sweden, S-161 11 BROMMA

flow region implies that instead of one structured grid, a number of coupled sub-grids are applied. The coupling introduces interface boundaries between contiguous blocks. Grid lines of two adjoining regions may or may not align [8]. A more general type of interface is overlapping grids. The disadvantage with overlapping techniques is that it is difficult to preserve conservation. This work is restricted to patched $C^1$-continuous grids [12], which means that continuity conditions are imposed on grid points, grid spacing and grid line orientation at grid interfaces between adjoining blocks as well as any other grid point. This implies that boundary grid cells at an interface boundary can be computed as interior grid cells. Bookkeeping of the addresses of adjoining grid cells is needed at the interface boundaries.

**2. Grid Generation.** A windtunnel model consisting of a 65 deg round leading edge delta wing with a sting, proposed for the International Vortex Flow Symposium in Stockholm on October 1 to 3, 1986 [14], serves as a test case for the multi-block flow solver. The sting is spool-shaped and is rounded off at the tail in order to enable an O-O topology around the both the sting and the wing. The O-O topology is shown to be very efficient for Euler computations around delta wings [2,5] . The sting geometry is described in terms of analytical expressions, whereas the wing geometry is described by spanwise cuts.

The purpose of grid generation is to distribute grid points in a way such that flow gradients are resolved properly. Grid defects like sudden changes in grid line spacing or grid line orientation as well as very skewed grid cells contribute to additional truncation errors. A combination of high flow gradients and metric discontinuities may generate false vorticity which can destroy the solution. The surface grid at the wing is generated by MESHP44, a standard code [3] for grid generation around delta wings. Spanwise grid lines on the sting surface must conform to both grid points, grid size and grid line orientation at the wing-sting cut. Preliminary grid points close to the sting are generated by transfinite interpolation based on the boundary grid points. The preliminary grid points are then projected onto the sting surface [4] . The surface grid for the wing-sting configuration is depicted in Fig. 1 below.

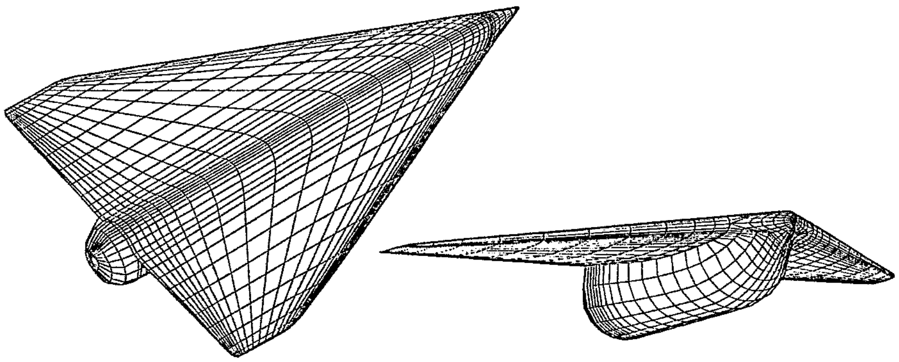

Figure 1. *Surface grid for the wing-sting configuration*

Besides the parabolic singularities at the wing tips and the polar singularities at the nose and the tail of the sting an additional type of mapping singularity appears at the wing-sting cut at both the leading and the trailing edge. The farfield boundary is chosen as a half sphere with a radius of five times the root chord of the wing. The corresponding surface grid is generated in an algebraic fashion, Fig 2.

The interior grid is generated by transfinite interpolation using the far-field and configuration surface grids and the configuration out-of-surface derivative as interpolation conditions [1,2,4]. The sudden change in grid line orientation at the intersection between the wing and the sting on the lower side propagates into the interior domain. The slope discontinuity is smoothed out with a post processor.

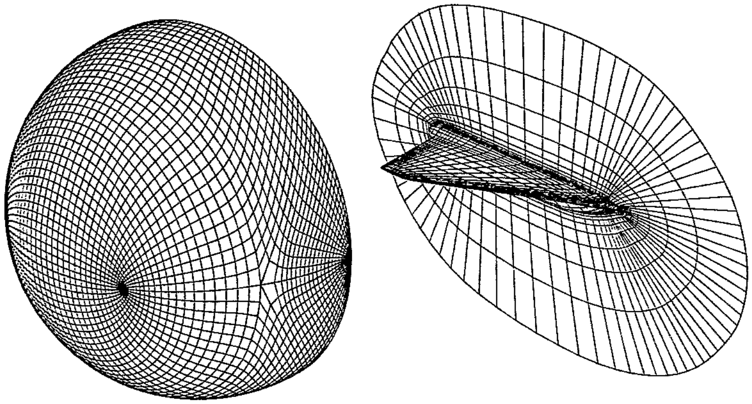

Figure 2. *Oblique view of the 3D grid.*

The topology described above is especially suitable for wing-fuselage configurations for which the fuselage extends far from the wing, because you can add spanwise gridlines on the nose and tail without increasing the size of the wing grid. The polar singularities implies good resolution of the nose and tail where high flow gradients occur. Generally an O-O grid around a delta wing has a slope discontinuity where the spanwise coordinate lines meet the symmetry plane. With a multi-block grid this can be avoided, i.e. the spanwise coordinate lines are all essentially perpendicular to the symmetry plane. Finally, the resolution at the nose and tail is very good due to the polar singularities. This is advantageous since very high flow gradients appear in these regions.

**3. Euler Flow Model.** The flow is restricted to that of a perfect gas with globally constant stagnation enthalpy $h_0$, so that the static pressure can be derived from the Bernoulli equation,

$$p = \frac{1}{2}\kappa\rho(2h_0 - u^2 - v^2 - w^2) \tag{1}$$

where u,v and w are the cartesian velocity components, $\rho$ is the density and $\kappa = \frac{(\gamma-1)}{\gamma}$, $(\gamma = \frac{C_p}{C_v})$ . The energy equation is therefore not needed, unless heat transfer is taken into account. The Euler equations for compressible 3D flow can then be written in the integral form as follows:

$$\int_{\Omega} \frac{\partial \vec{Q}}{\partial t} dV + \oint_{\partial\Omega} (n_x \vec{FX} + n_y \vec{FY} + n_z \vec{FZ})\, dS = 0 \tag{2}$$

with,

$$\vec{Q} \equiv \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \end{pmatrix}$$

$$\vec{FX} \equiv \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \end{pmatrix}, \ \vec{FY} \equiv \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \end{pmatrix}, \ \vec{FZ} \equiv \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \end{pmatrix}$$

where $\Omega$ is an arbitrary finite region, and $d\vec{S}$ is the outward pointing normal surface vector. The governing system of Eqs. (2) is hyperbolic.

**4. Numerical Method.** The numerical method used to integrate the Euler equations is a centered finite volume method for the spatial discretization together with an explicit one-step four-stage time integration scheme [5-7]. Since only steady solutions are of interest the local time step is used to accelerate the convergence to steady state.

**4.1 Spatial Discretization.** A centered finite volume discretization is obtained by applying the integral formulation of the governing Eqn. (3) to each grid cell and assuming that $\frac{\partial \vec{Q}}{\partial t}$ is constant within each grid cell and that the fluxes are constant across each cell surface,

$$VOL_{i,j,k} \ \frac{\partial \vec{Q}}{\partial t} + \vec{FI}_{i+\frac{1}{2},j,k} - \vec{FI}_{i-\frac{1}{2},j,k} + \vec{FJ}_{i,j+\frac{1}{2},k} -$$

$$-\vec{FJ}_{i,j-\frac{1}{2},k} + \vec{FK}_{i,j,k+\frac{1}{2}} - \vec{FK}_{i,j,k-\frac{1}{2}} = 0 \tag{3}$$

where,

$$\vec{FI} \equiv SIX \ \vec{FX} + SIY \ \vec{FY} + SIZ \ \vec{FZ}$$

$$\vec{FJ} \equiv SJX \ \vec{FX} + SJY \ \vec{FY} + SJZ \ \vec{FZ}$$

$$\vec{FK} \equiv SKX\ \vec{FX} + SKY\ \vec{FY} + SKZ\ \vec{FZ}$$

are the integrated fluxes and, SIX, SIY, SIZ, SJX, SJY, SJZ, SKX, SKY and SKZ are the metric coefficients, defined as the cartesian components of the cell surface vectors. The non-integer indices refer to cell surfaces whereas integer indices refer to cell centers. Non-integer index values of the fluxes $\vec{FX}, \vec{FY}$ and $\vec{FZ}$ are taken as the arithmetic means of neighboring cell centered values. Thus, the only quantities needed for the coordinate transformation are the x, y and z components of the grid points.

**4.2 Artificial Viscosity.** The Euler equations need artificial viscosity to avoid nonphysical shocks and sawtooth waves (with a wavelength of two grid lengths) in the solution. The model for artificial viscosity uses a combination of a variable-coefficient second order difference operator and a constant-coefficient fourth order difference operator [6]. The artificial viscosity terms are added to the semi-discrete scheme so that the resulting system becomes

$$\frac{\partial \vec{Q}}{\partial t} = N(\vec{Q}) - \chi\ (\delta_i[SWI(\vec{Q})\delta_i] + \delta_j[SWJ(\vec{Q})\delta_j] + \delta_k[SWK(\vec{Q})\delta_k])\vec{Q} -$$

$$-\beta\ CFL\ (\delta_i^4 + \delta_j^4 + \delta_k^4)\ \vec{Q} \qquad (4)$$

where N is a nonlinear vector function, $\chi$ and $\beta$ are filter constants, SWI, SWJ and SWK are coefficients proportional to the second derivative of the pressure and $\delta^2$ and $\delta^4$ are second and fourth order centered difference operators.

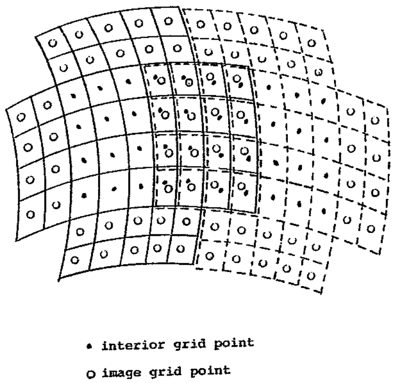

Figure 3. *Merging of two blocks.*

**5.1 Multi-Block Implementation.** To tailor new codes for each new topology is both time consuming and increases the possibilities of errors that are difficult to find. A general purpose multi-block flow solver greatly facilitates the application of the flow solver for different topologies [11,12]. The basic concept is to add to each block extra layers of grid cells outside the block, so that boundary grid points can be treated as interior grid points.

Since the second order filter operates on the pressure, we need image values of not only the flow variables but also the pressure.

Both the fourth and the second order filter requires two extra layers of image values outside the flow region. Each image grid cell will share the same storage cell as an interior grid cell of the contiguous block, Fig. 3. All anomalies of the scheme can then be located to subroutines dealing with boundary conditions. Subroutines updating the time step, flow variables and fourth and second order differences are completely symmetric in all curvilinear coordinate directions i, -i, j, -j, k and -k. The decomposition of the flow field into computational blocks is done so that each side of a computational block has precisely one type of boundary condition [10,12]. This means that we introduce more computational blocks than necessary but the handling of the boundary conditions is simplified. Subroutines dealing with boundary conditions are written to handle any coordinate surface. This means that the curvilinear coordinates can be chosen completely arbitrarily by specifying input parameters.

**5.2 Boundary Conditions.** As mentioned above, all boundary conditions are to be expressed in terms of image values of the flow variables outside the computational block. In our case four different types of boundary conditions are applied: solid wall, symmetry plane, inflow-outflow boundary and interface boundary.

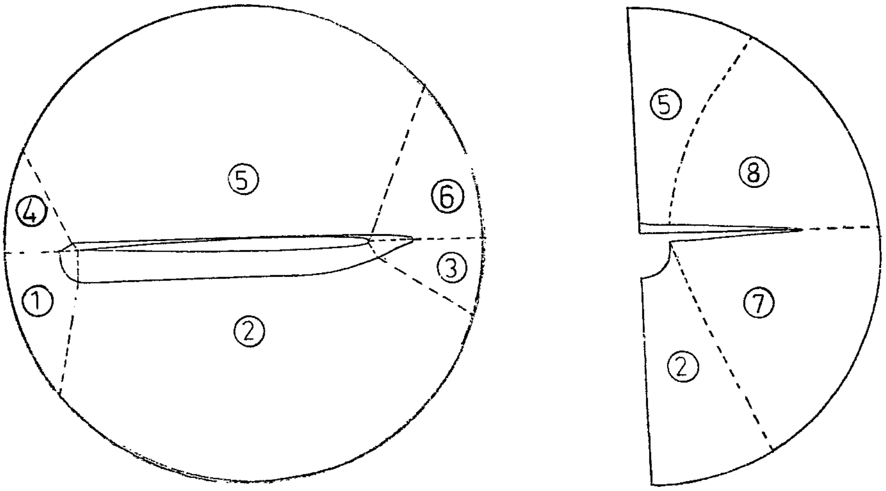

Figure 4. *Decomposition of the physical domain.*

At a solid wall the mass flux is zero while the pressure forces balances the momentum flux. As far as the fluxes are concerned, a symmetry plane boundary condition is the same as a solid wall boundary condition. At the inflow-outflow boundary absorbing boundary conditions [5] in their simplest form are applied. The governing equations are linearized locally and the characteristic variables are computed in the normal direction. The characteristic variables advected into the domain are fixed to the free-stream values whereas those which are advected out of the domain are linearly extrapolated from the interior to the boundary. The resulting set of characteristic variables is then transformed back to the primitive variables. The numerical boundary conditions for the second and fourth order filters is described in ref. [6] .

Finally, we have interface boundaries between computational blocks contiguous in the physical space. Since the same continuity restrictions of the computational grid are imposed at block interfaces as for the interior grid, these boundary grid cells can be treated as any other grid cell. The flux boundary conditions and the numerical boundary conditions for the fourth and second order difference operators automatically fall out by using image values from adjacent blocks. This can be obtained by either using indirect addressing and simply assigning the same address for corresponding grid cells or by updating the image values each stage of the time step. As shown in Fig. 4 above the flow region around the wing-sting configuration is decomposed into eight blocks so that each block surface has precisely one type of boundary condition. A schematic representation of the corresponding multiply-connected computational blocks is shown in Fig. 5. An interface boundary is specified by a pairwise enumeration of block surfaces contiguous in the physical space. However two coordinate surfaces can be merged together in essentially four different ways. It is consequently necessary for each coupling to also assign the type of merging.

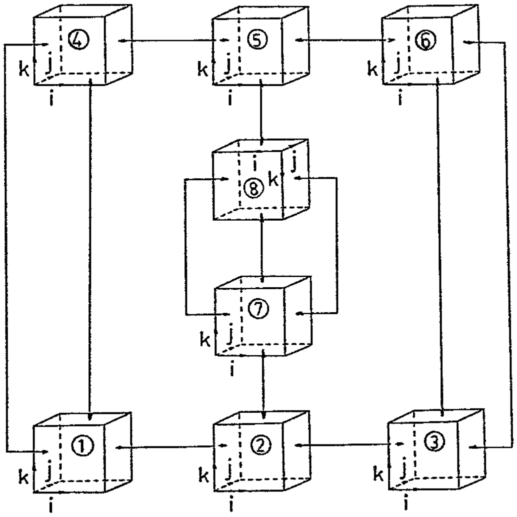

Figure 5. *Flow chart of couplings between computational blocks*

**6. Computational Results.** The flow around the wing-sting configuration was computed at 10 deg angle of attack and a free stream Mach number of 0.85. The grid contains 111,000 interior grid points and an oblique view of the grid was shown in Fig 2. The flow calculations were carried out with $\chi = 0.2$, $\beta = 0.01$ and $CFL = 1.0$. The initial flow field was extrapolated from computations on a coarser grid. Velocity vectors in three vertical planes are shown in Fig. 6. The flow properies have been interpolated onto intersection points between the grid and the plane. The velocity components have further been interpolated to an equispaced grid in this plane. The velocity vector plots show clearly the development of the vortex on the upper side of the wing. Figure 5 reveals that the vortex starts somewhere downstream of the 0.3 section.

Measurements of surface pressure at a Reynolds number of $1.3 \times 10^7$ [14], have been made along the previously mentioned chord sections. Computations around the wing-sting configuration are also compared with a similar computation around a wing alone with 286,000 grid points [13]. The boundary layer in the experiment separates under the primary vortex and rolls into a secondary vortex which reduces the strength of the primary vortex. The suction peaks will therefore tend to be higher in the Euler calculations than in the experiments.

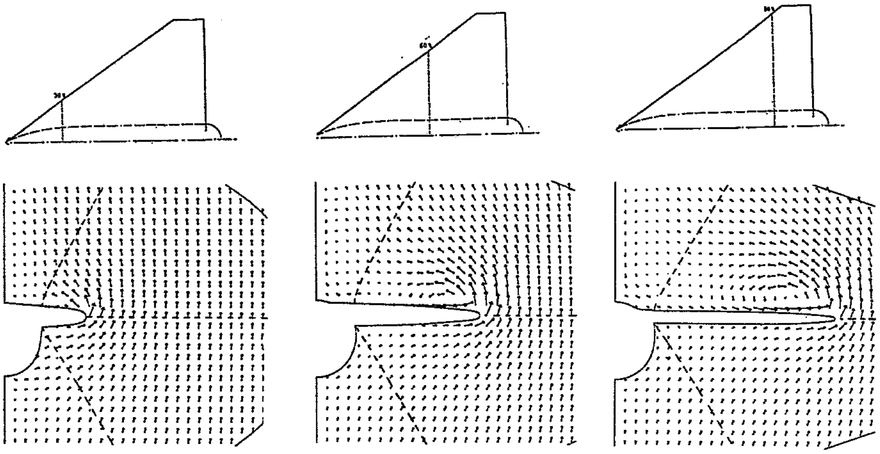

Figure 6. *Velocity vectors at* $\frac{x}{c_{root}} = 0.3,\ 0.6,$ *and* $0.8$ .

Figure 7 shows measured and computed $C_p-$ distributions at the sections $\frac{x}{c_{root}} = 0.3, 0.6$ and 0.8. As expected, it can be seen that for 0.3 section, the $C_p-$ curve on the lower side shows better agreement with experiment than the corresponding curve for the wing alone computation. On the upper side of the wing the difference is very small between the wing and the wing-sting computations. Also at the sections $\frac{x}{c_{root}} = 0.6$ and 0.8, the agreement to the experiments is improved compared with corresponding computation around a wing alone, especially on the lower side of the configuration where the sting extends most. Another thing that is striking is that the suction peak for the wing-sting computation is much lower than for the alone wing computation. This can be explained by the fact that the strength of the vortex grows with a better resolution i.e., with the number of grid points, however the position of the vortex does not change. Also the absence of sharp bends on the the upper $C_p-$ curve for the wing-sting computation probably has to do with the coarser grid. Finally a $C_p$- distribution on the configuration surface can be seen in Fig. 8. Traces of the vortex can be seen on the wing.

**7. Conclusions.** A general purpose Euler code which offers possibilities to compute generic multi-block grid topologies by specifying just a few input parameters, is developed. The multi-block grid approach is an important tool in generating efficient computational grids around complex configurations, such as a realistic aircraft. The computational results for a wing-sting configuration show that the grid topology used here efficiently resolves high flow gradients especially at the nose and tail and captures the vortex on the upper side of the wing. The grid topology is especially appropriate simple wing-fuselage configurations for which the fuselage extends far from the wing, such as space shuttles. Comparisons between wing-fuselage and wing alone computations show, as expected, that the former gives a better agreement with experimental values. Euler computations around more complex configurations remains for the future. Even though the effort in describing the grid topology to the Euler code is very small, the amount of work to generate a $C^1-$ continuous multiblock grid is substantial. To get rid of the continuity restriction at grid interfaces would not only facilitate the grid generation a lot, but also make it possible to refine the grid in each block independently. A desirable extension for the future of the multi-block Euler code is therefore implementation of a more general interfacing.

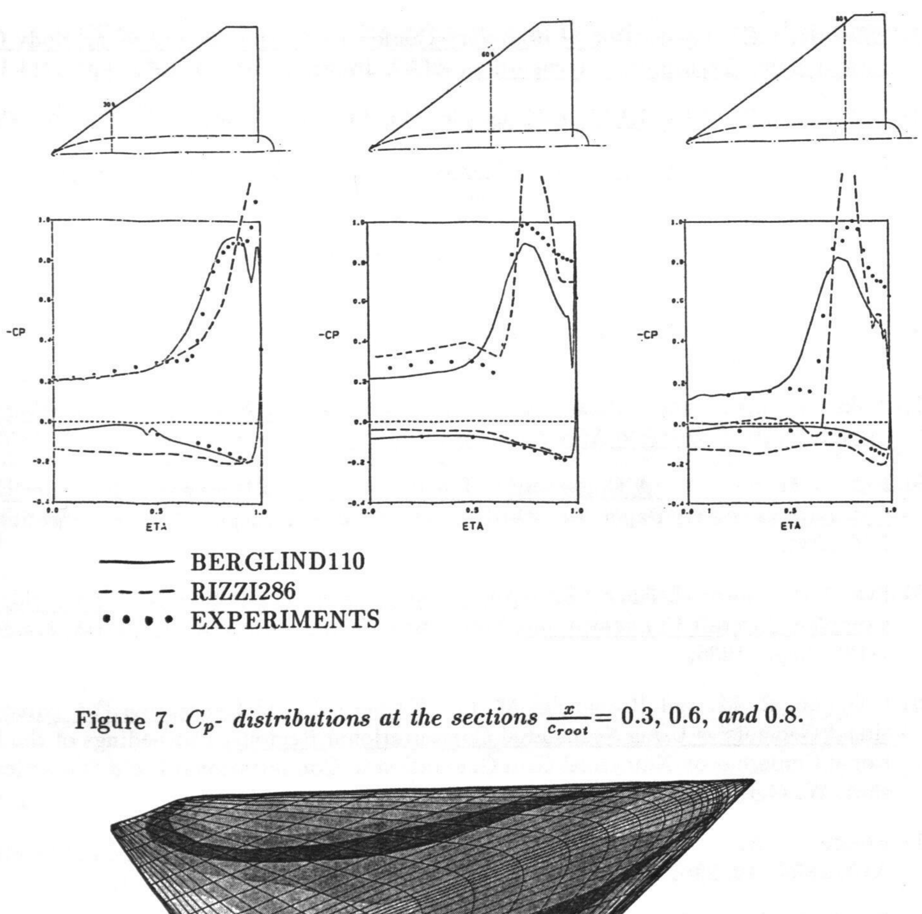

Figure 7. $C_p$- *distributions at the sections* $\frac{x}{c_{root}} = 0.3, 0.6,$ *and* 0.8.

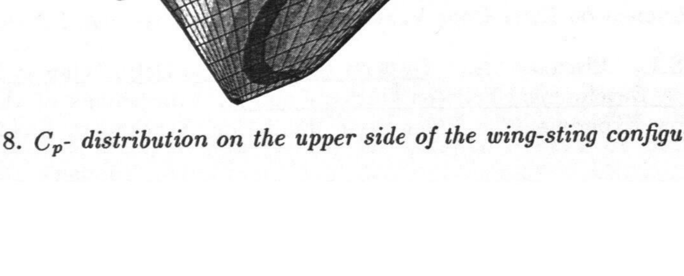

Figure 8. $C_p$- *distribution on the upper side of the wing-sting configuration.*

**Acknowledgements.** The author would like to thank Dr. Arthur Rizzi for valuable advices in the code development and Peter Eliasson and Tomas Strid for use of their flow visualization codes and for their help in extending them to multi-block topologies.

## REFERENCES

(1) Eriksson, L.E. - Three-Dimensional Spline-generated Coordinate Transformations for Grids around Wing-Body Configurations, Numerical Grid Generation Techniques, NASA CP 2166, 1980.

(2) Eriksson, L.E. - Generation of Boundary-Conforming Grids around Wing-Body Configurations Using Transfinite Interpolation, AIAA Journal, Vol. 20, 1982, pp. 1313-1320.

(3) Eriksson, L.E. - FFA 3D Mesh Generation Application - Users Manual, FFA - 830307.

(4) Berglind T. - A Comparison of Single-Block and Multi-Block Grids around Wing-Fuselage Configurations, FFA TN 1986-42, Stockholm 1986.

(5) Rizzi, A. and Eriksson, L-E. - Computation of Flow around Wings Based on the Euler Equations, J. Fluid. Mech., Vol. 148, pp 45-71, 1985.

(6) Eriksson L.E. - Boundary Conditions for Artificial Dissipation Operators, FFA TN 1984-53, Stockholm 1984.

(7) Eriksson L.E. and Rizzi A. - Computer-Aided Analysis of the Convergence to Steady State of Discrete Aproximation Problems, J. Comp. Phys. 2, 12-26, 1967.

(8) Rai M. - A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations, AIAA Paper No. 84-0164, AIAA 22nd Aerospace Sciences Meeting, Reno, NV, 1984.

(9) Fritz W. and Leicher S. -Numerical Solution of 3-D Inviscid Flow Fields around Complete Aircraft Configurations, Proceedings of the 15th ICAS Congress, London, UK, 7-12th Sept. 1986.

(10) Coleman, R. M. and Brabanski, M. L. - Numerical Grid Generation For Three-Dimensional Geometries Using Segmented Computational Regions, Proceedings of the International Conferens on Numerical Grid Generation in Computational Fluid Dynamics, Landshut, W. Germany, 14-17th July, 1986.

(11) Eberle A. - 3D Euler Calculations Using Characteristic Flux Extrapolation, AIAA-85-0119, 23rd Aerospace Science Meeting, Reno, NV, 1985.

(12) Berglind T. - Compressible Euler Solution on a Multi-Block Grid around a Wing-Fuselage Configuration, FFA TN 1987-46, Stockholm 1987.

(13) Rizzi A., Drougge G., Purcell C.J., Computations with the Euler Code WINGA2 for Vortex Flow around the 65 Degree Delta, Proceedings of the International Vortex Flow Experiment on Euler Code Validation, Stockholm, Sweden, 1-3 October 1986.

(14) Boersen S.J., Elsenaar A., Tests on the 65 Degree Delta Wing at NLR: A Study of Vortex Flow Development between Mach=.4 and 4., Proceedings of the International Vortex Flow Experiment on Euler Code Validation, Stockholm, Sweden, 1-3 October 1986.