EFFICIENT SOLUTION OF THREE-DIMENSIONAL EULER EQUATIONS USING EMBEDDED GRIDS

ICAS-86-1.3.3

R. Radespiel

DFVLR, Institute for Design Aerodynamics
3300 Braunschweig, Fed. Rep. Germany

ABSTRACT

For an efficient solution of three-dimensional Euler equations block structured grid generation and Euler codes have been developed which use local refinement of a coarse base grid to resolve the flow in regions with high gradients. Coordinate grids are generated by numerical solution of an elliptic system. A new iterative technique for grid control yields smooth and well distributed grids even near singular lines of H-type sections. The flow solver is based on a well known finite volume Runge-Kutta time stepping scheme . Different procedures for the treatment of the zonal boundaries are investigated. The codes are used to calculate transonic flow around DFVLR F4 wing. The results show that the present grid embedding technique yields a substantial reduction of computational expense without loss of accuracy.

INTRODUCTION

With the advent of fast vector computers a substantial advancement has been achieved in the development of efficient methods for the solution of the Euler equations. Solutions of the Euler equations give a more physical representation of inviscid subsonic, transonic and supersonic flow fields compared to potential flow methods. In particular the position of vortex sheets behind trailing edges of a lifting surface needs not to be specified but comes out as a part of the solution. In regions, where shock waves are present, the Euler equations allow entropy rise through shock waves while mass, momentum and energy are conserved. Solutions of the Euler equations are also considered as stepping stones towards the solution of the time averaged Navier-Stokes equations. On the other hand the numerical solution of the Euler equations requires high computational efforts in terms of CPU and storage. For an accurate prediction of transonic flow fields a sufficiently dense coordinate grid is required for regions where the flow is varying rapidly. Towards the far field where the flow is varying smoothly it is then desirable to increase the grid spacing. Following these ideas three-dimensional Euler codes often use a grid structure, where the block in the computational domain is wrapped around the wing in the physical domain, so that one complete boundary plane of the cube lies on the wing surface (O-grid). However, these grid structures are not flexible enough for an integration of additional components as bodies, nacelles, struts etc.. For these applications an H-type grid structure is suitable where the wing surface is located between a lower and an upper computational block. H-grids will give less resolution near the wing compared to O-grids with the same number of total points. Furthermore, the treatment of more complex configurations will require additional grid lines to resolve the interaction between the components. It seems unlikely, that an efficient solution of the Euler equations for a complete aircraft configuration can be achieved using a single global grid, even if an appropriate grid stretching to the far field could be specified.

To overcome these difficulties several papers have been addressed to the development of local grid refinement techniques.A general patched grid interface algorithm for 2-D problems has been given in Ref. 1. The formulation of interface schemes is much simplified if the global base grid is retained and the patch is refined by doubling the grid density there. Several recent papers[2,3,4,5,6] have followed this strategy to adapt the computational grid to local high gradients of the flow. Significant improvements in overall efficiency have been obtained without introducing too complicated additional logic into the codes.

In the present study we also follow this idea. The multiblock structured Euler code used here is based on the finite volume scheme of Jameson,Schmidt and Turkel [7]. Baker et al [4] have given 3-D calculations with grid embedding based on the scheme[7] using rather crude grids.Here we present accurate results for a transonic flow around large aspect ratio DFVLR F4 wing [8] which have been obtained on a fine H-grid. The basic cell centered Euler scheme requires a smooth coordinate grid in regions of large gradients of the flow.Therefore,the grid generation method is described in greater detail.The grids are generated by numerical solution of three Poisson equations for the coordinates x,y,z as proposed by Thompson et al [9]. Grid control is achieved by iteratively adjusting the source terms in the Poisson equations.

The purpose of the present paper is to judge accuracy and effiency of the cell centered finite volume scheme [7] with grid embedding. As H-type grids are flexible to treat complex geometries the present work forms a basis for the development of a code for a complete transport aircraft.

GRID GENERATION METHOD

Elliptic System

In the present work body fitted grids are generated by solution of three Poisson equations for curvilinear coordinates $\vec{\xi} = [\xi,\eta,\zeta]^T$,

$$\vec{\xi}_{xx} + \vec{\xi}_{yy} + \vec{\xi}_{zz} = \vec{P} \tag{1}$$

where $\vec{P} = [P,Q,R]^T$ represents source terms to control the grid spacing. This technique was first proposed by Thompson et al [10] and has been used subsequently by many researchers. Coordinates which are generated by eqn.(1) are distributed smoothly in the physical domain, even if the distribution of points at the boundaries is not smooth. Furthermore the source terms $\vec{P}$ can be choosen in such a way that coordinate lines are clustered in regions where high gradients of the flow are expected. Therefore, the choice of source terms is the main problem in generating suitable grids.

It is evident that eqn. (1) can be more easily solved by interchanging the role of dependent and independent variables as follows:

$$A(\vec{x}_{\xi\xi} + \frac{J^2P}{A}\vec{x}_\xi) + B(\vec{x}_{\eta\eta} + \frac{J^2Q}{B}\vec{x}_\eta) + \tag{2}$$
$$+ C(\vec{x}_{\zeta\zeta} + \frac{J^2R}{C}\vec{x}_\zeta) + 2(D\vec{x}_{\xi\eta} + E\vec{x}_{\xi\zeta} + F\vec{x}_{\eta\zeta}) = 0$$

where J denotes the Jacobian $J = \partial\vec{x}/\partial\vec{\xi}$ and the coefficients A to F are functions of the transformation coefficients which are given in [9]. Specifying the values of $\vec{x}$ at the boundaries of the computational domain, eqn. (2) can be solved in the interior.

Grid Control

A simple method to derive source terms from a prescribed point distribution has been given by Middlecoff and Thomas [10]. Let η = const. be a plane at which a point distribution is prescribed. Assume that the coordinate line on which η varies is crossing that plane orthogonally and that the curvature of the coordinate line on which η varies is zero. The source term P then writes

$$\frac{J^2}{A}P = \frac{s_{\xi\xi}}{s_\xi} \tag{3}$$

where s denotes the arc length distribution. Similar expressions can be derived for source terms Q and R respectively. Once the source terms P,Q and R have been evaluated on certain planes of the computational domain they can be interpolated in the whole region.

If one wishes to attract grid lines to other lines in the computational domain (i.e. leading or trailing edges) one can use additional source terms as given in [9]. Assume that the lines on which ζ varies are to be attracted to the line $\vec{\xi} = [\xi_1,\eta_1,\zeta]^T$. The source term P will then write

$$P = a\,\mathrm{sign}(\xi-\xi_1)\exp\left\{-\,b\left[(\xi-\xi_1)^2+(\eta-\eta_1)^2\right]^{1/2}\right\} \tag{4}$$

with empiral constants a,b. Analogous expressions for source terms Q follow directly.

For O- and C-type grids around wings the use of eqn.(3) yields smooth and well distributed coordinates in the physical domain. In the case of an H-type grid difficulties arise at the leading edge of the wing where a grid singularity occurs. In Fig.1 the grid structure used in the present investigation is sketched. The wing lies between an upper and a lower computational cube forming an H-sectionwise structure. The wing tip is wrapped in an O-type manner which there gives a better resolution of the flow compared to an H-type spanwise grid. The grid structure of Fig. 1 shows inner cuts between upper and lower cubes. Both coordinates and source terms have to be specified at these inner boundaries of the computational domain to allow a solution of eqn. (2). Fig. 2a shows a view of a sectionwise H-grid, which was generated with fixed values of the coordinates on the plane "ABCD" of Fig. 1 throughout the solution process. At the inner cut there is a slope discontinuity which deteriorates the order of accuracy of the flow solver. An alternative would be not to fix the coordinates on the inner cut throughout the solution, but to treat this coordinates as interior points. Fig. 2b shows the result of this strategy. Here the grid is smooth on the inner cut but is badly distributed near the leading edge. The failure of both alternatives is due to the assumption of orthogonality and vanishing curvature when specifying the source terms at the inner cut according to eqn. (3).

Iterative Grid Control

As both skewness and curvature come out as a part of the solution and therefore cannot be calculated beforehand an iterative determination of source terms has been developed. In order to obtain a smooth grid the solution algorithm is operating on all interior points including inner cuts. In order to obtain an appropriate point distribution the source terms are adjusted throughout the whole solution process. For this purpose we first choose target planes on which the final solution should coincide with a specified point distribution. In the present application the target plane "ABCD" of Fig. 1 is choosen. Target values of grid stretching $(s_{\xi\xi}/s_\xi)_o$ can be calculated from the desired point distribution on the target plane. Eqn. (3) will usually be solved by a relaxation method, which yields new values of the coordinates after each iteration. From these new coordinates actual values of $s_{\xi\xi}/s_\xi$ can be computed. The difference between target values and actual values of the grid stretching is used to adjust the source terms at the target plane as follows:

$$(\frac{J^2}{A}P)_{n+1} = (\frac{J^2}{A}P)_n + c_P\left[(\frac{s_{\xi\xi}}{s_\xi})_o - (\frac{s_{\xi\xi}}{s_\xi})_n\right] .$$

To obtain a stable iteration scheme it is necessary to add a damping term using the derivative of the difference between target values and actual values with respect to the iteration number. The final iteration formula then reads:

$$(\frac{J^2}{A}P)_{n+1} = (\frac{J^2}{A}P)_n + c_P\left[(\frac{s_{\xi\xi}}{s_\xi})_o - (\frac{s_{\xi\xi}}{s_\xi})_n\right] - \tag{5}$$
$$- c_T\left[(\frac{s_{\xi\xi}}{s_\xi})_{n-1} - (\frac{s_{\xi\xi}}{s_\xi})_n\right]$$

Once the source terms $(P\,J^2/A)_{n+1}$ have been obtained on the target plane they can be interpolated in the entire domain and the relaxation method can proceed to the next iteration and so forth. In practice coefficient values of c_P=0.05 and c_T=0.5 have been found to give good convergence properties in combination with a line relaxation scheme to solve eqn.(2). The converged solution will yield a coordinate grid for which $(s_{\xi\xi}/s_\xi)=(s_{\xi\xi}/s_\xi)_o$ is valid on the target plane and which is smooth in the entire domain. Several views on a grid which has been generated using iterative grid control will be given in the following paragraph.

Grid Generation Code

In order to enable the treatment of complex configurations a multiblock structured grid generation code has been developed. This approach was proposed by Lee and Rubbert[12] and has recently been used by Weatherill et al [13] and Fritz [14]. A complicated multiply connected computational domain is split into a number of simply connected cubes. Therefore, the structure of the basic relaxation routine is independant of the problem under consideration. The logic which is incorporated into the present code allows a high flexibility with respect to grid topologies. The boundaries of each block are divided into an arbitrary number of segments. Either Dirichlet boundary conditions can be prescribed or the segment can be treated as inner cut on which the coordinates are generated by the solution algorithm.

For an efficient grid generation successive grid refinement is used. The solution of the coarse grid is interpolated and used as input for the refined grid calculation. Furthermore there is an option available for a refinement of an embedded patch of the grid only. For the present application the surface grids on the wing, on the far field boundaries and on the inner cuts of the two block grid structure in Fig. 1 have been generated algebraically. In particular the surface grid of the wing tip has been smoothed by applying a series of superellipses [15] to the planform and the thickness of the wing tip. Fig. 3 shows the planform of DFVLR F4 wing and a view on the surface grid of the wing tip. The far field boundaries have been located around 16 chords away from the wing surface. Initial source terms have been specified on all boundaries using eqn. (3). To obtain a more dense grid near the leading edge of the wing spanwise grid lines have been attracted to the leading edge using eqn. (4).

Once the surface grids and surface source terms have been specified 3-D linear transfinite interpolation is used to obtain the initial values in the interior of the computational cubes. The iterative solution of eqn. (2) is then advanced by a conventional successive line relaxation method. Usually overrelaxation is used for a fast convergence of these methods. For the present application where large aspect ratio cells are occurring at the singular lines of the grid and near the far field the relaxation factor had to be reduced to a value around 0.7 to obtain a converged solution. Iterative grid control according to eqn. (5) has been used on the plane "ABCD" of Fig. 1. The coordinates on the plane "EFGH" have been fixed throughout the solution because of the finite trailing edge thickness of DFVLR-F4 wing.

The convergence history of the grid generation calculation is illustrated in Fig. 4. Three successive grids have been generated. The maximum change of x-values in the fine embedded patch is substantially lower compared to the coarse global grids, because the aspect ratio of the cells in the patch is smaller compared to those near the far field. The CPU required for the complete calculation was less than 100 s on CRAY 1S, which is about 5 percent of the CPU of a converged flow analysis. Fig. 5 shows the sectional grid near the root of the wing. A view of sectional and spanwise grid distributions near the tip is given in Fig. 5d. It can be seen that the present method of grid-control yields smooth and well distributed coordinates in the leading edge region of the wing.

EULER EQUATION METHOD

The well known finite volume multistage time stepping scheme of Ref. [7] is used in a multiblock structured 3-D Euler code. The development of the global grid version of this code has been described by Radespiel and Kroll [16] including details how to treat boundaries of the computational domain and how to handle the I/O problems of an out of core code. In the present paper we will briefly introduce the basic scheme and then turn over to the treatment of embedded patch boundaries.

Basic Scheme

In integral form the Euler equation are written as

$$\frac{\partial}{\partial t}\iiint_{\Omega} \vec{W}\, d\,\Omega + \iint_{\delta\Omega} \bar{\bar{F}}\;\; \vec{n}\, d\, s = 0 \qquad (6)$$

where Ω denotes a fixed region with boundary $\delta\Omega$ and outer normal $\vec{n}$. $\vec{W}$ represents the vector of conserved quantities, namely mass, momentum and total energy, and $\bar{\bar{F}}$ is the corresponding flux tensor. Semi-discretization of eqn. (6) separates spatial and time discretization. The final volume scheme is

$$\frac{\partial}{\partial t}(h_{i,j,k}\;\vec{W}_{i,j,k}) + \vec{Q}_{i,j,k} - \vec{D}_{i,j,k} = 0 \qquad (7)$$

where $h_{i,j,k}$ denotes the volume of quadrilateral cells. $Q_{i,j,k}$ represents the Euler flux balance for each cell. The fluxes at the cell faces are calculated from averages of the values $\vec{W}$ assigned to the cell centers. To damp out high frequency oscillations, dissipative terms $D_{i,j,k}$ are added which are formed by a blend of fourth and second differences. Multistage Runge-Kutta time stepping schemes are used to advance to the steady state. In the present study a 5-stage scheme with two evaluations of dissipative terms is used. Several techniques to accelerate the convergence to the steady state are applied. These are: local time stepping, enthalpy damping [7], implicit residual averaging[17] and successive grid refinement.

Interface Schemes

In the present investigation embedded patches are formed by dividing the spacing of the base global grid. Therefore, rather simple interpolation formulas can be used to transfer information between the zones. Fig. 6 shows a view of the patch interface where the grid size ratio is 2. Assume that the finite volume scheme in the fine grid requires the value of the variable U at the midpoint of the cell face S. A first order interpolation in the computational domain then yields

$$U_S = \frac{2}{3} U_{il,j,k} + \frac{3}{16} U_{1,n,m} + \frac{1}{16} U_{1,n+1,m} + \\ + \frac{1}{16} U_{1,n,m+1} + \frac{1}{48} U_{1,n+1,m+1} \tag{8}$$

The fluxes at the interface cells of the coarse grid are formed by adding the four corresponding fluxes of the fine grid. This interface scheme has been given by Eriksson [5] and is denoted by Scheme I. Allmaras and Baron [6] have pointed out that Scheme I allows unstable modes for waves entering the embedded region (downwind weighting of eqn.(8)). They propose a simple zeroth order interpolation which has already been used in Ref. [4]:

$$U_S = \frac{1}{2}(U_{il,j,k} + U_{1,n,m}) \tag{9}$$

Eqn. (9) is called Scheme II. Conservation at the interface is enforced in the same way as with Scheme I. Scheme II is neutrally stable for all flow directions [6].

In analogy to the treatment of Euler fluxes at a patch interface, special interface formulas for the disipative flux balance $D_{i,j,k}$ have to be provided. For both Scheme I and Scheme II the dissipative fluxes which are used to form the fourth difference dissipative terms are set to zero at patch interfaces.The fluxes, which are used to form the second difference terms are determined by taking first differences of each conservation variable. At the patch interfaces these first differences are taken using first order and zeroth order interpolation in connection with Scheme I and Scheme II respectively. This procedure is used for both coarse and fine grid interface cells and therefore it is not conservative. Alternatively some calculations were performed setting the first differences to zero at the interfaces. Very similar results were obtained in both the cases.

Multi-Block Euler Code

Multi-block structured flow codes exhibit a high flexibility with respect to grid topologies and grid sizes. Nevertheless the structure of the basic flow solver is simple and does not require any I/O transfer. Furthermore a block structured Euler code can be extended to allow embedded regions, where time avaraged Navier-Stokes equations are solved [18]. The present code is an extension of the block structured global grid Euler code CATS [16]. The logic, which is incorporated in CATS divides the boundaries of each block in an arbitrary number of segments. In the present version the following boundary conditions can be applied on each segment:

- far field
- solid wall
- inner cut, grid size ratio 1:1
- inner cut, grid size ratio 2:1
- inner cut, grid size ratio 1:2

The user,who wants to select appropriate boundary conditions for a particular block structure has to specify a small number of integer values for each segment. These integer values steer the boundary condition routines and link corresponding inner cuts to each other.

RESULTS

To judge the accuracy and efficiency of the present code several computations of the transonic flow around DFVLR F4 wing [8] are presented. This configuration has been choosen because it represents a typical rear loaded transonic design. It has also been used as test case for transonic flow computations of two GARTEUR Action Groups where relative large differences were observed between the results of the participants.

Fig. 7 shows the convergence history of the present method. Three successive grids have been used. The coarse and the medium grids are global grids. The solutions on the fine grid have been obtained using both a global grid with 128x80x28= 286720 cells (128x28 cells on the wing surface) and a refined patch which is embedded in the medium grid with 77882 cells in total (see Fig. 5). The reduction in averaged $\partial\rho/\partial t$ is slightly better for the embedded grid calculation with interface Scheme I than for the global fine grid. This may be due to the larger time steps which can be taken for the cells outside the embedded patch. If interface Scheme II was applied the CFL number had to be slightly reduced in order to get a converged solution. This stands in contradiction to the theoretical results of Ref. [6], which were obtained for a one-dimensional linear model problem.

The influence of finite distance to the far field boundary of the coordinate grid can be easily checked with the present H-type sectional grid. We choose a flow with $\alpha=0^{\circ}$ and compare the solutions with free stream boundary conditions at the horizontal far field boundaries with those when the far field boundaries were treated as solid walls. It is well known that solid wall boundary conditions will overestimate the lift, whereas free stream boundary conditions result in less total lift when compared to the flow in an infinite domain. For the present grid, where the far field was located about 16 mean chords away from the wing,Table 1 shows that the influence of far field distance on the lift seems to be less than 1%.

Table 2 gives the lift and drag coefficients for solutions with and without grid embedding. Note that the coefficients had been frozen to at least three figures when the computations were terminated. The results for interface Scheme I are closer to the global fine grid results than those of Scheme II. In Fig. 8 pressure distributions are given for wing sections near root, kink and tip of the wing.The main features of the flow are well resolved on the medium grid. The results of the global fine grid and the embedded fine grid (scheme I) are close together exept for the pressure peak at the very blunt leading edge of the root section, where the embedded patch should be slightly enlarged in upstream direction to resolve the flow more accurately (see Fig. 5). Note that the pressure distribution is smooth near the H-grid singularity at the leading edge. A more detailed comparison between the solutions is given in Fig. 9-10. Surface pressure distributions and Mach contours are shown at a midspan section where the spanwise lift distribution has its maximum. In the contour plots the boundaries of the embedded patches are indicated by dashed lines. Global grid and embedded grid (Scheme I) results compare very well. The Mach contours for Scheme II show an unsmooth behaviour at the interfaces. Therefore this scheme should be discarded. In Fig. 11 distributions of total pressure loss at the midspan station are presented. The total pressure loss is well below one percent upstream of the shock except a small peak at the leading edge with two percent at maximum. The results of Fig. 8-11 indicate that the present numerical solution of the Euler equations is quite accurate.

In Table 3 CPU-timings on CRAY 1-S vector computer are given. For engineering applications it should be sufficient to terminate the computations when 99,5% of the final lift has been obtained. If we adopt this criterion the solution with local grid refinement is 3.7 times cheaper than the solution on a global fine grid.A further improvement may be obtained if more time steps are executed on the embedded patch than on the coarse global grid. For example the sweep over the blocks can be arranged in such a way that always two time steps are taken on the embedded patch blocks whereas one time step is executed on the coarse global grid blocks. Now the solution with local grid refinement is 4.4 times faster than the solution on a global fine grid.

CONCLUSIONS

For an efficient solution of the three-dimensional Euler equations both grid generation and Euler codes have been developed which use locally refined patches to resolve high gradient regions of the transonic flow around wings. The grid generator described here provides smooth and well distributed H-O-grids. A cell centered finite volume Euler code has been extended to include treatment of zonal boundaries, where the grid spacing changes. Different boundary schemes have been investigated for the transonic flow around DFVLR-F4 wing. The results show that local grid refinement yields a substantial reduction of computational expense without loss of accuracy.

As both grid generation and Euler codes are block structured it is easy to adapt them to new or more complex configurations. Therefore the present work forms a basis for the development of a code for complete aircraft calculations.

ACKNOWLEDGEMENTS

The author would like to thank DORNIER Company for providing a basic two-dimensional Euler code and the basic line relaxation algorithm to solve the elliptic system in the grid generation method. Furthermore the author is indebted to Dr. A. Kumar from NAL Bangalore, India who mainly developed the graphic programs used in this paper.

REFERENCES

[1] Rai, M.M.: A Conservative Treatment of Zonal Boundaries for Euler Equation Calculations. AIAA Paper No. 84-0164 (1984).

[2] Useb, W.; Murman, E.M.: Embedded Mesh Solutions of the Euler Equations Using a Multiple-Grid Method. AIAA Paper No. 83-1946 (1983).

[3] Berger, M.J.; Jameson, A.: Automatic Grid Refinement for the Euler Equations. MAE Report No. 1633, Princeton University (1983).

[4] Baker, T.J.; Jameson, A.: Three-Dimensional Euler Solutions with Grid Embedding. AIAA Paper No. 85-0121 (1985).

[5] Eriksson,L.E.: Euler Solutions on O-O-Grids Around Wings Using Local Refinement. in: Notes on Numerical Fluid Mechanics, Volume 13, 1986.

[6] Allmaras, S.R.; Baron, J.R.: Embedded Mesh Solutions of the 2-D Euler Equations: Evaluation of Interface Formulations. AIAA Paper No.86-0509 (1986).

[7] Jameson, A.; Schmidt, W.; Turkel, E.: Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time Stepping Schemes. AIAA Paper 81-1259 (1981).

[8] Redeker, G.; Schmidt. M.; Müller, R.: Design and experimental Verification of a Transonic Wing for a Transport Aircraft. AGARD-CP-285, pp. 13-1-13-14, (1980).

[9] Thompson, J.T.; Warsi, Z.U.A.; Mastin, G.W.: Numerical Grid Generation. North Holland, New York, 1985.

[10] Thompson, J.F.; Thames, F.G.; Mastin, C.M.: Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing any Number of Arbitrary Two-Dimensional Bodies. J. Comp. Physics, Vol. 15, pp. 299-319, (1974).

[11] Middlecoff, J.F.; Thomas, P.D.: Direct Control of the Grid Point Distribution in Mesh Generated by Elliptic Equations. AIAA Journal, Vol. 18, pp. 652-656, (1980).

[12] Lee, K.D.; Rubbert, P.E.: Transonic Flow Computations Using Grid Systems with Block Structure. 7th Int. Conf. on Numerical Methods in Fluid Dynamics, pp. 266-271, (1980).

[13] Weatherill, N.P.; Shaw, J.A.; Forsey, C.R.; Rose, K.E.: A Discussion on a Mesh Generation Technique Applicable to Complex Geometries. AGARD-CP-412 (1986).

[14] Fritz, W.: Numerical Grid Generation Around Complete Aircraft Configurations. AGARD-CP-412 (1986).

[15] Stoffers, G.: Grundkurven zur Formerzeugung. DFVLR-IB 129-83/8 (1983).

[16]Radespiel, R.; Kroll, N.: Progress in the Development of an Efficient Finite Volume Code for the Three-Dimensional Euler Equations. DFVLR-FB-85-31 (1985).

[17]Jameson, A.; Baker, T.J.: Solution of the Euler Equations for Complex Configurations. AIAA-Paper No. 83-1929 (1983).

[18]Holst, T.L.; Gundy, K.L.; Flores, J.; Chaderjian, N.M.: Numerical Solution of Transonic Wing Flows using an Euler/Navier-Stokes Zonal Approach. AIAA-Paper No. 85-1640, (1985).

	c_L	c_D
Free stream horizontal far field boundaries	0.7367	0.02507
Solid wall horizontal far field boundaries	0.7473	0.02490

Table 1: Lift and drag coefficients obtained with different far field boundary conditions, $M_\infty = 0.75$, $\alpha = 0^o$, 64x20x14 grid

	c_L	c_D
32x20x7 global grid	0.809	0.0422
64x40x14 global grid	0.848	0.0352
128x80x28 global grid	0.878	0.0331
64x40x14 global grid 78x22x28 embedded grid Interface SCHEME I	0.877	0.0330
64x40x14 global grid 78x22x28 embedded grid Interface SCHEME II	0.875	0.0335

Table 2: Total forces for DFVLR F4 wing, $M_\infty = 0.75, \alpha = 0.84^o$

	99.5 % c_L	99.9 % c_L
128x80x28 global grid	6780 s	8830 s
64x40x14 global grid 78x22x28 embedded grid	1810 s	2740 s
64x40x14 global grid, n time steps 78x22x28 embedded grid, 2n time steps	1550 s	2310 s

Tʀble 3: CPU-timings on CRAY 1-S vector computer

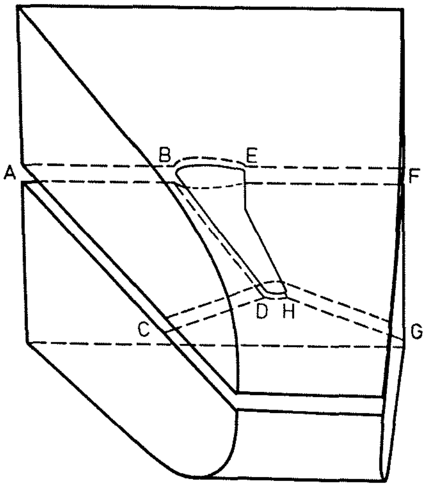
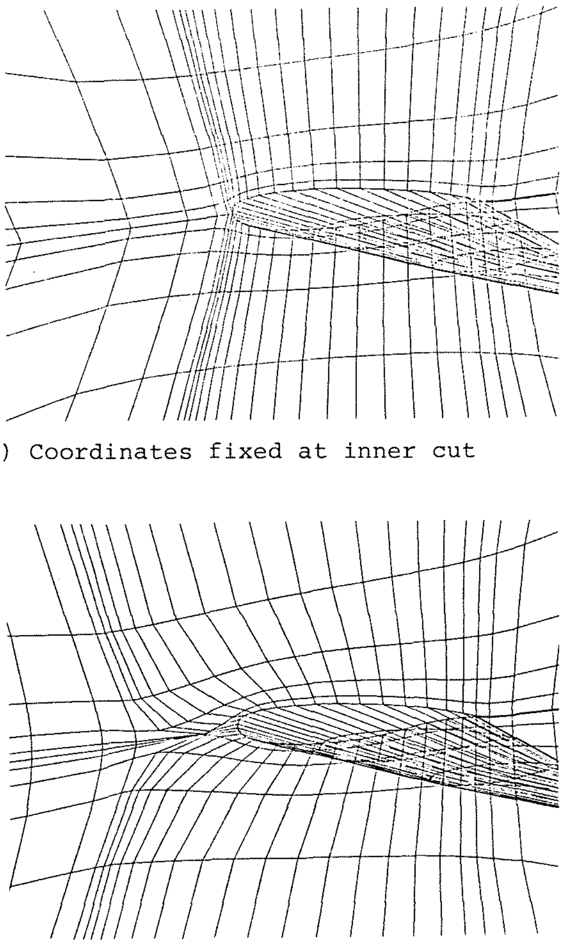


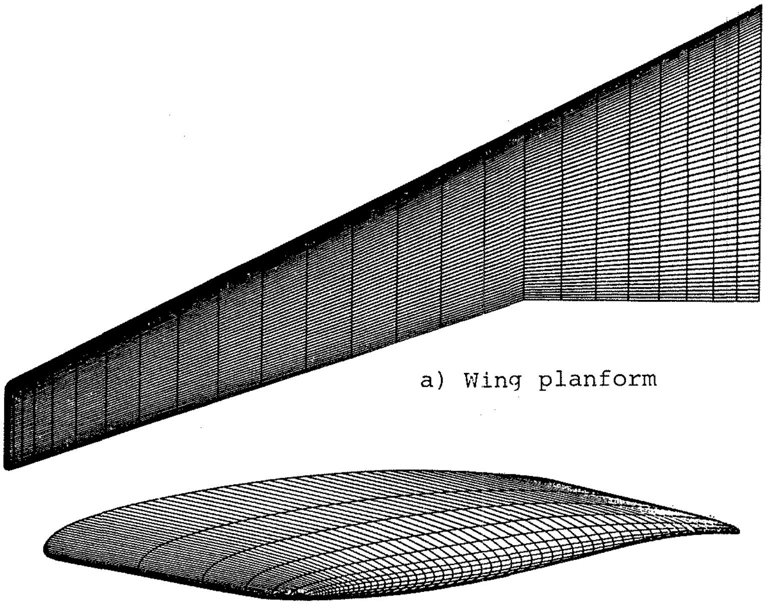
Figure 1: H-O-grid structure around large aspect ratio wing



a) Coordinates fixed at inner cut

b) Coordinates free at inner cut

Figure 2: Grid control using eqn. (3)



a) Wing planform

b) Wing tip

Figure 3: 129x29 Surface grid of DFVLR F4 wing

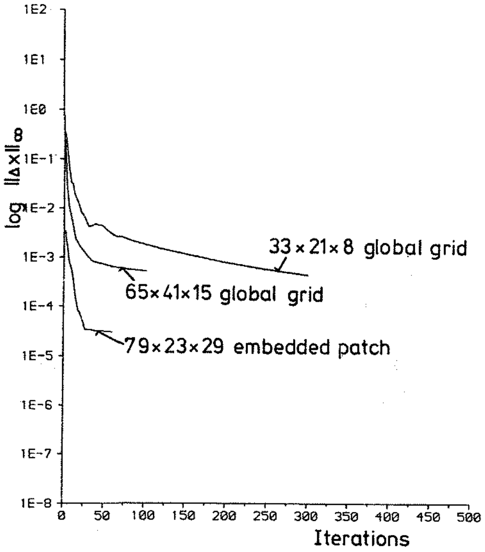
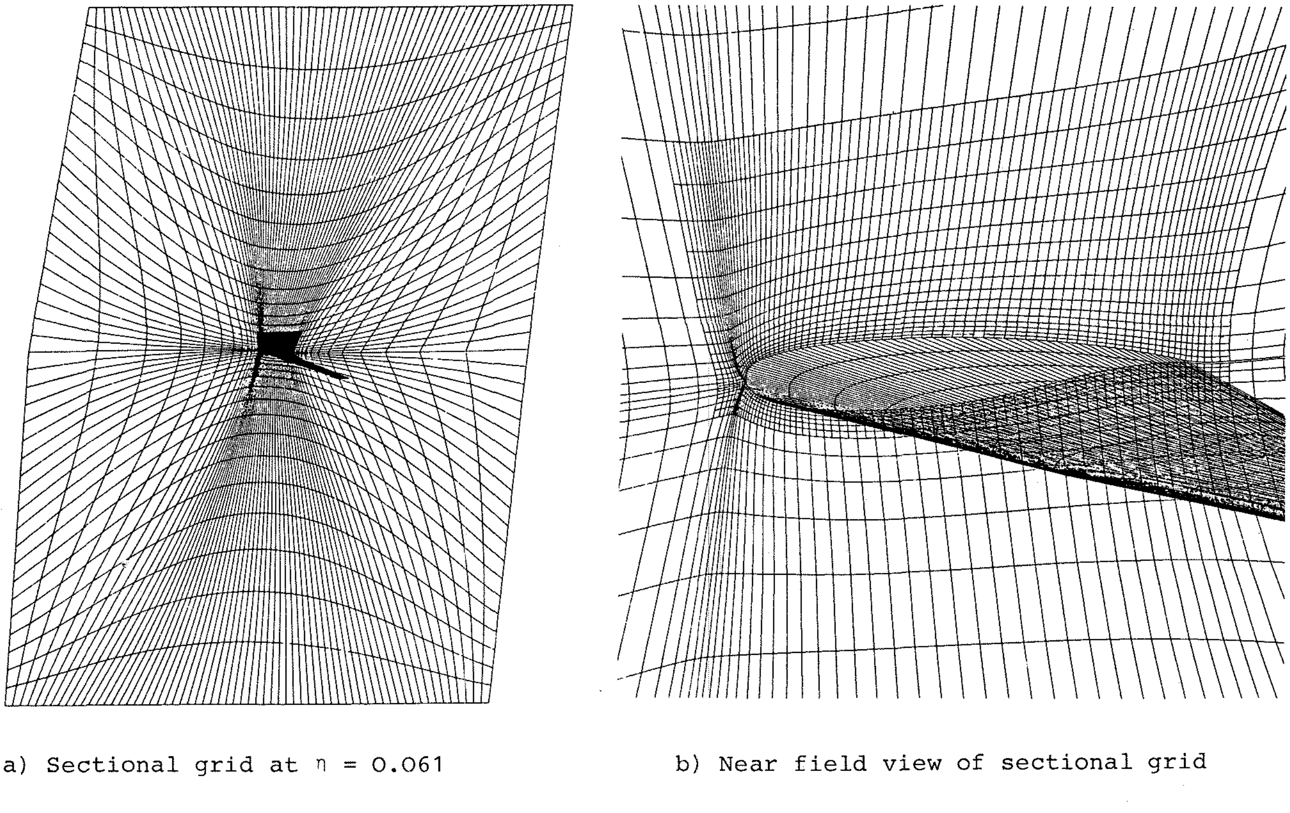


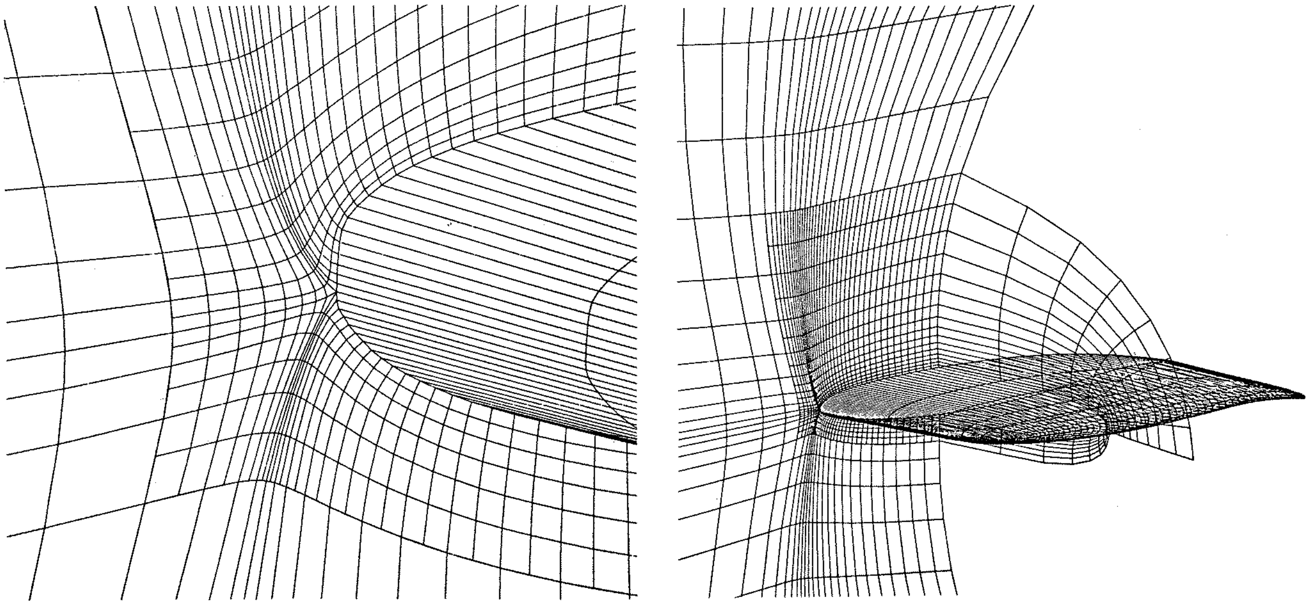
Figure 4: Convergence of grid generation method with iterative grid control



a) Sectional grid at η = 0.061

b) Near field view of sectional grid

Figure 5: Views on grid around DFVLR F4 wing



c) View on leading edge region

d) Wing tip region

Figure 5: continued

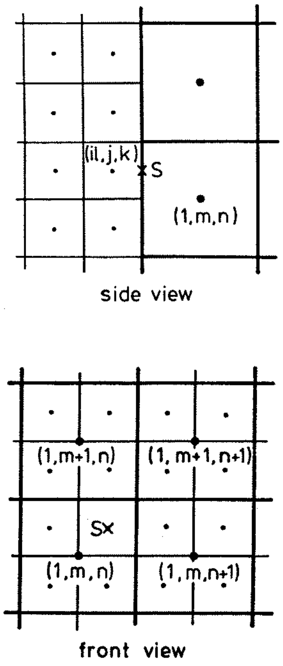
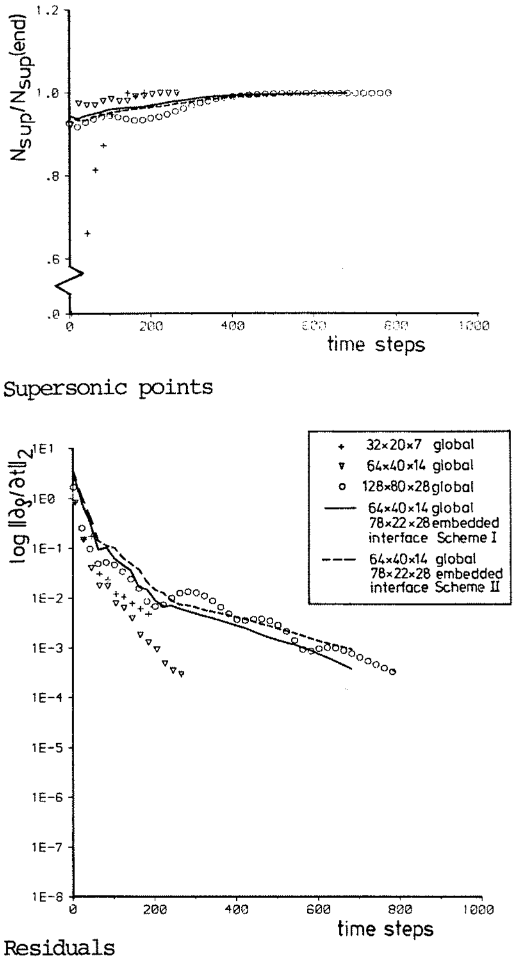


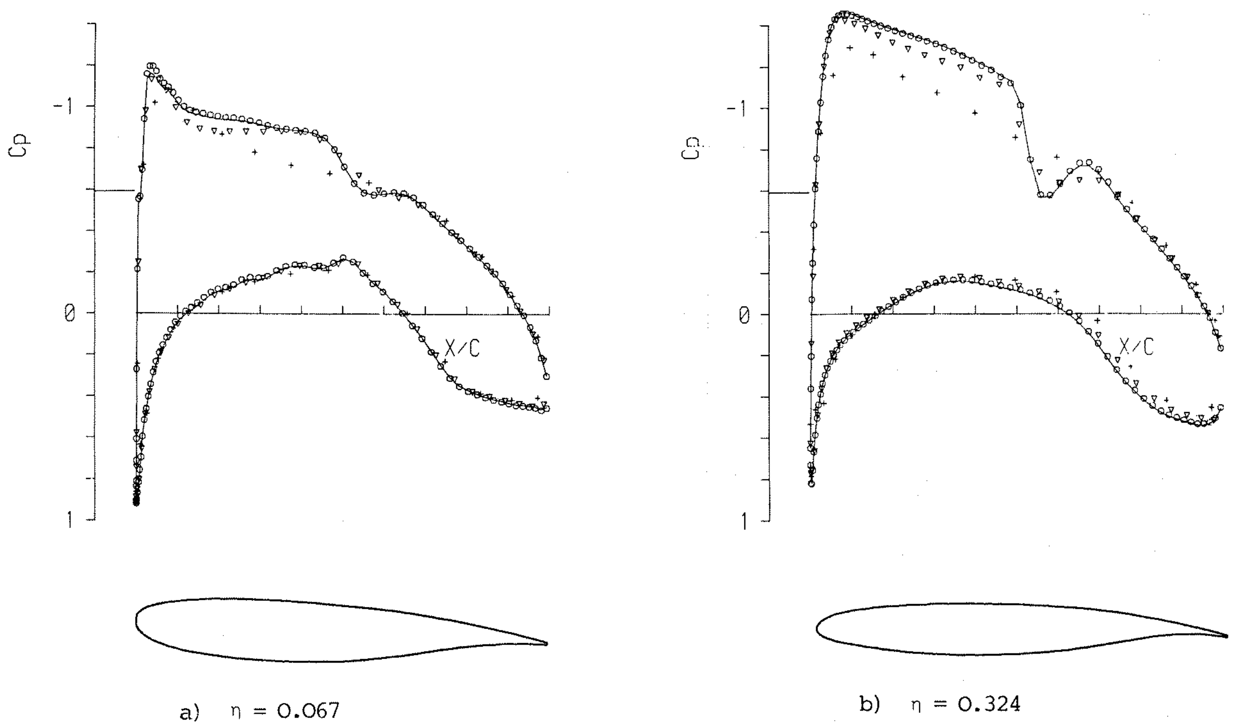
Figure 6: Embedded patch boundary in the computational domain



a) Supersonic points

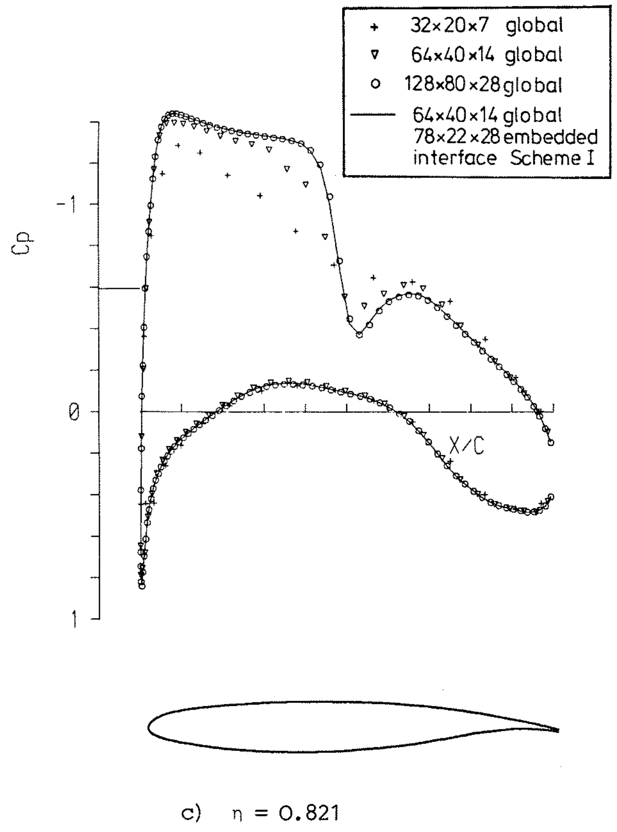
b) Residuals

Figure 7: Convergence history with and without local grid refinement, DFVLR F4 wing, $M_\infty = 0.75$, $\alpha = 0.34^\circ$



a) η = 0.067

b) η = 0.324



c) η = 0.821

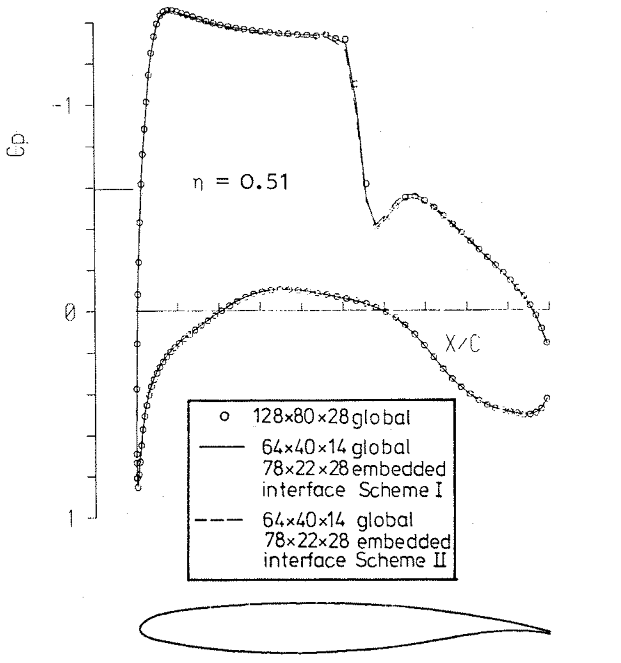
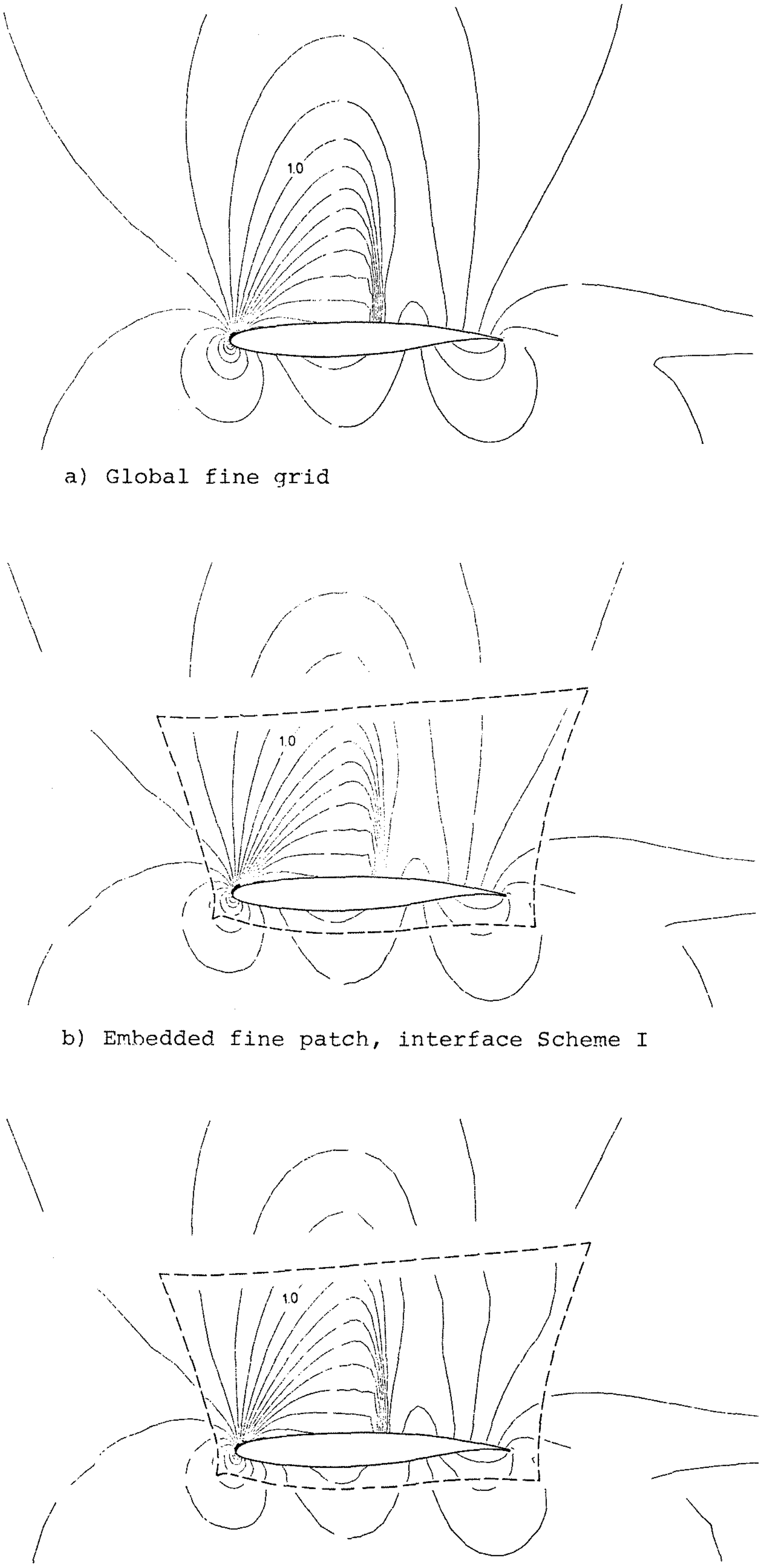


Figure 8: Pressure distributions with and without grid embedding DFVLR F4 wing, $M_\infty = 0.75$, $\alpha = 0.84^\circ$

Figure 9: Pressure distributions at midspan station: Influence of interface schemes



a) Global fine grid

b) Embedded fine patch, interface Scheme I

c) Embedded fine patch, interface Scheme II

Figure 10: Mach contours at $\eta = 0.51$, DFVLR F4 wing, $M_\infty = 0.75, \alpha = 0.84^\circ$

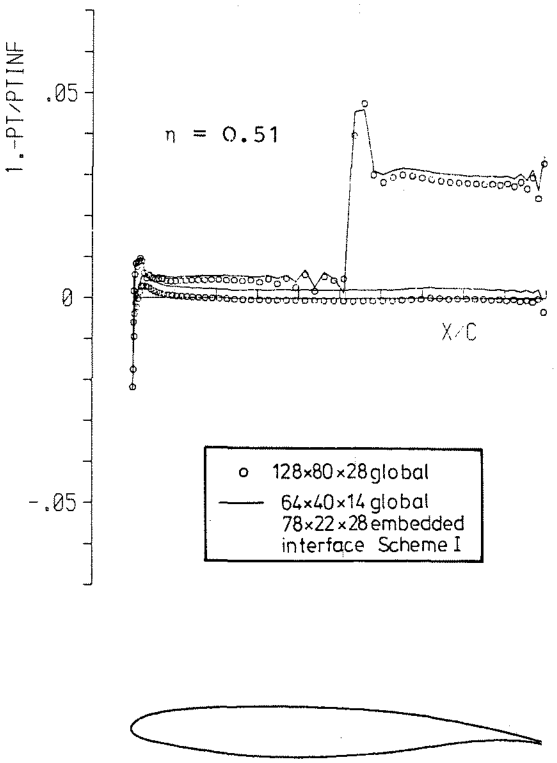


Figure 11: Distribution of total pressure at midspan station DFVLR F4 wing, $M_\infty = 0.75$, $\alpha = 0.84^\circ$