EVALUATION AND REDUCTION OF NUMERICAL DIFFUSION EFFECTS IN VISCOUS AEROFOIL FLOW CALCULATIONS

by

P.Tattersall

Submitted for the degree of Doctor of Philosophy in the Department of Transport Technology, Loughborough University

July 1993

Dedication

To Sue, all the others I love, and to long-term scientific research.

Summary

The Reynolds-averaged Navier-Stokes (RANS) equations form the most accurate model of viscous flow which can currently be solved computationally on a routine basis for practical engineering problems, given the size and cost of present-day computers. Before RANS solution methods can be used with confidence for the design of aircraft components, a number of areas related to solution accuracy must be investigated, one of which is numerical diffusion. Numerical diffusion, arising from the discrete solution method employed, is necessary to ensure numerical stability, but if too much is included the ability to predict physical phenomena (particularly diffusive ones) accurately can be seriously impaired, with obvious implications for the rational assessment of turbulence models.

The amount of numerical diffusion in solutions of the RANS equations is evaluated in the present work using two currently popular algorithms, for aerofoil flow test cases. The effect of the numerical diffusion on the prediction of physical processes is investigated, as is the behaviour of the numerical diffusion and corresponding solution when grid quality and algorithm smoothing parameters are varied. Results are presented in two ways, line diagrams giving detailed information along individual grid lines, and contour plots (showing a quantity called the Numerical Diffusion Ratio, NDR) giving overall information on accuracy of the solution throughout the field. The level of numerical diffusion in certain parts of the solution is shown to be unacceptably high in a number of cases.

Methods for modifying the NDR are investigated, with the aim of making it suitable for use as a "weighting function" for guiding automatic grid adaptation, to improve solution accuracy. It is shown that some of the modified forms of NDR can be used successfully in this manner. The advantages and disadvantages of using such a solution-accuracy measure (as opposed to the usual solution-activity measures) are discussed and some conclusions and recommendations are made.

Acknowledgement

I am responsible for the work submitted in this thesis. The original work is my own except as specified in the following acknowledgements. Sections of the thesis (notably Chapter 2 and Section 4.2) which are necessary for completeness, but which purely describe work due to others, are clearly indicated as such.

The work was carried out at the Royal Aerospace Establishment, Farnborough with the support of funding from the Department of Trade and Industry and the Procurement Executive of the Ministry of Defence. Thanks are due to a number of people within the Basic Aerodynamics Division at RAE who have helped me along the way, particularly to Prof B R Williams and Dr P D Smith for providing me with a smooth path, and to Dr D Catherall for help and advice with the use of his grid adaptation codes.

My thanks also to all those at RAE, Imperial College and Loughborough University who have struggled with the vagaries and complexities of my registration at various confusing points along the way.

Finally, this work simply would not have been completed (or perhaps even started) without the help of my Supervisor, Prof J McGuirk. His advice and suggestions, and particularly his enthusiastic commitment, have kept me going at times when I could easily have become disillusioned with the whole enterprise.

List of Contents

Page

Summary (i)
Acknowledgement (ii)
List of Contents (iii)
List of Figures (vi)
List of Symbols (xiv)

1. Introduction 1

1.1 Computation of Viscous Flows 2
1.2 Current Status of Methods for RANS Equations 4
1.2.1 Numerical Diffusion 9
1.3 Previous Work on Accuracy of RANS Methods 11
1.4 Aims of Present Work 13

2. Solution Algorithms for RANS Equations 15

2.1 Equations to be solved 16
2.2 Solution Algorithms to be used 19
2.2.1 Implicit Algorithm 20
2.2.2 Explicit Algorithm 22
2.2.2.1 Smoothing for Explicit Algorithm 24
2.3 Summary and Comments on the Smoothing Algorithms 25

3. Effects of Numerical Diffusion on Solution Accuracy 27

3.1 Numerical Diffusion Estimation 28
3.1.1 Numerical Diffusion Estimation in Indirectly-Added Diffusion Schemes 29
3.2 Numerical Diffusion Effects - Implicit Algorithm 29
3.2.1 Aspects of Grid Quality 30
3.2.2 Numerical Smoothing Coefficient 34
3.2.3 Grid Density Effects 35
3.3 Numerical Diffusion Effects - Explicit Algorithm 37
3.3.1 Subsonic Flow 37
3.3.2 Transonic Flow 39
3.4 Summary 39

4. Forms of Numerical Diffusion Estimates 41

4.1 Measures for Guiding Grid Adaptation 42
4.2 Grid Adaptation Methods to be used 43
4.3 Modification of Existing Measure (NDR) 47
4.3.1 Cut-off NDR Approach 49
4.3.2 Filtering Approach 50
4.4 Directionality 53
4.5 Summary 54

5. Results of Grid Adaptation 55

5.1 Grid Adaptation Examples 56
5.1.1 Example using Solution Activity Measure 58
5.1.2 Example using Solution Accuracy Measure - NDF 60
5.1.3 Use of Combined Solution Activity/Accuracy Measure 62
5.2 Use of Coarser Grids 63

5.2.1 Very Coarse Grids 64
5.2.2 Intermediate Grids 65
5.3 Summary and Implications 66

6. Conclusions 68

6.1 Conclusions and Implications of Present Work 69
6.2 Recommendations for Future Work 71

References 72
Tables 81
Figures 89
Appendices 154

List of Figures

1.1 Schematic showing practical limits of methods for calculation of two-dimensional viscous flow around aerofoils.

2.1 Coordinate systems for aerofoil calculation.

2.2 Smoothing fluxes for cell-centre scheme.

3.1 Sketch of grid extent and plotting locations.

3.2 Grid A - trailing-edge region.

3.3 Balance of terms in solution - Grid A - station 1.

3.4 Balance of terms in solution - Grid A - station 2.

3.5 Balance of terms in solution - Grid A - station 3.

3.6 Particle paths - Grid A - recirculation region.

3.7 Grid A - NDR.

3.8 Grid B - trailing-edge region.

3.9 Balance of terms in solution - Grid B - station 2.

3.10 Balance of terms in solution - Grid B - station 3.

3.11 Particle paths - Grid B - distinct recirculation regions.

3.12 Grid C - stretched to reduce flow/gridline angle.

3.13 Particle paths - Grid C - recirculation region.

3.14 Grid C - NDR.

3.15 Balance of terms in solution - Grid C - station 3.

3.16 Balance of terms in solution - Grid C - station 2.

3.17 Jacobian distribution along station 2, Grid C.

3.18 Balance of terms in solution - Case C(i) - station 2.

3.19 Particle paths - Case C(i) - recirculation region.

3.20 Case C(i) - NDR.

3.21 Balance of terms in solution - Case C(ii) - station 1.

3.22 Balance of terms in solution - Case C(ii) - station 2.

3.23 Particle paths - Case C(ii) - recirculation region.

3.24 Case C(ii) - NDR.

3.25 Balance of terms in solution - Grid D - station 2.

3.26 Particle paths - Grid D - some recirculation present.

3.27 Grid D - NDR.

3.28 Balance of terms in solution - Grid E - station 1.

3.29 Balance of terms in solution - Grid E - station 2.

3.30 Particle paths - Grid E - recirculation region.

3.31 Grid E - NDR.

3.32 Balance of terms in solution - Grid C - station 2 - explicit algorithm.

3.33 Balance of terms in solution - Grid C - station 3 - explicit algorithm.

3.34 Grid for compressible explicit case - trailing-edge region.

3.35 Balance of terms in solution - Compressible explicit case - station 2.

3.36 Particle paths - Compressible explicit case - recirculation region.

3.37 Compressible explicit case - NDR.

3.38 Particle paths - Grid B (compressible explicit case) - recirculation regions.

3.39 Particle paths - Grid C (compressible explicit case) - recirculation region.

3.40 Balance of terms in solution - Grid B (compressible explicit case) - station 2.

3.41 Balance of terms in solution - Grid B (compressible explicit case) - station 3.

3.42 Balance of terms in solution - Grid C (compressible explicit case) - station 2.

3.43 Balance of terms in solution - Grid C (compressible explicit case) - station 3.

3.44 Mach number contours - Turbulent case - explicit algorithm.

3.45 x/c = 0.25 - Turbulent case - explicit algorithm.

3.46 x/c = 0.25 - Turbulent case - explicit algorithm, detail close to surface.

4.1 Equidistribution principle in 1D.

4.2 Weighting functions for grid adaptation.

4.3 Spring analogy in 2D.

4.4 Physical diffusion / max(Convection, Pressure gradient) - station 3, Case A.

4.5 NDR - station 3, Case A.

4.6 Physical diffusion / max(Convection, Pressure gradient) - x/c = 0.25, Turbulent case - explicit algorithm.

4.7 NDR - x/c = 0.25, Turbulent case - explicit algorithm.

4.8 Physical diffusion / max(Convection, Pressure gradient) - x/c = 5.0, Turbulent case - explicit algorithm.

4.9 NDR - x/c = 5.0, Turbulent case - explicit algorithm.

4.10 Contour for 10% NDR cutoff - Case A.

4.11 Original NDI - trailing-edge region (Grid B).

4.12 Original NDR - trailing-edge region (Grid B).

4.13 Original NDI - station 3 (Grid B).

4.14 Physical diffusion / max(Convection, Pressure gradient) - station 3 (Grid B).

4.15 Original NDR - station 3 (Grid B).

4.16 Adapted grid - trailing-edge region.

4.17 Particle paths after continuation run on adapted grid.

4.18 NDI after calculation on adapted grid.

4.19 Balance of terms in solution - Adapted solution - station 3.

4.20a Solution-activity weighting function (streamwise direction) - Grid A.

4.20b Solution-activity weighting function (normal direction) - Grid A.

4.21 NDI - Grid A.

4.22 NDI - Turbulent case (5°) - explicit algorithm.

4.23 Solution-activity weighting function - Turbulent case (5°) - explicit algorithm.

4.24 NDI - x/c = 0.45, Turbulent case (5°) - explicit algorithm.

4.25 NDR - x/c = 0.45, Turbulent case (5°) - explicit algorithm.

4.26 NDFξ - Turbulent case (5°) - explicit algorithm.

5.1 Alternative grid adaptation strategies.

5.2 Turbulent (3°) case - initial grid.

5.3 Turbulent (3°) case - initial Mach number contours.

5.4 Turbulent (3°) case - DMDP (solution activity) weighting function on initial grid.

5.5 Turbulent (3°) case - C_p plot for calculation on initial grid.

5.6 Turbulent (3°) case - Grid after first adaptation to DMDP.

5.7 Turbulent (3°) case - Mach number contours after first adaptation to DMDP.

5.8 Turbulent (3°) case - DMDP weighting function on first adapted grid.

5.9 Turbulent (3°) case - Grid after second adaptation to DMDP.

5.10 Turbulent (3°) case - Mach number contours after second adaptation to DMDP.

5.11 Turbulent (3°) case - DMDP weighting function on second adapted grid.

5.12 Turbulent (3°) case - NDFξ on initial grid.

5.13 Turbulent (3°) case - Grid after adaptation to NDF - $\lambda_E = 1.0$.

5.14 Turbulent (3°) case - Grid after first adaptation to NDF - $\lambda_E = 0.5$.

5.15 Turbulent (3°) case - Mach number contours after first adaptation to NDF.

5.16 Turbulent (3°) case - NDFξ on first adapted grid.

5.17 Turbulent (3°) case - Grid after fourth adaptation to NDF.

5.18 Turbulent (3°) case - Mach number contours after fourth adaptation to NDF.

5.19 Turbulent (3°) case - NDFξ on fourth adapted grid.

5.20 Turbulent (3°) case - Grid after second adaptation to COMB (combined measure).

5.21 Turbulent (3°) case - Mach number contours after second adaptation to COMB.

5.22 Turbulent (3°) case - COMB weighting function on second adapted grid.

5.23 Turbulent (5°) case - Mach number contours on grid c5 (initial).

5.24 Turbulent (5°) case - Mach number contours on grid c53.

5.25 Turbulent (5°) case - Mach number contours on grid c54.

5.26 Turbulent (5°) case - Mach number contours on grid c55.

5.27 Grid c55.

5.28 Turbulent (5°) case - coarse grid - initial DMDP weighting function.

5.29 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to DMDP.

5.30 Turbulent (5°) case - coarse grid - grid after fourth adaptation to DMDP.

5.31 Turbulent (5°) case - coarse grid - grid after fourth adaptation to NDF - Strategy A.

5.32 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to NDF - Strategy A.

5.33 Turbulent (5°) case - coarse grid - grid after fourth adaptation to NDF - Strategy B.

5.34 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to NDF - Strategy B.

5.35 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to DMDP.

5.36 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to DMDP.

5.37 Turbulent (5°) case - intermediate grid - Mach number contours after fifth adaptation to DMDP.

5.38 Turbulent (5°) case - intermediate grid - grid after fifth adaptation to DMDP.

5.39 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to NDF - Strategy A.

5.40 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to NDF - Strategy A.

5.41 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to NDF - Strategy B.

5.42 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to NDF - Strategy B.

C.1 Stagnation point flow - quantities used in Appendix C.

List of Symbols

a	speed of sound
$\hat{A}$, $\hat{B}$, $\hat{M}$	Jacobian matrices
$A(w)_{ij}$, $C(w)_{ij}$, $D(w)_{ij}$	contributions to flux change in explicit algorithm arising from numerical smoothing, convective processes and physical diffusion respectively
c_v, c_p	specific heat at constant volume, pressure
C_L, C_D, C_p	coefficients of lift, drag, pressure, respectively
$d_{i+½,j}$, etc.	numerical smoothing fluxes in explicit algorithm
D_i, D_e	numerical smoothing operators in implicit algorithm
e	internal energy
E	total energy
F,G	vectors of inviscid dependent variables
f	arbitrary quantity in Chapter 1
$F(U)$	vector of spatial derivatives (Chapter 1)
g_{11}, etc.	metrics of coordinate transformation in elliptic grid generation method
J	Jacobian of coordinate transformation
k	thermal conductivity coefficient
M, M_∞	local, free stream Mach numbers
n	unit normal
p	pressure
P, Q, $\overline{P}$, $\overline{Q}$	functions in elliptic grid generation method
Pr	Prandtl number
q	heat flux vector
R	univeral gas constant
R, S	vectors of viscous variables
Re	Reynolds number
s	entropy
S	scaling factor in explicit algorithm numerical smoothing
t	time

T	temperature
u, v, w	cartesian velocity components
U, V	contravariant velocity components
	U also vector of flow quantities (Chapter 1)
w	vector of flux variables
$\hat{w}$	curvilinear form of above (e.g)
W	weighting function for grid adaptation
x, y, z	cartesian coordinates
β_c	heat flux (2D)
δ_ξ, δ_η, $\bar{\delta}_\eta$, Δ, ∇	spatial difference operators (Chapter 2)
δ_{ij}	Kronecker delta
Δt	time step
Δx	grid spacing in x direction
ε_i, ε_e	smoothing coefficients in implicit algorithm
ε_2, ε_4	smoothing coefficients in explicit algorithm
γ	ratio of specific heats (=1.4 in air)
λ	bulk viscosity
λ_i, λ_j	spectral radii
λ_L, λ_P, λ_E	controlling parameters in LPE grid adaptation method
μ	viscosity coefficient
θ	variable in implicit/explicit scheme (Chapter 1)
ρ	density
σ_{xy}	viscous stresses (2D)
τ_{ij}	viscous stress tensor (3D)
Ω_{ij}, $\partial\Omega_{ij}$	domain and surface of computational cell labelled ij
ξ, η	general curvilinear coordinates
ξ_x, etc.	first derivative of ξ, etc.
∂	partial differential operator

Subscripts

i,i+1, etc.	discretisation counters in x direction (e.g.)

lam	laminar
tot	total
turb	turbulent

Superscripts

n,n+1, etc.	time levels
$\bar{p}$, p'	mean, fluctuating components of (e.g.) pressure from time-averaging
$\tilde{u}_i$, u_i''	mean, fluctuating components of (e.g.) velocity from mass-weighted averaging
$\overline{\rho u_i'' u_j''}$	Reynolds stresses

Page

1. Introduction 1

1.1 Computation of Viscous Flows 2

1.2 Current Status of Methods for RANS Equations 4

1.2.1 Numerical Diffusion 9

1.3 Previous Work on Accuracy of RANS Methods 11

1.4 Aims of Present Work 13

1.1 Computation of Viscous Flows

Computational Fluid Dynamics (CFD) involves the numerical solution of equations describing fluid motion. Its use in the practical design of aerodynamic shapes for aircraft components is becoming ever more widespread. Due to limitations of size and speed of the available computers, approximations to the complete governing equations of fluid motion have to be made in order to allow codes for use in this manner. These codes can still be of immense value to the aircraft designer, as long as they can produce consistent and reliable results. This has been proved by the success, in the design environment, of methods based on the transonic small perturbation (TSP) equation[1], and viscous-inviscid interaction models (e.g. BVGK[2]), both of which make restrictive assumptions about the flow. As computer technology and numerical techniques improve, the approximations which must be made continue to become less restrictive. Nevertheless, if current trends continue, it will still be many decades before the complete equations can be solved affordably for realistic aircraft geometries. Even then, cheaper methods, based on approximations to the full equations, will always be heavily utilised when modelling of the full complexity of the situation is not required - particularly in the initial stages of the design process where many different designs might be tested with only a limited range of attributes, such as lift and drag, in mind.

A considerable range, or hierarchy, of fluid flow models can be built up by starting with a complete description of the flow and progressively introducing additional approximations. The Navier-Stokes equations describe completely the fluid flow in the regimes of interest to an aircraft designer. The non-linear nature of this system of partial differential equations means that analytic solutions can be found only for simple cases of limited interest, so numerical solutions must be sought. However, the Navier-Stokes equations are currently unsolvable numerically at the Reynolds numbers of interest because the disparate range of time- and length-scales involved could only be simultaneously resolved accurately with computing resources many times greater than those currently available. To date, numerical solution of the full equations, termed Direct Simulation, and the closely related idea of Large Eddy Simulation (in which a "subgrid model" is supplied for the turbulent eddies which are too small to be resolved by the discrete grid being used), have only been attempted[3,4] for relatively low Reynolds numbers and very simple geometries (e.g. flow around a cube[4] at a Reynolds number of approximately $5x10^4$). This compares with Reynolds numbers as high as 10^8 in real aircraft flight.

In order to address the problem of computer-resource limitations, an important approximation to the full equations is obtained by assuming that the velocity components are composed of mean and fluctuating parts, which leads to the Reynolds-averaged Navier-Stokes (RANS) equations. Computation times are significantly lower than for the full equations but the most obvious penalty is the problem of evaluating the additional fluctuating terms introduced in the Reynolds-averaged form. This is the problem of turbulence modelling. The RANS equations are currently attracting widespread interest and the numerical accuracy of solution methods, along with the physical accuracy of the turbulence models, are important issues, both of which can affect how well the solution compares to that observed, for instance, in an experiment. The physical accuracy of the turbulence model depends on the assumptions made in deriving it and also on the fact that even the most sophisticated model requires the specification of certain constants which can only be supplied by comparison with experiment and can, therefore, be case-dependent. Some more detail is given in Sections 1.2 and 2.1 but turbulence modelling in itself is not considered further in this work.

The numerical accuracy, which is the subject of this thesis, is the accuracy with which the computed solution of the particular set of equations being used reproduces the solution which would be obtained if it were possible to solve these equations exactly. This is what is meant throughout the rest of this work by "numerical" or "solution accuracy." The difference between the computed solution and the "exact" solution is sometimes referred to as the "solution truncation error (STE)." If a sufficiently fine grid (which would usually be much finer than is affordable on a routine basis) is used, the numerical solution should be very close to the "exact" solution, i.e. the solution accuracy would be high and the STE would be low. This is explained in more detail in Section 1.2. It is worth noting that, with the definition used here, a solution can be considered to be accurate even if it does not agree with an experimental observation, the implication being that the differences not due to experimental error must then be due to the physical modelling, i.e. the equations being solved do not describe the flow sufficiently well. In the context of RANS solutions this means that the turbulence model is inadequate. Clearly it is not valid to make judgements about the relative performance of turbulence models if nothing is known of the solution accuracy, and this observation provides the motivation for the work described here.

Returning to the hierarchy of models, the next major assumption which can be made is to neglect the viscous nature of the fluid completely, leading to the Euler equations. This model has obvious drawbacks but can predict important phenomena, such as shock waves, well. Making one further assumption - irrotational flow - leads to the full-potential equations. These can again be solved more quickly than their predecessor in the hierarchy but the biggest price to be paid is the loss of the ability to resolve shock waves correctly. It is possible to "adjust" the shock position[5] but this is not completely successful.

Other approximations are possible but we will stop here. An important class of methods which fit somewhere between the RANS equations and the inviscid models are viscous-inviscid interaction (VII) techniques, which represent the flow as being composed of two parts. The flow near a body is assumed viscous, and usually represented by a boundary-layer (parabolic) approximation to the full RANS equations, and the flow outside this region is assumed to be inviscid. The inviscid region is usually represented by the full-potential equations[2] or the Euler equations[6]. A schematic of the practical limits of these methods for two-dimensional flow around a single aerofoil is shown in fig.1.1, which indicates the physical complexity which can be modelled by each approximation and the associated time for a single calculation, together with an indication of the situations for which the use of a more expensive method is worthwhile.

1.2 Current Status of Methods for RANS Equations

The design of aircraft is an inherently three-dimensional (3D) process and a long-term goal of CFD developers is the use of a 3D RANS solution method for routine design applications. Indeed, a number of exploratory calculations have been attempted along these lines already. For example, particularly impressive results have been shown by Rizk and Gee[7] for a time-accurate RANS solution method, in which the unsteady interaction of a vortex from a leading-edge wing extension with the tail fin of an F-18 aircraft was modelled. Both inviscid[8] and viscous[9] calculations around representative configurations have also been presented by a number of authors, which represent very impressive achievements. However, quite apart from the computational expense of such calculations (some of these can use more than a day of computing time on a dedicated "supercomputer" such as a Cray), comprehensive attempts have rarely been made to demonstrate their accuracy. This is partly because many of the calculations have been performed purely to demonstrate a state of the art capability rather than for serious predictive purposes, and the expense is also a factor because grid refinement exercises, in which the trend of the changes in the solution on successively finer grids is considered, are very costly.

Two-dimensional (2D) RANS methods are, therefore, of great interest, and are concentrated on in this thesis, for a number of reasons. Firstly, 2D flows are much cheaper to calculate than 3D ones. Second, the range and availability of solution methods and test cases in 2D is currently much greater, with a lot of work having been done on validation[10] through use of test problems. Finally, the issues of interest in this work apply in 3D without any extra conceptual complications, so that use of 2D does not unduly simplify the problem. Also, only "steady" as opposed to "unsteady" flows are considered. This means that the variation of the solution in time is not important, only the final, unchanging or converged solution. Steady flows are of real importance in many applications and the justifications given above for studying 2D, rather than 3D, flow apply similarly to the examination of steady, in preference to unsteady, flow. The state of CFD has advanced to the stage where it is now possible to contemplate the use of RANS solution algorithms in steady two-dimensional situations for design purposes, to check flow characteristics at selected locations. It is certainly still the case that RANS solution methods are computationally expensive for routine use in design, a typical calculation taking 5-10 minutes of Central Processor Unit (CPU) time on a supercomputer as opposed to a few seconds for an Euler or even a VII calculation. However, past and present trends in computer speed and algorithm efficiency suggest that computing times for 2D RANS are likely to be down to a few seconds within a decade[11]. Before the use of these methods for design can become a reality, a large measure of confidence must be built in their ability to give consistent, reliable and accurate results rapidly. That is, the limitations of the current RANS schemes and the turbulence models they contain must be established. The issues which remain to be resolved before RANS algorithms can be used with confidence can be split into two clearly different but, nonetheless, interdependent areas: grid generation methods and flow solution algorithms. These can be thought of as relating, respectively, to geometric complexity and to flow complexity, but both can affect the accuracy of the solution.

The grids used in CFD problems are almost always "body-fitted" or "surface-aligned" - they are aligned with solid surfaces to facilitate the implementation of boundary conditions there. Also, for reasons of economy, the majority of points are usually clustered in certain parts of the grid, on the assumption that more points are needed where there is most "activity" in the solution. For example, there will be high velocity gradients across the viscous region near a solid surface, and in the region of a shock wave. Grid points can be clustered a priori where it is known that they will be needed, as in the case of a solid surface. With the example of a shock wave, however, it is not usually known in advance precisely where this will be, but points can be made to cluster near the shock wave during the course of the computation, by the process of grid adaptation,[12] in which the shock wave is detected by some means, in the course of the solution, and extra grid points are moved into the vicinity. The use of grid adaptation to improve solution accuracy is discussed in more detail in Chapter 4.

A grid is generally considered to be one of two types, structured[13,14] or unstructured[15,16]. In a 2D structured grid, for example, the grid points in the two directions (call these i,j) can be assigned labels (i-th point, j-th point) and the points (i,j), (i+1,j), (i,j+1), (i+1,j+1) will form the corners of a computational cell for a grid composed of quadrilaterals. In unstructured grids the region is usually filled with triangles and the labelling relationship described above cannot necessarily be applied. The structured grid is the most obvious and natural for simple geometries because it can be generated quite easily, and labelling of grid nodes is simple which leads to more efficient flow solution algorithms than is the case with unstructured grids. However, for complex 2D and particularly for 3D geometries, the generation of structured, body-fitted grids is very difficult because of the need to align the grid to surfaces which can meet in awkward ways, such as wing/pylon intersections, for example. An unstructured grid is far easier to generate in these circumstances but there are drawbacks. Apart from the degradation of flow algorithm efficiency already mentioned, it is yet to be convincingly demonstrated that a similar standard of solution accuracy can be achieved as with a structured grid. Therefore, some variation on the structured grid approach is still the most widely used for complex geometries. A detailed description of each variation is not relevant here, but the possibilities fall, very broadly, into two categories. In the multiblock[17] approach, the region is split into an unstructured collection of blocks, one associated with each particular feature (solid surface, viscous wake, etc.), and each block is filled with a structured[17] or possibly an unstructured[18] grid. The grid in each block is generated such that points on a block boundary coincide with points on the boundary of an adjacent block so that the result can be thought of as a single grid with internal boundaries. The second category involves multiple mesh approaches such as FAME[19], in which one grid can overly another (others) in any given part of the field. One background grid is used for most of the field with separate grids again being associated with particular features. Unlike the multiblock approach, the result cannot be thought of as a single grid, and some interpolation of quantities between the separate grids is necessary during a flow calculation. Only structured grids are used throughout this work as they are much more common, for the reasons given, especially for 2D calculations, but unstructured grids do have some special merits for adaptation which are discussed in Chapter 5.

Flow solution algorithms take an even larger number of forms than do grid generation techniques. In this thesis, only steady flows are considered, and the flow regime of most interest involves distinct regions of subsonic and supersonic flow. As a result, the governing equations are of "mixed-type," which means that disturbances propagate through the subsonic region in a different manner to those propagating through the supersonic region, and solution techniques which are appropriate for the one case are not always appropriate for the other. By far the most widely used and highly developed technique in such cases is to retain time as an "auxiliary variable" so that the solution is progressed from an initial guess to a final steady-state solution via incremental steps in "pseudo-time." The term pseudo-time is used to emphasise that, when only the steady-state solution is required, convergence acceleration techniques such as multigrid[20] can be used which means that, whilst the accuracy of the final solution is unaffected, the approximate solution after each "time-step" is not the one that would be achieved if the computation were done in a time-accurate manner. The time-stepping is done in one of two ways - explicitly or implicitly - as described briefly here. When the partial differential equations governing the flow are discretised in space, the result is a set of first-order, non-linear, ordinary differential equations which can be written

$$\frac{dU}{dt} = F(U)$$

where U is the vector of unknowns and F is some vector depending on U. The vector U is often very large, as it contains a number of quantities (4 for 2D Euler or RANS), associated with every one of the points of the computational grid. A simple, first-order accurate time-stepping technique can be cast in the form

$$\frac{U^{n+1} - U^{n}}{\Delta t} = [\theta F(U^{n+1}) + (1-\theta)F(U^{n})] .$$

In the explicit method, $\theta = 0$ and information at any grid node (call this i) at time level n+1 is deduced purely from information at the previous time step n, usually from a restricted number of grid nodes, say the node i and its immediate neighbours i-1,i+1. In the implicit method, $\theta \neq 0$ and the information at the node i at time level n+1 cannot be deduced solely from information at the previous level. This requires the inversion of an influence matrix which can be expensive, but the attraction is that a larger time step can be taken[21] and, hence, the steady-state solution can be reached much more quickly. The matrix inversion time can be reduced significantly by using methods like approximate factorisation[22] in which one complicated matrix is split into 2 simpler ones (3 in 3D), each associated with a single coordinate direction. The descriptions implicit and explicit refer to the time-level at which the discretisation of the spatial-derivative terms in the equations is effected. The two approaches are both widely used and one representative of each type of method is examined in this thesis. There is also some choice regarding the type of discretisation used for the spatial derivatives at the chosen time-level. If the differential form of the equations is discretised the method is referred to as finite-difference. This is done in one of two ways. With a central-difference, a derivative at a point i, say, is approximated using information at points on either side, usually i-1 and i+1. Alternatively, the derivative can be approximated using information from one side only; for example, if information at the points i and i-1 is used, this is called an upwind, or backward, difference. A more sophisticated type of upwinding, due largely to the work of Roe[23], involves adding second-order corrections to a basic first-order upwind difference. This method is particularly effective for calculating discontinuities. The second choice for the form of the equations is to write them in integral form before discretisation - this is a finite-volume method. The two methods to be examined also represent these possibilities, the implicit method being of finite-difference type and the explicit one being finite volume.

In addition to these mathematical considerations, the physical modelling included in the equations to be solved is a matter of some choice. One relevant example of this, which is used in one of the solution methods studied later, is the thin-layer Navier-Stokes approximation in which the viscous layer is assumed to be so thin that velocity gradients in the direction across the layer are much greater than those in the direction of the layer (because the flow speed changes from zero at the wall to nearly the free-stream speed at the outer edge of the layer). This approximation holds for many flows, breaking down mainly when large separated regions are present, and means that a number of terms can be dropped from the equations, as shown in the next chapter, leading to lower computation times. A good analysis of the thin-layer Navier-Stokes equations, relating them to the simpler and often used boundary layer equations, is given by Blottner.[24]

At present, the most intensely studied issue with regard to physical modelling is that of turbulence. Turbulence models represent the effects of the fluctuating terms as extra diffusion in addition to the molecular diffusion. This is the factor which will limit the accuracy of RANS computer codes, given a sufficiently precise numerical discretisation procedure, and is the focus of a great deal of current research. The variety of turbulence models is so great that even a cursory review would take a number of pages. Some remarks are included in the next chapter, but the interested reader should see ref.25, for example, for a good introduction to the subject. In both algorithms studied later, one of the most basic models, that of Baldwin and Lomax[26] is used. This is very simple and consequently cheap, and works well as long as considerable separation is not present, as in the cases studied. As has already been stated, the aim of this thesis is to investigate to what extent the numerical solutions obtained from various current schemes are sufficiently accurate to allow a confident assessment of the validity of certain turbulence models. The actual validity of the various models is not investigated in this work.

The major topic of this work, numerical diffusion, leads to the introduction of diffusion-like effects into the solution as a result of the discretisation process. This can be affected by all of the factors listed above, such as the solution method and the type of grid used. It is of great importance particularly because it is invalid to make observations regarding physical modelling if the amount of error introduced by the numerical modelling is not known. Many terms such as numerical/artificial viscosity, numerical diffusion, numerical smoothing are used and are appropriate, under certain circumstances, for essentially the same phenomenon. These terms are explained in the next section.

1.2.1 Numerical Diffusion

In the previous section the term "solution truncation error (STE)" was introduced as representing the difference between the computed solution and the exact solution. Unfortunately, it is not usually possible to derive an expression for the STE because the exact solution is not known. It is possible, however, to estimate the solution truncation error indirectly by finding the truncation error of the numerical discretisation - the equation truncation error (ETE). The discretisation of the governing partial differential equations leads to a system of difference equations which includes terms, with no counterpart in the original system, which result from the truncation of the approximations to the derivatives. The combination of all these extra terms, from all the derivatives in a particular equation, gives the ETE for that equation. Although it cannot be formally proved that, for example, the STE is of the same order of accuracy as the ETE (it may be lower), if the ETE is substantially reduced the STE will be reduced also. Thus the accuracy of the solution may be estimated by calculating the ETE, and this is what is done in this thesis.

For example, if a gradient $\partial f/\partial x$ is discretised using a first-order accurate forward difference, the resulting Taylor series is given by:

$$\frac{\partial f}{\partial x} \approx \frac{f_{i+1} - f_i}{\Delta x} = \left.\frac{\partial f}{\partial x}\right|_i + \underline{\frac{\Delta x}{2!}\left.\frac{\partial^2 f}{\partial x^2}\right|_i} + \frac{(\Delta x)^2}{3!}\left.\frac{\partial^3 f}{\partial x^3}\right|_i + \frac{(\Delta x)^3}{4!}\left.\frac{\partial^4 f}{\partial x^4}\right|_i + \dots\dots$$

This is first-order accurate because the leading term of the truncation error (underlined) contains the grid spacing raised to the power 1. This term has the same form, $A\,\partial^2 f/\partial x^2$ (where A is related to the mesh spacing by $A = \Delta x/2$), as the physical diffusion terms in the original differential equations, hence the description "numerical diffusion," referring to the whole term, or "artificial viscosity," which refers to the quantity A. Since this extra term is similar to the physical terms, it can behave in a similar way to them and, hence, diminish the agreement between the numerical solution and the solution of the partial differential equations. In general, not just this second derivative but all the even-order derivatives, have a "diffusive" nature (which means that they tend to attenuate propagating waves) and, therefore, are of concern because they can contaminate the solution in the same way. Also, in general, all the odd-order derivatives have a "dispersive" nature (they tend to introduce spatial oscillations into propagating waves). A good explanation of this is given by Degrez[27].

If the extra dissipative terms are not present, non-physical spatial oscillations can occur in the solution, causing the numerical scheme to be unstable.[27] Thus, the extra terms are necessary for stability, hence the phrase "numerical smoothing," but, at the same time, they should be small enough not to greatly affect the accuracy. The terms can be added in at least two ways. With a first-order accurate discretisation, the terms arise naturally from the differencing. With a second-order accurate discretisation, the numerical diffusion term does not arise, but a third-order dispersive term does, and so, since the numerical diffusion is needed for stability, it must be added in some form $B\,\partial^4 f/\partial x^4$. To distinguish between the two cases, the numerical diffusion is referred to, in the remainder of the text, as being either "indirectly-added", via a first-order discretisation, or "directly-added", via a "numerical smoothing" algorithm, in the second-order case. A discretisation such as that of Roe[23] is of the indirectly-added smoothing type, but has a much lower truncation error than the basic first-order upwind method, because of the correction procedure employed. It was in the context of Euler calculations that the issue of numerical diffusion first arose, since numerical practices resulting in numerical diffusion are essential for calculations including shock waves (unless shocks are "fitted" rather than "captured"). Too much numerical diffusion can be a problem in Euler calculations, resulting in smeared shock waves and incorrect values for such quantities as the lift and drag. In general, however, higher levels are permissible than for viscous calculations because physical diffusive processes are not modelled by the equations, and so are not required to be predicted.

A turbulence model must predict diffusive phenomena, such as boundary-layer growth and separation, and the presence of numerical diffusion can make the success of a particular model impossible to assess. Although the treatment of numerical diffusion was much improved over the years of development of efficient and accurate solutions of the Euler equations, many researchers (e.g. refs.28,29) are now aware that methods which work in this case are much less successful at coping with the RANS equations in which physical viscous processes are modelled.

1.3 Previous Work on Accuracy of RANS Methods

Over the past few years much work has been done on assessing and improving the numerical accuracy of RANS solutions. The amount of numerical diffusion present in a solution, and, hence, the accuracy of the solution, can be estimated in a number of ways. In the case of terms added directly via a numerical smoothing algorithm, the algorithm-specific parameters which control the size of these terms can be varied, and the effects on global quantities, such as lift and drag coefficients, tabulated[30], so as to give information on the most appropriate values of these parameters for general use. This is quite time-consuming and it does not allow the user to extrapolate beyond the limits of the test. A particularly common and useful technique[31,32] for upwind schemes, as well as other types, is to calculate solutions on successively finer grids until further refinement no longer affects such quantities, implying a "grid-independent solution". Just like the previous technique, this sets guidelines for the user on what is generally sufficient, but to be absolutely sure it ought to be carried out for each significantly different set of flow conditions, which is expensive. Richardson extrapolation is a useful method which can be used, in conjuction with such grid refinement, either to extrapolate to a much more accurate solution, or to guide grid adaptation[33,34]. Again, this requires evaluation of a solution on more than one grid, and only yields an error estimate on the coarser of two grids. The effect of varying numerical diffusion (which can also be achieved by using different grids) on local solution properties has been studied[35]. Some researchers[28,29,36] have explicitly evaluated the distribution of numerical diffusion in a solution, for example by Richardson extrapolation or by a truncation error analysis for upwind calculations, and observed how this is affected by use of global, rather than local, grid refinement, until, again, grid-independence can (or cannot) be claimed.

The importance of examining the contribution of various terms, including some estimate of the numerical diffusion, to the balance of the solution for each constituent equation, has been realised for a number of years[36]. More recently, during the time-span of this thesis, this technique has received a great deal of attention. It has been used as a tool in developing new algorithms[37], or as a means of evaluating more established algorithms[38,39]. Often the conclusion[38] is that well-established algorithms can suffer serious accuracy problems, requiring substantial changes to the manner in which numerical smoothing is introduced.

Once the accuracy of a solution has been assessed, it often needs to be improved in some way. This can be done simply by global refinement of the grid[31,32] but computationally this is usually expensive and always inefficient. A much more attractive proposition is local grid refinement. The refinement method is generally based on node enrichment,[40,41] in which the total number of grid points is increased as a result of extra ones being added in important areas, or node movement,[42,43] in which existing points are moved to the important areas and the total number of points remains fixed. The refinement is usually guided by a measure of solution activity such as Mach number gradient, which is taken as an indirect indication of solution accuracy. The basic principle is that more points are required to produce an "accurate" solution where activity - given by gradients of solution variables - is highest. This is explained more fully in Chapter 4.

An alternative method of improving solution accuracy is to improve the algorithm itself on a fixed grid. Ways of doing this have been devised for directly-added smoothing algorithms by modification of the smoothing algorithm, after numerical experimentation[44] (altering the values of the user-specified parameters) or use of heuristic arguments[45,46] (scaling the amount of smoothing by a factor which behaves in a manner thought to be desirable - the factor is smallest where the least smoothing is postulated to be required). The problem with the former approach tends to be that the smoothing parameters remain case-sensitive, whilst the examples cited of the latter have only led to a relatively minor improvement. The ability to examine solution accuracy closely, on a point-by-point basis, can provide evidence of the success of such attempts, and suggest alternative ideas.

1.4 Aims of Present Work

Since numerical practices which lead to numerical diffusion are very often a necessity to stabilise a calculation, its complete elimination is unfeasible and its reduction to an acceptable level is the only realistic target. This, however, requires a definition of acceptable, which, for the purposes of a RANS solution, will depend on the relative magnitudes of the numerical and physical diffusion terms, and will be, at least to some extent, arbitrary (dependent on the maximum affordable number of grid points, etc.). In view of this and previous comments, it is clear that it is desirable to be able to assess the magnitude of the numerical diffusion present in a solution, and its relationship to the size of the physical diffusion terms.

The first part of this work, then, consists of trial calculations, using two algorithms representing, very broadly, the current range of popular solution methods for RANS, on grids which are typical of those which could normally be afforded for routine design purposes. The equations to be solved and the solution methods to be used are described in Chapter 2. In the first place, the use of these calculations allows an investigation of how the level of numerical diffusion can be assessed, and the results displayed, so as to provide useful information. Having assessed the amount of numerical diffusion in a solution, it is likely that it will also be desirable to reduce its magnitude. This is not certain, as it may be possible, given some experience, to make a judgement about the validity of a solution where the level of numerical diffusion is known, without recourse to further calculation. Indeed, by and large, this corresponds to current practice. Whether or not the typical numerical diffusion levels are too high is investigated in Chapter 3, as is the way in which these levels are affected by the sorts of factor which might easily be varied, such as the global fineness of the grid (i.e. the number of grid points) and the amount of directly added smoothing which is introduced (this can be controlled by user-specified parameters for algorithms of "directly-added" type, of which both methods investigated are examples).

There will, without doubt, be some instances in which the numerical diffusion levels are too high, and, although it may be possible to remedy the situation in a "hands on" manner, by global grid refinement (one of the possibilities investigated in Chapter 3), for example, it will be highly desirable, for routine use, to have an automatic method for improving solution accuracy. Logically, within the context of this work, it is appropriate to investigate a method for improving the accuracy which is based directly on the method used to assess the accuracy, which is the course taken in Chapters 4,5, where the option of local grid refinement via node movement is investigated. Rather than use solution-activity-based measures for guiding grid adaptation, functions based on the estimate of the numerical diffusion developed in the earlier part of the thesis, are used. The factors affecting the use of such a function are discussed in detail and the results of the alternative functions are compared with each other and with the typical solution-activity functions normally used. The current best attempt at using such a solution-accuracy based function within an automatic grid adaptation environment is investigated in Chapter 5, examining the advantages and disadvantages of such an approach, and finally some conclusions are drawn, and recommendations for further work are made.

Page

2. Solution Algorithms for RANS Equations 15

2.1 Equations to be Solved 16

2.2 Solution Algorithms to be used 19

2.2.1 Implicit Algorithm 20

2.2.2 Explicit Algorithm 22

2.2.2.1 Smoothing for Explicit Algorithm 24

2.3 Summary and Comments on the Smoothing Algorithms 25

The equations to be solved are described first, followed by the two solution methods to be used. The description of the important points of the algorithms, given below, is taken largely from the original authors. Various remarks are added where they bear on the uses to which the algorithms will be put in the later chapters, and the similarity of the two is emphasised. Full details of each algorithm are available in the references given below. It is not an objective of this research to produce a new RANS solution method, but some deficiencies in existing ones will be highlighted and possible improvements suggested in the later chapters.

2.1 Equations to be Solved

The Navier-Stokes equations (ignoring gravitational body forces) in cartesian coordinates x_j, j=1,2,3 ($x_1 = x$, $x_2 = y$, $x_3 = z$), in three dimensions, are given by

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} \qquad (2.1)$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(u_j(\rho E + p)) = \frac{\partial}{\partial x_j}(u_i \tau_{ij} - q_j)$$

where

$$E = e + \tfrac{1}{2}u_i u_i, \quad p = \rho R T \quad \text{and} \quad e = c_v T.$$

ρ is the density, the u_i are the cartesian velocity components, p is the pressure, e is the internal energy and E is the total energy. T is the temperature, R is the universal gas constant and c_v is the specific heat at constant volume. The subscripts i and j represent repeated summation over i,j= 1,2,3. The stress tensor τ_{ij} and the heat-flux vector q_j are given by

$$\tau_{ij} = \lambda\, \delta_{ij} \frac{\partial u_m}{\partial x_m} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

$$q_j = -k\frac{\partial T}{\partial x_j}$$

where λ is the bulk viscosity which is usually, as here, evaluated by Stokes' hypothesis, $\lambda = -\tfrac{2}{3}\mu$, μ is the viscosity, δ_{ij} is the Kronecker delta [=1 when i=j, =0 otherwise], and k is the thermal conductivity. The conservation equation for total energy is sometimes replaced by a simple relation, derived by assuming constant stagnation enthalpy (= specific heat at constant pressure (c_p) x stagnation temperature), which is valid as long as thermal effects are negligible (e.g. no external heating/cooling applied at solid surfaces, and no surface heating due to very high speed flows). In both the methods used here, however, the energy conservation equation is included.

To obtain the Reynolds-averaged Navier-Stokes equations from the above, the pressure and density variables are decomposed using a conventional time average[47], e.g. pressure $p = \bar{p} + p'$, where $\bar{p}$ is a mean component and p' is a fluctuating component, and the energy and each velocity component u_i are decomposed into a mass-weighted[47] average component and a fluctuating component so that, for example, $u_i = \tilde{u}_i + u_i''$, where $\tilde{u}_i = \overline{\rho u_i}/\bar{\rho}$. The resulting quantities are inserted into the equations (2.1), which are time averaged, giving the equations

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j) = 0$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{u}_i) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_i\tilde{u}_j) = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial \bar{\tau}_{ij}}{\partial x_j} - \frac{\partial}{\partial x_j}(\overline{\rho u_i'' u_j''}) \qquad (2.2)$$

$$\frac{\partial}{\partial t}(\bar{\rho}\tilde{E}) + \frac{\partial}{\partial x_j}(\tilde{u}_j(\bar{\rho}\tilde{E} + \bar{p})) = \frac{\partial}{\partial x_j}(\tilde{u}_i\bar{\tau}_{ij} - \bar{q}_j)$$
$$- \frac{\partial}{\partial x_j}(\tilde{u}_i\,\overline{\rho u_i'' u_j''}) - \frac{\partial}{\partial x_j}(\overline{\rho u_j'' e''}) + \frac{\partial}{\partial x_j}(\overline{u_i''(-p\delta_{ij} + \tau_{ij} - \tfrac{1}{2}\rho u_i'' u_j'')})$$

Equations (2.2) are the Reynolds-averaged Navier-Stokes equations in cartesian coordinates. The mean energy dissipation term $\overline{u_i''(-p\delta_{ij} + \tau_{ij} - \tfrac{1}{2}\rho u_i'' u_j'')}$ is usually neglected on an order of magnitude consideration. If the following definitions are adopted:

$$(\tau_{ij})_{tot} = \bar{\tau}_{ij} - \overline{\rho u_i'' u_j''}$$

$$(q_j)_{tot} = \bar{q}_j + \overline{\rho u_j'' e''}$$

the averaged equations (2.2) reduce to the same form as the instantaneous equations (2.1). The question of how to define the additional terms involving the fluctuating quantities is the problem of turbulence modelling. In the methods used later, the total stress tensor is assumed to be:

$$\tau_{ijtot} = \mu_{tot}\left(\frac{\partial \tilde{u}_i}{\partial x_j} + \frac{\partial \tilde{u}_j}{\partial x_i} - \frac{2}{3}\frac{\partial \tilde{u}_m}{\partial x_m}\right)$$

and the total heat flux vector is

$$(q_j)_{tot} = -c_p\left[\frac{\mu_{lam}}{Pr} + \frac{\mu_{turb}}{Pr_{turb}}\right]\frac{\partial \tilde{T}}{\partial x_j}\,.$$

Pr, Pr_{turb}, are the molecular and turbulent Prandtl numbers. $\mu_{tot} = \mu_{lam} + \mu_{turb}$, where μ_{lam} is the laminar (molecular) viscosity coefficient and μ_{turb} is the so-called eddy-viscosity coefficient. In the simplest turbulence models (algebraic models), as in that of Baldwin and Lomax[26] which is used in both solution methods described below, the eddy-viscosity coefficient is composed of an empirical constant, a characteristic length scale and a characteristic velocity scale. The velocity and length scales are correlated to the mean flow properties through algebraic laws. The Baldwin-Lomax model is outlined in Appendix A. More sophisticated turbulence models use extra differential equations to find the length and velocity scales. In a different class of models, Reynolds-stress transport models, a large number of extra equations must be solved, leading to much increased computing time. More comprehensive details of these turbulence modelling issues can be found, for example, in ref.25. The important point is that, with the eddy viscosity approach, the Reynolds-averaged equations, used for turbulent flows, and the Navier-Stokes equations to be solved for laminar flows, have exactly the same form (2.1), with the addition of a method to calculate the extra viscosity coefficient in the turbulent case. In this study, only 2D cases are considered so the governing equations are simplified accordingly (i.e. $u = u(x,y)$, etc.) and, in addition, they are non-dimensionalised[24]. Thus, the equations to be solved, dropping the averaging notation for simplicity, are:

$$\frac{\partial w}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = Re^{-1}\left\{\frac{\partial R}{\partial x} + \frac{\partial S}{\partial y}\right\} \tag{2.3}$$

where

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \qquad F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho vu \\ (\rho E + p)u \end{pmatrix}, \qquad G = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{pmatrix}$$

The viscous terms are given by

$$R = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \beta_x \end{pmatrix}, \qquad S = \begin{pmatrix} 0 \\ \sigma_{yx} \\ \sigma_{yy} \\ \beta_y \end{pmatrix}$$

where

$$\sigma_{xx} = 2\mu \frac{\partial u}{\partial x} + S_o , \qquad \sigma_{yy} = 2\mu \frac{\partial v}{\partial y} + S_o$$

$$\sigma_{xy} = \sigma_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)$$

$$S_o = \lambda \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$$

$$\beta_c = (\mu/Pr)(\partial T/\partial c) + u\sigma_{cx} + v\sigma_{cy}$$

where c = x or y, and Re is the Reynolds number.

These equations must be supplemented by physical boundary conditions at a solid surface - the no-slip condition $u = v = 0$ is appropriate for the momentum equations and an adiabatic wall (zero normal temperature gradient) condition is generally sufficient for the energy equation. In addition, numerical boundary conditions are needed at the edges of the computational region. These are not described in the following section as they are not an issue in this work, but details can be found in the references given for the two algorithms.

2.2 Solution Algorithms to be used

Throughout this thesis two RANS solution algorithms are used. To differentiate between the two, they are referred to in the remainder of the text by the terms implicit or explicit, as defined in Chapter 1. Both are obvious candidates for investigation as they are typical forms of algorithm which are widely used as research codes and as such are readily available. The explicit one, in particular, is being heavily investigated and developed by aircraft company research departments[48,49] with a view to regular use in the design of aircraft. Both algorithms have distinctive features but can still be classed together as, in the terminology introduced in the previous chapter, directly-added numerical diffusion algorithms. Both algorithms are used unmodified except for the addition of extra output routines which are detailed in Chapter 3, and, hence, any attempts to improve solution accuracy, outlined in Chapters 4 and 5, are undertaken by means of grid adaptation methods, using a fixed given solution algorithm.

2.2.1 Implicit Algorithm

The first algorithm to be described and used is an implicit, approximately factored, finite-difference algorithm adapted from Beam and Warming[22] or Briley and McDonald[50]. The code was originally written by Steger[51], subsequent developments having been reported by Steger and Pulliam[52,53]. The version of the code used is an early one and few of the improvements reported in ref.53, designed to increase efficiency, accuracy and convergence rate, had been incorporated. The improvements related to accuracy are to aid the calculation of shock waves, and, since this algorithm is only used here for subsonic flows, the omission of these improvements is acceptable.

For this algorithm, the Navier-Stokes equations are written in general curvilinear coordinates ξ and η [54] (see fig.2.1), with a thin-layer approximation (viscous derivatives in the normal direction are neglected) having been made. Dropping the averaging notation for simplicity, the equations to be solved are:

$$\frac{\partial \hat{w}}{\partial t} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} = Re^{-1}\frac{\partial \hat{S}}{\partial \eta} \qquad (2.4)$$

where $\hat{w} = w/J$ and

$$\hat{F} = J^{-1}\begin{pmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ (\rho E + p)U - \xi_t p \end{pmatrix}, \qquad \hat{G} = J^{-1}\begin{pmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ (\rho E + p)V - \eta_t p \end{pmatrix}$$

with

$$U = \xi_x u + \xi_y v, \qquad V = \eta_x u + \eta_y v.$$

U,V are the contravariant (i.e. grid-line oriented) velocities. The metrics of the transformation from Cartesian to curvilinear space are defined as

$$\begin{array}{ll} \xi_x = Jy_\eta & \xi_y = -Jx_\eta \\ \eta_x = -Jy_\xi & \eta_y = Jx_\xi \end{array}$$

and J is the transformation Jacobian

$$J = \xi_x\eta_y - \xi_y\eta_x = 1/(x_\xi y_\eta - x_\eta y_\xi) .$$

The vector of viscous terms is:

$$\hat{S} = J^{-1} \begin{pmatrix} 0 \\ \mu m_1 u_\eta + (\mu/3) m_2 \eta_x \\ \mu m_1 v_\eta + (\mu/3) m_2 \eta_y \\ \mu m_1 m_3 + (\mu/3) m_2 (\eta_x u + \eta_y v) \end{pmatrix}$$

with

$$m_1 = \eta_x^2 + \eta_y^2$$

$$m_2 = \eta_x u_\eta + \eta_y v_\eta$$

$$m_3 = \frac{1}{2}(u^2 + v^2)_\eta + Pr^{-1}\frac{\partial T}{\partial \eta} .$$

Equation (2.4) is the form of the RANS equations to be used with the implicit algorithm for laminar flows, with $\mu = \mu_{lam}$, and for turbulent flows with $\mu = \mu_{tot}$. The approximate-factorisation implicit method used to advance the solution from time t_n to time t_{n+1} (denoted by superscripts n,n+1) is given (see ref.51 for derivation) by

$$(I + \Delta t\delta_\xi \hat{A}^n - \Delta t D_i\big|_\xi)(I + \Delta t\delta_\eta \hat{B}^n - \Delta t Re^{-1}\bar{\delta}_\eta J^{-1}\hat{M}^n J - \Delta t D_i\big|_\eta)(\hat{w}^{n+1} - \hat{w}^n)$$

$$= -\Delta t(\delta_\xi \hat{F}^n + \delta_\eta \hat{G}^n - Re^{-1}\bar{\delta}_\eta \hat{S}^n + D_e \hat{w}^n) \qquad (2.5)$$

(1) (2) (3) (4)

where $\hat{A}$, $\hat{B}$ and $\hat{M}$ are the Jacobian matrices $\partial\hat{F}/\partial\hat{w}$, $\partial\hat{G}/\partial\hat{w}$ and $\partial\hat{S}/\partial\hat{w}$. δ_ξ and δ_η are second-order accurate central difference approximations. For the viscous terms a midpoint operator, $\bar{\delta}_\eta$, $(\bar{\delta}_\eta s_j \equiv (s_{j+\frac{1}{2}} - s_{j-\frac{1}{2}})/\delta\eta)$ is used. The added numerical smoothing terms D_e and D_i are defined as

$$D_e = \varepsilon_e \Delta t J^{-1}\left[(\nabla\Delta)_\xi^2 + (\nabla\Delta)_\eta^2\right] J \tag{2.6a}$$

and

$$D_i\big|_\xi = \varepsilon_i \Delta t J^{-1}(\nabla\Delta)_\xi\, J, \qquad D_i\big|_\eta = \varepsilon_i \Delta t J^{-1}(\nabla\Delta)_\eta\, J \tag{2.6b}$$

∇ and Δ are simple backward and forward differences. The strength of these terms is controlled via user determined parameters ε_e and ε_i; the effect of these on solution accuracy is one aspect to be studied below. This procedure can be termed "constant-coefficient smoothing," because the ε_e, for example, is isotropic in that it takes the same value for each coordinate direction. An alternative is described below, in section 2.2.2.1. This description of the algorithm is sufficiently complete for the purposes of this work, equations (2.5) and (2.6) being made use of below. Other details may be found in ref.51.

2.2.2 Explicit Algorithm

The second algorithm used in this work, which is the one utilised for all transonic flow cases, is a cell-centred, finite-volume algorithm using Runge-Kutta time-stepping, with acceleration to the steady-state solution by multigrid and residual averaging. The algorithm is due to Martinelli[55], and development and performance of the algorithm are well documented[56,57]. Finite-volume schemes have become very popular in the solution of steady flow problems using both Euler[58,59] and Navier-Stokes[37,55] equations. A cell-centred scheme is one in which the flow variables are associated with the centre of a computational cell, as opposed to a cell-vertex scheme, in which the flow variables are stored at cell vertices. Much development work has been put into both types of finite-volume scheme and their accuracy must be of major concern. The algorithm is formulated in a different manner to the implicit one, but the governing equations are equivalent to the differential form (2.3). The Navier-Stokes equations, in two dimensions, are written in integral form as:

$$\frac{d}{dt}\int_{\Omega}\rho d\Omega + \int_{\partial\Omega}\rho u\cdot n d(\partial\Omega) = 0$$

$$\frac{d}{dt}\int_{\Omega}\rho u d\Omega + \int_{\partial\Omega}\rho uu\cdot n d(\partial\Omega) = -\int_{\partial\Omega} p n d(\partial\Omega) + \int_{\partial\Omega}\sigma\cdot n d(\partial\Omega) \qquad (2.7)$$

$$\frac{d}{dt}\int_{\Omega}\rho E d\Omega + \int_{\partial\Omega}\rho E u\cdot n\, d(\partial\Omega) = -\int_{\partial\Omega} p u\cdot n d(\partial\Omega) + \int_{\partial\Omega} u\sigma\cdot n d(\partial\Omega) - \int_{\partial\Omega} q\cdot n d(\partial\Omega)$$

where σ is the viscous stress tensor and q is the heat flux vector. ρ is the density, u is the velocity vector, and E is total energy (as above). The domain Ω is a computational cell, fixed with respect to a Cartesian frame of reference and bounded by the surface $\partial\Omega$.

The surface integrals appearing in (2.7) are discretised by a flux balance over the finite volume Ω, which results in the problem of integrating in time a system of ordinary differential equations of the general form:

$$\frac{d(w\Omega)_{ij}}{dt} + C(w)_{ij} - D(w)_{ij} + A(w)_{ij} = 0 \qquad (2.8)$$

where w is the vector of dependent variables, $C(w)_{ij}$ and $D(w)_{ij}$ represent the contributions from the approximation of the convective and diffusion operators, respectively, at the ij-th location, and $A(w)_{ij}$ represents an artificial smoothing term which must be applied to this algorithm, as to the implicit one, because undamped spatial oscillations can otherwise arise. Ω_{ij} is the area of the control volume surrounding the ij-th point. The time integration is accomplished by making use of explicit multistage Runge-Kutta time-stepping schemes[55]. Equation (2.8) is very similar to (2.5) in which the entire left-hand side implicit operator corresponds to a time step (stabilised by the smoothing operator D_i), and the right hand side corresponds to the remaining terms, $C(w)_{ij} - D(w)_{ij} + A(w)_{ij}$, in (2.8). Specifically, $A(w)_{ij}$ corresponds to term 4 in (2.5), $D(w)_{ij}$ corresponds to term 3 and $C(w)_{ij}$ corresponds to terms 1 and 2 combined. This similarity means that the two algorithms can be treated virtually identically in the remainder of the text, in terms of the methodology used to examine solutions obtained with them.

2.2.2.1 Smoothing for Explicit Algorithm

The smoothing fluxes for the explicit algorithm are used later, in section 4.4, in connection with the development of weighting functions based on numerical diffusion. They are included in the same form for each of the equations:

$$A(w)_{ij} = d_{i+\frac{1}{2},j} - d_{i-\frac{1}{2},j} + d_{i,j+\frac{1}{2}} - d_{i,j-\frac{1}{2}}$$

where each of the d's is a smoothing flux through a cell side as indicated in fig.2.2, and is based on a combination of first and third differences of the flow variables. For the continuity equation, for example, one of the fluxes is:

$$d_{i+\frac{1}{2},j} = S(\varepsilon_2 - \varepsilon_4 \delta_x^2)(\rho_{i+1,j} - \rho_{i,j}) \tag{2.9}$$

where δ_x^2 is the second difference operator, ε_2 and ε_4 are adaptive coefficients (they are dependent on the solution), and S is a scaling factor proportional to an estimate of the maximum local wave speed[55]. The coefficient ε_2 controls shock capturing and, for viscous flows, is given by:

$$\varepsilon_2 \propto \left| \frac{s_{i+1,j} - 2s_{i,j} + s_{i-1,j}}{s_{i+1,j} + 2s_{i,j} + s_{i-1,j}} \right|$$

where s is the entropy. The coefficient ε_4 is related to ε_2, but for subsonic cases where ε_2, can be set to zero, ε_4 is a constant. Thus, for subsonic cases the smoothing in the explicit algorithm is very close to that of the implicit algorithm. The scaling factor S is important because it attempts to mimic the smoothing behaviour exhibited by a Roe-type scheme[23] and produce an anisotropic smoothing, the form of S being different for each coordinate direction. The form used for S is the one preferred by Martinelli for high aspect ratio cells:

$$S_i = \Phi(r)\,\lambda_i$$

$$S_j = \Phi\left(\frac{1}{r}\right)\lambda_j\,,$$

one associated with each coordinate direction i and j, where $r = \lambda_i/\lambda_j$, and $\Phi(r) = 1 + r^{\alpha}$, where $\alpha < 1$.

$$\lambda_i = |uy_\eta - vx_\eta| + a(x_\eta^2 + y_\eta^2)^{1/2}$$

$$\lambda_j = |vx_\xi - uy_\xi| + a(x_\xi^2 + y_\xi^2)^{1/2}$$

where ξ and η are the coordinate directions in the computational plane and a is the speed of sound. Kunz and Lakshminarayana[45] have noted that this scaling is particularly essential for three-dimensional problems, such as rotor calculations, and have produced a form which is slightly improved from that of Martinelli[55] and the three-dimensional extension of this by Radespiel et al[60]. The work of Allmaras[38] suggests strongly that the Martinelli scaling is inadequate, and only a strongly anisotropic smoothing such as that resulting from Roe-type schemes[23] may be acceptable.

2.3 Summary and Comments on the Smoothing Algorithms

Both of these algorithms, expressed in the forms (2.5), (2.8) are very similar and can be regarded as being of directly-added numerical diffusion type. Central differencing is used with the implicit algorithm and the finite-volume formulation of the explicit algorithm can be regarded as being equivalent to a central-difference discretisation on a regular computational grid. The undamped spatial oscillations referred to in the previous section can give rise to "odd-even decoupling"[61] which can hinder or even destroy convergence. Therefore, numerical smoothing is added to both algorithms to overcome this problem. For the implicit method, which is only used for subsonic flow cases, a background, fourth-difference numerical smoothing is added, as described above. The smoothing of (2.6) is referred to as being a "constant-coefficient" type as the controlling parameters are varied on a global, not a local basis. For the explicit algorithm, the smoothing is added in the manner of Jameson et al[58]. This adaptive dissipation scheme includes a second-difference term, proportional to a feature of the flow solution, which allows the oscillation-free computation of flows involving shock waves. There are two user-specified parameters, ε_2, ε_4. For cases not involving shock waves, the second-order smoothing term can be set to zero, and the remaining fourth-order coefficient ε_4 is a constant. Also the adaptive dissipation includes a scaling factor, S, to improve the accuracy on grids containing high aspect ratio cells. In the subsonic cases investigated later, the maximum aspect ratio of the cells in the computational grids employed is generally low (usually considerably less than 100) so that the scaling of Martinelli does not make a great deal of difference†. Thus, for subsonic cases (2.9) is very similar to the form (2.6a).

Although the adaptive dissipation includes the scaling factor introduced by Martinelli which means that it is not strictly a constant-coefficient type, work has already been performed[46] which suggests that it is insufficient to qualify it as a truly anisotropic scheme. During the time span of this thesis, much more work has been done on this point[38,39,45], some of which[39] is detailed in Chapter 3.

The two algorithms represent a large proportion of the RANS codes in current use and, therefore, it is necessary that their performance is examined very critically. Although the preceding remarks have cast doubt on some aspects of their construction they are still very powerful tools by current standards, and it could be viewed as a testament to the improvement of CFD technology over the last two decades that such criticisms are now being made. The following chapter will provide evidence as to how valid these criticisms are.

---

† The scaling was introduced with meshes for turbulent calculations particularly in mind. Some of the laminar flow cases examined later have quite thick viscous regions (≈ 10% chord at the trailing edge) and the meshes used to resolve them typically have the first grid point at ≈ $2.\text{x}10^{-4}$ chords. The resulting maximum cell aspect ratios are no more than 100. Because grid points must be very close to the surface for adequate resolution of turbulent flows, in which the viscous layers are usually very thin (in the cases examined later, the first grid point is at ≈ $4.\text{x}10^{-6}$ chords), maximum cell aspect ratios of 1000 or more can result. These are the kinds of grid for which the Martinelli scaling might be expected to make a substantial impact on solution accuracy.

Page

3. Effects of Numerical Diffusion on Solution Accuracy 27

3.1 Numerical Diffusion Estimation 28

3.1.1 Numerical Diffusion Estimation in Indirectly-Added Diffusion Schemes 29

3.2 Numerical Diffusion Effects - Implicit Algorithm 29

3.2.1 Aspects of Grid Quality 30

3.2.2 Numerical Smoothing Coefficient 34

3.2.3 Grid Density Effects 35

3.3 Numerical Diffusion Effects - Explicit Algorithm 37

3.3.1 Subsonic Flow 37

3.3.2 Transonic Flow 39

3.4 Summary 39

In this chapter, a method for estimating the numerical diffusion is first described and then used in particular test cases, using both the implicit and the explicit algorithms. The chapter is essentially a self-contained investigation of numerical diffusion levels in the various solutions, with the object of finding out whether these constitute a problem, and if so, the nature and magnitude of that problem. However, the NDR parameter which is introduced is also used in the remainder of the text.

3.1 Numerical Diffusion Estimation

The method used to evaluate the numerical diffusion is described particularly for the implicit algorithm. A virtually identical process is carried out for the explicit algorithm. Recall that the form of the implicit algorithm is such that the left-hand side (l.h.s.) of equation (2.5) operates on the change in the solution variables $\Delta\hat{w} \equiv \hat{w}^{n+1} - \hat{w}^{n}$ where n, n+1 denote values at the respective time levels n, n+1. If the solution is assumed to be well converged, then the l.h.s., since it operates on the change in the solution, can be regarded as small, and the algorithm is of the form:

$$0 = -\Delta t\,(\underbrace{\delta_\xi \hat{F}^n}_{(1)} + \underbrace{\delta_\eta \hat{G}^n}_{(2)} - \underbrace{Re^{-1}\,\bar{\delta}_\eta \hat{S}^n}_{(3)} + \underbrace{D_e \hat{w}^n}_{(4)})\,. \qquad (3.1)$$

This balance equation applies at each point in the solution field. Thus, at each point, the solution can be split into a number of component parts which sum to zero (to within the convergence criterion). For the implicit algorithm, the solution is split into four terms. In the case of the equation for the component ρu of $\hat{w}$, the four terms are: one associated with the pressure gradient and one associated with the convection - these are isolated from (1) & (2) in equation (3.1); a term associated with the physical diffusion - term (3) in equation (3.1); and the numerical diffusion term - term (4) in (3.1). These terms are simply taken from the code on the final iteration of the calculation , at the points in the code where they are normally evaluated anyway. Although, in principle, all members of the above equation set should be analysed, for demonstration purposes only the equation for the component, ρu, of $\hat{w}$ is investigated here. In the cases where the equation for the component, ρv, of $\hat{w}$ was examined, similar results were obtained.

3.1.1 Numerical diffusion estimation in indirectly-added diffusion schemes

The sort of process described above can be carried out for indirectly-added numerical diffusion algorithms, for instance by following ref.36, using a Taylor-series analysis to estimate the terms to be investigated. Consideration of the difference between the Taylor-series for central and upwind differencing of an arbitrary variable leads to the isolation of an "out-of-balance residual" term which is the estimate of the numerical diffusion for such schemes. Fuller details are given in Appendix B.

3.2 Numerical Diffusion Effects - Implicit Algorithm

The amount of numerical diffusion is first assessed for a test case of flow around a symmetric aerofoil, the NACA 0012, at zero incidence. The Mach number at infinity is set to 0.2 and the Reynolds number is 10000. This case is chosen because it is one of those shown previously by Steger[51] to substantiate the numerical algorithm. The main feature is the appearance of a large recirculation region around the trailing edge of the aerofoil. No experimental results are available for comparison but this is not important here.

All the grids used are "C"-grids which are generated initially using an algebraic grid generation code written by Rizzi[62]. Various refinements are later made to the initial grid to investigate the effects of certain grid features. The extent of the grid is indicated in fig.3.1. Where actual grids and computed results are shown, the "plotting window" is always $0.75 < x/c < 1.25$, $-0.25 < y/c < 0.25$, where the aerofoil leading and trailing edges are located at $(x/c, y/c) = (0,0)$ and $(1,0)$ respectively (see fig.3.1).

The aim of this section is to investigate the level of numerical diffusion in solutions from the implicit algorithm, and to see how this level varies with changes in certain factors, such as grid quality and user-specified input parameters, on typical meshes. Grid quality is not a particularly well-defined phrase. Here, it is used, in a somewhat restricted sense, to mean geometric aspects of the lines and computational cells of a given grid. These can be such things as the smoothness, or variation in cell size from one to the next, the maximum skewness, or departure of the angle between two intersecting gridlines from some ideal value (90° for a quadrilateral grid), or the maximum cell aspect ratio. These are all things which can be quantified[63]. Grid density is categorised separately in this thesis - a grid of

20x20 squares would have a higher density (x4) than a grid of 10x10 squares but, by the measures suggested above, would have exactly the same quality. The major point to bear in mind about the concept of grid quality is that, whilst geometric measures are useful, a grid cannot be described as being of truly "high quality" in the absence of information about the solution. For example, square grids as mentioned above would have higher geometric quality than a grid which was highly stretched to give very long, thin cells near one boundary. However, the latter would, in general, be very much better for the computation of a viscous flow over a flat plate, because it would allow much greater resolution of the physical properties of the viscous boundary layer. Where some geometric aspects of the grid are varied in the next section, it will be seen that the quality of the grid is clearly linked to the quality of the solution.

3.2.1 Aspects of grid quality

The first grid used is termed Grid A, and is shown in part in fig.3.2. The complete grid consists of 129 grid nodes along "C" grid lines, with 97 on the aerofoil surface, and 41 nodes in the direction normal to the aerofoil. The far-field boundary is located approximately 15 chord lengths from the aerofoil. The first grid line in the normal direction is a distance 0.0002 chords from the aerofoil surface. The cell aspect ratio at the trailing edge is 30 and at the leading edge is 10.

Terms representing convection, pressure gradient, physical diffusion and numerical diffusion are determined from the fully converged solution, in this case from the equation for the component ρu of $\hat{w}$. To represent the balance of these terms, three individual grid lines in the direction (approximately) normal to the aerofoil are selected. The grid lines variously used are at $x/c = 0.0$, 0.9 and 1.0. For convenience these are referred to in the text, and labelled on fig.3.1, as stations 1,2,3 respectively. These locations were chosen to provide data at the leading edge and trailing edge and at a station expected to be relatively near the separation point. Each of the four terms is then plotted at each grid point on this line from the aerofoil surface out to a distance of 8% of the chord length of the aerofoil. 8% is chosen simply because it allows all the important features of the solution to be displayed in the particular cases shown here. These plots show whether or not the solution is adequately converged, and indicate the relative size of each term, demonstrating which physical processes are important locally and confirming that these are being correctly predicted by the solution where the numerical diffusion is small. This situation is shown in fig.3.3 for the grid line from the leading edge to the upstream boundary (station 1). Very close to the aerofoil, the flow can be considered to be a stagnation point flow for which there exists an analytical solution. (See, for example, ref.64.) Using this theory (see Appendix C) gives an estimate for the boundary layer thickness at the leading edge of approximately 0.2% of the chord length and this is indicated on fig.3.3. The analytical theory also allows the calculation of the behaviour of the various physical terms considered. For example, the physical viscous term should decrease from a non-zero value at the surface to a value of nearly zero at the edge of the boundary layer and, further out, to zero. Examination of the numerically predicted physical diffusion term in fig.3.3 indicates that this behaviour does occur until the edge of the theoretical boundary layer is reached. Then the computed term changes sign before eventually falling to zero. Although it is difficult to decide precisely where the computed edge of the boundary layer is, the behaviour of the terms seems consistent with predictions, as far as the theoretical edge of the boundary layer. This, along with the negligible level of numerical diffusion, suggests that this is a good example of a situation where the added numerical diffusion terms have had little contaminating effect on solution accuracy. In fig.3.4 station 2 is used. The aerofoil surface here is very roughly tangential to the free-stream direction and the pressure gradient exhibits the expected constancy across the viscous layer. It is clear that the numerical diffusion is much larger here, and this can be seen just as clearly at station 3 in fig.3.5 although in different regions across the profile. These last two stations are examples of zones where the added numerical smoothing terms have contaminated the solution.

The major feature of the flow is the recirculation region and this may be visualised using particle paths. The approximate extent of the recirculation zone is shown by the closed path lines. The particle paths in fig.3.6 show a well developed region of recirculating flow. The slight constriction of the paths in the trailing edge region seems unphysical and it is not known why this should occur. Station 3 is indicated on this figure, showing the extent of the region for which terms are displayed. For ease of visualisation, and to assist in developing a quantity for use within a grid adaptation scheme, a second method is now used to represent the four terms mentioned above. This method is to evaluate the ratio of the numerical diffusion to the largest of the four terms at every grid point in the field. Thus, the magnitude of this ratio is never greater than one. Contours of this quantity (which will be referred to as the numerical diffusion ratio, NDR) are then shown, indicating where in the field the numerical diffusion term is large. Fig.3.7 shows the areas where the numerical diffusion term is relatively high for this first test case, and it is clear that the validity of the solution is not assured. Several regions of high NDR are visible, some of which have a crucial bearing on the accuracy of the solution and some of which do not. From the trial calculations to follow it becomes clear that the important area in this particular case is the one indicated by the long, thin, solid-shaded region in fig.3.7 (and its continuation downstream), which approximately covers the outer portion of the viscous layer. A discussion on why there should be other regions of high NDR (such as the large "blob" further outboard), and how to differentiate between the important and unimportant ones, is better placed in section 4.3, where possible use of the NDR for grid adaptation is discussed. Very briefly, however, the high NDR areas within regions where physical viscosity is important are a cause for concern. On the other hand, high NDR areas can arise in the inviscid part of the flow, because a small amount of numerical smoothing is needed for stability reasons, as explained in Section 1.2.1, but the prediction of physical processes is generally not contaminated in these areas.

Noting that the constriction of particle paths corresponds closely with areas where the numerical diffusion is relatively high, a second calculation was performed using a Grid B. This grid was generated by the Rizzi method with 161 points in the streamwise direction and 57 in the normal direction, most of the additional points being allocated to the trailing-edge region. As can be seen in fig.3.8, there is indeed a greater density of points in this area, but one side-effect is a marked increase in slope discontinuity of the streamwise grid lines along the x/c = 1.0 line. Fig.3.9 indicates that the numerical diffusion at station 2 has been reduced slightly, in comparison to fig.3.4, with the addition of the extra grid points in the normal direction. The most significant point is that the peak value of numerical diffusion at $y/c \approx 0.03$ has been reduced to the same magnitude as the peak value of physical diffusion. Fig.3.10 tells a somewhat different story, however, for station 3. The numerical diffusion is substantially (85%) reduced near the aerofoil (compare fig.3.5), but at values of y/c of around 0.03 and above, where the discontinuity in slope of the grid lines is at its highest, the numerical diffusion has been increased (50%). This is manifested in a dramatic way in fig.3.11, where the particle paths indicate separate recirculation regions, with forward flow between. The numerical diffusion ratio is too little changed to give any extra information in this case.

The relative size of the numerical diffusion is clearly demonstrated by this example to be affected by grid fineness and grid quality. In this case grid refinement has worsened the grid quality which has led to an accentuation of the pinching of streamlines noticed before, to such an extent that the nature of the solution is completely changed. Two ways in which the quality has been degraded are the increase in slope discontinuity and also an increase in the angle between grid lines and the expected direction of the streamlines near the trailing edge. An obvious question arising from the above evidence is "Can the quality of Grid B be (in any sense) improved to provide the better solution quality expected from the additional grid points?" An answer is obtained by considering the two grid qualities mentioned. The normal spacing of the grid lines downstream of the trailing edge may be adjusted such that the grid lines lie more closely parallel to the expected direction of the streamlines than before. The slope discontinuity may then also be removed by taking each streamwise line of Grid B in turn, retaining the x/c locations of the points, but altering the y/c locations of 5-10 points either side of the trailing edge by use of cubic splines. This gives Grid C, which is shown in fig.3.12. Looking first at the particle paths produced by a calculation on this grid (fig.3.13), the recirculation region is more realistically represented than before and fig.3.14, showing the numerical diffusion ratio contours, suggests that this result is the "best" so far obtained in the sense that the regions of high NDR have a smaller extent than in the previous calculations. Comparing it with fig.3.7, the regions of highest numerical diffusion are much more confined near the edge of the viscous layer and the overall levels are much reduced (the proportion of grid points at which the numerical diffusion is at least 50% of the largest term has been reduced by 66%). The profiles shown in fig.3.15 for station 3 can be seen, by comparison with fig.3.10 and fig.3.5, to demonstrate a return to the higher numerical diffusion of Grid A near the wall, but a much reduced (maximum reduced by 90%) level further out in the region of $y/c = 0.04$, near the edge of the viscous region, which seems to be critical for satisfactory computation of this case. One odd feature can be noted in the profiles in fig.3.16 at station 2. The pressure gradient and physical diffusion exhibit a noticeable "blip" near the wall which was not present before. The cause, if not the explanation, comes from fig.3.17. Here, the distribution of the Jacobian along the grid line is shown. The sudden change in this quantity, which is the reciprocal of grid cell area, occurs at the same point as the blip in the profiles. It arises because of the way the grid was generated, giving a non-monotonic increase in cell area from the surface to the outer boundary for a portion of the grid near the trailing edge - this is another measure of grid quality in the geometric sense.

In this section, evidence has been presented of the influence of geometric grid quality on solution quality. Certain grid qualities have been shown to alter the magnitude of numerical diffusion effects dramatically. Different qualities may well be important in other cases and generation of grids with this in mind, if it is possible, could be fruitful. For example, grids can be generated such that measurable (desirable) properties, such as orthogonality, are maximised, using optimisation techniques[65,66]. However, the optimal grid will be dependent on the flow-field. This can be allowed for, to some extent, by including measures of solution activity in the optimisation procedure[67], or, conceivably by associating a mathematically-derived estimate of the truncation error of a scheme with features of the grid[68]. Lee and Tsuei[68] provide a relationship between the truncation errors from second-order upwind differencing of the convection terms and properties such as grid size, angle between grid lines, and flow angle. It might be possible to use this information specifically to produce an adapted grid (rather than the normal procedure of clustering more points in certain areas and then checking that the truncation errors have been reduced).

3.2.2 Numerical smoothing coefficient

The two calculations performed in this section are to be compared against the final one of the previous section which will be referred to as the standard result. They are both made using the same Grid C. Inspection of the solution algorithm shows that control of the level of numerical diffusion for a given grid can only be exercised via the coefficients ε_e, ε_i. In fact, $\varepsilon_i = 2.5\varepsilon_e$ is used so that there is effectively only one parameter, and, in any case, only ε_e matters in the steady-state solution (see equation (3.1)). The value used in the standard calculation is $\varepsilon_e=1$. This value is successively halved to check that the value used for the standard case is acceptably low. In Case C(i) the value is 1/2. Fig.3.18 shows that the numerical diffusion has been reduced (maximum reduced by 17%) throughout the profile at station 2 (compare fig.3.16) but there is some evidence, in the convection and physical diffusion profiles very near the wall, of spatial oscillations. The recirculation region of fig.3.19 is about 15% longer than that in fig.3.13 and fig.3.20 shows that the areas of large relative numerical diffusion are generally reduced (the number of grid points at which the numerical diffusion is at least 25% of the largest term has been reduced by 13%), although the areas of very high relative numerical diffusion have actually increased (by about 14%). This is particularly noticeable in the areas downstream of the trailing edge.

In Case C(ii) the value of ε_e is halved again to 1/4. Station 1 is shown again in fig.3.21. There is now a distinct quantitative change from all the previous profiles for this station which were very similar to those of fig.3.3. The peak values of the pressure gradient and physical diffusion are much increased and the convection profile now has two local maxima. This appears to be related to a large increase in the peak negative value of the physical diffusion. In previous cases the numerical diffusion was sufficient to stabilise the calculation at this station without changing the solution. As was stated earlier, numerical diffusion often cannot be eliminated completely. Here the amount present has been reduced, via the controlling parameter, to such a low level that spatial oscillations start to appear with the result that the physical diffusion is larger and changes sign and the other terms respond. Fig.3.22 shows spatial oscillations starting to occur throughout the pressure gradient profile at station 2, with particularly large oscillations near the wall. The particle paths of fig.3.23 indicate that the recirculation region is again increased in length (13%) and breadth (3%) from the previous solution. Fig.3.24 demonstrates a corresponding decrease in overall level of numerical diffusion (25%). With ε_e reduced again to 1/8 no converged solution could be obtained. Since the value of ε_e could not be reduced by much more than a factor of two without the appearance of instability, it is clear that the value $\varepsilon_e=1$ used in the standard case is a reasonable one for stability purposes. This value is, therefore, used in the remaining calculations.

It is clear from the above results that with this algorithm on this grid the levels of numerical diffusion present in the solution cannot be reduced below those shown here, with this constant coefficient smoothing algorithm. An algorithm which allows the use of locally variable coefficients may allow the levels to be reduced. In this study, however, the only remaining possibility for improved results is grid refinement.

3.2.3 Grid density effects

An obvious factor affecting the quality of a solution is grid density. The first two results of section 3.2.1 indicate that an increase in grid density does not automatically lead to an improved solution if other aspects of the grid are not considered simultaneously. In order to investigate the grid density aspect, two more grids, one coarser and one finer than the standard grid, are produced. The first, Grid D, is obtained by removing every alternate point in each direction to obtain a relatively coarse grid of 81 x 29 points. In fig.3.25, at station 2, the blip due to the Jacobian distribution is still present with this grid and the numerical diffusion level is generally higher (maximum value 57%) than for the standard case. In fig.3.26 a tiny recirculation region is indicated and it is quite clear from the numerical diffusion ratios of fig.3.27 that the levels of this quantity are very high almost everywhere.

The next grid, Grid E, is produced by a refinement of Grid C in the normal direction. An extra grid point is placed midway between each existing one as far as the outer edge of the region of highest numerical diffusion ratio, corresponding approximately with the edge of the boundary layer. This results in a grid of 161 x 83 points†. Fig.3.28 shows that the quantities at station 1 are similar to those for the standard case but there is some evidence of spatial oscillations. The clearest example of this can be seen at station 2 in fig.3.29. It is now obvious that the level of numerical diffusion has become too low because the coefficient has remained constant at ε_e while the contribution from the cell area (J^{-1} in equation (2.6a)) has diminished. This effect is analogous to reducing the value of ε_e on a given grid (constant J^{-1}), as was done in the previous section, until the level of numerical diffusion goes below that required for stability. Note that the "blip" near the wall is still visible and is, not surprisingly, more pronounced since refinement in the normal direction accentuates the sudden change in the Jacobian distribution. The particle paths of fig.3.30 show a recirculation region which is slightly thinner (10%) than that of the standard case upstream of the trailing edge but longer (15%) downstream. The evidence of the numerical diffusion ratios in fig.3.31 suggests that, of those presented, this result is the best by that criterion, but the presence of oscillations in the solution should be taken as a warning against absolute reliance on this. Certainly, the evidence indicates that the form of smoothing used here does not allow sufficient grid refinement to produce an acceptable level of accuracy without stability being compromised.

Refinement of the grid in the streamwise direction was also tried. Intermediate points were placed midway between those existing in Grid C for the region shown in the results, giving a grid of 227 x 57 points. No significant benefits were gained by this refinement, indicating that the streamwise resolution is adequate in this case.

---

† The most appropriate way to carry out a grid refinement exercise is to produce each grid by removing points from one extremely fine grid so as to retain similar grid quality in terms of expansion ratios, etc. Examination of all the results suggests that the oscillations experienced on Grid E are a more severe form of those seen in other results, rather than a consequence of inappropriate refinement.

Some encouragement can be drawn from the grid refinement exercise, namely that the pattern of the numerical diffusion ratio contours suggested that refinement in the direction normal to the aerofoil was effective. The impact of refinement on the numerical stability with the current algorithm made a clear interpretation difficult.

3.3 Numerical Diffusion Effects - Explicit Algorithm

It is important to establish that the results of the preceding section apply to other algorithms. Here the explicit algorithm is used to investigate whether this is the case. The grids are generated in the same way as for the implicit algorithm - indeed two of them, Grids B and C, have been used with that algorithm. The plotting window for these results is not necessarily the same as the one used before.

3.3.1 Subsonic Flow

In general, different cases are examined with this algorithm from those for which the implicit one was used. The first of the new cases introduces the effects of compressibility, and the second new case involves a shock wave, as well as a turbulent, rather than a laminar, boundary layer. However, the previous case ($M_\infty = 0.2$, $Re = 10^4$) was calculated with the explicit algorithm on Grid C, to see how closely the results resemble those obtained with the implicit algorithm. Just two figures are shown to demonstrate the similarity of the results. Figs.3.32 and 3.33 show stations 2 and 3 respectively (and should be compared with figs.3.16 and 3.15 respectively). It is quite clear that, apart from the scales differing by a constant scaling factor, the results are very similar. The only noteworthy difference is that the "blip" in fig.3.32 is even more pronounced than in fig.3.16. The correspondence of the results is entirely to be expected because, for subsonic cases, the second-difference component of the smoothing in the explicit algorithm (see Section 2.2.2.1) is not used. Also, the scaling introduced by Martinelli[55] for high aspect ratio grids (~1000) would not be expected to make much difference for this grid, in which the maximum aspect ratio is only about 30. Therefore, in this case the two smoothing algorithms are virtually identical.

Now a case is examined which is similar to the previous one, except that the higher Mach number introduces significant compressibility effects. This is one of the cases from a workshop on laminar flows[69] which has been calculated by the originator of the explicit algorithm[55]. The aerofoil is a NACA 0012, with $M_\infty = 0.85$ and Re = 2000. No shock wave is present. There is a recirculating flow region at the trailing edge, as in the previous case, but this time it is even larger. The first grid used is generated by a modified Rizzi method. There are 128x32 cells with 81 points on the aerofoil. The far-field is at 18 chords and the first grid line is at 0.002 chords from the aerofoil. The trailing-edge region of the grid is shown in fig.3.34. The plotting window for this flow case is $0.5 < x/c < 1.5$, $-0.5 < y/c < 0.5$. In this case, for all the calculations, $\varepsilon_2 = 0.$ and $\varepsilon_4 = 0.2$. The value of ε_4 was the lowest that could be used, without convergence problems being experienced. Fig.3.35 shows, for station 2, that similar behaviour to that of the previous calculations is predicted. The recirculation region (fig.3.36) shows the slight pinching of streamlines in the trailing edge region seen before. It should be noted that the predicted length of the recirculation region is around twice that of the previous case, because of the different flow conditions. The NDR distribution (fig.3.37) again shows similar behaviour, with the highest values of this quantity being concentrated in the outer part of the viscous layer rather than the inner recirculating flow region.

Two more calculations are included with the explicit algorithm. The grids B and C were used, still for the $M_\infty = 0.85$ case flow case. Figs.3.38 and 3.39 show the respective computed recirculation regions. The results are qualitatively similar to those obtained for the implicit code, in that separate recirculation regions are predicted on the poor grid as before, with the results being much improved by use of the better grid. The line diagrams for stations 2 and 3 for the grids B and C are shown in the figs.3.40 - 3.43. No comments are really necessary, save that the general behaviour noted for the implicit case is present here also. Given the similarity of the two smoothing algorithms for low aspect ratio grids, this is not a great surprise. It is worth noting that the behaviour is now seen for a higher Mach number case in which compressibility effects are not negligible, as well as for the earlier low Mach number case. It is clear that the implications of the earlier results remain true for this case, with a different solution method and different flow conditions. This widely used algorithm again produces levels of numerical diffusion which are unacceptably high on the type of grids which are affordable in realistic design calculations and cannot be reduced to tolerable levels by altering parameters in the existing numerical smoothing method. This does not imply that the solution could not be improved by use of much finer grids (globally) than would normally be affordable, or by grids refined locally by adaptation, so long as stability problems are not encountered.

3.3.2 Transonic flow

Next a transonic flow case is examined. The aerofoil is still the NACA 0012. The Mach number at infinity is set to 0.7, the Reynolds number is 9×10^6, and the incidence is 3°. This time a finer grid (321x65 points) is used, and the values of the smoothing parameters are $\varepsilon_2 = 0.25$, $\varepsilon_4 = 1.0$. The flow-field is indicated by fig.3.44, which shows the Mach number contours in the vicinity of the aerofoil. The shock wave can be seen at approximately 30% chord. If the profiles are examined in the same manner as for previous cases some extremely surprising and disturbing behaviour is evident. Fig.3.45 shows profiles at the grid line closest to x/c = 0.25. (Stations other than those used previously are chosen since the flow field is completely different). This grid line is ≈ 5-10% chord upstream of the shock wave on the upper surface. In fig.3.45, values are plotted right out to one chord length from the aerofoil. Throughout the profile, except very close to the aerofoil (which is difficult to visualise on this scale), the physical diffusion is very small, the convection and pressure gradient terms balance each other as they should, and the numerical diffusion is acceptably small. Nearer the aerofoil, up to about 0.15 chords from the surface, the numerical diffusion is clearly non-zero and much larger than the physical diffusion, but it is still much smaller than either the convection or pressure gradient contributions. In fig.3.46, values are only plotted to ≈ 1% chord away from the surface. Wild spatial oscillations are present in the numerical and physical diffusion profiles very close to the surface. This is only reflected to a small extent in the pressure gradient profile. It is, perhaps, difficult to imagine that a solution may be achieved with such poor behaviour present. Similar levels of numerical diffusion for this algorithm have been reported elsewhere[38] and suggestions of the same oscillations are present in the published paper. This evidence strongly suggests that the smoothing algorithm is inadequate for this case, despite the Martinelli scaling which is designed for the higher aspect ratio (~1000) cells of this type of grid.

3.4 Summary

The major conclusion from the illustrative test case calculations discussed here is (as suggested in other work[38,45,46]) that neither smoothing algorithm is adequate in its present form. For the subsonic flow cases, where the grids are not too highly stretched, the algorithms are virtually the same and the results tend to corroborate one another. In the transonic case, where the Martinelli scaling might be expected to make a difference, it is clear that there are still problems. Indeed, it is possible that the numerical smoothing method cannot be satisfactorily rescued by any type of ad hoc fix which does not attempt to introduce some anisotropy into the smoothing (see Section 2.3). Nevertheless, the more exhaustive results of the implicit method suggest that the smoothing algorithm does behave in a predictable manner under grid refinement, as long as stability problems are not encountered. It has already been stated that the global grid refinement is not an option for routine use, for reasons of economy, but local refinement through adaptivity is. This is, in fact, the only possibility for automatically improving the solution accuracy which does not involve altering the algorithm. Since the grid refinement exercises already carried out have shown that the solution can be improved, there is sufficient encouragement to investigate the possibility of improving solution accuracy by grid adaptation, which is the task undertaken in the remainder of this text.

Page

4. Forms of Numerical Diffusion Estimates 41

4.1 Measures for Guiding Grid Adaptation 42

4.2 Grid Adaptation Methods to be used 43

4.3 Modification of Existing Measure (NDR) 47

4.3.1 Cut-off NDR Approach 49

4.3.2 Filtering Approach 50

4.4 Directionality 53

4.5 Summary 54

There is much evidence in the previous chapter that the amount of numerical diffusion present in RANS solutions, on grids of a density affordable for routine purposes, is far too high. Some of the evidence points to the conclusion that the amount of numerical diffusion can only be reduced to genuinely low levels on this density of grid by making substantial changes to the numerical smoothing algorithm. However, there is evidence that levels can be reduced to acceptable proportions by appropriate local grid refinement, and the algorithms should provide a useful test-bed for the investigation of the use of numerical diffusion as input for an automatic grid adaptation method.

4.1 Measures for Guiding Grid Adaptation

An automatic grid adaptation method can be one of two types, node enrichment[40,41] or node movement[42,43], or conceivably a mixture of the two. Both types work by placing extra grid points in regions where they are deemed to be needed, which is more efficient than a simple global refinement since the computational overhead of the adaptation algorithm is almost certainly less than the extra expense and memory requirements of increasing the total number of grid points unnecessarily. Where it is possible (in special cases) to estimate directly the truncation error in the solution (see appendix A of ref.43 and the discussion in Section 1.2.1), adjusting the position of grid nodes so as to try and minimise the maximum value of the error can be shown to greatly improve the accuracy of the results. In general, as stated in Section 1.2.1, it is not possible to estimate the solution truncation error directly, but it is usually possible to estimate it indirectly by finding the truncation error of the numerical discretisation method - the equation truncation error. This is, in effect, what has been done in the early part of this thesis, and in much other work. The aim of the process is to, in some sense, distribute the error in the solution equally amongst the grid nodes. Intuitively, this seems to be the best course of action, and this "equidistribution principle" has been used extensively with a great deal of success. One might ask whether there could be situations in which an error generated in one part of the field (perhaps at a shock wave) would have such an overriding impact on the global solution that a different strategy should be developed. However, in the absence of an obvious alternative, the equidistribution principle is retained here.

In one dimension it is easy to set up an equidistribution scheme, based on the idea of associating a "weighting function" with an interval and altering the size of each interval so as to satisfy the relation

$$W \, \Delta x = \text{constant} \tag{4.1}$$

where W is the weight associated with the interval and Δx is the length of the interval. The situation is shown in fig.4.1. A weighting function should be able to differentiate clearly between areas which are deemed important and those which are not. It should be well behaved in the sense that a reasonably smooth grid node distribution should be produced - numerical second derivatives, for example, can cause problems in this regard. Finally, the weighting function should behave in some predictable manner as the iterative process of solution → adaptation → solution, etc., proceeds, so that it is possible to judge when the optimum grid node arrangement has been achieved.

Generally, the weighting function is a "solution-activity" measure, i.e. it is based on such quantities as first derivative of Mach number, or pressure. A weighting function is associated with a particular coordinate direction. For example, if Mach number gradients are used in a curvilinear grid system (fig.4.2), there would be a function $\partial M/\partial \xi$ associated with the grid lines in the ξ-direction, and a function $\partial M/\partial \eta$ associated with the η-direction lines.

In this and the final chapter, various forms of weighting function are investigated which are based on the solution accuracy measures introduced in the previous chapter. The issues relating to their use are discussed and the results are compared with those achieved with the more usual solution-activity measures.

4.2 Grid Adaptation Methods to be used

In this work, use is made of two different grid adaptation methods, both due to Catherall[12,43]. Both are of the node movement type, in which a fixed number of points are used. The first method[43] is based on the "spring analogy" model for node movement in which the grid is considered to be analogous to a system of weights (the grid nodes) interconnected by springs with associated tensions. In this method, the tension associated with a particular spring (cell side) is found from the weighting function.

The idea of the spring analogy is introduced when considering a two- (or three-) dimensional situation. If the equidistribution principle (4.1) were to be implemented along each grid line in turn, highly skewed grids could result (ie. grid quality would suffer). This is because the i-th point of grid line j needs to be connected to the i-th point of line j+1, etc., in a structured grid environment and if (4.1) was applied completely independently along each line there would be no relationship between the i-th point on each line. This is shown in fig.4.3. The spring analogy is therefore introduced with the idea of providing a torsion spring between the original position of a grid point and its proposed new position, so as to restrain the point from moving too far (see fig.4.3). The tension t_i is related to the angle θ shown, so that highly skewed grids are legislated against.

This spring analogy method, although potentially very simple, actually leads to a fairly complicated algorithm because a number of safeguards need to be built in to avoid such problems as grid crossover; a number of other limitations are listed by Catherall[43]. As a result of this work, Catherall moved on to the investigation of a new method[12], based on elliptic grid generation. In view of this, the spring analogy method was used by the present author (see section 4.3.2 and ref.39) as a first step, partly because the more advanced method was unavailable in the early stages of the work and partly because the spring analogy method is quicker and cheaper.

The method used in later work (chapter 5, ref.70) relies on the observation that the elliptic grid generation equations can be written in a form similar to the equidistribution equation, as explained below. For a surface conforming grid ξ, η, the grid is produced by numerically solving Poisson equations:

$$\frac{\partial^2\xi}{\partial x^2} + \frac{\partial^2\xi}{\partial y^2} = \bar{P}; \qquad \frac{\partial^2\eta}{\partial x^2} + \frac{\partial^2\eta}{\partial y^2} = \bar{Q} \tag{4.2}$$

where x,y are cartesian coordinates and $\bar{P},\bar{Q}$ are functions to control grid stretchings. One desirable feature of this method is that if the equations are solved with $\bar{P},\bar{Q} = 0$, then smoothness and orthogonality are maximised[71], and the extremum principle applies which means that grid crossover is prohibited. With non-zero control functions these remarks are no longer formally applicable but in practice there is usually found to be a strong bias towards them[12]. Equations (4.2) are not in a suitable form for numerical solution and are normally inverted so that x and y become the dependent variables:

$$g_{22}\left(\frac{\partial^2 x}{\partial\xi^2} + \frac{\partial x}{\partial\xi}P\right) + g_{11}\left(\frac{\partial^2 x}{\partial\eta^2} + \frac{\partial x}{\partial\eta}Q\right) - 2g_{12}\frac{\partial^2 x}{\partial\xi\partial\eta} = 0 \tag{4.3}$$

$$g_{22}\left(\frac{\partial^2 y}{\partial\xi^2} + \frac{\partial y}{\partial\xi}P\right) + g_{11}\left(\frac{\partial^2 y}{\partial\eta^2} + \frac{\partial y}{\partial\eta}Q\right) - 2g_{12}\frac{\partial^2 y}{\partial\xi\partial\eta} = 0$$

The metrics of the transformation from cartesian to curvilinear space are:

$$g_{11} = \left(\frac{\partial x}{\partial\xi}\right)^2 + \left(\frac{\partial y}{\partial\xi}\right)^2$$

$$g_{22} = \left(\frac{\partial x}{\partial\eta}\right)^2 + \left(\frac{\partial y}{\partial\eta}\right)^2$$

$$g_{12} = \frac{\partial x\partial x}{\partial\xi\partial\eta} + \frac{\partial y\partial y}{\partial\xi\partial\eta}$$

$$J = \frac{\partial x\partial y}{\partial\xi\partial\eta} - \frac{\partial x\partial y}{\partial\eta\partial\xi}$$

and P,Q are related to $\overline{P},\overline{Q}$ by

$$P = \frac{g_{22}}{J}\overline{P}$$

$$Q = \frac{g_{11}}{J}\overline{Q}$$

The one-dimensional form of (4.3) is:

$$x'' + Px' = 0 \tag{4.4}$$

where $x' = {}^{\partial x}/_{\partial\xi}$, etc. The equidistribution equation may be written as

$$Wx' = \text{constant}$$

Differentiating this gives

$$Wx'' + W'x' = 0$$

which is identical to (4.4) if P is W'/W. Using this observation, the equations (4.3) are made into an adaptive grid scheme by putting

$$P = (\partial W_i/\partial\xi)/W_i; \quad Q = (\partial W_j/\partial\eta)/W_j$$

where W_i and W_j are weighting functions which measure solution activity along η=constant grid lines, and ξ=constant grid lines respectively. Catherall[12] found that this method has some very desirable features, but concluded that the adaptivity (4.4) and smoothing (4.2 with $\bar{P}$, $\bar{Q} = 0$) elements need to be combined in a different way before satisfactory grids can be obtained. He, therefore, proposed[12] the new LPE method. In this method a combination of Laplace, Poisson and equidistribution equations are solved to obtain a new grid.

The equations used are as follows, where r represents x or y:

Inverted Laplace equations are obtained by setting $P = Q = 0$:

$$g_{22}\frac{\partial^2 r}{\partial\xi^2} + g_{11}\frac{\partial^2 r}{\partial\eta^2} - 2g_{12}\frac{\partial^2 r}{\partial\xi\partial\eta} = 0 \qquad \text{(A)}$$

Inverted Poisson equations in the same form are:

$$g_{22}\left(\frac{\partial^2 r}{\partial\xi^2} + \frac{\partial r}{\partial\xi}P\right) + g_{11}\left(\frac{\partial^2 r}{\partial\eta^2} + \frac{\partial r}{\partial\eta}Q\right) - 2g_{12}\frac{\partial^2 r}{\partial\xi\partial\eta} = 0 \qquad \text{(B)}$$

The equidistribution equation may be written in the form

$$W_i\frac{\partial S_i}{\partial\xi} = \text{constant}; \qquad W_j\frac{\partial S_j}{\partial\eta} = \text{constant}$$

where S_i and S_j are the arc lengths measured along i-varying and j-varying grid lines respectively. These two equations are differentiated along their respective grid lines, making use of the identity $dS^2 = dx^2 + dy^2$, to give

$$\frac{1}{g_{11}}\left(\frac{\partial x\partial^2 x}{\partial\xi\partial\xi^2} + \frac{\partial y\partial^2 y}{\partial\xi\partial\xi^2}\right) + \frac{1}{W_i}\frac{\partial W_i}{\partial\xi} = 0 \qquad \text{(C1)}$$

$$\frac{1}{g_{22}}\left(\frac{\partial x\partial^2 x}{\partial\eta\partial\eta^2} + \frac{\partial y\partial^2 y}{\partial\eta\partial\eta^2}\right) + \frac{1}{W_j}\frac{\partial W_j}{\partial\eta} = 0 \qquad \text{(C2)}$$

The LPE method involves solving a combination of these equations:

$$\lambda_L(A) + \lambda_P(B) + \lambda_E\left[g_{22}\frac{\partial r}{\partial\xi}(C1) + g_{11}\frac{\partial r}{\partial\eta}(C2)\right] = 0 \qquad \text{(D)}$$

where λ_L, λ_P and λ_E are user-specified weights. These weights can be varied according to circumstances and experience. The Poisson component is used to include some weighting on the original grid. When the initial grid points are input to the code, the P and Q functions corresponding to this particular configuration are calculated (or if the grid was formed by an elliptic generator they will be known). If the P and Q functions are fed back into equation (D) with the weights $\lambda_P = 1$, $\lambda_L = \lambda_E = 0$, then the initial grid will be recovered. Thus, the final grid can be biased more or less towards the initial grid, by varying λ_L, λ_P, λ_E, depending on how desirable this is felt to be.

4.3 Modification of Existing Measure (NDR)

The NDR, introduced in the previous chapter, has been used as the basis of a weighting function in this work. It is not suitable as a weighting function itself, as it contains too little (or too much) information. Specifically, it does not differentiate between areas where the numerical diffusion is large compared to the physical terms and this is of concern, and areas where it is large in relative terms but this is not of concern (recall that this was mentioned in section 3.2.1).

To understand this statement, consider the solution at each point as a balance of four terms - the physical and numerical diffusion terms, and the convection and pressure gradient terms, as has been done in Chapter 3. In the region near a solid surface, for instance, where the physical diffusion is appreciable, it is desirable to have the numerical diffusion term as small as possible, and a value of the NDR of around one would be of great concern. However, in a far-field region, for example, where the physical diffusion is negligible, it is necessary that the numerical diffusion term should balance the Euler term (convection and pressure gradient together) so as to avoid the numerical problem of "odd-even decoupling". Thus a value for the NDR of one would be possible in far-field regions, even though there is only cause for concern (in the sense of the prediction of the local physical processes being unacceptably degraded) if the actual magnitude of the numerical diffusion term itself (rather than the magnitude of the NDR) is much higher than it needs to be from numerical stability considerations.

In table 4.1, the values of the four terms are listed from the aerofoil surface to the far-field boundary for station 3, Case A of chapter 3, along with the NDR value and a parameter termed NDI (to be defined in section 4.3.2). The important information is extracted and presented graphically also. In fig.4.4 the physical diffusion is divided by the larger of the convection and pressure gradient terms at each point and plotted from surface to far-field. A logarithmic scale has had to be used for distance from the surface to show the information clearly. It seems clear that the edge of the viscous region should be considered to be at approximately 10% chord ($\log_{10}(y/c) = -1.$) from the surface. The streamlines of fig.3.6 tend to support this conclusion. Now the NDR distribution along this particular grid line is shown in fig.4.5. It can be seen that the NDR can take large values in the far-field, even though the magnitude of numerical diffusion is much smaller than in the region close to the surface (see table 4.1). It has not been possible to pinpoint the reason why these large values arise, but they could be due to inadequate convergence or faulty boundary condition implementation, or they could just be necessary for stability. Also, it may be noticed that regions which are similarly difficult to explain are present closer to the aerofoil. One of these is the NDR = 1 peak nearest the $\log_{10}(y/c) = 0$ axis on the negative side (indicated by an arrow), which is at ≈ 15% chord from the surface and corresponds to the outer "blob" mentioned in section 3.2.1 with regard to fig.3.7. Again, this may indicate inadequate convergence, or simply that the algorithm provides too much smoothing in this region. Due to the expense of running this code, these solutions have not been continued further, but close examination of the individual profiles, along with other evidence such as convergence histories, strongly suggests that convergence is acceptable. The obvious conclusion, therefore, is that the smoothing algorithm does not perform particularly well in the inviscid part of this particular flow case.

In table 4.2, similar information is presented for the $x/c \approx 0.25$ grid line considered in section 3.3.2 for the transonic, 3° incidence, case. Fig.4.6 shows that the viscous region for this case, and at this particular station, is at most ≈ 0.3% chord thick, but the NDR (fig.4.7) is again appreciable further out. Nevertheless, this particular distribution suggests that the NDR might be acceptable as a weighting function since it does not take high values throughout most of the outer region (apart from the non-negligible value of 0.06 near the far-field boundary which could, perhaps, be overlooked). However, the information contained in table 4.3 suggests a different conclusion. This station is approximately 4 chord lengths downstream of the trailing edge. The edge of the viscous shear layer (fig.4.8) is no more than 10% chord from the surface but there are peaks of high NDR out to at least 1 chord from the surface (fig.4.9), and the value near the far-field also rises sharply to 0.26. This particular station is much more reminiscent of that shown for the previous case (figs.4.4 and 4.5) and again shows that the NDR highlights regions well away from the viscous part of the flow.

In view of the preceding remarks on far-field NDR values, it is clear that the information contained in the NDR distribution needs to be modified in some way to provide a useful guide for grid adaptation.

4.3.1 Cut-off NDR Approach

One method, which was investigated in some detail but never actually used within a grid adaptation method, is to divide the NDR into two by identifying a boundary between inner viscous and outer inviscid regions. The two regions could then be treated separately. In the inner viscous region the NDR could be used in its unadulterated form as this would then be a situation in which a high value for the NDR indicated where the numerical diffusion was high relative to the physical terms, and the numerical diffusion would almost certainly be contaminating the prediction of physical diffusive processes. In the outer inviscid region some other kind of procedure would still need to be implemented, however. The outer region could not simply be ignored, in general (by resetting the NDR to some suitably low value, for example), because important, non-diffusive, physical processes such as shock waves could be present and it would be essential to be able to separate these areas from those, described above, where an NDR of one existed simply because of the odd-even decoupling phenomenon. Work would also need to be done to make sure that the inner viscous region and the outer inviscid region (appropriately treated) were put "on the same footing", so that movement of grid nodes in the two regions was of a similar magnitude in comparable situations, particular at the boundary between the two.

Many of these issues were not, in the end, addressed in this work, because it proved to be very difficult, in practice, to provide a sufficiently reliable method for positioning the boundary between the two regions. Since the aim of the work was to investigate the possibility of using a solution accuracy rather than a solution activity measure for grid adaptation it seemed preferable to try and base the dividing line on the behaviour of the numerical diffusion term. The largest part of the investigation concentrated on one particular method which was:

(i) isolate, in turn, each grid line in the direction normal to the aerofoil, (ii) traverse each line from aerofoil surface to far-field boundary, noting the maximum value of a chosen quantity such as numerical diffusion, (iii) find the point at which the magnitude of the quantity drops below a certain percentage of its maximum value along that line, (iv) join these points together to form a boundary between inner and outer regions.

This method also proved not to be particularly successful, even when quantities other than the numerical diffusion were tried. The percentage to be used for the cut-off was arbitrarily chosen and the value giving the boundary at a sensible distance changed from case to case. The location of the boundary was also quite sensitive to the chosen percentage. In addition, the shape of the boundary was not, in general, well behaved. A typical result is shown in fig.4.10. The boundary is for Case A from Chapter 3 with a 10% cutoff.

It may have proved possible to get further by using some quantity such as the displacement surface to define the boundary, as this should be well behaved, even if the solution is not particularly accurate. The earlier objections regarding the outer region would still apply, of course. In addition, a yet further, though in this context less important objection, regards any extension of the work to more complicated geometries, such as multi-element aerofoils. In such cases, certain grid lines extend from one solid surface to another rather than from a solid surface to a far-field boundary, thus requiring another revision of the treatment to cope with multiple cut-off surfaces.

4.3.2 Filtering Approach

In view of the comments of the last section, a "cut-off" approach to the problem cannot be completely dismissed, but it is also clear that a method which can deal with the whole field in one go, if it exists, is distinctly preferable. This would consist of filtering the NDR in some way, to remove unimportant areas whilst retaining the important ones. One method[39] is to filter NDR by multiplying it by the local magnitude of the numerical diffusion term to produce a parameter called here the numerical diffusion indicator (NDI).

Considering, in the first instance, the first laminar flow test case investigated in section 3.2.1, using Grid B, the NDI is shown in fig.4.11. This quantity only takes large values in the region concentrated on in the earlier results, in the outer part of the viscous region. Note that the contour values are different, emphasising that this is a different quantity, and it is clear that the important region has been reduced (compare with fig.4.12, showing the NDR for this case). In fig.4.13, the NDI is plotted solely at station 3. This figure corresponds to fig.4.14, which shows the relative importance of the physical diffusion term throughout the profile, and fig.4.15, which shows the NDR distribution in the profile. The largest peak in

NDI is associated with one of the NDR peaks, as are the other, smaller ones. The largest NDI peak is in the region where the physical nature of the solution is undoubtedly incorrect (fig.3.11), and it is worth noting that the relative importance of the other peaks is diminished in both directions - the NDI process is not simply a scaling of the NDR with distance from the surface.

This NDI distribution was used as input to the spring-analogy-based grid adaptation scheme of Catherall[43]. The implementation of the NDI as a weighting function for this method was not investigated in a rigorous manner as it was always intended (see section 4.2) to use a more flexible grid adaptation scheme[12] in due course. Rather than inputting Mach number and pressure distributions to the code from which a function based on derivatives of these quantities would be evaluated, as in ref.12, the numerical diffusion indicator was input and used unchanged as the "weighting function". Some experimentation with various parameters was required, including an artificial reduction of the effect of the weighting function downstream of the trailing edge, and the method was not very robust; these difficulties reinforce the preference for the alternative adaptation method in subsequent work.

Fig.4.16 shows part of the adapted grid, in which the original streamwise distribution of points has been kept. The grid lines have tended to move away from the aerofoil towards the outer edge of the viscous region. The computed flow quantities which produced fig.3.11 were interpolated to the new grid positions within the adaptation program, and the Navier-Stokes calculation restarted and run to a good level of convergence once more. The resulting particle paths are shown in fig.4.17; clearly the solution is improved in terms of the physical realism of the predicted recirculating flowfield. In fig.4.18 the new distribution of the numerical diffusion indicator is shown and should be compared with fig.4.11. This shows that the numerical diffusion has been reduced in the critical area which is the edge of the recirculation region at the trailing edge station. Fig.4.19, at station 3, shows the presence of spatial oscillations similar to previous results. Fig.4.19 also confirms that the numerical diffusion is indeed reduced at y/c around 0.03 and above (compare with fig.3.10).

In subsequent cases, the solution-accuracy-based measures will be compared with solution-activity distributions obtained by the preferred formula of Catherall[12,43]. This is referred to here as DMDP (from Derivative of Mach number or Derivative of Pressure) and is defined by:

$$W = \max_n(W_n)$$

$$W_n = \left|\frac{dq}{ds}\right| + \beta \left|\frac{d^2q}{ds^2}\right| , q = q_1,...,q_n ,$$

s is distance between points along a given grid line and, in all the cases here, n = 2 with q_1 = Mach number, M, and q_2 = pressure, p. In general, for a viscous flow, the Mach number will be automatically selected in the direction normal to a solid surface, because Mach number varies rapidly across a boundary layer whilst the pressure is approximately constant. The opposite will apply, for example, along the solid surface grid line because pressure varies significantly in this direction whilst the Mach number is zero everywhere on the surface. First, an indication of solution activity is given for the low Mach number laminar flow case, using Grid A, in figs.4.20a & 4.20b. These figures correspond to gradients in the C (body-conforming) grid direction and the body-normal direction respectively. The NDI distribution for the same calculation is shown in fig.4.21. It can be seen that the NDI is relatively high in areas which are also emphasised in one of the solution activity figures, but is more selective in its extent. In the region downstream of the trailing edge, for example, the solution activity measure for the normal direction would imply that extra grid refinement is necessary. The NDI analysis shows that no significant numerical diffusion is present here so grid resolution is probably acceptable.

To explore the generality of the observation that NDI is a more selective measure of the need to refine the grid than is the solution activity measure, a different flow type and RANS code combination was considered. The explicit algorithm was used to calculate the flow at a Mach number of 0.7, an incidence of 5° and a Reynolds number of $9. \times 10^6$, on a grid of 321x65 points. This is a turbulent flow case involving a shock wave strong enough to produce a small separated region at its foot. The resulting NDI distribution is shown in fig.4.22 and the solution activity measure for the C-direction is shown in fig.4.23. Normal-direction solution activity highlights only a very thin boundary layer region around the aerofoil. Both measures take high values in the shock-wave region, but NDI does not highlight the leading-edge region. This again seems to be because there are sufficient points to resolve the leading-edge suction peak relatively well compared to other areas, despite the high flow gradients. In addition, in this case NDI highlights some areas not present in the solution activity measure. For example, two "ribbons" of NDI can just be made out. Table 4.4 shows the quantities along the gridline at 45% chord which is downstream of the shock wave. The two ribbons correspond to local peaks of numerical diffusion at ≈1% and 3% chord from the surface, which are also visible as two small bumps between $\log_{10}(y/c) =$ -1. and -2. in fig.4.24. The related NDR is shown in fig.4.25 and it can be seen that there is again a correspondence between the peaks. These ribbons may, perhaps, be a legacy of the much higher peaks close to the aerofoil. One possibility is that incorrect quantities are being specified at the boundary for the numerical diffusion. (Since this is a fourth-difference, derivatives up to order 3 must be specified in some way). One or two alternatives were tried with no noticeable effect, but a careful study might produce some improvement.

4.4 Directionality

The above results indicate that the NDI gives similar information to that obtained from solution activity, but is more selective. The main disadvantage of the NDI is that it makes no distinction between directions of importance for grid adaptation. It indicates which cells in the field need refinement, but does not give any information about how the refinement is best carried out, unlike the function based on solution activity which makes use of gradient information in each coordinate direction.

Therefore, modified parameters have been constructed, called numerical diffusion functions (NDFξ, NDFη), which reintroduce directional information into the error estimate. The functions are obtained by multiplying NDI by coordinate-direction-dependent factors. For the explicit scheme used here these factors have been taken as the smoothing flux through the cell side in a given direction (e.g $d_{i+\frac{1}{2}j}$ as described in section 2.2.2.1), divided by a relevant distance. This form was chosen because the fluxes include first differences of flow quantities and as such should behave similarly to a solution activity measure, giving high values (and thus indicating a need for refinement) across high flow gradient regions. In the turbulent case, NDFξ (fig.4.26) is similar to NDI, except that the outer part of the shock wave is emphasised more†.

---

† The maximum values in the inner and outer solid-shaded regions of fig.4.26 are of similar size; the maximum values in the inner region of fig.4.22 are greater than the maximum values in the outer region. Unfortunately, picking a scale to show this would result in some of the other information contained in fig.4.22 being lost.

NDFη (not shown) is similar to the solution activity measure for the normal direction, with very high values near the surface. This form of NDF has achieved the objective of introducing directional information into the solution accuracy measure, indicating that refinement is necessary in the normal direction across the boundary layer, and in the streamwise direction in the vicinity of the shock wave.

Another method for introducing directional information into the new weighting function would be to combine it in some way with a typical solution activity measure. This alternative is tried in Chapter 5, although it departs to some extent from the philosophy which has been applied throughout the text, so far, of basing the function entirely on a solution accuracy estimate.

4.5 Summary

In this chapter, weighting functions based on solution accuracy measures have been developed. These have been shown to be justifiable in their construction, and have also been shown to highlight more or less the same information as solution activity measures whilst being more selective. In Section 4.3.2., one measure (NDI) has been used successfully to guide grid adaptation with a corresponding increase in the accuracy of the solution.

It was stated earlier that a weighting function should behave in some predictable manner during a process which may contain a number of cycles of solution calculation followed by adaptation. This issue is addressed in the next chapter, where the NDF formulation is taken as the best available alternative which has emerged from the work to this point. The behaviour of this will be compared with that exhibited by the solution activity approach.

Page

5. Results of Grid Adaptation 55

5.1 Grid Adaptation Examples 56

5.1.1 Example using Solution Activity Measure 58

5.1.2 Example using Solution Accuracy Measure - NDF 60

5.1.3 Use of Combined Solution Activity/Accuracy Measure 62

5.2 Use of Coarser Grids 63

5.2.1 Very Coarse Grids 64

5.2.2 Intermediate Grids 65

5.3 Summary and Implications 66

In this final chapter of results, the use of the NDF weighting function developed in Chapter 4 is investigated with the LPE grid adaptation method[12]. The intention is to ascertain whether or not it can be used successfully to guide grid adaptation in a self-consistent manner, in such a way that the iterative framework of RANS solution → error estimation → adaptation → RANS solution → error estimation → adaptation, etc., converges to a final solution of high accuracy. For the theory of grid adaptation to work usefully in practice, it must be possible to terminate such an iterative process in some logical and consistent manner, without much reliance on the experience of the user. Use of the NDF is also compared with the use of a solution-activity-based measure[12,43] (called DMDP, as introduced in Section 4.3.2) and one particular combination (COMB) of the two is tried.

In section 5.1, these are used with an original grid which is already globally fine. The prime interest in this case is how the adapted grids react to the presence of a shock wave. In section 5.2, the important question of what kind of computational savings can be expected from adaptation is addressed, by considering the solutions on grids of different densities of points, and attempting adaptation exercises on coarser grids. The behaviour of various physical quantities during successive adaptations is also considered.

5.1 Grid Adaptation Examples

If a weighting function is to be used to guide grid adaptation by node movement, as here, the most desirable behaviour is for it to:

(i) take high values in regions of the initial grid where refinement is desirable to increase accuracy,

(ii) as the grid is refined in problem areas, the function values remain approximately constant. This behaviour might occur, for example, at a shock wave. A tendency for the values to increase with refinement is acceptable, although some limit might need to be imposed on the minimum size of grid cell permitted, as is the case with the LPE method.

The DMDP function exhibits this type of behaviour and has been used successfully within the LPE method[12], where the procedure is, effectively, to assign maximum and minimum spacings to each grid line ξ=constant or η=constant, either by choice or by finding the maximum and minimum values on the initial grid, so that no matter how much the weighting function might increase locally, the refinement cannot continue indefinitely. (For computational convenience, the actual procedure is to find the values of the weighting function corresponding to the maximum and minimum grid spacings, and constrain the weight function associated with each grid point to take a value between these two extrema. This has a very similar effect to constraining the grid spacings directly.) A second, alternative, behaviour for a weighting function is to commence as in (i), and, tend to zero in those regions as the mesh density becomes adequate. It has been shown[72] that, whilst the first type of function is undesirable for node enrichment, the second type is ideal for that purpose.

The first type of weighting function behaviour is easily incorporated in the LPE method, in the manner illustrated in fig.5.1. As described in section 4.2, the LPE method uses a series of weights λ_L, λ_P and λ_E, which place proportional emphasis respectively on smoothness and orthogonality, the grid which was input to the adaptation method, and adaptation to the weighting functions. In the normal scheme, using the DMDP measure, the grid which is input is the initial one. At each stage, a new solution is produced and, if the process has not converged, this solution is used, along with the initial grid, to produce a new adapted grid. This procedure will tend to converge because the DMDP measure will tend to give larger values (for shock waves, thin boundary layers) as the grid is refined locally. The second type of behaviour of a weighting function could possibly be accommodated in a node movement scheme with a slight modification. This is also illustrated in fig.5.1. At each stage, the adapted grid from the previous stage is used as the input grid for the present adaptation. This method would tend to converge if the weighting function tended to reduce in magnitude under local grid refinement. These two alternatives are referred to from here onwards as strategy A and strategy B respectively. In the remainder of section 5.1, strategy A is used with each of the weighting functions tried. The first example (Section 5.1.1) shows how the method of ref.12 works, the weighting function used being the one recommended there. This will be used as a benchmark for the remaining two examples.

The flow case examined in this section is the one discussed in section 3.3.2, a turbulent flow at a Mach number of 0.7, an incidence of 3°, and a Reynolds number of 9×10^6, for the NACA0012 aerofoil. The initial grid is shown in fig.5.2. It is a relatively fine grid of 321x65 points which gives $C_L = 0.5001$ and $C_D = 0.0086$, and the Mach number contours shown in fig.5.3, using the explicit algorithm. The broken line in fig.5.3 represents M=1, indicating the calculated position of the shock wave, and the spacing of the contours gives an indication of how well it is captured.

5.1.1 Example Using Solution Activity Measure

The weighting function, based on Mach number and pressure, associated with the initial solution, is shown for the ξ-coordinate direction in fig.5.4. In this example, as in all the others, the weighting functions are displayed by finding the maximum value of the quantity in the field and using this to normalize all the quantities. Thus, the weighting function always ranges from 0 to 1, irrespective of the absolute values calculated by simply evaluating $\partial M/\partial\xi$, etc. This makes it easier to see the contrast between various parts of the field, and is permissible because absolute values are much less important than relative values in this context. Picking the best scale to show all the interesting features is not always easy, but enough comments are included in the text to give sufficient explanation.

Using the equidistribution principle, large values of the weighting function, W, do not necessarily indicate a need for refinement. This is only necessary if W x ds (grid spacing) is locally larger than the average for a given grid line. However, displaying the quantity (W x ds) would not be particularly informative (unless the adaptation process tended to diverge for some reason), because a contour plot of the ideal situation (W x ds = constant everywhere) would be completely featureless. The plots of the function W, which is always the quantity shown here, are useful because they make it possible to see the areas of the solution which are forcing the adaptation. Now, referring to fig.5.4, three areas are noticeable. The weighting function takes significant values in:

(i) the leading-edge region. In fact, the highest value here is close to the largest in the field.

(ii) the upper surface shock-wave region. The largest value here is only about 1% of the maximum, as the shock is "smeared" over quite a large region.

(iii) the trailing-edge region. The maximum value in the field is here. In virtually all calculations performed with the explicit algorithm, a "blip" in the surface pressure coefficient distribution exists. Its legacy is also noticeable in similar distributions along adjacent grid lines, the effect dying away with distance. In fig.5.5, the surface C_p plot for the trailing-edge region is shown. Although the pressure blip does not look particularly large on a plot covering the whole aerofoil, the actual pressure gradients are very large, and often, in fact, the largest in the field. The cause is not known, but the behaviour does not seem to be affected significantly by increases in grid resolution on an affordable scale and so it would seem to be an underlying feature of the solution algorithm.

The LPE method is now used to produce a grid adapted to the initial flowfield, using the solution activity weighting function, shown in fig.5.4 (and the corresponding function for the normal direction which is not shown because values drop so quickly from the largest ones at the surface, due to the thin viscous layer for this high Reynolds number case, that it is difficult to produce any helpful visual display). The parameters which have been used, for each of the adaptation stages of the process, are $\lambda_L = 1.0$, $\lambda_P = 1.0$, $\lambda_E = 1.0$. It is their relative size that matters, not the absolute values. The adapted grid is shown in fig.5.6. In the field, it can be seen that some points have been drawn to the shock wave region (compared to fig.5.2).

Flow variables from the solution on the initial grid are now interpolated onto the adapted grid, and the solution is continued. The resulting Mach number contours are shown in fig.5.7, and it should be clear that the shock wave is better resolved than on the initial grid. This is confirmed by the weighting function in fig.5.8, where the maximum value in the field has increased slightly, and the value in the centre of the shock wave region has increased to over 2% of this maximum. This weighting function is used to guide a second adaptation. The second adapted grid is shown in fig.5.9. The grid lines are closer still in the shock wave region, and the Mach number contours of fig.5.10, resulting from a continuation of the flow solution on the new grid, confirm this (compare with fig.5.7). Once more the weighting function is shown, in fig.5.11. This is the function which would be used as input to a third grid adaptation, but a sequence is already becoming clear. The maximum value in the field has again increased slightly, and the value in the shock wave region has again increased significantly, to over 4% of this maximum.

It seems likely that this behaviour will continue indefinitely, with more points being attracted to the shock wave region, until eventually the minimum cell size limit will halt the process. Deciding on a criterion for terminating this sequence depends on what is considered to be important. The calculated lift and drag coefficients, plus those from the next two cycles, for which other figures are not included, are shown in table 5.1. It would seem that, as far as these quantities are concerned, there is actually very little difference between any of the results (the maximum difference between any two of them is 1% in C_L and 2% in C_D), and the main motivation for continuing to adapt in such a situation would be an interest in the fine detail of the shock-wave/boundary-layer interaction. Given the tiny thickness of a real shock wave, it is not obvious that this could ever be resolved adequately in a calculation of a whole aerofoil.

The results of this section suggest that the benefits of adaptation, when the original mesh is as fine as the one used here, are quite limited. However, it is still interesting to compare the behaviour of the NDF as a weighting function with that of the results obtained so far.

5.1.2 Example using Solution Accuracy Measure - NDF

Therefore, using the initial grid of fig.5.2, the distribution of the parameter NDFξ, to be used as the weighting function, is shown in fig.5.12. There are two obvious differences from the figures seen previously. The first is that much of the area around the leading-edge which is highlighted by the solution activity measure in fig.5.4 is not emphasized, suggesting that, whilst gradients of Mach number and pressure may be high there, the accuracy of the solution in that region is acceptable. However, there are some high values of the NDFξ very near the surface, near the leading-edge. The second difference is that the shock wave region appears different. It is seen as two parts. The values in the inner part are higher than those in the outer. Also very high values still arise in the trailing-edge region, suggesting that the pressure gradients there are associated with an accuracy problem.

This weighting function was used to adapt the initial grid, using the values λ_L = 1.0, λ_P = 1.0, λ_E = 1.0 as before. The resulting grid is shown in fig.5.13, where it can be seen that many more points have been drawn to the shock wave region than with the measure used in the previous section. This is because the function values in the outer part of the shock wave, whilst being smaller than those in the inner part, are still much larger than values elsewhere in the field along the same streamwise grid lines - the equidistribution part of the adaptation is done on a line by line basis. An unfortunate side-effect of this is visible just upstream and downstream of the strongly adapted region. There is some very poor grid quality in the form of highly skewed cells, caused by the fact that along the surface there are three regions in which the NDFξ is relatively high, whereas in the field the only region of high activity is near the shock wave. This tendency for a node movement method to produce highly skewed cells provides a potential argument in favour of unstructured grids, although the "quality" of completely unstructured grids may not be any better. A "semi-unstructured" system of embedded grids[73] may have advantages here. Another noteworthy feature is the presence of two streaks of slightly more concentrated grid nodes emanating from the trailing edge. By comparison with fig.5.2, these can be seen to correspond to a slight fanning out of lines in the initial grid. Presumably, a slight inaccuracy in the solution results from this which is picked up because of the lack of anything else happening along those particular "C" lines. Thus, the NDF seems, if anything, to be too sensitive to inaccuracies in the solution. For the rest of this section, the value of $\lambda_E = 0.5$ is used, to try to lessen the effect.

The adapted grid resulting from using this value of λ_E, with the other values the same as before, is shown in fig.5.14. The strength of the adaptation is suitably diminished, such that grid quality in the susceptible regions is not too bad. As before, the initial solution is interpolated onto this new grid and then continued. The distribution of Mach number contours is shown in fig.5.15 where it can be seen that the shock wave is more sharply defined than before, although the region of highest Mach number (>1.3) has diminished significantly. The solution seems even "better," in the sense of the shock wave definition, than the solution after two adaptations using the activity measure (fig.5.10), as one might expect from looking at the grids involved. The distribution of NDFξ corresponding to this new solution is shown in fig.5.16. The most important difference is that the values in the outer shock wave region are much higher as a percentage of the maximum value in the field, although this value has dropped substantially. The result is that the highest actual value in the shock wave region has approximately trebled since the shock wave is more sharply defined with a resulting need for more smoothing to prevent the solution becoming oscillatory. Again the sequence is continued, using this NDF distribution to produce a second adapted grid, etc. Results are shown for the fourth cycle, which is the last one performed. The fourth adapted grid is shown in fig.5.17 with the associated Mach number contours in fig.5.18 and the weighting function in fig.5.19. The grid has clearly adapted to the shock wave, which is defined more sharply than in any other result so far. It is just possible to discern a slightly patchy nature in the grid (this is much more noticeable in one of the examples in section 5.2). This is a genuine concern as it is certainly possible that a weighting function based on a numerical accuracy measure could adapt to transient numerical waves, which will exist as long as the solution is not completely converged, and this might conceivably slow down the convergence process, or even affect the converged solution. The C_L and C_D values for this series of cycles using the NDF is shown in table 5.2. Once again, all the results are very similar except that there is a tendency for the drag coefficient to rise with adaptation. One possible explanation for this is that the adaptation in the normal direction is inadequate, causing insufficient clustering of grid points near the surface leading, in turn, to a tendency to smear out the boundary layer with a consequent increase in the predicted drag. Partly because of concern over the patchy nature of the adapted grids, and partly because of the concern over adaptation in the normal direction, which will become more evident in section 5.2, another method to address the directionality question, described in section 4.4, is tried here.

5.1.3 Use of Combined Solution Activity/Accuracy Measure

The NDF parameter uses the NDI as a base and numerical diffusion fluxes to give directional information. In this section the NDI is used as a base, with the solution activity measure, as in section 5.1, used for directional information. This departs slightly from the idea of using accuracy measures alone, but could offer the best of both worlds. Just how to combine the two measures is not at all obvious. The method tried here is simply to multiply the values of the two measures at each point. This particular COMB measure requires that, the accuracy and activity measures being multiplied, both must be substantial for the product to be high. The desire to model areas of slow recirculating fluid better might not, therefore, be satisfied, although this has not been investigated here. Alternative forms of a combined measure, based on linear combinations, should be investigated in any future work.

The grid and associated quantities after the second adaptation are shown in figs.5.20-5.22, and the C_L and C_D values for four cycles are shown in table 5.3. Once again, the shock is sharpened from the initial solution and the COMB measure behaves in a similar manner to the NDF, tending to increase its magnitude near the shock wave relative to its value in the rest of the field whilst the maximum value decreases, suggesting that the accuracy measure tends to dominate the activity measure. Nevertheless, the activity measure has ameliorated the tendency of the accuracy measure to produce a wavy or patchy grid, providing evidence that this nature is a numerical, rather than a physical feature.

In all the results presented in this section, the shock wave is captured progressively better with adaptation, but it is difficult to counter the argument that, judged by C_L and C_D values alone, even the worst of the results is, in fact, very good. This question is addressed in the next section. It can, however, be said that all the measures tend to improve the resolution of the shock wave with both accuracy-based measures resolving this feature more strongly than the activity-based one. In addition, the NDF has a tendency to produce an increase in predicted drag, probably due to a smearing out of the boundary layer, and that the COMB measure produces smoother grids than the NDF for this case.

5.2 Use of Coarser Grids

In this section, a slightly different case is examined; all the parameters are the same except that the incidence is raised to 5°. The shock wave is now strong enough to separate the boundary layer at its foot. A different case was chosen because it was found that, with the very coarse grids used in the next section, there was so much numerical diffusion present in the 3° incidence case that the shock wave was not captured in the calculation on the original grid. Since the resulting values for all the weighting functions were so low in the region where the shock wave should have been, compared to values elsewhere in the field, points were not drawn to this region, the resolution of the region was not improved and the shock wave never appeared as a feature of the solution. The same initial 321x65 grid as was used in the previous section is used to give the flow field shown in fig.5.23. A number of grids were produced from this initial one by deleting alternate points in one or both directions from a previous grid in the series. The C_L and C_D values associated with a calculation on each of these grids are shown in table 5.4. Other than that obtained on the extremely coarse grid, none of the results can be said to be very bad in terms of these quantities (the maximum deviation in C_D from the fine grid solution is only 10%). The Mach number contours from the grids designated c53 (161x33 points), c54 (81x33 points) and c55 (81x17 points) are shown in figs.5.24-5.26 respectively. The major tendencies with decreasing number of grid points are for the shock wave to decrease in strength, become less vertical, and move forward. Also, from the evidence of these four plots, the area enclosed by the semi-circular contour below the aerofoil decreases as the number of points in the normal direction decreases (disappearing when only 17 points are used), the detail of the boundary-layer thickening diminishes with a reducing number of nodes, and the separated flow region at the foot of the shock tends to disappear. It is clear that the details of the result on the c55 grid are poorly represented with a misshapen supersonic region and no real hint of boundary-layer thickening at the foot of the M=1 contour. Therefore, the value of the results on adapted grids should not be judged solely by the C_L and C_D values, as correspondence of these to the fine grid results can be achieved with significantly different flow fields.

5.2.1 Very Coarse Grids

In the first instance, adaptation exercises have been carried out using grid c55. Obtaining a well-represented flow field and approximately correct values for lift and drag coefficients with such a coarse grid would be very impressive, if it were possible. Having only 17 points in the normal direction (to cover a distance of approximately 20 chord lengths) makes it particularly unlikely that all the features can be resolved. The initial grid is shown in fig.5.27, underlining just how coarse it is, and the DMDP weighting function is shown in fig.5.28. On this scale, the shock wave is not seen at all, although the values in that region are a little higher than the average in the field. A total of four adaptation cycles were carried out, for which C_L and C_D values are shown in table 5.5. These values bear little relation to those encountered on any of the unadapted grids, the movement of grid points having had much more of an effect on these quantities than the refinement/derefinement (coarsening) of the initial grid. The final Mach number contours and grid are shown in figs.5.29 and 5.30 respectively. The area of supersonic flow has increased, as has the maximum Mach number ahead of the shock. However, the flow field is still a long way from that produced on the fine grid. The adapted grid (fig.5.30) shows that there is little movement of grid nodes toward the shock region.

The use of the NDF was also investigated on this grid. Both strategies A and B for inclusion of previous grid information were tried. The C_L and C_D values for strategy A are shown in table 5.6 and the final grid and Mach number contours in figs.5.31 and 5.32 respectively. The corresponding quantities are shown for strategy B in table 5.7 and figs.5.33 and 5.34. It is particularly clear from the tables and from the grids that the results are not good, but a few general points can be made:

a) The position of the shock wave is further rearward (closer to its position in the fine grid result) in both cases than with the DMDP measure.
b) With strategy A the shock position is similar to that from the fine grid result, although it is at a completely different inclination.
c) Strategy B shows a much greater tendency to cluster points in the shock wave region than do either of the other methods.

d) The contour below the aerofoil is just resolved in both cases.
d) The normal direction is not treated correctly at all. It is particularly evident from fig.5.33 that clustering of grid points near the surface does not occur.

The tabulated values, along with e) above, suggest that the presence of the viscous region is not captured at all correctly. This provides some justification for the examination of something like the COMB measure considered in section 5.1.3.

From the examples of this section, one should not draw final conclusions about the suitability of the various measures in general cases, given the very coarse nature of the grid, but some remarks can be made with certainty. The first is, in fact, not at all surprising. This is that a grid adaptation scheme based on node movement will require a minimum number of points in order to resolve a complex flow field. If too coarse a grid is used, there is no chance of capturing shock waves sharply whilst retaining adequate resolution of leading and trailing edge regions. This is far less likely to be true of a node enrichment method. The NDF measure has been shown to capture behaviour in the normal direction particularly badly when a very coarse grid is used, although the streamwise features are probably captured about as well as is possible with this number of points.

5.2.2 Intermediate Grids

In this final section, the grid c53 (161x33 points) is used. This has sufficient points to resolve the features of the flow field quite well (compare figs.5.23 and 5.24). The DMDP measure is used first, with the C_L and C_D values shown in table 5.8. There are suggestions of an oscillatory nature here. The Mach number contours and grid from the fourth cycle are shown in figs.5.35 and 5.36, with those from the fifth cycle in figs.5.37 and 5.38. In the results of the fourth cycle, the supersonic region is larger with the shock wave further back, and the shock-induced boundary-layer thickening and separation is much reduced. Similar behaviour was noticed by Catherall[43] for a different case, and he concluded that the flow conditions were particularly sensitive to small changes. It is possible that this case is similar, particularly given the strength of the shock wave, but it is also possible that some experimentation with the various weights, λ, could alter the situation. This is suggested by the fact that this oscillatory behaviour is not observed when the NDF measure is used.

Once again, both strategies A and B were used. Tables 5.9 and 5.10 contain the values from A and B respectively, with the associated contours and grids from the final (fourth) cycle shown in figs.5.39-5.42. In both cases, the contours are similar to those from the fifth cycle with the DMDP measure, although the shock wave is captured most sharply in the final result. In all cases, the supersonic region is smaller, and the shock further forward, than in the fine grid results. Both the NDF results and the fifth cycle results with DMDP display the separated flow region at the foot of the shock wave which is present in the fine grid results, but is completely missing in the fourth cycle results with DMDP. The most striking feature of the NDF-produced grids is their poor geometric quality or "patchy" nature, particularly when contrasted with the grids produced using the DMDP measure. However, the use of NDF should not be rejected for this reason alone, especially as the Mach number contours and tabulated information suggest quite acceptable performance, with strategy B in particular giving C_L very close to the fine grid result, and strategy A doing similarly for C_D. In fact, considering all the evidence of overall flow field, shock position and strength, separated flow region and C_L and C_D values, the NDF result with strategy B is the best one obtained with this density of grid. Once again, although it has not been investigated here, it would seem that the use of a COMB-type measure would be worthy of consideration in this case.

5.3 Summary and Implications

In this chapter two weighting functions based on a solution accuracy measure have been compared with the normal solution activity measure for guiding grid adaptation. These two functions appear to work at least as well as, and normally better than, the one based on solution activity. This provides evidence that a weighting function based on numerical diffusion estimates should be used in preference to a solution-activity measure, although it is not clear that the NDF is the ideal form. The conclusions have been drawn, and the work progressed, largely on pragmatic grounds and, although the approach taken here seems to work quite well, no truly rigorous mathematical justification has been advanced. An alternative weighting function, based on a more mathematical approach to accuracy assessment[74] is being investigated elsewhere. From the work done here, it is clear that something like the COMB measure of Section 5.1.3 would be a strong candidate for any future investigation.

On the basis of the results presented in this chapter, there are clearly two boundaries to the grid density which would be appropriate for adaptation via node movement. Grids which are too coarse will never resolve all the features of a complex flow, and adaptation of grids which are very fine will not yield a sufficient improvement in accuracy to justify the extra work involved. Based on the results here, one could tentatively suggest that for single aerofoils, adaptation is likely to be fruitful for grids of a density in the approximate range 120x24 to 240x48 points, obviously dependent to some extent on the type of flow involved. It is not known how the node movement method on this density of grid would compare, economically or in terms of accuracy, with the alternative approach of using an initial coarse grid and a node enrichment method.

Both the functions based on numerical diffusion estimates have shown desirable features and performance in certain cases. They have tended to adapt the grid more quickly and more strongly to shock wave regions when used on a relatively fine initial grid. In addition, points are not drawn to the leading-edge region to the same extent as they are with the activity measure, suggesting that these measures can differentiate between areas where flow gradients are high but the accuracy is acceptable, and areas where accuracy is unacceptable. In the intermediate grid cases, they have also produced grids capable of capturing the separated flow region at the foot of a shock wave, which the activity measure has failed to do consistently. Thus, the results presented here provide considerable evidence of the advantages of an adaptation measure based on numerical accuracy estimates rather than gradients of flow variables, and show that the particular forms tested here display a number of the qualities required of such a measure.

Page

6. Conclusions 68

6.1 Conclusions and Implications of Present Work 69

6.2 Recommendations for Future Work 71

6.1 Conclusions and Implications of Present Work

The early part of this work (Chapter3, ref.39) has concentrated on solution accuracy assessment, showing that this can be done quite simply. Two methods have been used to visualise the results, and it has been shown that they can give detailed information on the behaviour of the solution. The method of examining the distribution of terms along individual grid lines is important, as it is then possible to check that the correct physical processes are (or are not) being predicted. Displaying the NDR term gives immediate warning of areas where the solution accuracy could be unacceptable. This type of detailed accuracy assessment is an extremely important part of algorithm development which has been, until recently, rather neglected. During the time-span of this work, this issue has been taken up and addressed by a number of other authors[37,38].

In this work, two commonly used algorithms have been investigated. The smoothing coefficients of these algorithms have been varied, either directly, or indirectly via grid refinement. The effect of reducing the controlling parameters was eventually noticeable as a failure of the solution to remain stable after a relatively large number of time steps, rather than an abrupt failure at the start of the calculation. The results of Mahajan et al[30] have also shown how this can occur if the explicit smoothing coefficient is reduced to a sufficiently small value.

It is clear, from the work done, that the use of a constant coefficient for the fourth-order smoothing term is inadequate. Reduction of local levels by reduction of the controlling coefficient can mean that the smoothing applied elsewhere in the field is insufficient to maintain a stable solution. With constant coefficient smoothing, reduction via grid refinement can also cause stability problems. If the grid is refined, the resolution of flow gradients increases and the amount of numerical diffusion present is no longer sufficient to prevent spatial oscillations in the solution.

One basic message of this work is that there are more question marks over the accuracy of these popular methods than might have been expected. These algorithms can produce results whose accuracy is unacceptably degraded by the presence of numerical diffusion. As suggested by other work[45,46] the use of variable coefficients for the fourth-order smoothing is essential for satisfactory accuracy to be obtained. Other work[38] tends to suggest that even this may not be enough, particularly for the second algorithm and that only some type of dissipation which is as truly anisotropic as that introduced by a Roe-type upwinding method[23] will do.

The results of Chapter 3 suggested that grid adaptation could improve the results. In the later chapters, the numerical diffusion estimate (NDR) has been used as the basis for a weighting function to guide grid adaptation. Various forms of weighting function, based on it, have been tried. All of these have met with some degree of success. For example, in a subsonic flow case (Chapter 4) the parameter NDI was used to guide grid adaptation, giving a physically improved solution. It is important to note that this was done in a self-consistent manner, i.e. that the distribution of NDI on the adapted grid also indicated that this was indeed an improved solution.

To address a concern about the usefulness of the NDI, on its own, for adaptation, due to its lack of selectivity of problem areas, modified versions (NDF, COMB) have been produced and their use demonstrated. These have been tested for transonic flow cases which include a shock wave. Here, the magnitude of the function tends to increase as the shock wave becomes better resolved, but the value elsewhere tends to even itself out. This type of behaviour works perfectly well in a structured adaptive grid procedure. The weighting functions based on accuracy appear to work as well as, if not better than that based on solution activity, at least for the shock wave region. They are, also, more selective in the areas which are identified as needing higher resolution.

The results of Chapter 5 have cast some doubt on the potential benefits of grid adaptation via node movement. When a relatively fine initial grid was used the results in terms of lift and drag coefficients were already quite good, and the main improvement due to adaptation was in better definition of the shock wave. When an initial grid was used which was coarse enough to give results significantly poorer than a reference fine-grid result, there were insufficient points available to resolve all the important features, even after adaptation. With an initial grid of intermediate density, some improvements in accuracy were achieved when the NDF measure was used. It could be inferred that the greatest economic benefits of adaptation are likely to be achieved either by using a node movement method and an initial grid within a medium range of densities (see Section 5.3) or by starting with a relatively coarse grid and using a node enrichment method.

6.2 Recommendations for Future Work

As far as the solution algorithms considered here are concerned, an improved numerical smoothing is essential. This means, at the least, providing a scaling which makes the smoothing truly anisotropic. Ad hoc methods of achieving this[45,46] have not worked well enough. It appears that only a truly anisotropic dissipation will be sufficient, but this may end up with the algorithm being no different to a Roe-type upwinding.

In terms of the use of numerical diffusion estimates to guide grid adaptation, there is clear encouragement that this could offer some practical advantages over currently used solution activity measures. It would be very advantageous to consider other test cases. For example, the hypothesis that accuracy based measures would be better than activity measures for a large recirculating flow region should be tested. One possibility is the $M_\infty = 0.8$, incidence = 10° case from the GAMM workshop[69], and it would, perhaps, be informative to examine non-aerofoil flows, such as large separated flow behind a bluff body or a backward facing step.

Based on the results of Chapter 5, the use of these weighting functions with a node enrichment method would be worthy of investigation, in order to avoid grid quality problems, and also possibly to exploit more fully the economic potential of grid adaptation.

Despite the improvements in solution algorithms over the last few years, there is still plenty of scope, indeed a definite necessity, to improve solution accuracy before the methods may be used routinely for design purposes. The work presented here has served to provide evidence to demonstrate and emphasise this message, and to suggest ways in which the accuracy may be improved without increased cost.

References

1. Albone,C.M., Catherall,D.C., Hall,M.G., Joyce,Gaynor "An improved numerical method for solving the transonic small-perturbation equation for the flow past a lifting aerofoil." RAE TR74056 (Aug74).

2. Ashill,P.R., Wood,R.F., Weeks,D.J. "An improved, semi-inverse version of the viscous, Garabedian and Korn method (VGK)." RAE TR87002 (Jan87).

3. Pauley,L.L., Moin,P., Reynolds,W.C. "The structure of two-dimensional separation." J.Fluid Mech. Vol.220, pp.397-411 (1990).

4. Werner,H., Wengler,H. "Large-eddy simulation of turbulent flow over and around a cube in a plate channel." Paper 19-4, 8th Symposium on Turbulent Shear Flows, Munich, 9-11 September, 1991.

5. Collyer,M.R., Lock,R.C. "Prediction of viscous effects in steady transonic flow past an aerofoil." Aero.Qu., Vol.30, pp.485-505 (Aug79).

6. Tattersall,P. "A viscous-inviscid interaction method using a cell-vertex multigrid solution of the Euler equations." RAE TR90042 (Aug90).

7. Rizk,Y.M., Gee,K. "Numerical prediction of the unsteady flowfield around the F-18 aircraft at large incidence." AIAA 91-0020 (Jan.90).

8. Peace,A.J. "Towards a cell-vertex Navier-Stokes algorithm for three-dimensional structured/unstructured meshes." ICFD Conference on Numerical Methods for Fluid Dynamics, Reading, 7-10 April 1992.

9. Ramakrishnan,S.V., Szema,K.Y., Chakravarthy,S.R., Bihari,B.L. "Supersonic flow computations for an ASTOVL aircraft configuration." AIAA 90-2997 (Jan.90).

10. Holst,T.L. "Viscous transonic airfoil workshop compendium of results." AIAA paper 87-1460 (1987).

11. Kutler,P., Steger,J.L., Bailey,F.R. "Status of Computational Fluid Dynamics in the United States." AIAA paper 87-1135 (1987).

12. Catherall,D. "The adaptation of structured grids to numerical solutions for transonic flow." International Journal for Numerical Methods in Engineering, Vol.32, pp.921-937 (1991).

13. Thompson,J.F., Warsi,Z.U.A., Mastin,C.W. "Boundary-fitted coordinate systems for numerical solution of partial-differential equations - a review." J. Comp. Phys., Vol.47, p.1, 1982.

14. Smith,R.E "Algebraic grid generation." in Numerical Grid Generation, ed. J.F.Thompson, North-Holland, p.137, 1982.

15. Löhner,R. "Generation of three-dimensional unstructured grids by the advancing-front method." AIAA 88-0515, (Jan.88).

16. Baker,T.J. "Automatic mesh generation for complex three-dimensional regions using a constrained Delaunay triangulation." in Engineering with Computers, No.5, pp.161-175, (1989).

17. Shaw,J.A., Georgala,J.M., Weatherill,N.P. "The construction of component adaptive grids for aerodynamic geometries." in Numerical Grid Generation in Computational Fluid Mechanics (eds. Sengupta,S. et al), Pineridge Press, (1988).

18. Shaw,J.A., Peace,A.J., Weatherill,N.P. "A three-dimensional hybrid structured-unstructured method: motivation, basic approach and initial results." Aircraft Research Association Report 78, (1990).

19. Albone,C.M. "Embedded meshes of controllable quality synthesised from elementary geometric features." AIAA 92-0663, (Jan.92).

20. Ni,R.H. "A multiple grid scheme for solving the Euler equations." AIAA 81-1025, (Jun.81).

21. Fletcher,C.A.J. "Computational Techniques for Fluid Dynamics, Vols.I and II." Springer Verlag (1988).

22. Beam,R.M., Warming,R.F. "An implicit factored scheme for the compressible Navier-Stokes equations." AIAA Paper 77-645 (June77)

23. Roe,P.L. "Approximate Riemann solvers, parametric vectors, and difference schemes." J. Comp. Phys., Vol.43, pp.357-372, (1981).

24. Blottner,F.G. "Significance of the thin-layer Navier-Stokes approximation." 3rd Symposium on Numerical and Physical Aspects of Aerodynamic Flows, Long Beach, (Jan.85).

25. Bradshaw,P., ed. "Turbulence." Topics in Applied Physics, Vol.12, Springer Verlag (1976).

26. Baldwin,B.S., Lomax,H. "Thin layer approximation and algebraic model for separated turbulent flows." AIAA 78-0257, (1978).

27. Degrez,G. "Implicit time dependent methods for inviscid and viscous compressible flows, with a discussion of the concept of numerical dissipation." from Introduction to CFD, VKI Lecture Series, Jan.15-19, 1990.

28. Kessler,R., Peric,M., Scheuerer,G. "Solution error estimation in the numerical predictions of turbulent recirculating flows." AGARD CP 434 (1988).

29. Forester,C.K., Tjonneland,E. "New guide for accurate Navier-Stokes solution of two-dimensional external compression inlet with bleed." ICAS-88-2.5.1 (1988).

30. Mahajan,A.J., Dowell,E.H., Bliss,D.B. "Role of artificial viscosity in Euler and Navier-Stokes solvers." AIAA Journal, Vol.29, No.4, pp.555-559 (1991).

31. Rumsey,C.L., Taylor,S.L., Thomas,J.L., Anderson,W.K. "Application of an upwind Navier-Stokes code to two-dimensional transonic airfoil flow." AIAA paper 87-0413 (1987).

32. Kuruvila,G., Anderson,J.D. "A study of the effects of numerical dissipation on the calculation of supersonic separated flows." AIAA paper 85-0301 (1985).

33. Berger,M.J., Jameson,A. "Automatic adaptive grid refinement for the Euler equations." AIAA Journal, Vol.23, No.4, pp.561-568, (1985).

34. Skamarock,W., Oliger,J., Street,R.L. "Adaptive grid refinement for numerical weather prediction." Journal of Computational Physics, Vol.80, pp.27-60, (1989).

35. Caughey,D.A., Turkel,E. "Effects of numerical dissipation on finite-volume solutions of compressible flow problems." AIAA paper 88-0621 (1988).

36. McGuirk,J.J., Taylor,A.M.K.P., Whitelaw,J.H. "The assessment of numerical diffusion in upwind difference calculations of turbulent recirculating flows." in Turbulent Shear Flows III (1982) Springer-Verlag.

37. Hall,M.G. "A vertex-centroid scheme for improved finite-volume solution of the Navier-Stokes equations." AIAA paper 91-1540 (1991).

38. Allmaras,S.R. "Contamination of laminar boundary layers by artificial dissipation in Navier-Stokes solutions." ICFD Conference on Numerical Methods in Fluid Dynamics, Reading, U.K., 7-10 April 1992.

39. Tattersall,P., McGuirk,J. "Evaluation of numerical diffusion effects in viscous flow calculations." To appear in Computers and Fluids.

40. Berger,M.J., Oliger,J. "Adaptive mesh refinement for hyperbolic partial differential equations." Manuscript NA-83-02, Stanford University, (March 1983).

41. Dannenhoffer,J.F., Baron,J.R. "Robust grid adaptation for complex transonic flows." AIAA paper 86-0495, (1986).

42. Nakahashi,K., Diewert,G,S. "A practical adaptive-grid method for complex fluid-flow problems." NASA T.M.85989, (June89).

43. Catherall,D. "A solution-adaptive-grid procedure for transonic flows around aerofoils." RAE TR88020 (Mar88).

44. O,Brien,P., Hall,M.G. "A comparison of the effects of grid distortion on finite-volume methods for solving the Euler equations." Proc. Int. Conf. Num. Meth. Fluid Dynamics 12, Oxford (1990).

45. Kunz,R.F., Lakshminarayana,B. "Three-dimensional Navier-Stokes computation of turbomachinery flows using an explicit numerical procedure and a coupled k-ε turbulence model." ASME Paper 91-GT-146, International Gas Turbine and Aeroengine Congress and Exposition, Orlando (1991).

46. Davis,R.L., Ni,R.H., Carter,J.E. "Cascade viscous flow analysis using the Navier-Stokes equations." AIAA paper 86-0033 (1986).

47. Cebeci,T., Smith,A.M.O. "Analysis of turbulent boundary layers." Academic Press (1974).

48. Jou,W.H., Wigton,L.B., Allmaras,S.R., Spalart,P.R., Yu,N.J. "Towards industrial-strength Navier-Stokes codes." in Numerical and Physical Aspects of Aerodynamic Flows V (ed. Cebeci,T.), Springer-Verlag, New York, to appear.

49. Benton,J. "Convergence acceleration for multiblock Reynolds-averaged Navier-Stokes." Presentation at AGCFM Workshop, Guildford, (Sep.91).

50. Briley,W.R., McDonald,H. "An implicit numerical method for the multi-dimensional compressible Navier-Stokes equations." Report M911363-6, United Aircraft Research Laboratories, 1973.

51. Steger,J.L. "Implicit finite difference simulation of flow about arbitrary geometries with application to airfoils." AIAA Paper 77-665 (June77)

52. Pulliam,T.H., Steger,J.L. "Implicit finite-difference simulations of three-dimensional compressible flow." AIAA Jnl., Vol.18, p.159 (1980).

53. Steger,J.L., Pulliam,T.H. "Recent improvements in efficiency, accuracy, and convergence for implicit approximate factorization algorithms." AIAA Paper 85-0360 (Jan85).

54. Peyret,R., Viviand,H. "Computation of viscous compressible flows based on the Navier-Stokes equations." AGARD-AG-212, 1975.

55. Martinelli,L. "Calculations of viscous flows with a multigrid method." Ph.D. Thesis, Department of Mechanical and Aerospace Engineering, Princeton University (1987).

56. Grasso,F., Martinelli,L., Jameson,A. "A multistage multigrid method for the compressible Navier-Stokes equations." GAMM Workshop on Numerical Simulation of Compressible Navier-Stokes Flows, INRIA, Sophia Antipolis, (1985).

57. Martinelli,L., Jameson,A., Grasso,F. "A multigrid method for the Navier-Stokes equations." AIAA Paper 86-0208 (Jan86).

58. 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).

59. Hall,M.G. "Fast multigrid solution of the Euler equations using a finite volume scheme of Lax-Wendroff type." RAE TR84013 (1984).

60. Radespiel,R., Rossow,C., Swanson,R.C. "Efficient cell-vertex multigrid scheme for the three-dimensional Navier-Stokes equations." AIAA Journal, Vol.28, No.8, pp.1464-1472, (Aug.90).

61. Jameson,A. "Transonic flow calculations." Princeton University Report, MAE 1651, (1984), included in Lecture Notes in Mathematics, 1127, edited by F.Brezzi, Springer Verlag, pp.156-242, (1985).

62. Rizzi,A. "Computational mesh for transonic airfoils." In Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves. (A.Rizzi and H.Viviand, eds.), pp.222-254, Vieweg, Braunschweig/ Weisbaden (1981).

63. Catherall,D.C., Fitzsimmons,C.I. "On the measurement of grid quality in CFD applications." RAE TM Aero2219, (Jul.91).

64. Schlichting,H. "Boundary Layer Theory." (1960) McGraw-Hill.

65. Kennon,S.R., Dulikravich,G.S. "A posteriori optimization of computational grids." AIAA paper 85-0483 (1985).

66. Carcaillet,R., Kennon,S.R., Dulikravich,G.S. "Optimization of three-dimensional computational grids." AIAA paper 85-4087 (1985).

67. Siebert,B.W., Dulikravich,G.S. "Grid generation using a posteriori optimization with geometrically normalized functionals." AIAA paper 90-3048 (1990).

68. Lee,D., Tsuei,Y.M. "A formula for estimation of truncation errors of convection terms in a curvilinear coordinate system." J.Comp.Phys., Vol.98, pp.90-100 (1992).

69. Bristeau,M.O., Glowinski,R., Periaux,J., Viviand,H. "Numerical simulation of compressible Navier-Stokes flows - a GAMM workshop." Notes on Numerical Fluid Mechanics, Vol.18, Vieweg (1987).

70. Tattersall,P., McGuirk,J. "Numerical diffusion estimates as input for grid adaptation." ICFD Conference on Numerical Methods in Fluid Dynamics, Reading, 7-10 April 1992.

71. Thompson,J.F., Warsi,Z.U.A., Mastin,C.W. "Numerical grid generation - foundations and applications." North Holland, 1985.

72. Warren,G.P., Anderson,W.K., Thomas,J.L., Krist,S.L. "Grid convergence for adaptive methods." AIAA 91-1592 (1991).

73. Warren,G.P. "Adaptive grid embedding for the two-dimensional Euler equations." AIAA 90-3049, (Aug.90).

74. Mackenzie,J.A., Mayers,D.F., Mayfield,A.J. "Error estimates for adaptive cell vertex finite volume algorithms." ICFD Conference on Numerical Methods in Fluid Dynamics, Reading, 7-10 April 1992.

75. McGuirk,J.J., Rodi,W. "A depth-averaged mathematical model for the near-field of side-discharges into open-channel flow." J.Fluid Mech. Vol.86, pp.761-781 (1978).

76. Hiemenz,K. "Die grenzscicht an einem in den gleichformigen flussigkeitsstrom eingetauchten geraden kreiszylinder." (Thesis Gottingen 1911). Dingl.Polytech.J., Vol.326, p.321 (1911).

77. Howarth,L. "On the calculation of the steady flow in the boundary layer near the surface of a cylinder in a stream." ARC R&M 1632 (1935).

Tables

Convection	Pressure gradient	Physical diffusion	Numerical diffusion	y/c	NDR	NDI
-0.311522E-03	-0.559869E-03	0.198779E-02	-0.111633E-02	0.200000E-03	0.561593E+00	0.626923E-03
-0.265649E-03	-0.518109E-03	0.166115E-02	-0.877403E-03	0.457000E-03	0.528190E+00	0.463435E-03
-0.222202E-03	-0.533815E-03	0.139277E-02	-0.636750E-03	0.786000E-03	0.457181E+00	0.291110E-03
-0.184467E-03	-0.468688E-03	0.105982E-02	-0.406627E-03	0.120800E-02	0.383674E+00	0.156012E-03
-0.155902E-03	-0.477463E-03	0.830066E-03	-0.196681E-03	0.174900E-02	0.236947E+00	0.466030E-04
-0.137259E-03	-0.397585E-03	0.566012E-03	-0.311735E-04	0.244400E-02	0.550756E-01	0.171690E-05
-0.130310E-03	-0.396475E-03	0.428110E-03	0.986728E-04	0.333500E-02	0.230485E+00	0.227426E-04
-0.129489E-03	-0.298240E-03	0.261100E-03	0.166659E-03	0.447800E-02	0.558807E+00	0.931300E-04
-0.133643E-03	-0.281613E-03	0.198265E-03	0.216980E-03	0.594500E-02	0.770490E+00	0.167181E-03
-0.131030E-03	-0.159578E-03	0.111634E-03	0.178973E-03	0.782600E-02	0.100000E+01	0.178973E-03
-0.145518E-03	-0.126847E-03	0.163853E-03	0.108540E-03	0.102390E-01	0.662422E+00	0.718993E-04
-0.248118E-03	0.121129E-04	0.297145E-03	-0.611338E-04	0.133340E-01	0.205737E+00	0.125775E-04
-0.553849E-03	0.111558E-04	0.511320E-03	0.313866E-04	0.173040E-01	0.566700E-01	0.177868E-05
-0.107444E-02	0.615572E-04	0.562339E-03	0.450546E-03	0.223980E-01	0.419330E+00	0.188928E-03
-0.166276E-02	-0.252365E-03	0.391000E-03	0.152411E-02	0.289320E-01	0.916610E+00	0.139701E-02
-0.144393E-02	-0.310362E-03	0.133064E-04	0.174094E-02	0.373130E-01	0.100000E+01	0.174094E-02
0.648005E-03	-0.248717E-03	-0.262520E-03	-0.136808E-03	0.480640E-01	0.211122E+00	0.288833E-04
0.154902E-02	-0.172854E-03	-0.239043E-03	-0.113716E-02	0.618560E-01	0.734121E+00	0.834816E-03
0.802685E-03	-0.197199E-03	-0.112278E-03	-0.493210E-03	0.795490E-01	0.614450E+00	0.303052E-03
0.152383E-03	-0.291802E-03	-0.327003E-04	0.172113E-03	0.102244E+00	0.589827E+00	0.101517E-03
-0.877614E-05	-0.361958E-03	-0.419969E-05	0.374905E-03	0.131359E+00	0.100000E+01	0.374905E-03
0.111118E-03	-0.385911E-03	0.135667E-05	0.273425E-03	0.168707E+00	0.708519E+00	0.193727E-03
0.267504E-03	-0.368046E-03	0.971012E-06	0.995628E-04	0.216619E+00	0.270518E+00	0.269335E-04
0.334612E-03	-0.329070E-03	0.259663E-06	-0.581870E-05	0.278082E+00	0.173894E-01	0.101184E-06
0.309582E-03	-0.281842E-03	0.105813E-07	-0.277543E-04	0.356931E+00	0.896510E-01	0.248820E-05
0.242268E-03	-0.227281E-03	-0.212028E-07	-0.149656E-04	0.458083E+00	0.617729E-01	0.924468E-06
0.171276E-03	-0.167654E-03	-0.101740E-07	-0.363710E-05	0.587851E+00	0.212353E-01	0.772349E-07
0.111278E-03	-0.111622E-03	-0.142358E-08	0.327000E-06	0.754330E+00	0.292952E-02	0.957953E-09
0.653244E-04	-0.671953E-04	0.203503E-08	0.187520E-05	0.967905E+00	0.279067E-01	0.523307E-07
0.335845E-04	-0.367059E-04	0.262689E-08	0.310870E-05	0.124190E+01	0.846921E-01	0.263282E-06
0.144819E-04	-0.178981E-04	0.206370E-08	0.341690E-05	0.159340E+01	0.190909E+00	0.652315E-06
0.473974E-05	-0.735447E-05	0.128731E-08	0.260600E-05	0.204432E+01	0.354342E+00	0.923416E-06
0.658721E-06	-0.204977E-05	0.688349E-09	0.138910E-05	0.262278E+01	0.677687E+00	0.941376E-06
-0.633829E-06	0.211779E-06	0.327271E-09	0.424600E-06	0.336483E+01	0.669897E+00	0.284438E-06
-0.803633E-06	0.937488E-06	0.140163E-09	-0.127500E-06	0.431674E+01	0.136002E+00	0.173402E-07
-0.614604E-06	0.101296E-05	0.534017E-10	-0.400000E-06	0.553784E+01	0.394882E+00	0.157953E-06
-0.357696E-06	0.858857E-06	0.170897E-10	-0.507100E-06	0.710426E+01	0.590436E+00	0.299410E-06
-0.136949E-06	0.608019E-06	0.357364E-11	-0.474900E-06	0.911363E+01	0.781062E+00	0.370926E-06
0.169266E-07	0.311575E-06	-0.880379E-12	-0.325400E-06	0.116912E+02	0.100000E+01	0.325400E-06
0.169266E-07	0.311575E-06	-0.880379E-12	-0.325400E-06	0.149977E+02	0.100000E+01	0.325400E-06

Table 4.1

Convection	Pressure gradient	Physical diffusion	Numerical diffusion	y/c	NDR	NDI
-0.378677E+01	-0.936037E+00	-0.160822E+03	0.165543E+03	0.200000E-05	0.100000E+01	0.165543E+03
-0.161033E+01	-0.664395E+00	0.140472E+03	-0.138197E+03	0.650000E-05	0.983800E+00	0.135958E+03
-0.189786E+01	-0.541240E+00	0.460427E+02	-0.436029E+02	0.125000E-04	0.947010E+00	0.412924E+02
-0.306359E+01	-0.695370E+00	0.241887E+01	0.134120E+01	0.200000E-04	0.437788E+00	0.587161E+00
-0.374893E+01	-0.629485E+00	-0.106629E+02	0.150422E+02	0.290000E-04	0.100000E+01	0.150422E+02
-0.588570E+01	-0.616256E+00	-0.173453E+02	0.238478E+02	0.405031E-04	0.100000E+01	0.238478E+02
-0.753557E+01	-0.635810E+00	-0.121855E+02	0.203574E+02	0.550091E-04	0.100000E+01	0.203574E+02
-0.772474E+01	-0.644702E+00	-0.859102E+01	0.169606E+02	0.730154E-04	0.100000E+01	0.169606E+02
-0.699513E+01	-0.649446E+00	-0.160334E+01	0.924805E+01	0.955209E-04	0.100000E+01	0.924805E+01
-0.595162E+01	-0.645902E+00	0.252465E+01	0.407290E+01	0.124025E-03	0.684335E+00	0.278723E+01
-0.569943E+01	-0.635806E+00	0.929223E+00	0.540603E+01	0.160538E-03	0.948522E+00	0.512774E+01
-0.567827E+01	-0.637823E+00	-0.969094E+00	0.728522E+01	0.206549E-03	0.100000E+01	0.728522E+01
-0.499805E+01	-0.646621E+00	0.777209E+00	0.486748E+01	0.264068E-03	0.973876E+00	0.474032E+01
-0.418392E+01	-0.644767E+00	0.318553E+01	0.164318E+01	0.336095E-03	0.392738E+00	0.645340E+00
-0.393690E+01	-0.639121E+00	0.298699E+01	0.158903E+01	0.427117E-03	0.403626E+00	0.641375E+00
-0.394287E+01	-0.641250E+00	0.228487E+01	0.229926E+01	0.542144E-03	0.583144E+00	0.134080E+01
-0.378438E+01	-0.643044E+00	0.211510E+01	0.231234E+01	0.686686E-03	0.611020E+00	0.141288E+01
-0.340690E+01	-0.643472E+00	0.196004E+01	0.209032E+01	0.868242E-03	0.613556E+00	0.128253E+01
-0.287808E+01	-0.644330E+00	0.168226E+01	0.184015E+01	0.109681E-02	0.639366E+00	0.117653E+01
-0.210222E+01	-0.647617E+00	0.136362E+01	0.138621E+01	0.138439E-02	0.659400E+00	0.914065E+00
-0.117013E+01	-0.652179E+00	0.105151E+01	0.770794E+00	0.174601E-02	0.658725E+00	0.507741E+00
-0.235631E+00	-0.660600E+00	0.673526E+00	0.222702E+00	0.220064E-02	0.330651E+00	0.736365E-01
0.553271E+00	-0.674724E+00	0.276682E+00	-0.155232E+00	0.277231E-02	0.230067E+00	0.357138E-01
0.114647E+01	-0.699545E+00	0.616526E-01	-0.508569E+00	0.349152E-02	0.443597E+00	0.225600E+00
0.145275E+01	-0.737047E+00	0.754999E-02	-0.723251E+00	0.439529E-02	0.497849E+00	0.360069E+00
0.143975E+01	-0.789921E+00	0.604021E-03	-0.650428E+00	0.553062E-02	0.451766E+00	0.293841E+00
0.124664E+01	-0.860911E+00	-0.651837E-05	-0.385723E+00	0.695704E-02	0.309409E+00	0.119346E+00
0.107439E+01	-0.951967E+00	-0.523644E-04	-0.122371E+00	0.874907E-02	0.113898E+00	0.139378E-01
0.103071E+01	-0.106569E+01	-0.366453E-04	0.350139E-01	0.109997E-01	0.328556E-01	0.115041E-02
0.111303E+01	-0.120635E+01	-0.158538E-04	0.933380E-01	0.138255E-01	0.773723E-01	0.722178E-02
0.126978E+01	-0.137904E+01	-0.282467E-05	0.109257E+00	0.173725E-01	0.792267E-01	0.865603E-02
0.146339E+01	-0.158751E+01	0.252485E-05	0.124101E+00	0.218232E-01	0.781736E-01	0.970144E-02
0.168403E+01	-0.183130E+01	0.328621E-05	0.147274E+00	0.274071E-01	0.804204E-01	0.118438E-01
0.193047E+01	-0.210186E+01	0.193524E-05	0.171391E+00	0.344131E-01	0.815425E-01	0.139756E-01
0.219237E+01	-0.237831E+01	-0.937718E-07	0.185944E+00	0.432091E-01	0.781830E-01	0.145376E-01
0.243894E+01	-0.262500E+01	-0.192464E-05	0.186064E+00	0.542704E-01	0.708816E-01	0.131886E-01
0.262178E+01	-0.279479E+01	-0.278603E-05	0.173017E+00	0.682258E-01	0.619070E-01	0.107110E-01
0.268202E+01	-0.283039E+01	-0.200226E-05	0.148371E+00	0.859166E-01	0.524209E-01	0.777775E-02
0.254586E+01	-0.265184E+01	0.730070E-06	0.105987E+00	0.108424E+00	0.399673E-01	0.423602E-02
0.211179E+01	-0.215906E+01	0.506003E-05	0.472614E-01	0.137016E+00	0.218898E-01	0.103454E-02
0.131740E+01	-0.131746E+01	0.949032E-05	0.504256E-04	0.173038E+00	0.382750E-04	0.193004E-08
0.315252E+00	-0.307061E+00	0.115024E-04	-0.820287E-02	0.217897E+00	0.260200E-01	0.213439E-03
-0.487075E+00	0.479782E+00	0.971992E-05	0.728414E-02	0.273142E+00	0.149549E-01	0.108934E-03
-0.746743E+00	0.732236E+00	0.570794E-05	0.145004E-01	0.340445E+00	0.194183E-01	0.281574E-03
-0.518982E+00	0.509570E+00	0.223866E-05	0.941073E-02	0.421384E+00	0.181330E-01	0.170645E-03
-0.147959E+00	0.145567E+00	0.555039E-06	0.239167E-02	0.517351E+00	0.161643E-01	0.386597E-04
0.955474E-01	-0.935640E-01	0.131524E-06	-0.198325E-02	0.629863E+00	0.207567E-01	0.411657E-04
0.170270E+00	-0.166842E+00	0.106667E-06	-0.342756E-02	0.761036E+00	0.201302E-01	0.689973E-04
0.154967E+00	-0.152027E+00	0.888143E-07	-0.294045E-02	0.914140E+00	0.189746E-01	0.557939E-04
0.119182E+00	-0.117613E+00	0.454389E-07	-0.156848E-02	0.109358E+01	0.131604E-01	0.206417E-04
0.885352E-01	-0.879559E-01	0.145830E-07	-0.579340E-03	0.130478E+01	0.654361E-02	0.379098E-05
0.648465E-01	-0.646568E-01	0.275290E-08	-0.189719E-03	0.155453E+01	0.292566E-02	0.555052E-06
0.460733E-01	-0.459810E-01	0.531722E-10	-0.922163E-04	0.185125E+01	0.200151E-02	0.184572E-06
0.315692E-01	-0.315227E-01	-0.360510E-09	-0.465200E-04	0.220534E+01	0.147359E-02	0.685512E-07
0.210564E-01	-0.210534E-01	-0.430477E-09	-0.300890E-05	0.262964E+01	0.142897E-03	0.429963E-09
0.138364E-01	-0.138581E-01	-0.384616E-09	0.217517E-04	0.313998E+01	0.156960E-02	0.341414E-07
0.901606E-02	-0.904167E-02	-0.278411E-09	0.255969E-04	0.375593E+01	0.283099E-02	0.724647E-07
0.583451E-02	-0.585418E-02	-0.176328E-09	0.196755E-04	0.450170E+01	0.336093E-02	0.661278E-07
0.374811E-02	-0.375997E-02	-0.104901E-09	0.118695E-04	0.540724E+01	0.315679E-02	0.374694E-07
0.238572E-02	-0.239096E-02	-0.610480E-10	0.523870E-05	0.650970E+01	0.219104E-02	0.114782E-07
0.150529E-02	-0.149228E-02	-0.351159E-10	-0.130084E-04	0.787573E+01	0.864177E-02	0.112416E-06
0.917422E-03	-0.902160E-03	-0.197681E-10	-0.152615E-04	0.959543E+01	0.166352E-01	0.253879E-06
0.529033E-03	-0.522147E-03	-0.102831E-10	-0.688805E-05	0.117718E+02	0.130201E-01	0.896830E-07
0.280924E-03	-0.298723E-03	-0.370954E-11	0.177995E-04	0.145413E+02	0.595854E-01	0.106059E-05

Table 4.2

Convection	Pressure gradient	Physical diffusion	Numerical diffusion	y/c	NDR	NDI
0.680477E-01	-0.808618E-03	-0.392865E-01	-0.279480E-01	0.200000E-05	0.410712E+00	0.114786E-01
0.113047E+00	-0.808589E-03	-0.801730E-02	-0.104217E+00	0.763217E-05	0.921886E+00	0.960761E-01
0.129046E+00	-0.808554E-03	0.199617E-02	-0.130231E+00	0.196023E-04	0.100000E+01	0.130231E+00
0.113462E+00	-0.808509E-03	0.196482E-02	-0.114616E+00	0.393859E-04	0.100000E+01	0.114616E+00
0.112714E+00	-0.808467E-03	0.262779E-02	-0.114532E+00	0.673684E-04	0.100000E+01	0.114532E+00
0.922201E-01	-0.808423E-03	0.258196E-02	-0.939920E-01	0.103712E-03	0.100000E+01	0.939920E-01
0.793339E-01	-0.808378E-03	0.130774E-02	-0.798319E-01	0.148703E-03	0.100000E+01	0.798319E-01
0.705015E-01	-0.808329E-03	0.112077E-02	-0.708131E-01	0.201802E-03	0.100000E+01	0.708131E-01
0.583433E-01	-0.808276E-03	0.888151E-03	-0.584226E-01	0.263318E-03	0.100000E+01	0.584226E-01
0.487156E-01	-0.808221E-03	0.497580E-03	-0.484047E-01	0.334225E-03	0.993619E+00	0.480959E-01
0.423968E-01	-0.808160E-03	0.360439E-03	-0.419489E-01	0.414269E-03	0.989434E+00	0.415057E-01
0.354740E-01	-0.808099E-03	0.248107E-03	-0.349139E-01	0.504072E-03	0.984211E+00	0.343627E-01
0.306780E-01	-0.808029E-03	0.161231E-03	-0.300313E-01	0.605005E-03	0.978919E+00	0.293982E-01
0.262195E-01	-0.807950E-03	0.903971E-04	-0.255020E-01	0.717866E-03	0.972635E+00	0.248041E-01
0.234360E-01	-0.807860E-03	0.383769E-04	-0.226667E-01	0.844362E-03	0.967176E+00	0.219227E-01
0.212938E-01	-0.808428E-03	0.122203E-04	-0.204978E-01	0.986061E-03	0.962620E+00	0.197316E-01
0.196838E-01	-0.807636E-03	0.232800E-05	-0.188788E-01	0.114632E-02	0.959101E+00	0.181067E-01
0.180321E-01	-0.807488E-03	0.733266E-05	-0.172321E-01	0.132966E-02	0.955638E+00	0.164677E-01
0.159492E-01	-0.807310E-03	0.242328E-04	-0.151663E-01	0.154080E-02	0.950912E+00	0.144218E-01
0.129011E-01	-0.807103E-03	0.437997E-04	-0.121379E-01	0.178743E-02	0.940845E+00	0.114199E-01
0.884549E-02	-0.807073E-03	0.538032E-04	-0.809232E-02	0.207956E-02	0.914852E+00	0.740327E-02
0.410417E-02	-0.806537E-03	0.448240E-04	-0.334251E-02	0.243049E-02	0.814418E+00	0.272220E-02
-0.483330E-03	-0.806159E-03	0.251486E-04	0.126429E-02	0.285676E-02	0.100000E+01	0.126429E-02
-0.371831E-02	-0.805933E-03	0.135357E-04	0.451067E-02	0.337912E-02	0.100000E+01	0.451067E-02
-0.437527E-02	-0.805132E-03	0.426955E-05	0.517611E-02	0.402533E-02	0.100000E+01	0.517611E-02
-0.162868E-02	-0.804597E-03	-0.335499E-04	0.246681E-02	0.482807E-02	0.100000E+01	0.246681E-02
0.452181E-02	-0.803656E-03	-0.180043E-03	-0.353812E-02	0.582816E-02	0.782458E+00	0.276843E-02
0.129995E-01	-0.802655E-03	-0.552096E-03	-0.116448E-01	0.707699E-02	0.895786E+00	0.104312E-01
0.217849E-01	-0.801425E-03	-0.107182E-02	-0.199117E-01	0.863718E-02	0.914012E+00	0.181995E-01
0.284308E-01	-0.800142E-03	-0.133922E-02	-0.262915E-01	0.105858E-01	0.924752E+00	0.243131E-01
0.306285E-01	-0.798654E-03	-0.864400E-03	-0.289654E-01	0.130177E-01	0.945702E+00	0.273927E-01
0.268556E-01	-0.796642E-03	0.282974E-03	-0.263420E-01	0.160512E-01	0.980872E+00	0.258381E-01
0.169559E-01	-0.794327E-03	0.775701E-03	-0.169372E-01	0.198332E-01	0.998902E+00	0.169186E-01
0.286103E-02	-0.791372E-03	0.100251E-02	-0.307216E-02	0.245451E-01	0.100000E+01	0.307216E-02
-0.118092E-01	-0.787530E-03	0.995456E-03	0.116013E-01	0.304125E-01	0.982392E+00	0.113970E-01
-0.228256E-01	-0.782338E-03	0.637626E-03	0.229702E-01	0.377158E-01	0.100000E+01	0.229702E-01
-0.268752E-01	-0.774980E-03	0.267293E-03	0.273828E-01	0.468034E-01	0.100000E+01	0.273828E-01
-0.230130E-01	-0.764150E-03	-0.256631E-04	0.238027E-01	0.581093E-01	0.100000E+01	0.238027E-01
-0.131409E-01	-0.748275E-03	-0.194767E-03	0.140839E-01	0.721712E-01	0.100000E+01	0.140839E-01
-0.118323E-02	-0.726994E-03	-0.243564E-03	0.215379E-02	0.896590E-01	0.100000E+01	0.215379E-02
0.861580E-02	-0.702583E-03	-0.203031E-03	-0.771020E-02	0.111405E+00	0.894891E+00	0.689979E-02
0.133859E-01	-0.679365E-03	-0.101224E-03	-0.126053E-01	0.138445E+00	0.941684E+00	0.118702E-01
0.125826E-01	-0.660052E-03	0.928946E-05	-0.119319E-01	0.172066E+00	0.948281E+00	0.113148E-01
0.789068E-02	-0.643712E-03	-0.818923E-07	-0.724691E-02	0.213867E+00	0.918414E+00	0.665566E-02
0.217938E-02	-0.623315E-03	0.557678E-07	-0.155613E-02	0.265838E+00	0.714024E+00	0.111111E-02
-0.197218E-02	-0.586823E-03	0.866527E-07	0.255892E-02	0.330454E+00	0.100000E+01	0.255892E-02
-0.341538E-02	-0.519946E-03	0.669138E-07	0.393525E-02	0.410791E+00	0.100000E+01	0.393525E-02
-0.262743E-02	-0.412346E-03	0.370266E-07	0.303973E-02	0.510674E+00	0.100000E+01	0.303973E-02
-0.981789E-03	-0.266691E-03	0.152796E-07	0.124846E-02	0.634862E+00	0.100000E+01	0.124846E-02
0.272683E-03	-0.980707E-04	0.435970E-08	-0.174618E-03	0.789287E+00	0.640368E+00	0.111820E-03
0.623543E-03	0.839546E-04	0.553243E-09	-0.707498E-03	0.981401E+00	0.100000E+01	0.707498E-03
0.258523E-03	0.292302E-03	-0.189084E-09	-0.550826E-03	0.122072E+01	0.100000E+01	0.550826E-03
-0.357116E-03	0.549768E-03	-0.145792E-09	-0.192652E-03	0.151968E+01	0.350424E+00	0.675098E-04
-0.877131E-03	0.852659E-03	-0.755100E-10	0.244717E-04	0.189463E+01	0.278997E-01	0.682753E-06
-0.120451E-02	0.115403E-02	-0.643434E-10	0.504857E-04	0.236646E+01	0.419138E-01	0.211604E-05
-0.139167E-02	0.138880E-02	-0.679019E-10	0.286887E-05	0.295994E+01	0.206145E-02	0.591404E-08
-0.149551E-02	0.151220E-02	-0.650890E-10	-0.166933E-04	0.370195E+01	0.110391E-01	0.184278E-06
-0.151619E-02	0.151519E-02	-0.543628E-10	0.100535E-05	0.462015E+01	0.663076E-03	0.666624E-09
-0.142210E-02	0.140918E-02	-0.383711E-10	0.129144E-04	0.574409E+01	0.908120E-02	0.117278E-06
-0.120151E-02	0.120379E-02	-0.212887E-10	-0.228136E-05	0.710837E+01	0.189515E-02	0.432352E-08
-0.913389E-03	0.921687E-03	-0.677638E-11	-0.829710E-05	0.877041E+01	0.900208E-02	0.746911E-07
-0.658279E-03	0.631241E-03	-0.439916E-12	0.270352E-04	0.108113E+02	0.410696E-01	0.111032E-05
-0.510736E-03	0.435270E-03	-0.604270E-11	0.754663E-04	0.133295E+02	0.147760E+00	0.111509E-04
-0.322823E-03	0.436429E-03	-0.796395E-11	-0.113606E-03	0.164530E+02	0.260307E+00	0.295724E-04

Table 4.3

Convection	Pressure gradient	Physical diffusion	Numerical diffusion	y/c	NDR	NDI
-0.627533E-01	0.262643E+01	0.284455E+02	-0.310115E+02	0.206155E-05	0.100000E+01	0.310115E+02
-0.196762E+00	0.262553E+01	-0.217818E+02	0.193544E+02	0.657647E-05	0.888557E+00	0.171975E+02
-0.294832E+00	0.261209E+01	-0.127822E+02	0.104655E+02	0.120416E-04	0.818756E+00	0.856870E+01
-0.371898E+00	0.261734E+01	-0.935619E+01	0.711194E+01	0.190591E-04	0.760132E+00	0.540602E+01
-0.463154E+00	0.262674E+01	-0.540059E+01	0.323783E+01	0.285701E-04	0.599533E+00	0.194119E+01
-0.544380E+00	0.261560E+01	-0.370816E+01	0.163786E+01	0.400780E-04	0.441691E+00	0.723429E+00
-0.583387E+00	0.262295E+01	-0.285998E+01	0.821171E+00	0.541133E-04	0.287125E+00	0.235779E+00
-0.630448E+00	0.261819E+01	-0.192828E+01	-0.589853E-01	0.726395E-04	0.225291E-01	0.132888E-02
-0.667608E+00	0.262090E+01	-0.195301E+01	0.189142E-03	0.956582E-04	0.721667E-04	0.136497E-07
-0.711253E+00	0.262075E+01	-0.199818E+01	0.888798E-01	0.124697E-03	0.339139E-01	0.301426E-02
-0.752264E+00	0.261988E+01	-0.145790E+01	-0.409516E+00	0.161251E-03	0.156311E+00	0.640119E-01
-0.793222E+00	0.261836E+01	-0.867499E+00	-0.957624E+00	0.207319E-03	0.365735E+00	0.350236E+00
-0.839482E+00	0.261772E+01	-0.165200E+01	-0.126210E+00	0.265396E-03	0.482138E-01	0.608507E-02
-0.867187E+00	0.261388E+01	-0.268350E+01	0.936785E+00	0.338007E-03	0.349091E+00	0.327023E+00
-0.865143E+00	0.261253E+01	-0.297854E+01	0.123113E+01	0.430142E-03	0.413335E+00	0.508871E+00
-0.844203E+00	0.261172E+01	-0.292691E+01	0.115940E+01	0.546297E-03	0.396117E+00	0.459256E+00
-0.810611E+00	0.261020E+01	-0.277639E+01	0.976798E+00	0.692017E-03	0.351823E+00	0.343660E+00
-0.768297E+00	0.260888E+01	-0.266439E+01	0.823810E+00	0.875290E-03	0.309192E+00	0.254716E+00
-0.719759E+00	0.260775E+01	-0.263315E+01	0.745162E+00	0.110610E-02	0.282993E+00	0.210876E+00
-0.664780E+00	0.260600E+01	-0.266178E+01	0.720557E+00	0.139701E-02	0.270705E+00	0.195059E+00
-0.605916E+00	0.260450E+01	-0.271907E+01	0.720467E+00	0.176303E-02	0.264968E+00	0.190901E+00
-0.542472E+00	0.260282E+01	-0.282901E+01	0.768663E+00	0.222316E-02	0.271707E+00	0.208851E+00
-0.485200E+00	0.259682E+01	-0.311674E+01	0.100513E+01	0.280199E-02	0.322494E+00	0.324148E+00
-0.447794E+00	0.259172E+01	-0.394259E+01	0.179866E+01	0.353001E-02	0.456214E+00	0.820574E+00
-0.452838E+00	0.259535E+01	-0.439438E+01	0.225187E+01	0.444527E-02	0.512444E+00	0.115396E+01
-0.676259E+00	0.260006E+01	-0.303371E+01	0.110991E+01	0.559636E-02	0.365859E+00	0.406070E+00
-0.121098E+01	0.258826E+01	-0.122353E+01	-0.153742E+00	0.704383E-02	0.593997E-01	0.913224E-02
-0.188955E+01	0.254745E+01	-0.608908E-01	-0.597004E+00	0.886329E-02	0.234354E+00	0.139910E+00
-0.261886E+01	0.246211E+01	0.994917E+00	-0.838163E+00	0.111503E-01	0.320049E+00	0.268253E+00
-0.335394E+01	0.230349E+01	0.152777E+01	-0.477323E+00	0.140240E-01	0.142317E+00	0.679313E-01
-0.381693E+01	0.202836E+01	0.146703E+01	0.321546E+00	0.176344E-01	0.842421E-01	0.270877E-01
-0.352191E+01	0.159222E+01	0.101710E+01	0.912595E+00	0.221702E-01	0.259119E+00	0.236471E+00
-0.230417E+01	0.995465E+00	0.452587E+00	0.856117E+00	0.278664E-01	0.371551E+00	0.318091E+00
-0.701780E+00	0.301893E+00	0.875044E-01	0.312382E+00	0.350185E-01	0.445128E+00	0.139050E+00
0.577771E+00	-0.404293E+00	-0.443746E-02	-0.169040E+00	0.439984E-01	0.292573E+00	0.494565E-01
0.134127E+01	-0.107818E+01	-0.407802E-02	-0.259012E+00	0.552709E-01	0.193109E+00	0.500174E-01
0.182023E+01	-0.170148E+01	-0.737824E-03	-0.118003E+00	0.694217E-01	0.648287E-01	0.764999E-02
0.221383E+01	-0.222605E+01	-0.706510E-04	0.122875E-01	0.871928E-01	0.551985E-02	0.678248E-04
0.249228E+01	-0.255684E+01	0.139226E-04	0.645493E-01	0.109538E+00	0.252457E-01	0.162959E-02
0.250764E+01	-0.258461E+01	0.201222E-04	0.769585E-01	0.137722E+00	0.297757E-01	0.229149E-02
0.217935E+01	-0.224816E+01	0.171318E-04	0.688032E-01	0.173461E+00	0.306042E-01	0.210567E-02
0.154894E+01	-0.158774E+01	0.127993E-04	0.387863E-01	0.218961E+00	0.244286E-01	0.947497E-03
0.745128E+00	-0.755962E+00	0.820237E-05	0.108258E-01	0.276666E+00	0.143205E-01	0.155031E-03
-0.358812E-01	0.268015E-01	0.417470E-05	0.907549E-02	0.348684E+00	0.252932E+00	0.229548E-02
-0.574591E+00	0.554230E+00	0.136388E-05	0.203600E-01	0.436362E+00	0.354340E-01	0.721437E-03
-0.758602E+00	0.737462E+00	-0.168694E-07	0.211427E-01	0.540534E+00	0.278706E-01	0.589261E-03
-0.658731E+00	0.647907E+00	-0.358392E-06	0.108255E-01	0.662274E+00	0.164339E-01	0.177906E-03
-0.442416E+00	0.441497E+00	-0.242286E-06	0.920822E-03	0.803507E+00	0.208135E-02	0.191655E-05
-0.242502E+00	0.245606E+00	-0.688241E-07	-0.310456E-02	0.967491E+00	0.126404E-01	0.392429E-04
-0.108425E+00	0.111440E+00	0.194337E-07	-0.301449E-02	0.115869E+01	0.270504E-01	0.815433E-04
-0.343968E-01	0.361429E-01	0.349410E-07	-0.174614E-02	0.138257E+01	0.483121E-01	0.843597E-04
0.110071E-02	-0.296996E-03	0.240234E-07	-0.803732E-03	0.164600E+01	0.730193E+00	0.586880E-03
0.154488E-01	-0.150789E-01	0.124177E-07	-0.369921E-03	0.195740E+01	0.239449E-01	0.885774E-05
0.190178E-01	-0.188147E-01	0.575960E-08	-0.203098E-03	0.232721E+01	0.106794E-01	0.216895E-05
0.175820E-01	-0.174628E-01	0.258249E-08	-0.119212E-03	0.276828E+01	0.678037E-02	0.808304E-06
0.143198E-01	-0.142630E-01	0.110470E-08	-0.568568E-04	0.329645E+01	0.397050E-02	0.225750E-06
0.108498E-01	-0.108365E-01	0.422349E-09	-0.132678E-04	0.393131E+01	0.122286E-02	0.162246E-07
0.785043E-02	-0.785746E-02	0.128101E-09	0.700947E-05	0.469711E+01	0.892079E-03	0.625300E-08
0.549657E-02	-0.550390E-02	0.164140E-10	0.732607E-05	0.562384E+01	0.133107E-02	0.975150E-08
0.374039E-02	-0.373805E-02	-0.183886E-10	-0.234209E-05	0.674873E+01	0.626162E-03	0.146652E-08
0.247993E-02	-0.243974E-02	-0.230137E-10	-0.401979E-04	0.813960E+01	0.162093E-01	0.651581E-06
0.155585E-02	-0.151432E-02	-0.182681E-10	-0.415281E-04	0.988802E+01	0.266916E-01	0.110845E-05
0.897560E-03	-0.884382E-03	-0.108491E-10	-0.131809E-04	0.120977E+02	0.146852E-01	0.193564E-06
0.456074E-03	-0.506310E-03	-0.287355E-11	0.502363E-04	0.149062E+02	0.992204E-01	0.498447E-05

Table 4.4

Cycle	C_L	C_D
0	0.5001	0.0086
1	0.4963	0.0084
2	0.4980	0.0085
3	0.5011	0.0086
4	0.5042	0.0086

Table 5.1 DMDP

Cycle	C_L	C_D
0	0.5001	0.0086
1	0.4837	0.0087
2	0.4882	0.0090
3	0.4997	0.0092
4	0.4981	0.0091

Table 5.2 NDF

Cycle	C_L	C_D
0	0.5001	0.0086
1	0.4912	0.0086
2	0.4945	0.0088
3	0.4906	0.0087
4	0.4945	0.0087

Table 5.3 COMB

Grid Dimensions	C_L	C_D	Designation
321x65	0.7364	0.0365	c5
161x65	0.7289	0.0367	c51
81x65	0.7393	0.0413	c52
161x33	0.7531	0.0374	c53
81x33	0.7606	0.0406	c54
81x17	0.7464	0.0403	c55
41x9	0.5292	0.0616	c56

Table 5.4 Turbulent case (5°)

Cycle	C_L	C_D
0	0.7464	0.0403
1	0.8084	0.0411
2	0.8259	0.0422
3	0.8387	0.0435
4	0.8455	0.0443

Table 5.5 DMDP

Cycle	C_L	C_D
0	0.7464	0.0403
1	0.8132	0.0501
2	0.8448	0.0462
3	0.8385	0.0522
4	0.8705	0.0492

Table 5.6 NDF - Strategy A

Cycle	C_L	C_D
0	0.7464	0.0403
1	0.8132	0.0501
2	0.8316	0.0447
3	0.8201	0.0487
4	0.8079	0.0448

Table 5.7 NDF - Strategy B

Cycle	C_L	C_D
0	0.7531	0.0374
1	0.6985	0.0348
2	0.7627	0.0374
3	0.7179	0.0350
4	0.7671	0.0375
5	0.7050	0.0346

Table 5.8 DMDP

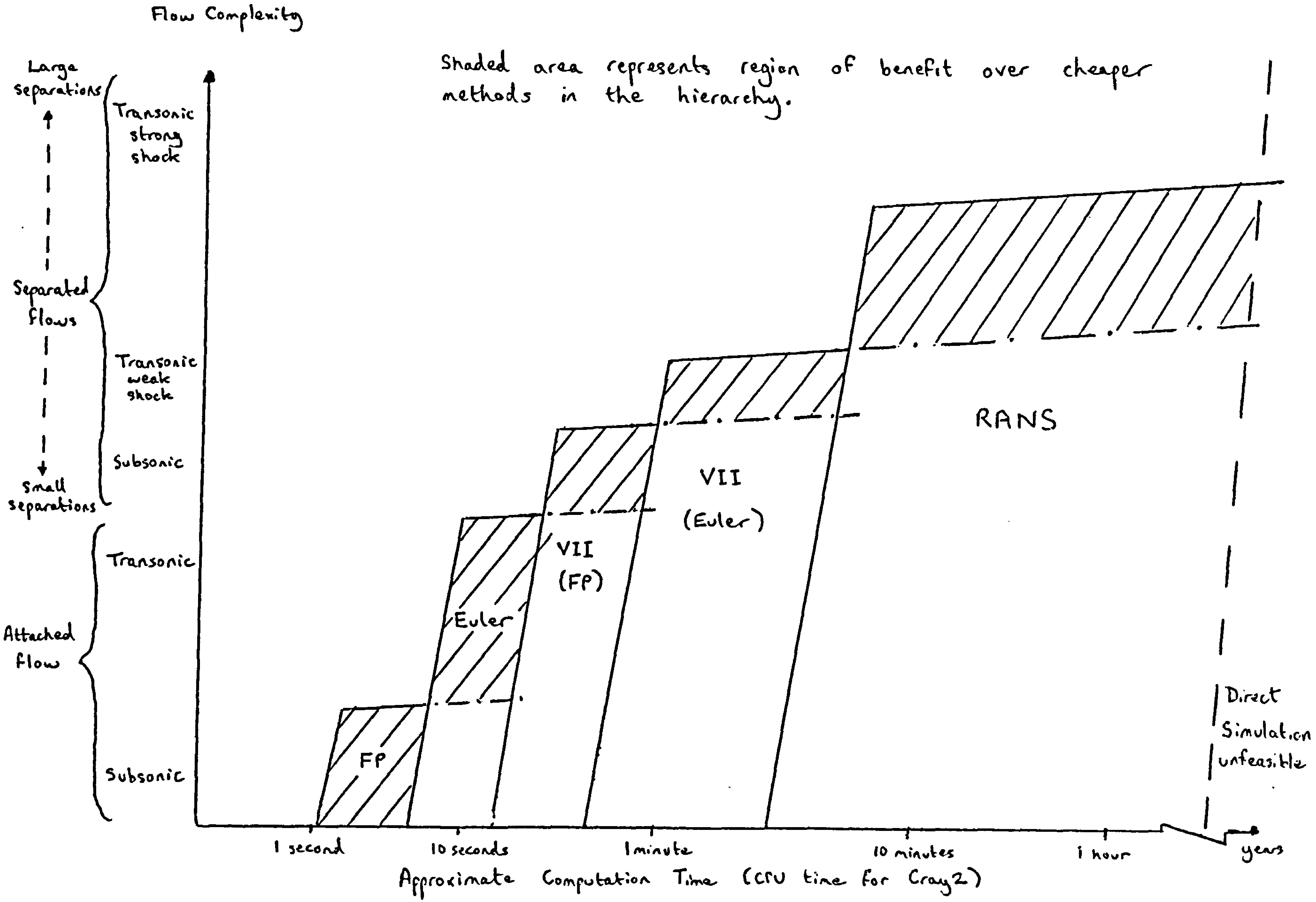
Cycle	C_L	C_D
0	0.7531	0.0374
1	0.7339	0.0375
2	0.7025	0.0365
3	0.7173	0.0363
4	0.7196	0.0367

Table 5.9 NDF - Strategy A

Cycle	C_L	C_D
0	0.7531	0.0374
1	0.7339	0.0375
2	0.7228	0.0381
3	0.7348	0.0376
4	0.7338	0.0377

Table 5.10 NDF - Strategy B

Figures



Shaded area represents region of benefit over cheaper methods in the hierarchy.

Fig.1.1 *Approximate Capability (2D) 1993 - single aerofoil* x,y are cartesian coordinates, usually with origin at aerofoil leading - edge

ξ,η are boundary - conforming curvilinear coordinates

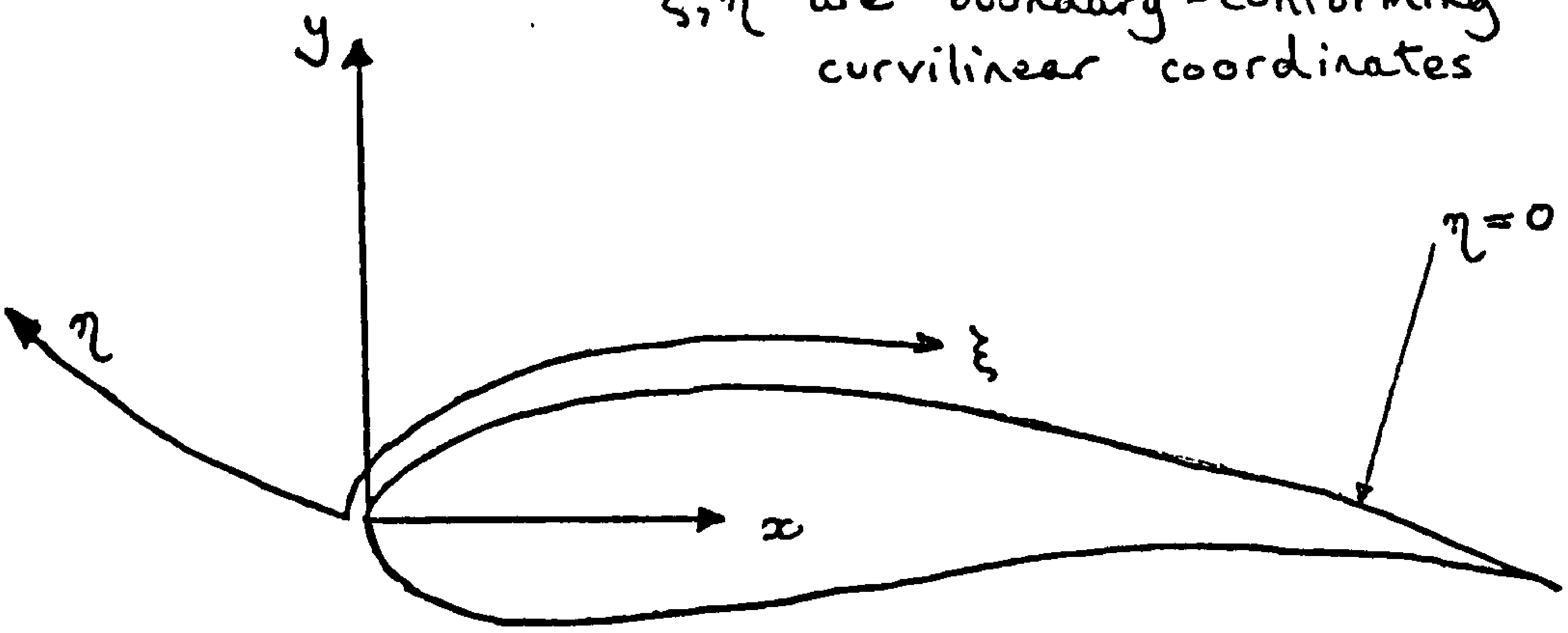
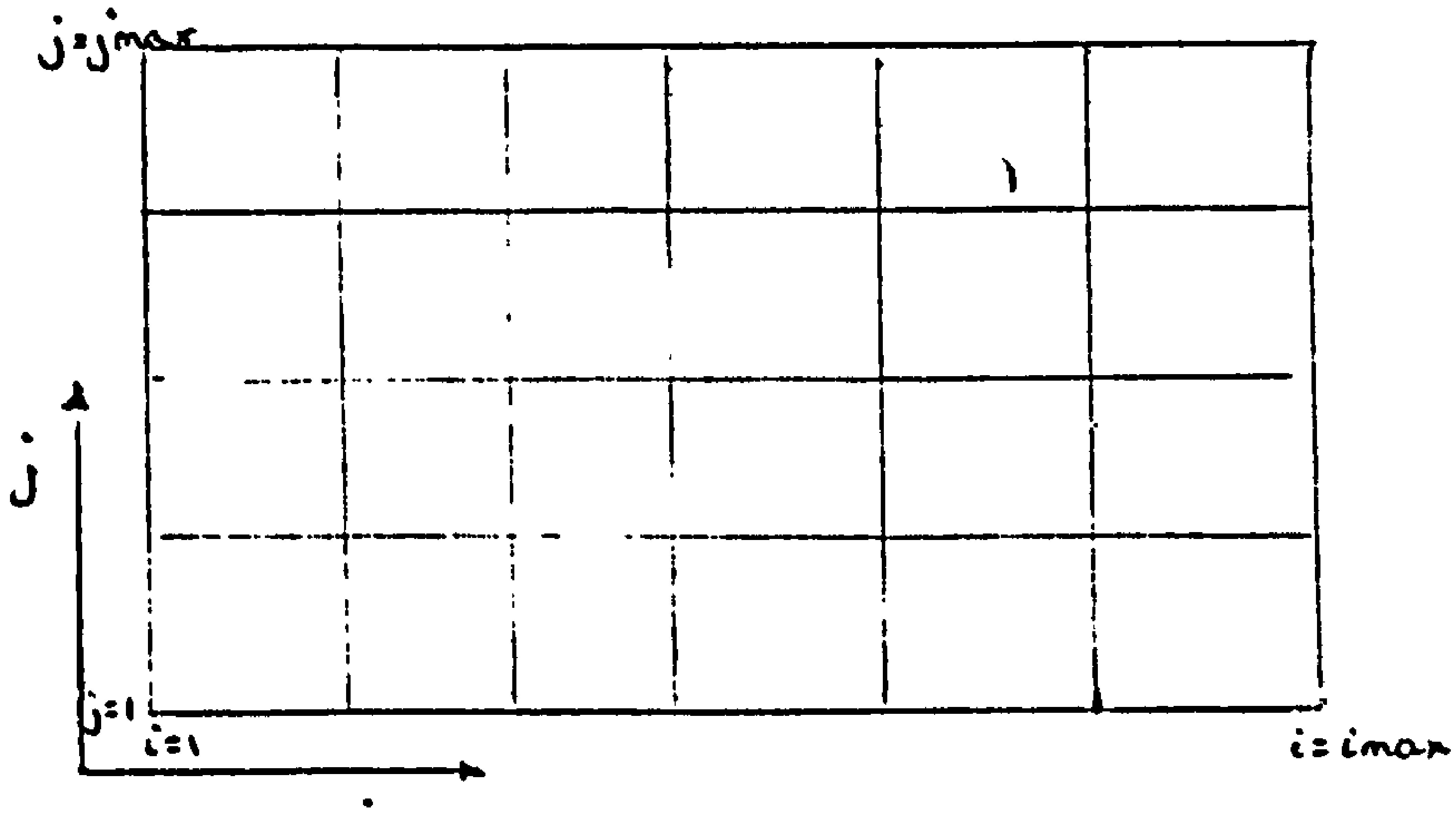


Fig.2.1 Relationship of curvilinear to cartesian coordinates



Physical space usually transformed to a regular computational space

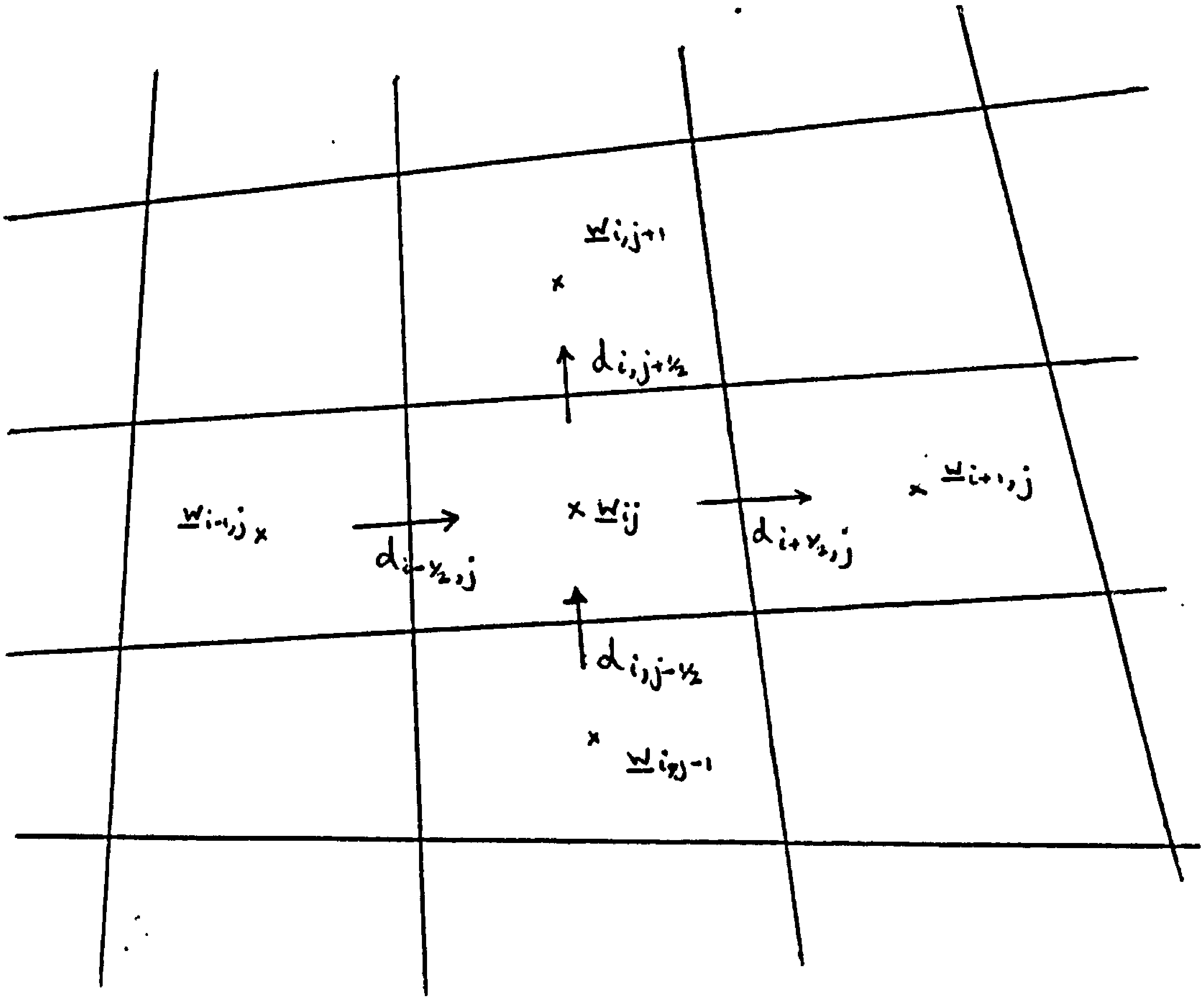


Fig.2.2 Smoothing fluxes associated with a computational cell in cell-centred explicit scheme

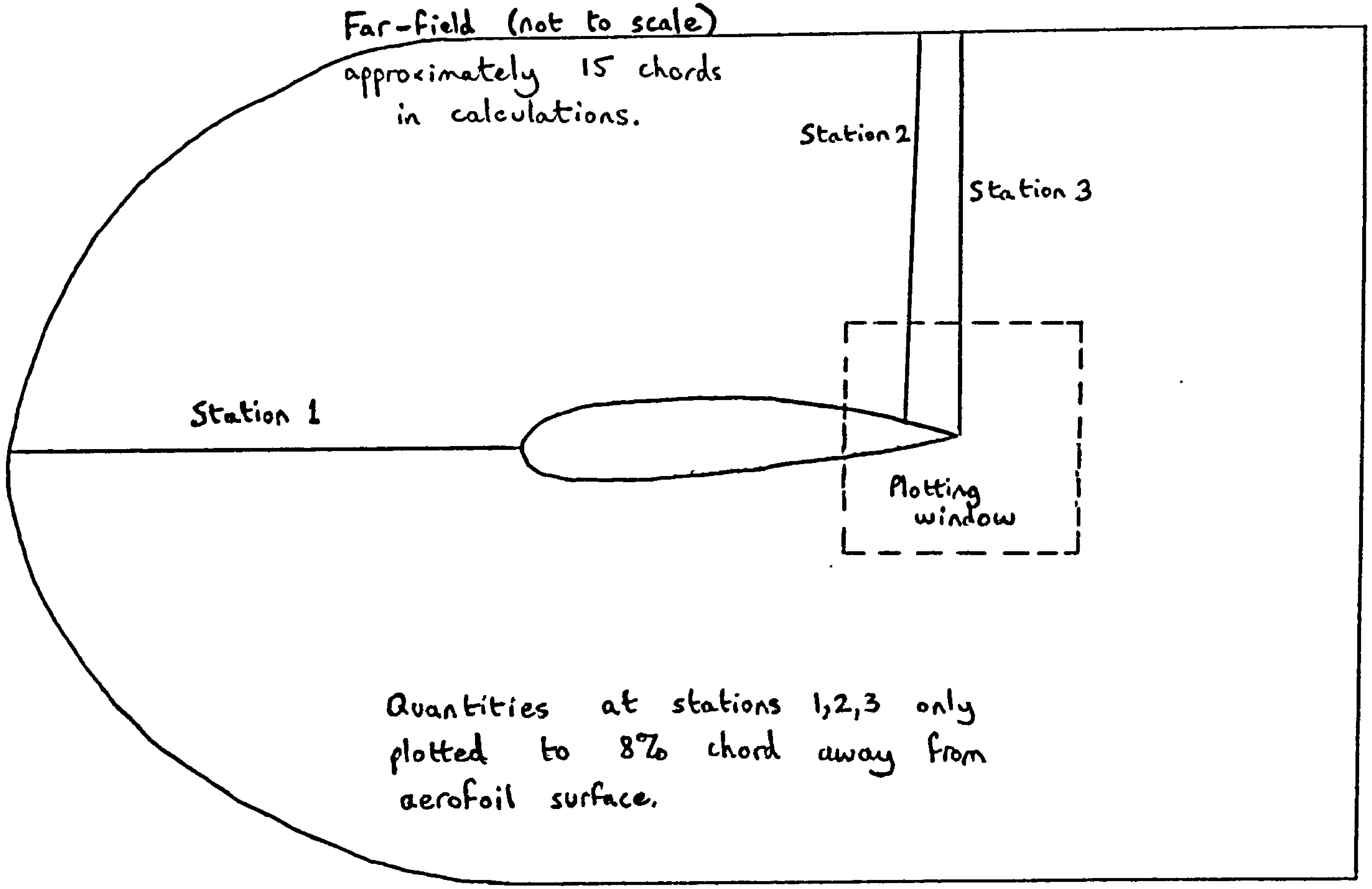


Fig.3.1 Sketch of grid extent and plotting locations.

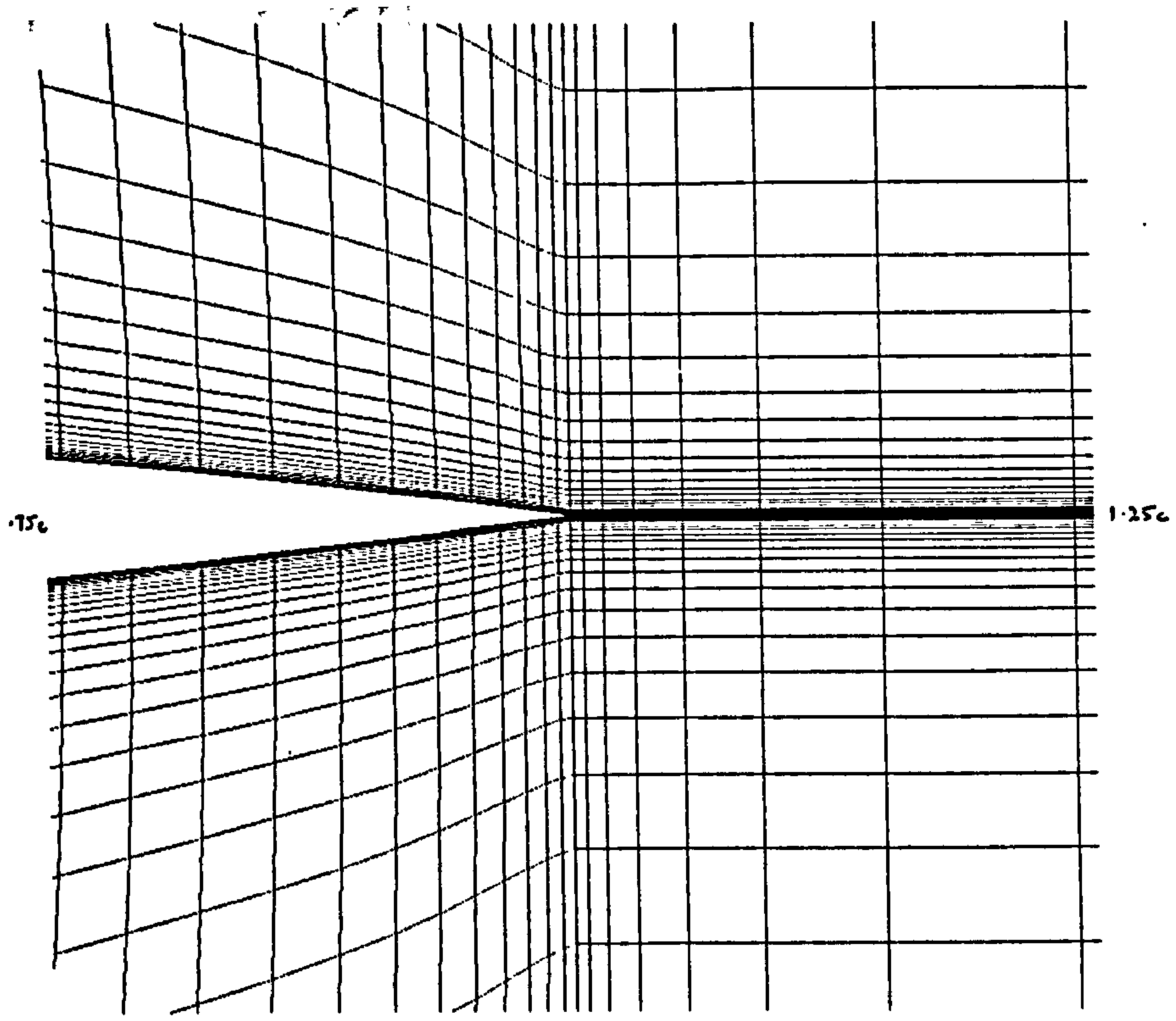


Fig.3.2 Grid A - trailing-edge region.

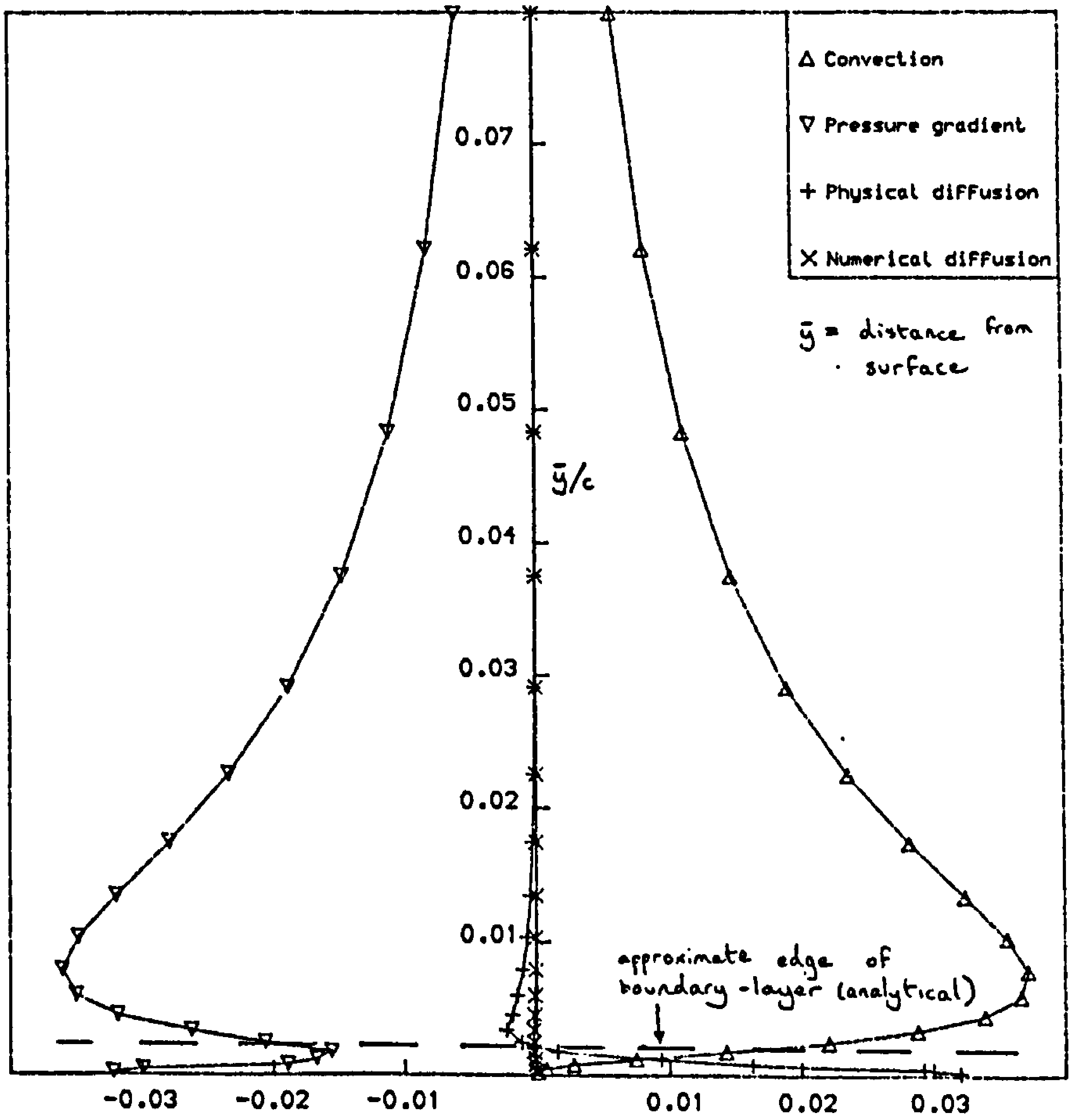


Fig.3.3 Balance of terms in solution - Grid A - station 1.

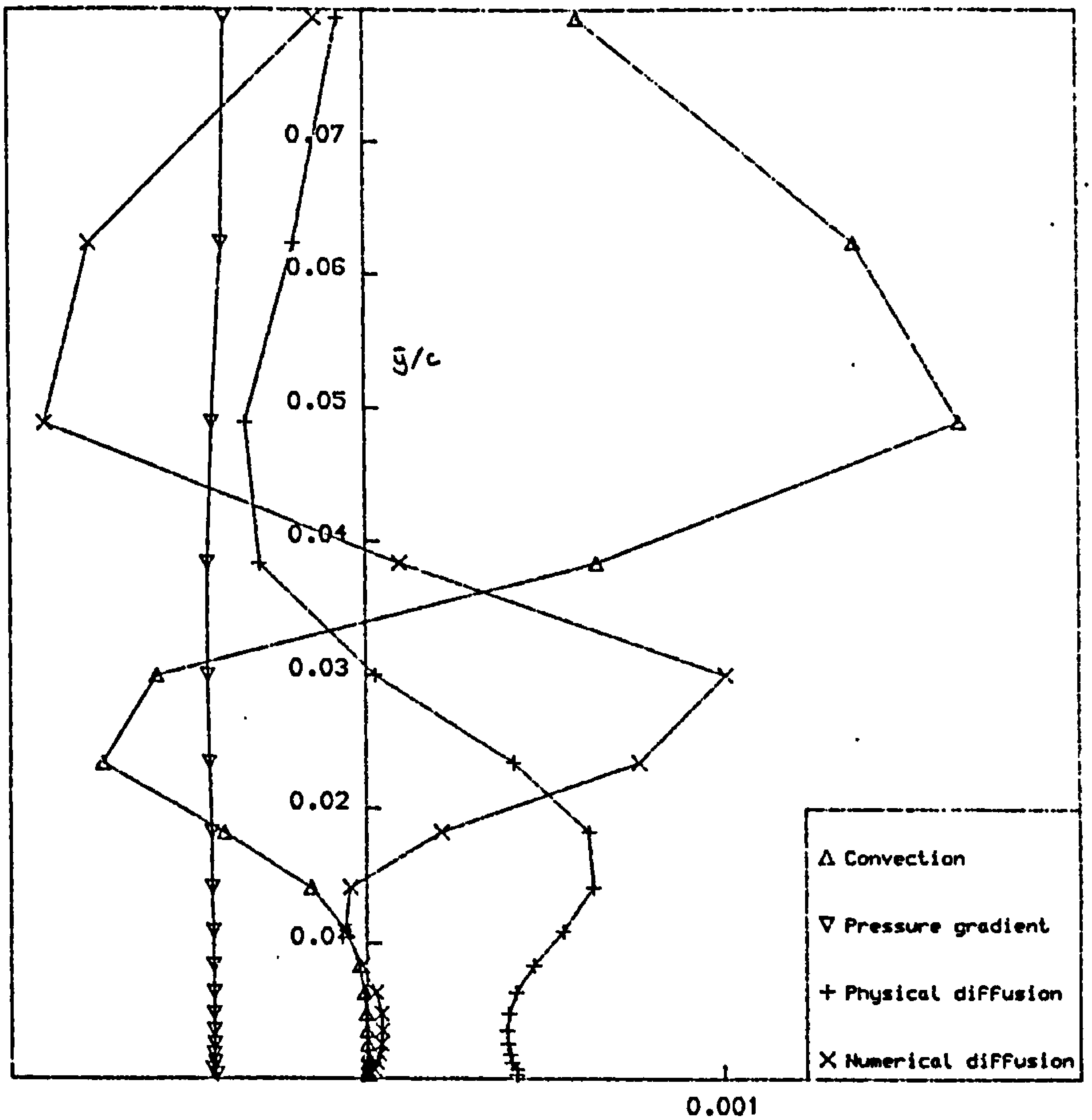


Fig.3.4 Balance of terms in solution - Grid A - station 2.

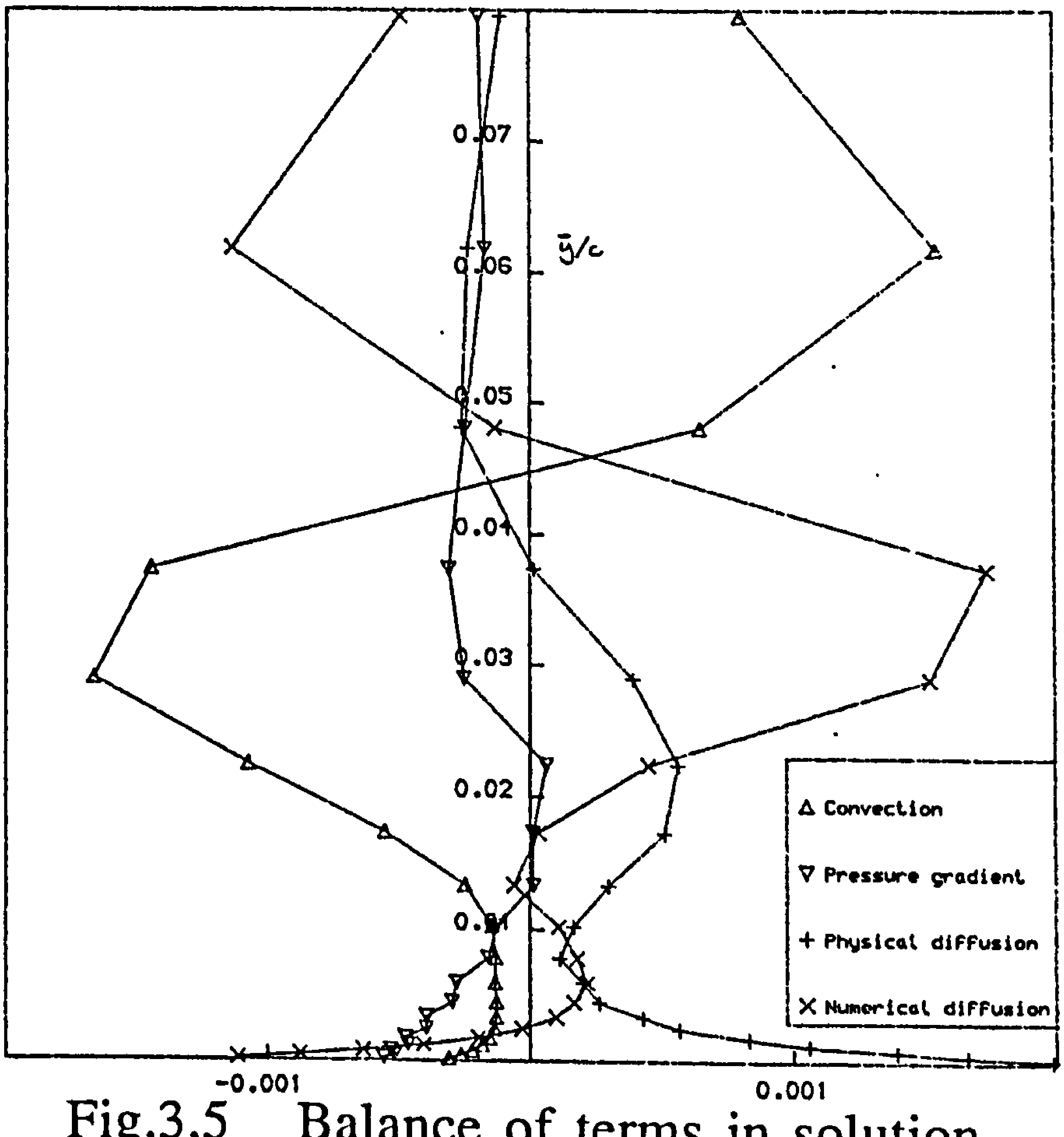


Fig.3.5 Balance of terms in solution - Grid A - station 3.

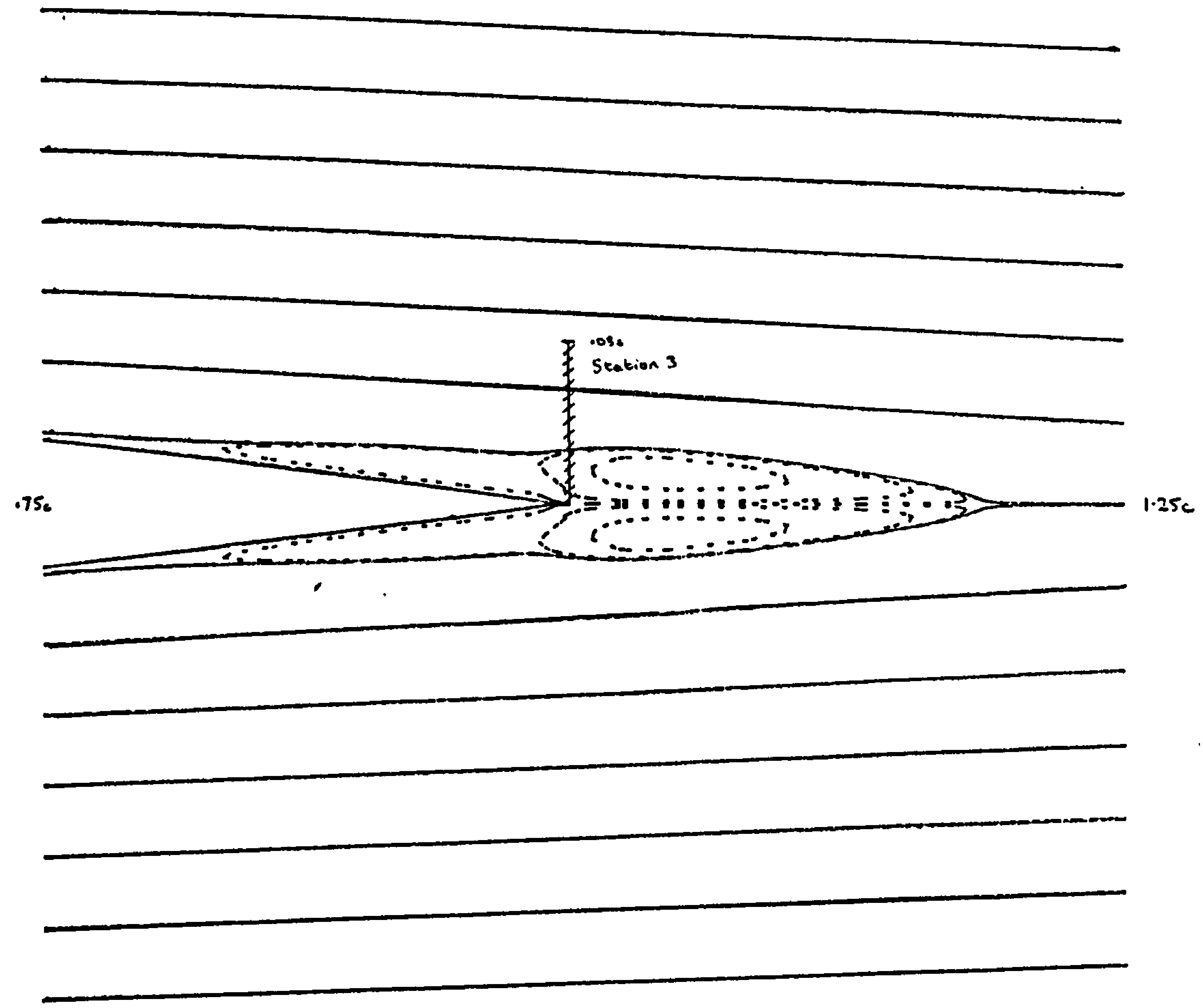


Fig.3.6 Particle paths - Grid A
- recirculation region.

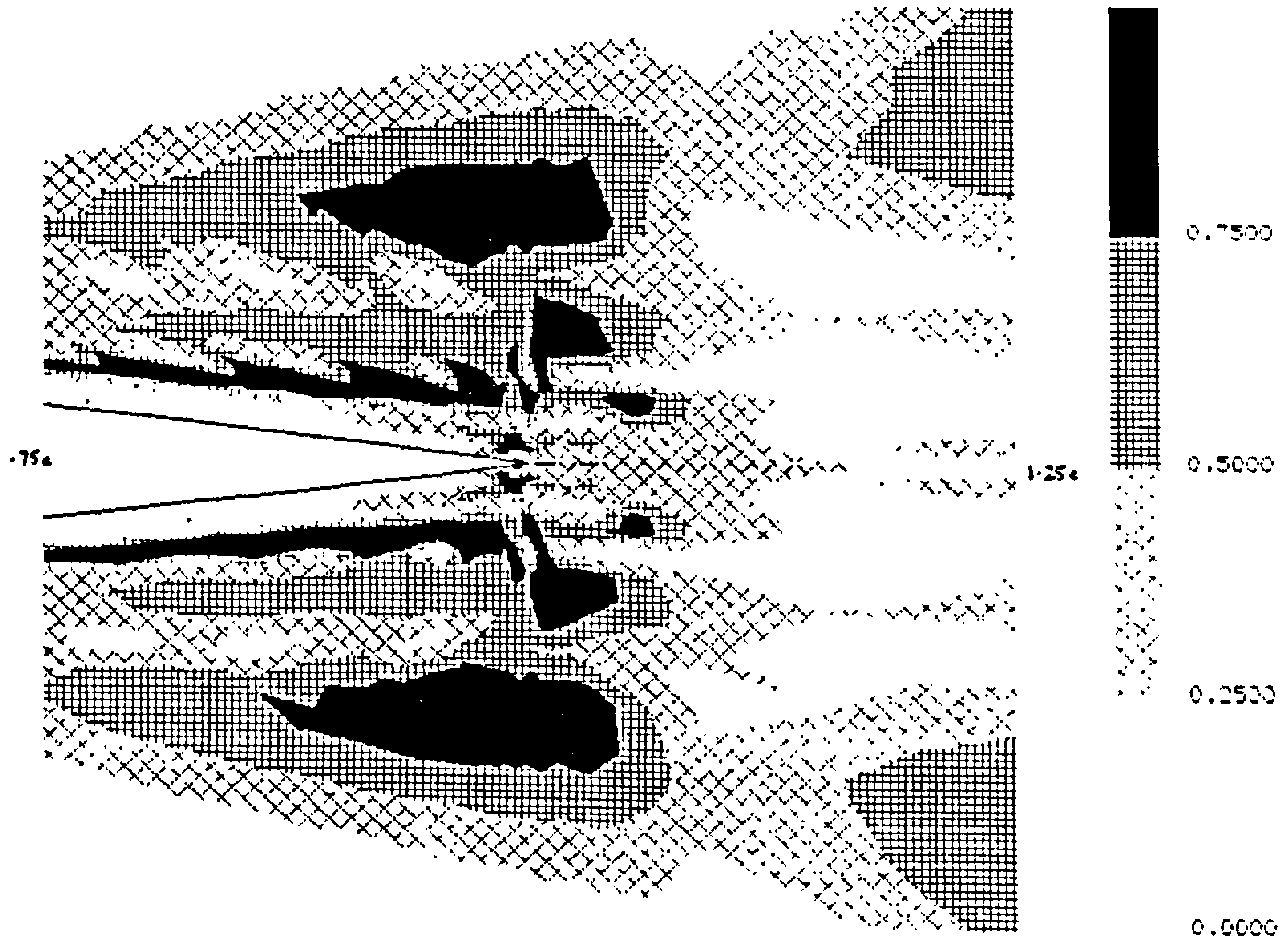


Fig.3.7 Grid A - NDR.

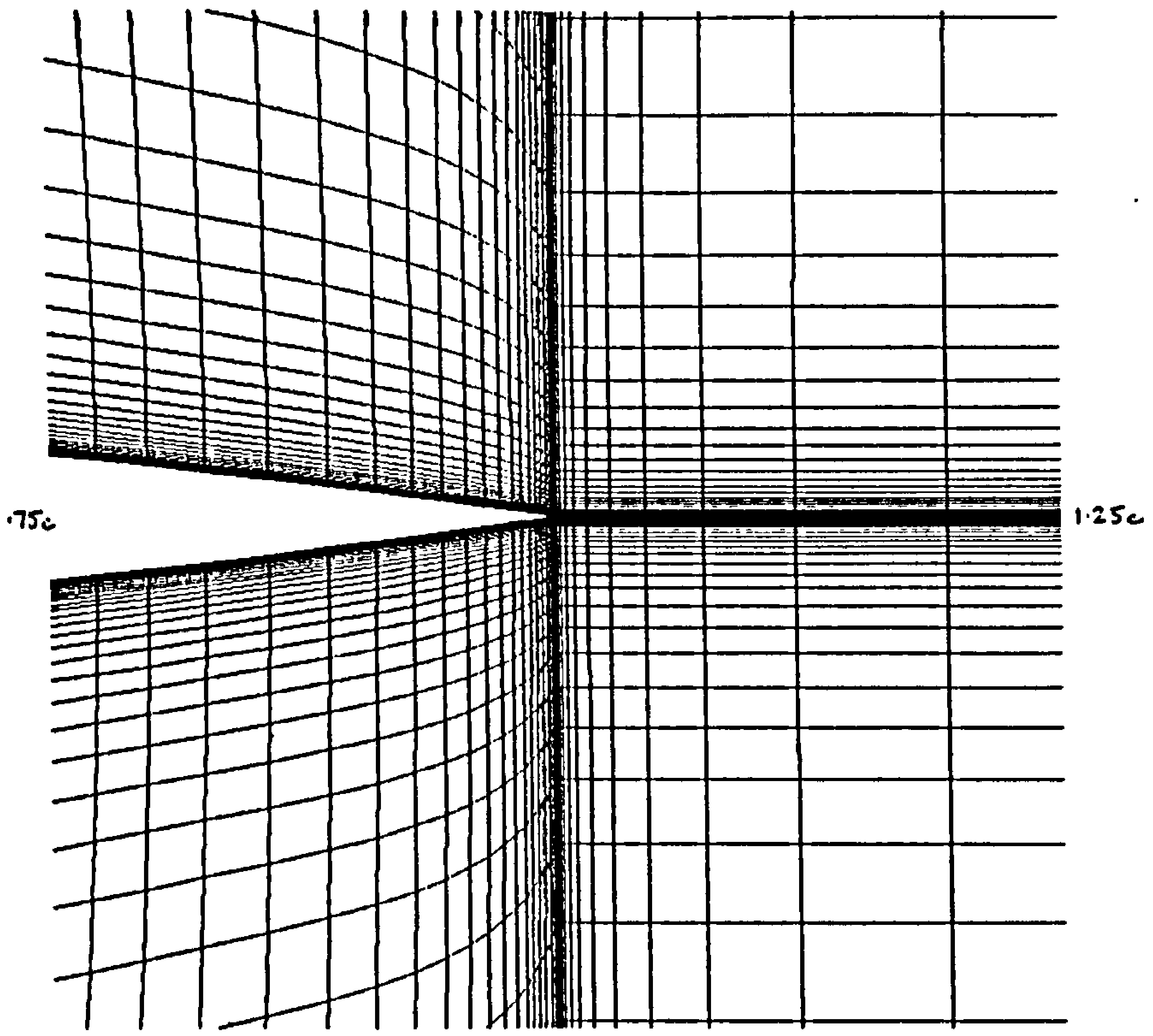


Fig.3.8 Grid B - trailing-edge region.

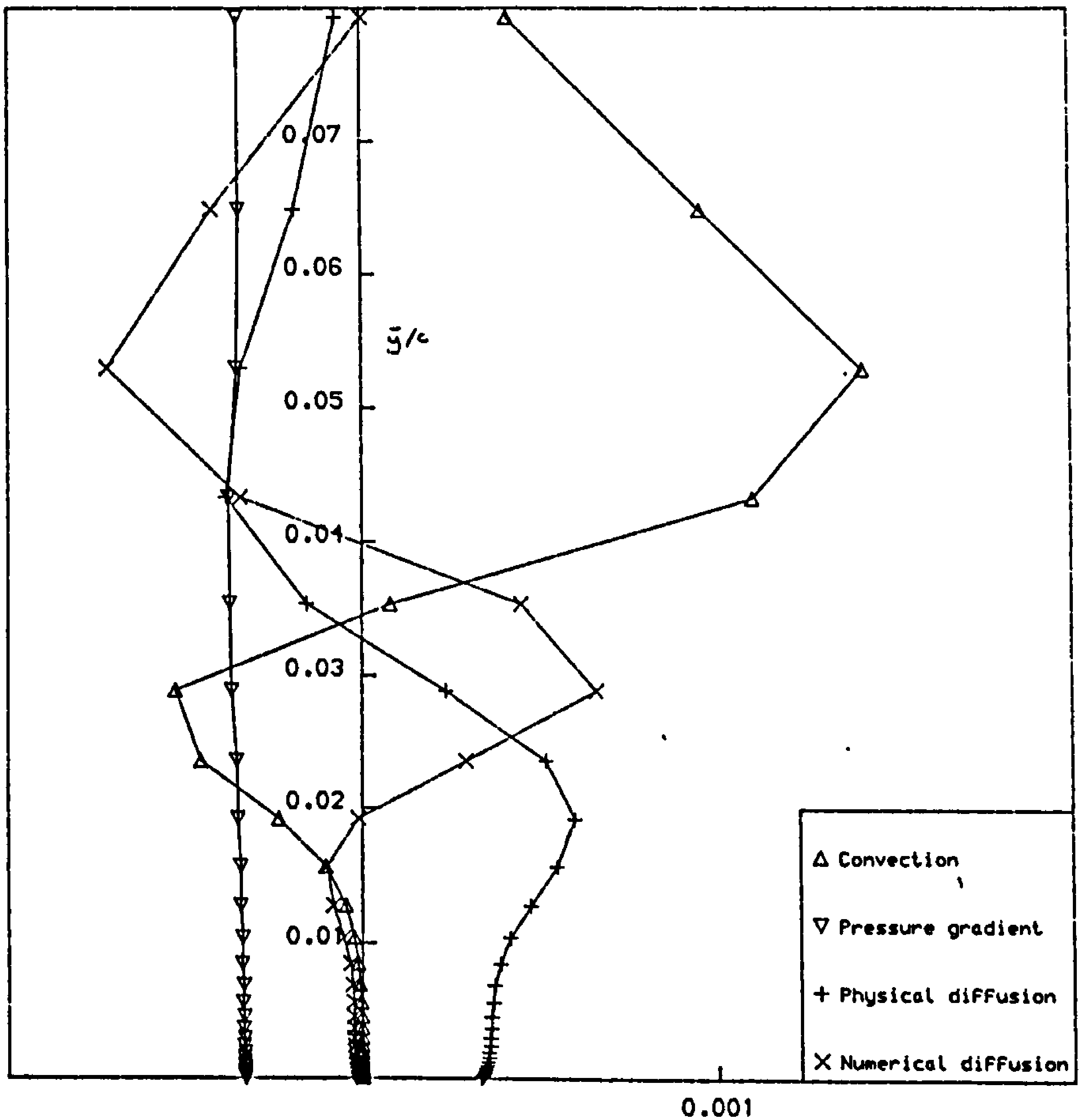


Fig.3.9 Balance of terms in solution
- Grid B - station 2.

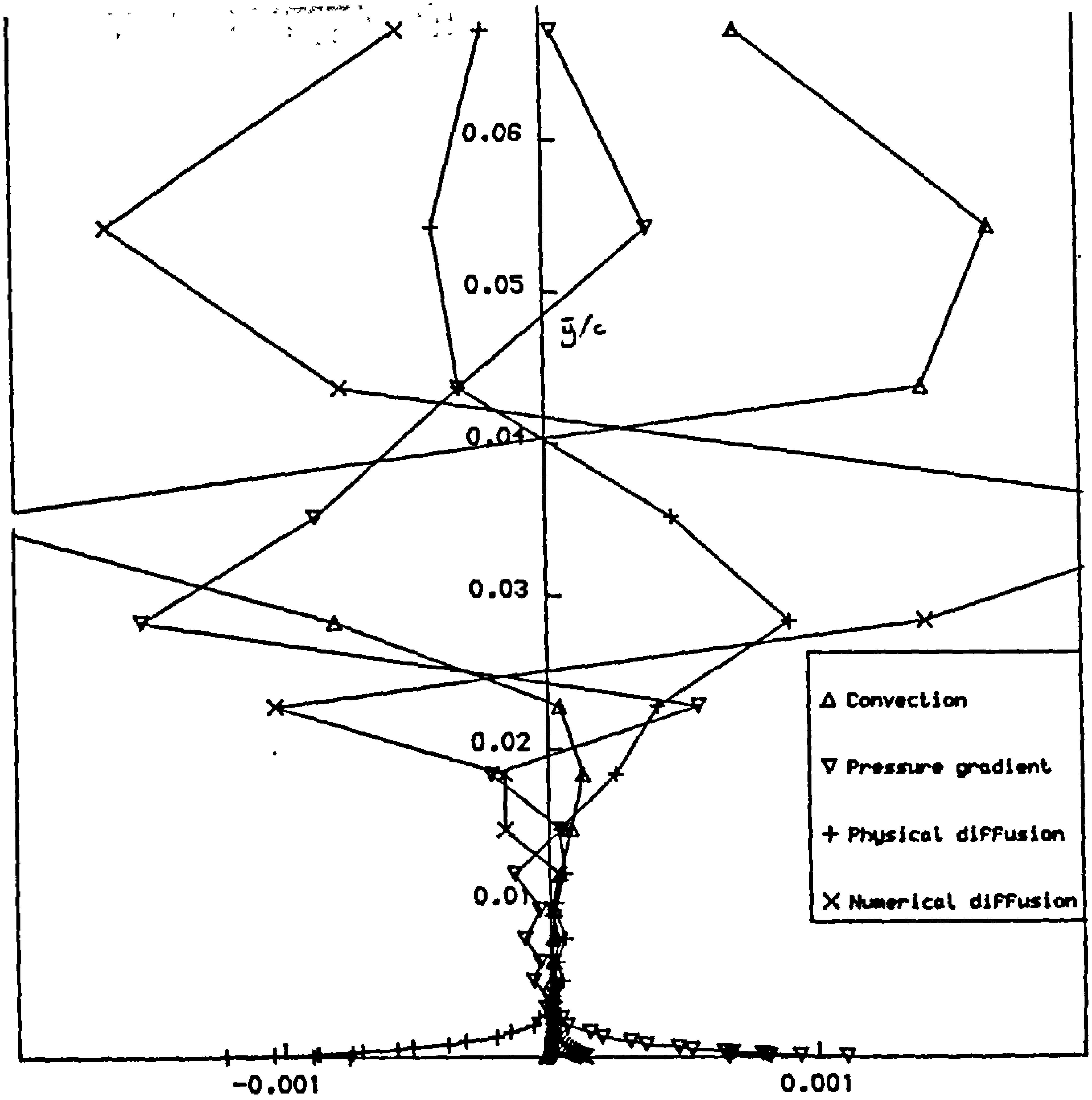


Fig.3.10 Balance of terms in solution
- Grid B - station 3.

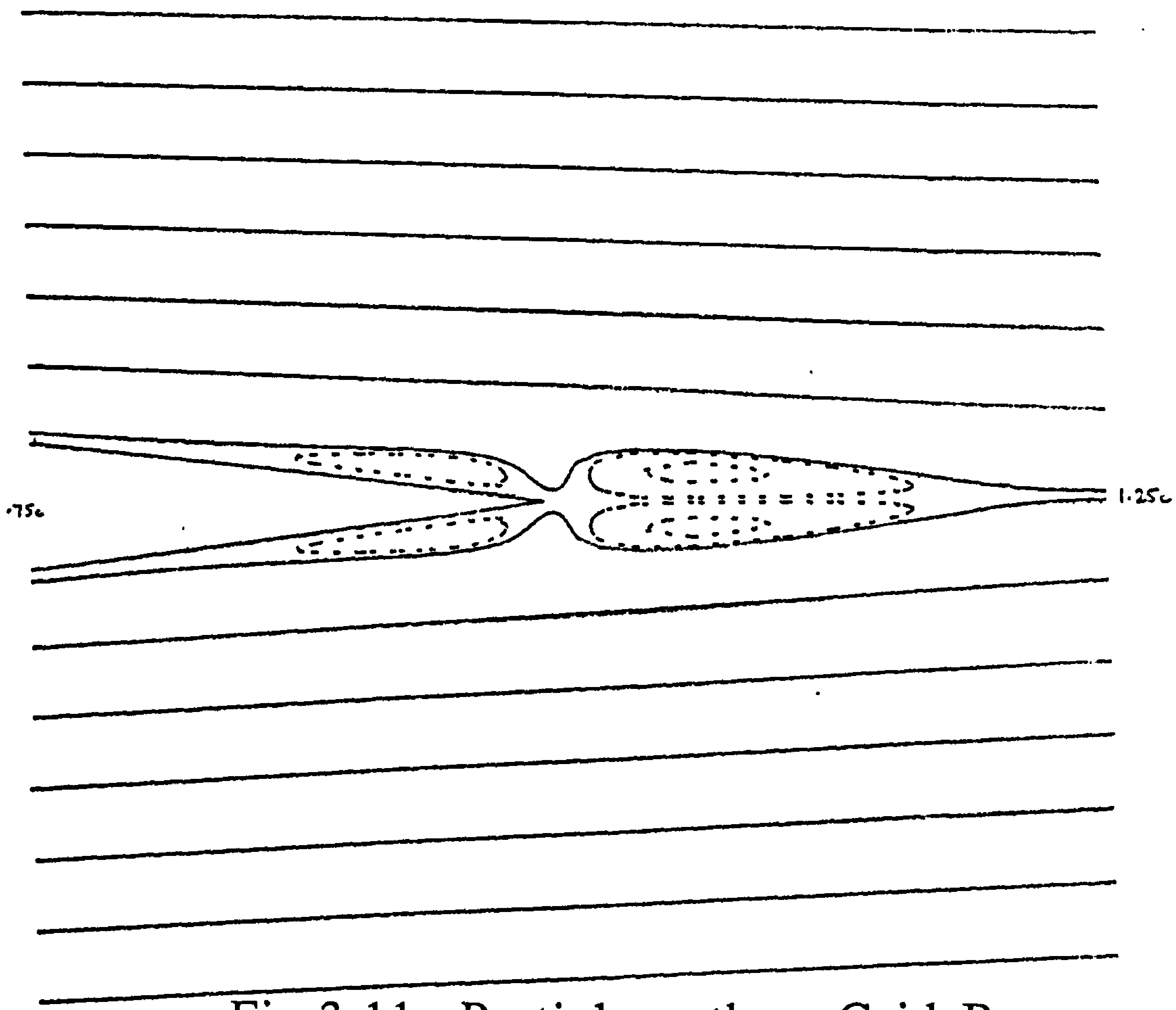


Fig.3.11 Particle paths - Grid B
- distinct recirculation regions.

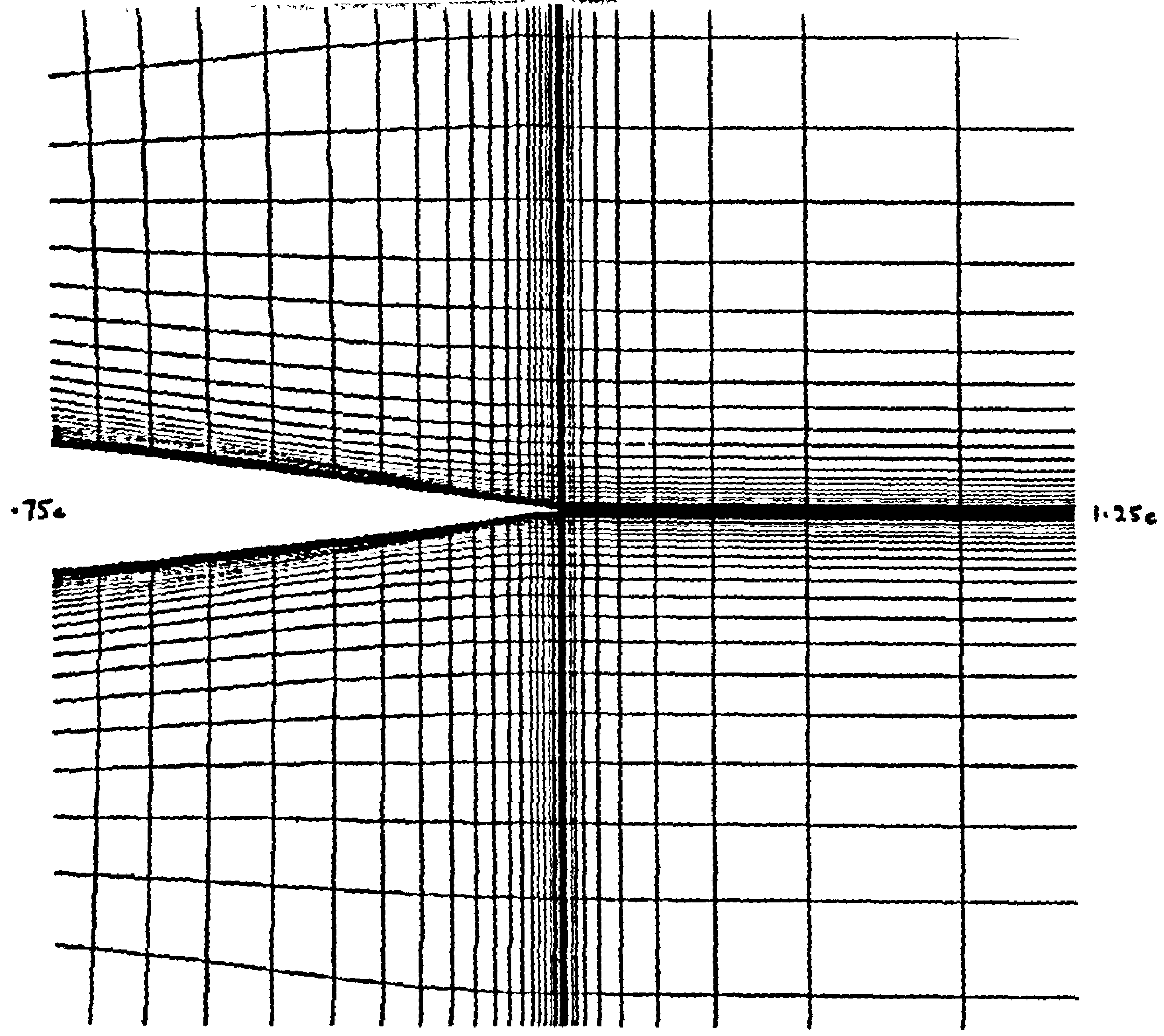


Fig.3.12 Grid C - stretched to reduce flow/gridline angle.

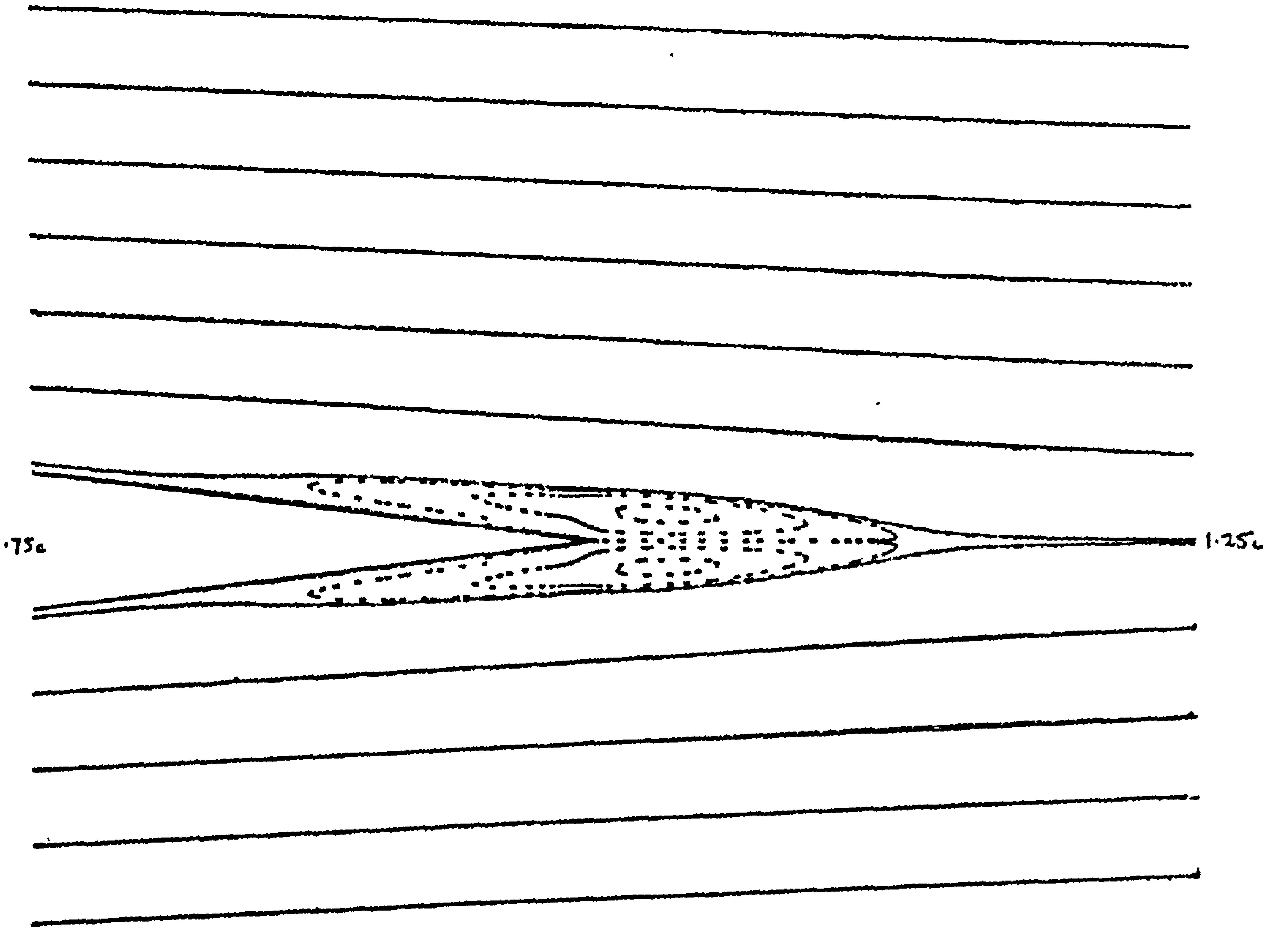


Fig.3.13 Particle paths - Grid C - recirculation region.

DAMAGED
TEXT
IN
ORIGINAL

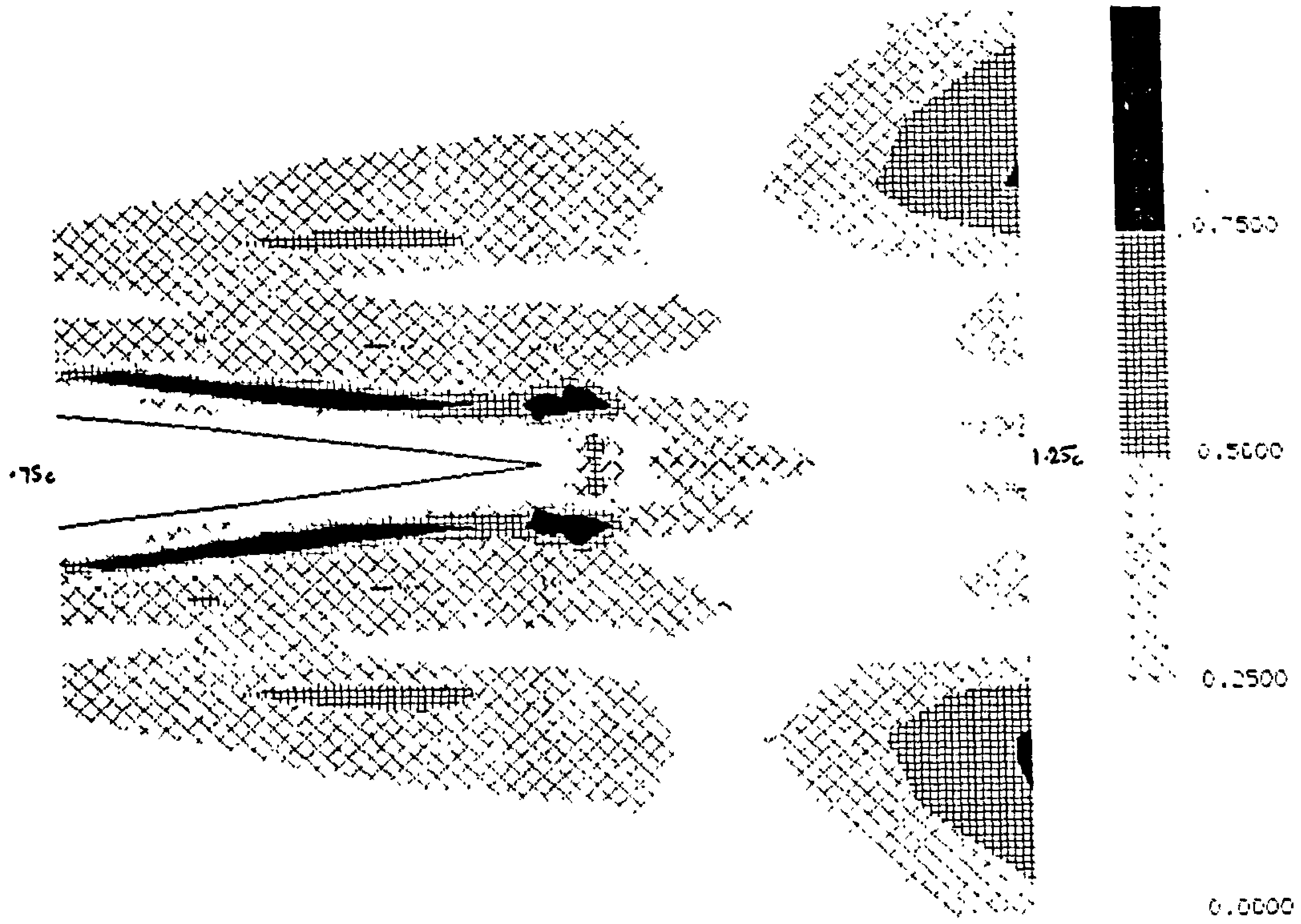


Fig.3.14 Grid C - NDR.

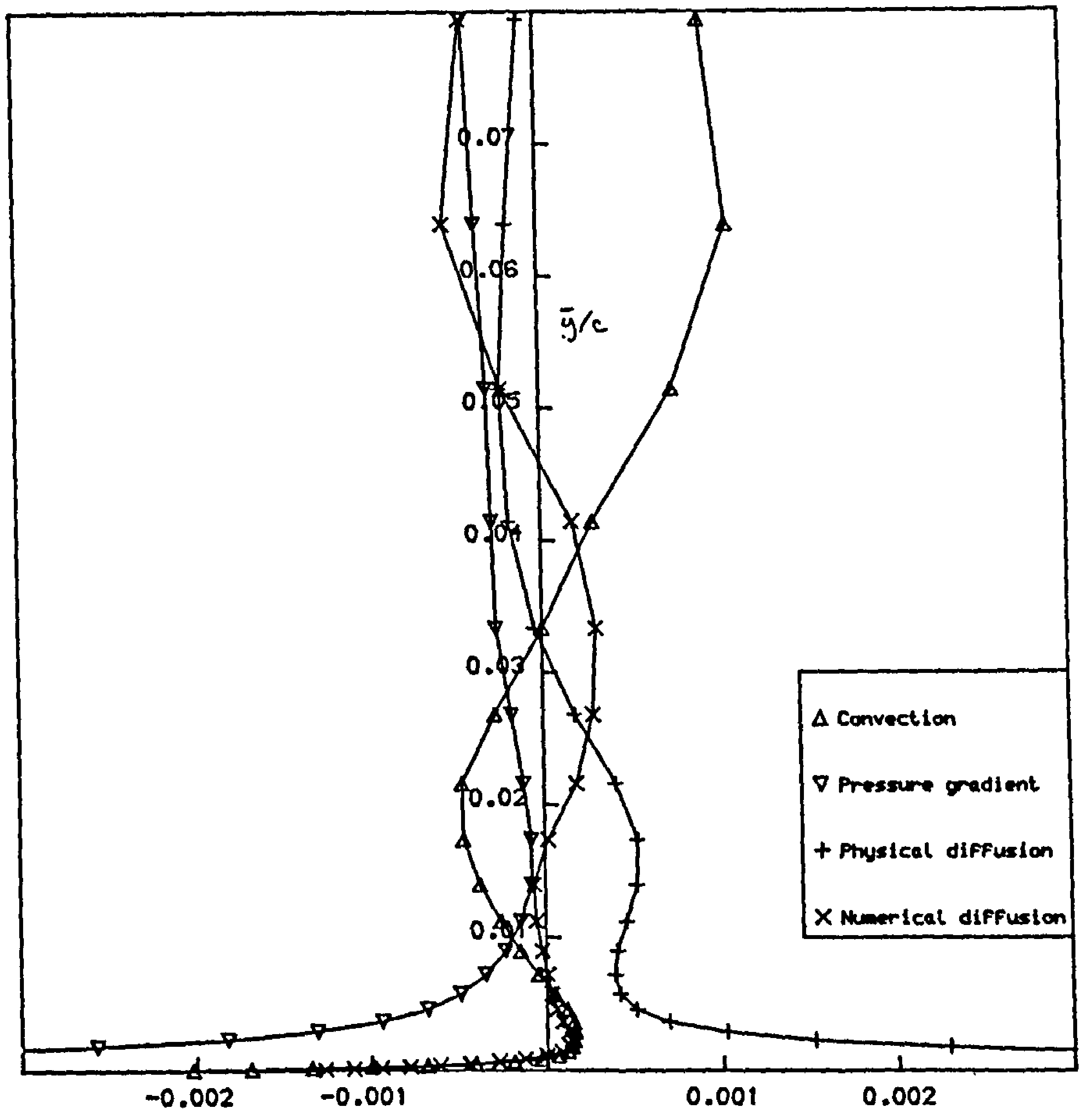


Fig.3.15 Balance of terms in solution - Grid C - station 3.

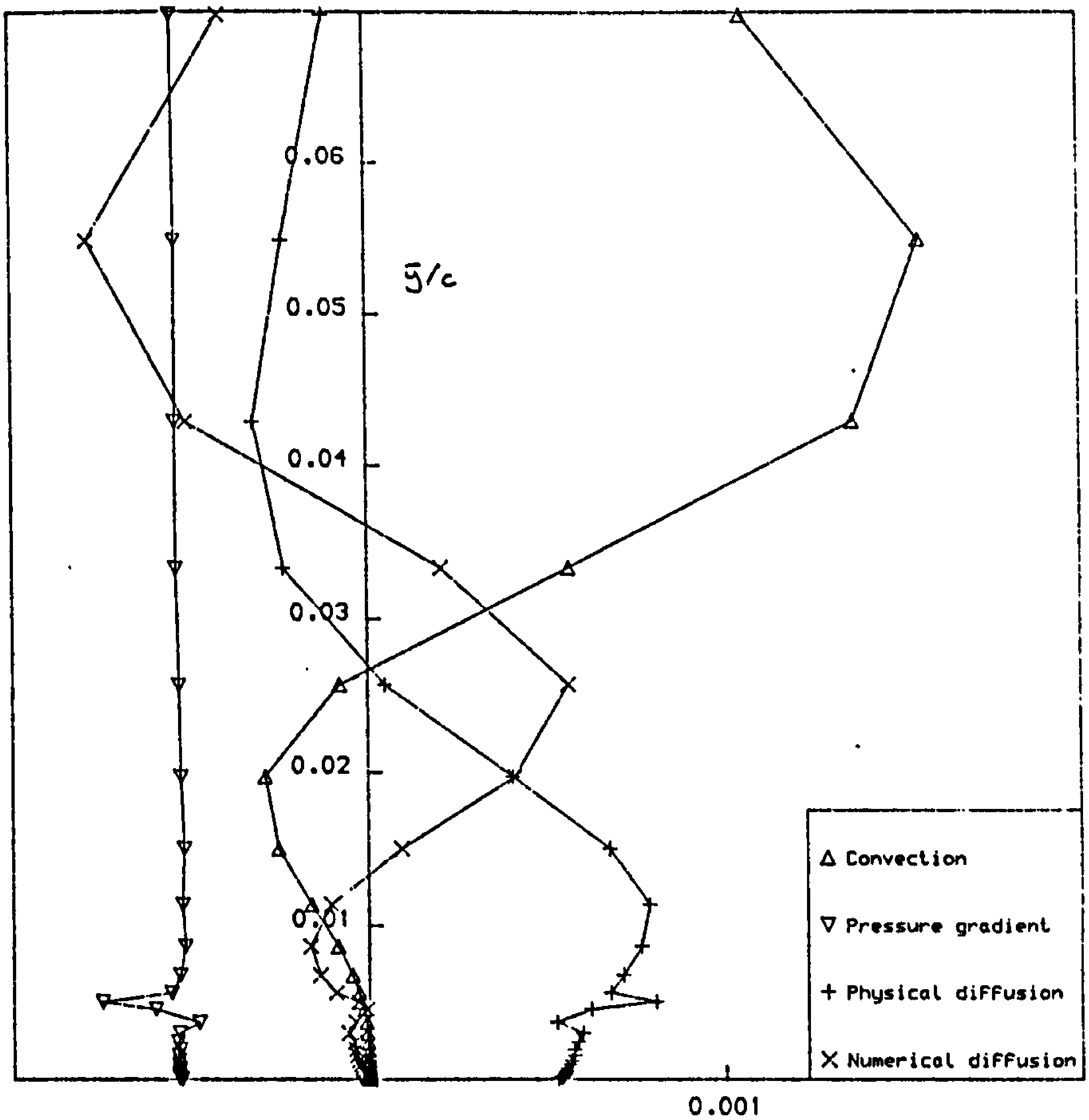


Fig.3.16 Balance of terms in solution - Grid C - station 2.

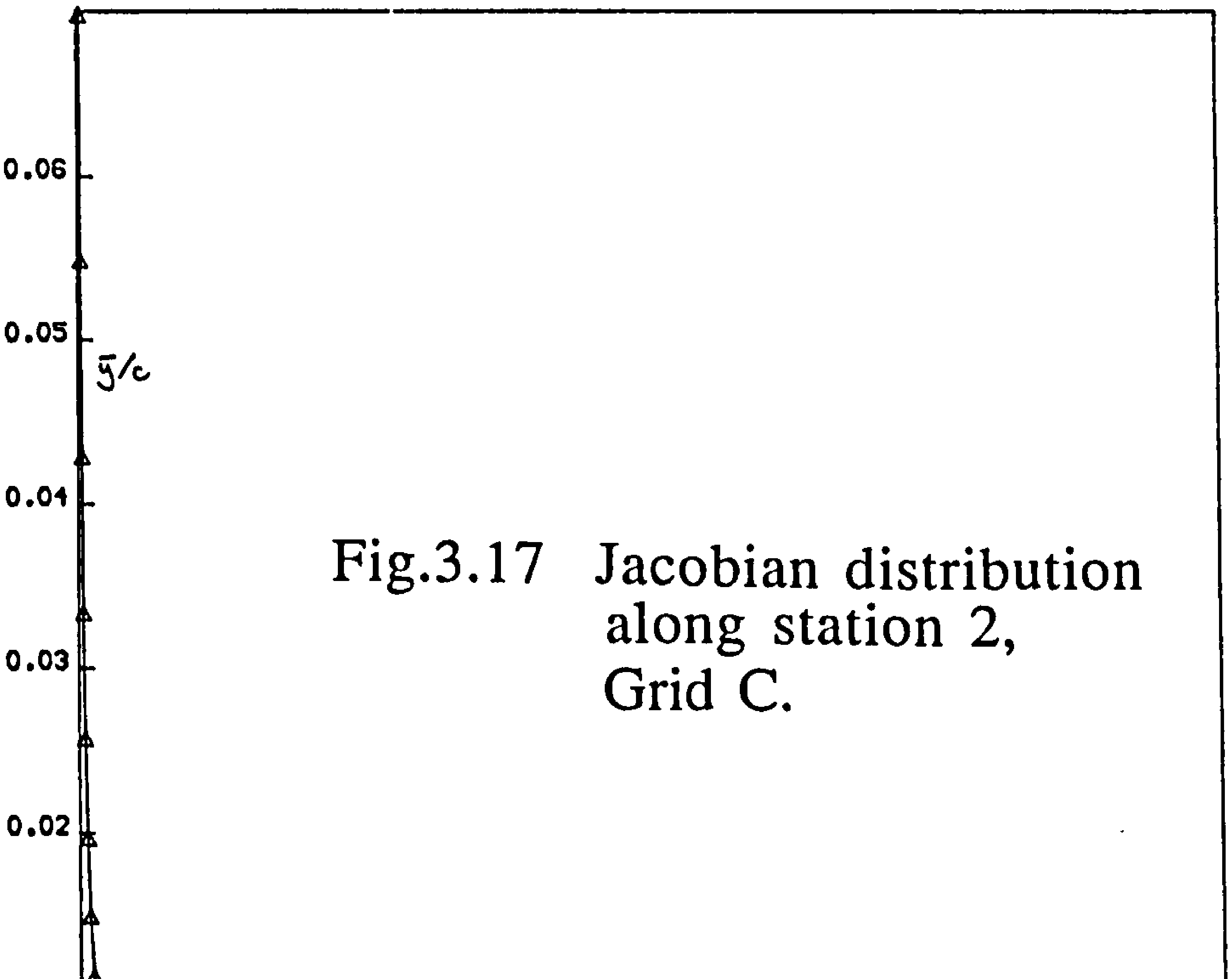


Fig.3.17 Jacobian distribution along station 2, Grid C.

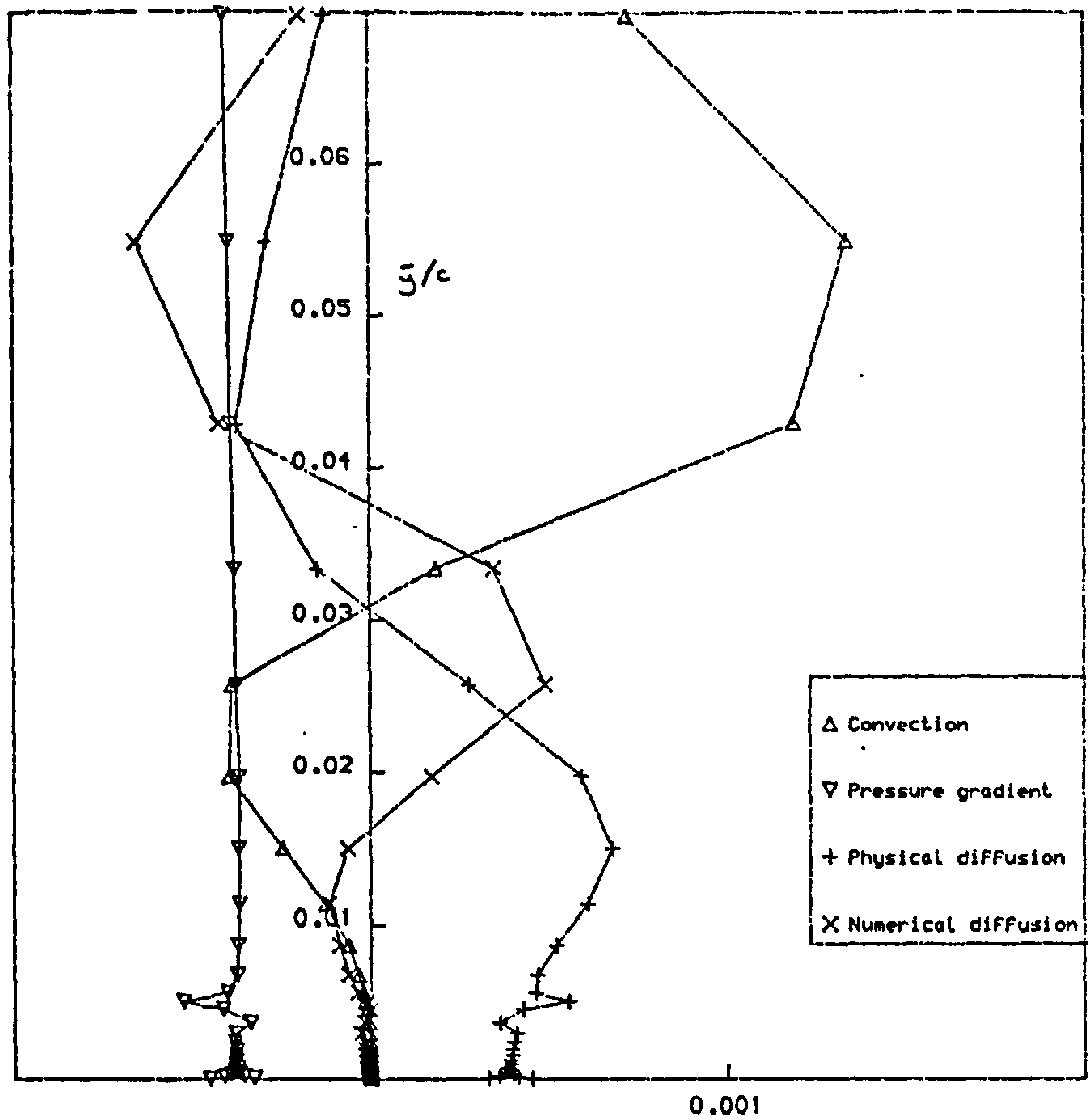


Fig.3.18 Balance of terms in solution - Case C(i) - station 2.

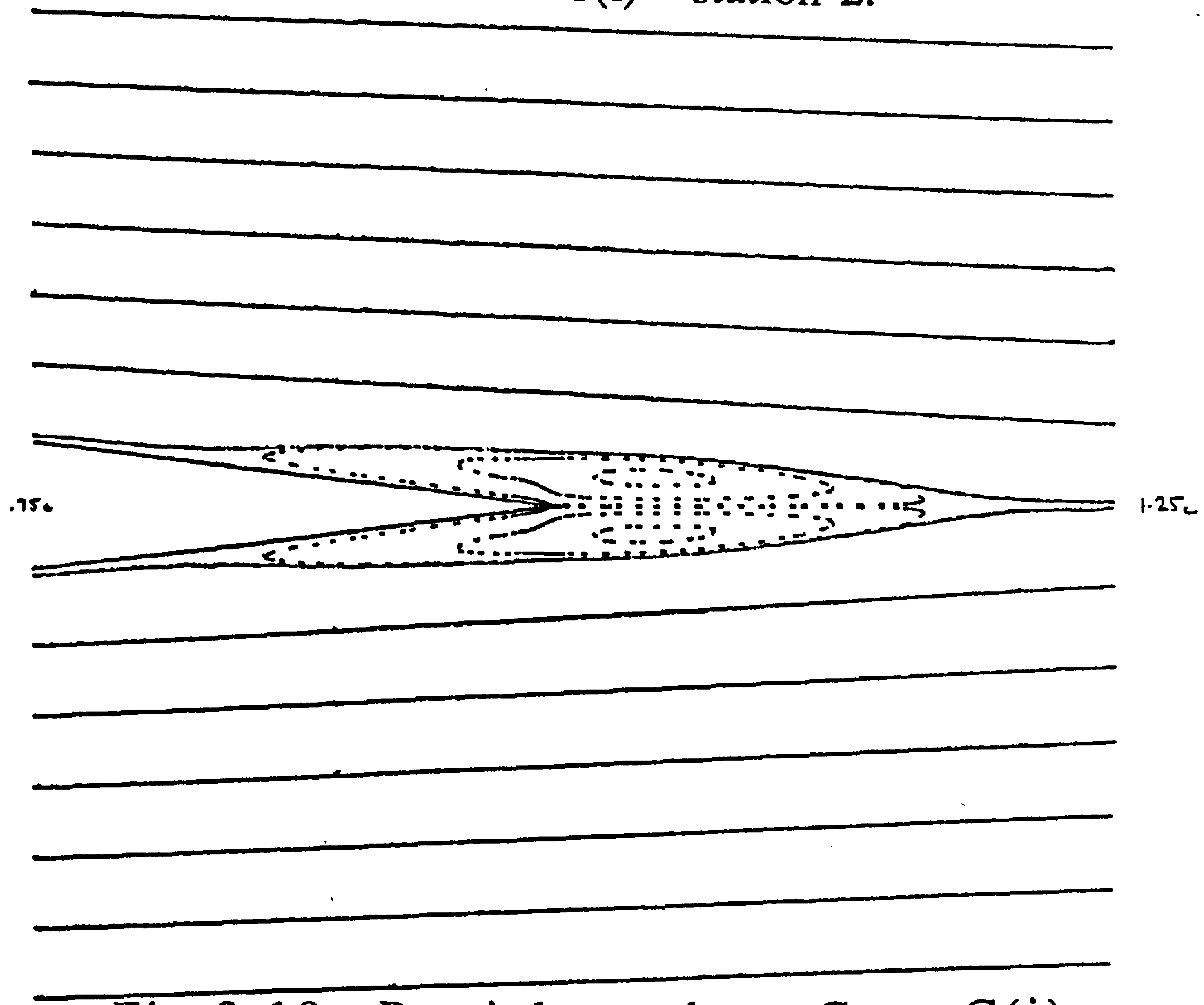


Fig.3.19 Particle paths - Case C(i) - recirculation region.

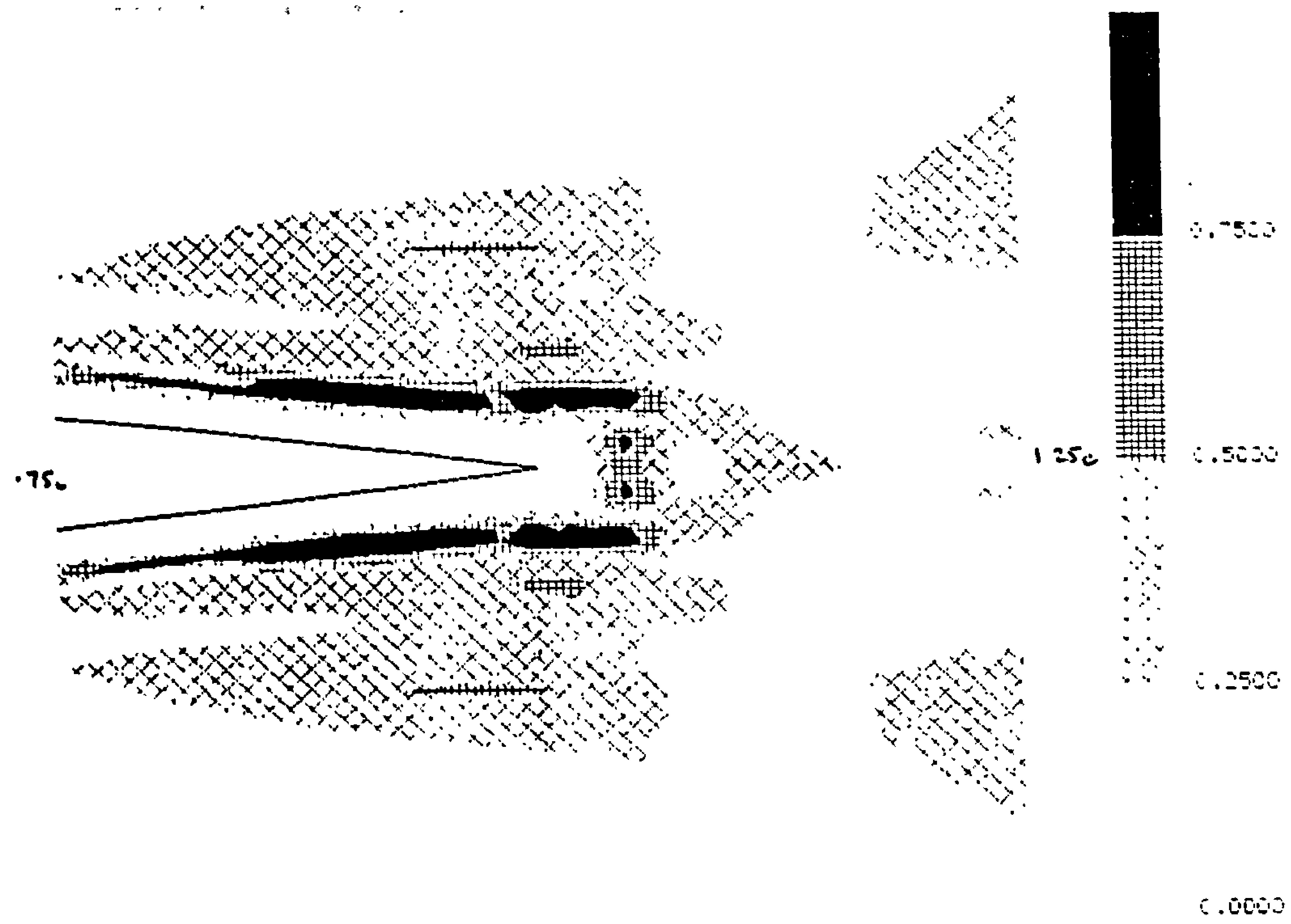


Fig.3.20 Case C(i) - NDR.

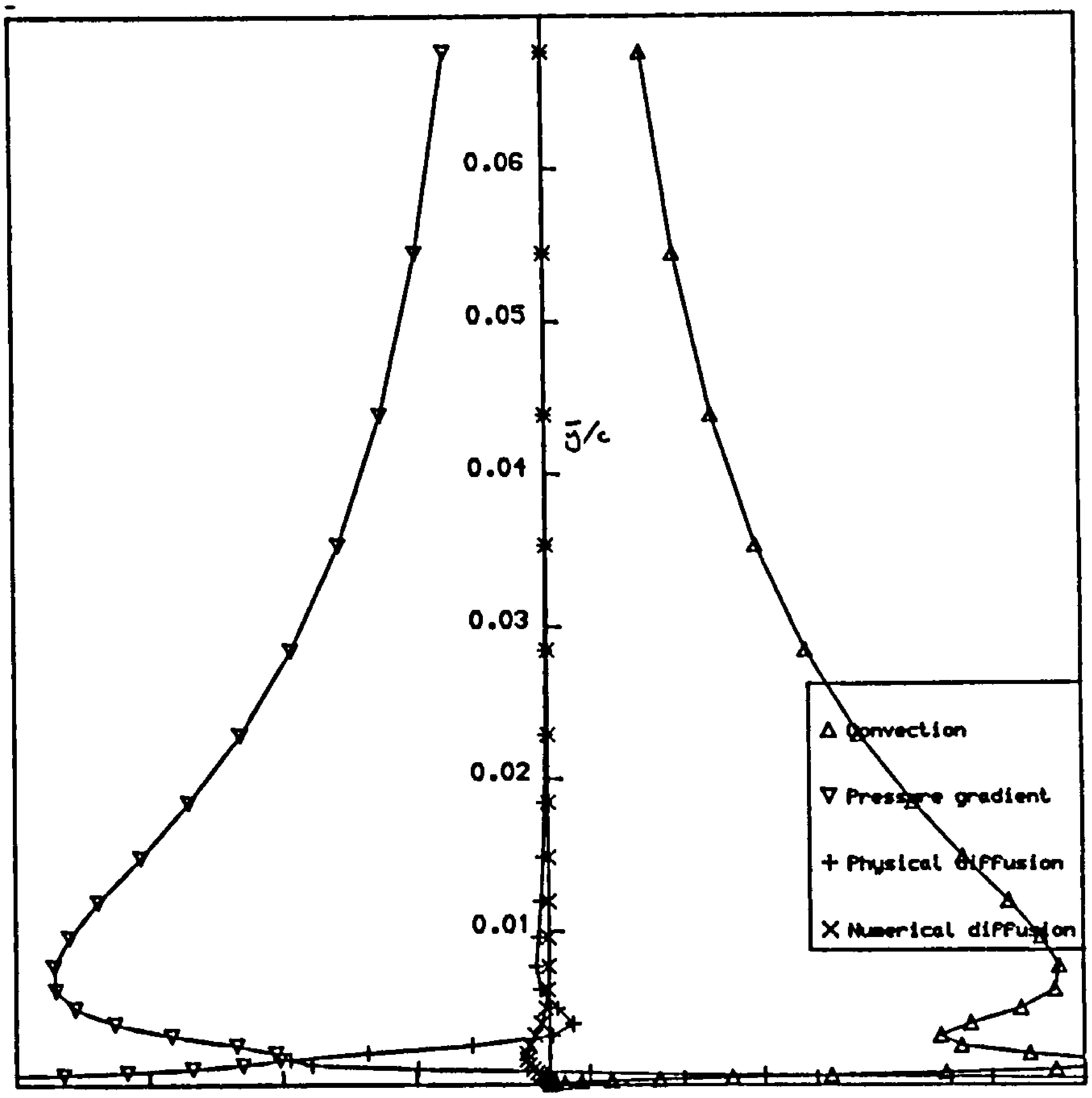


Fig.3.21 Balance of terms in solution - Case C(ii) - station 1.

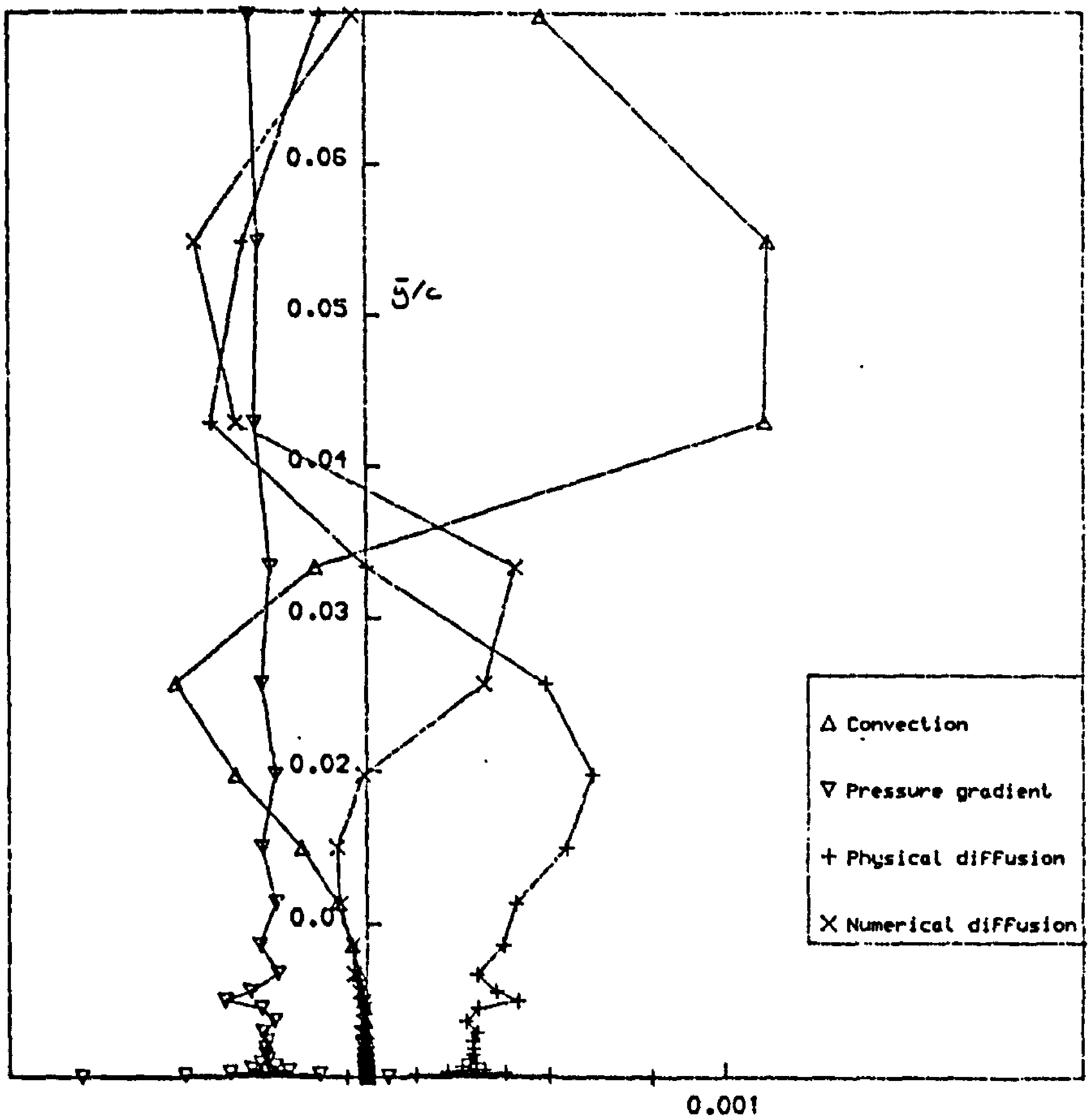


Fig.3.22 Balance of terms in solution - Case C(ii) - station 2.

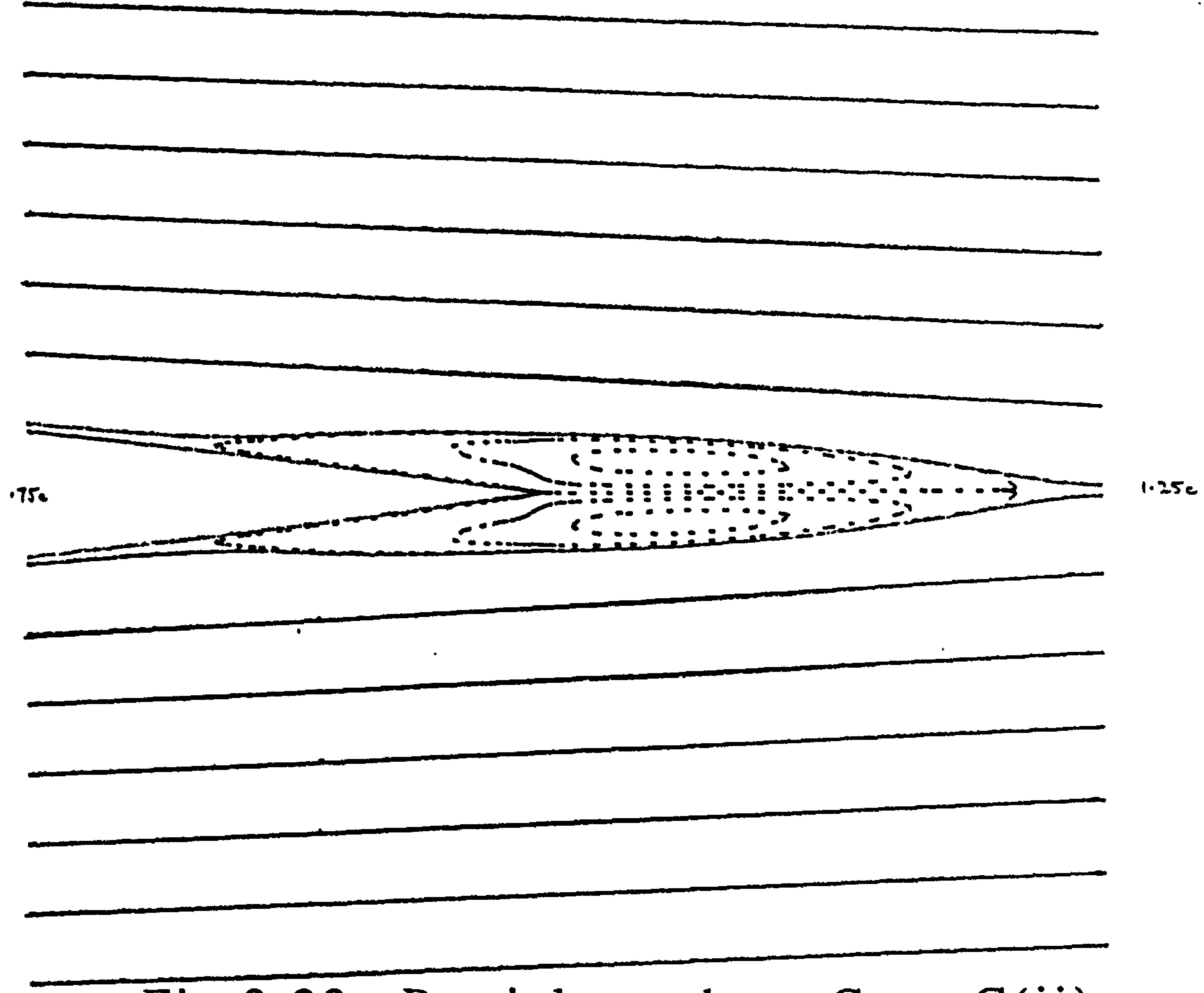


Fig.3.23 Particle paths - Case C(ii) - recirculation region.

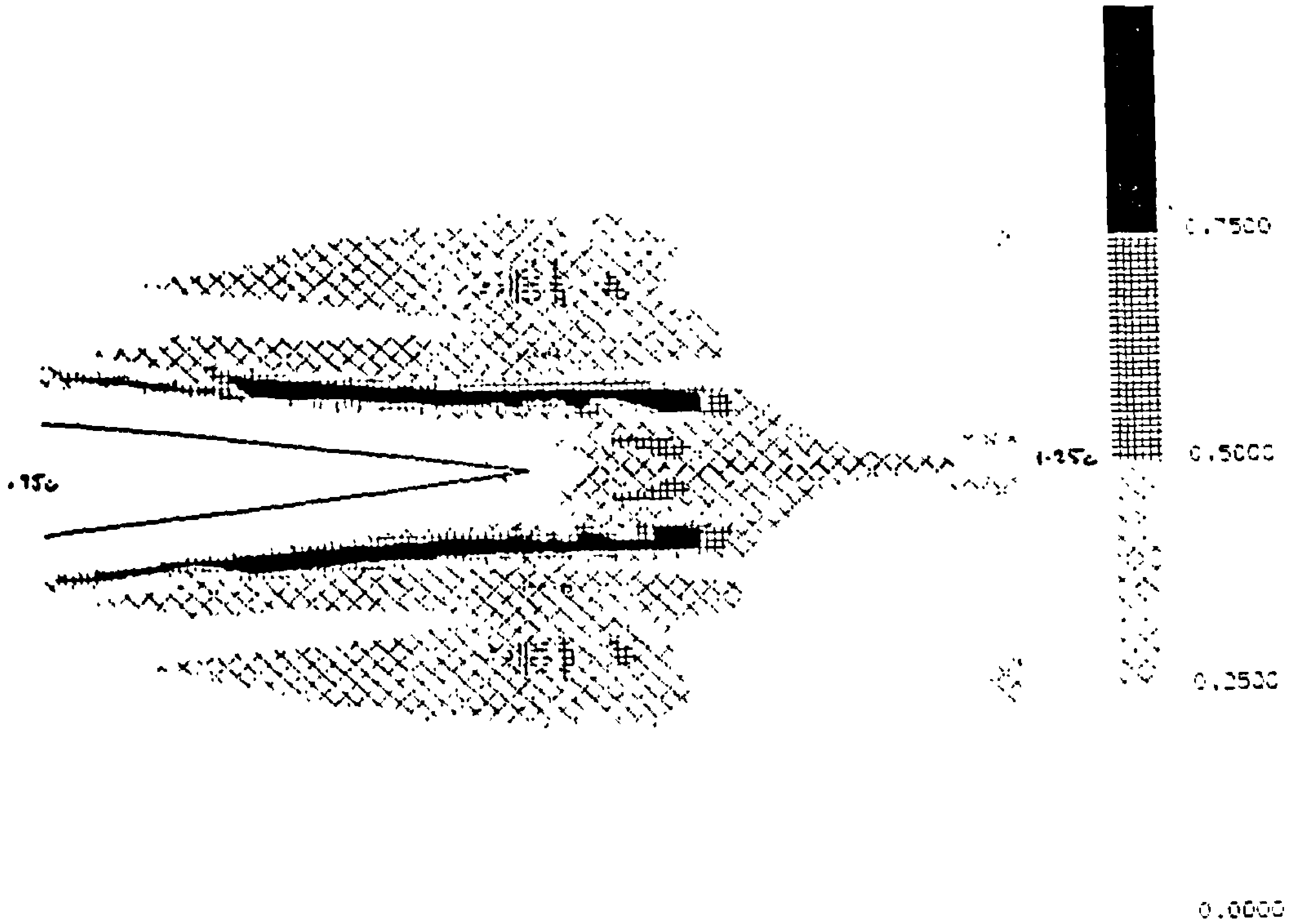


Fig.3.24 Case C(ii) - NDR.

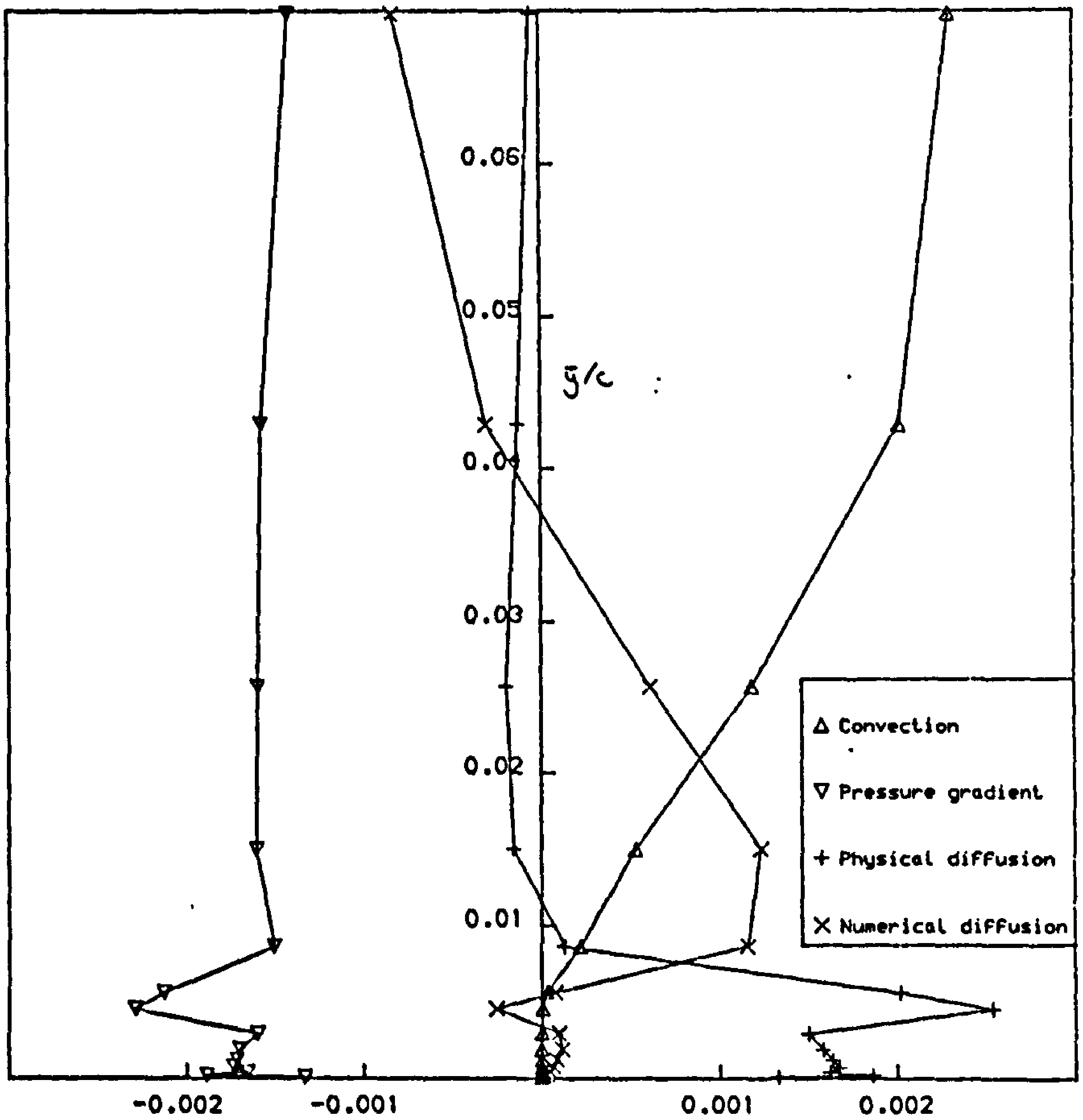


Fig.3.25 Balance of terms in solution - Grid D - station 2.

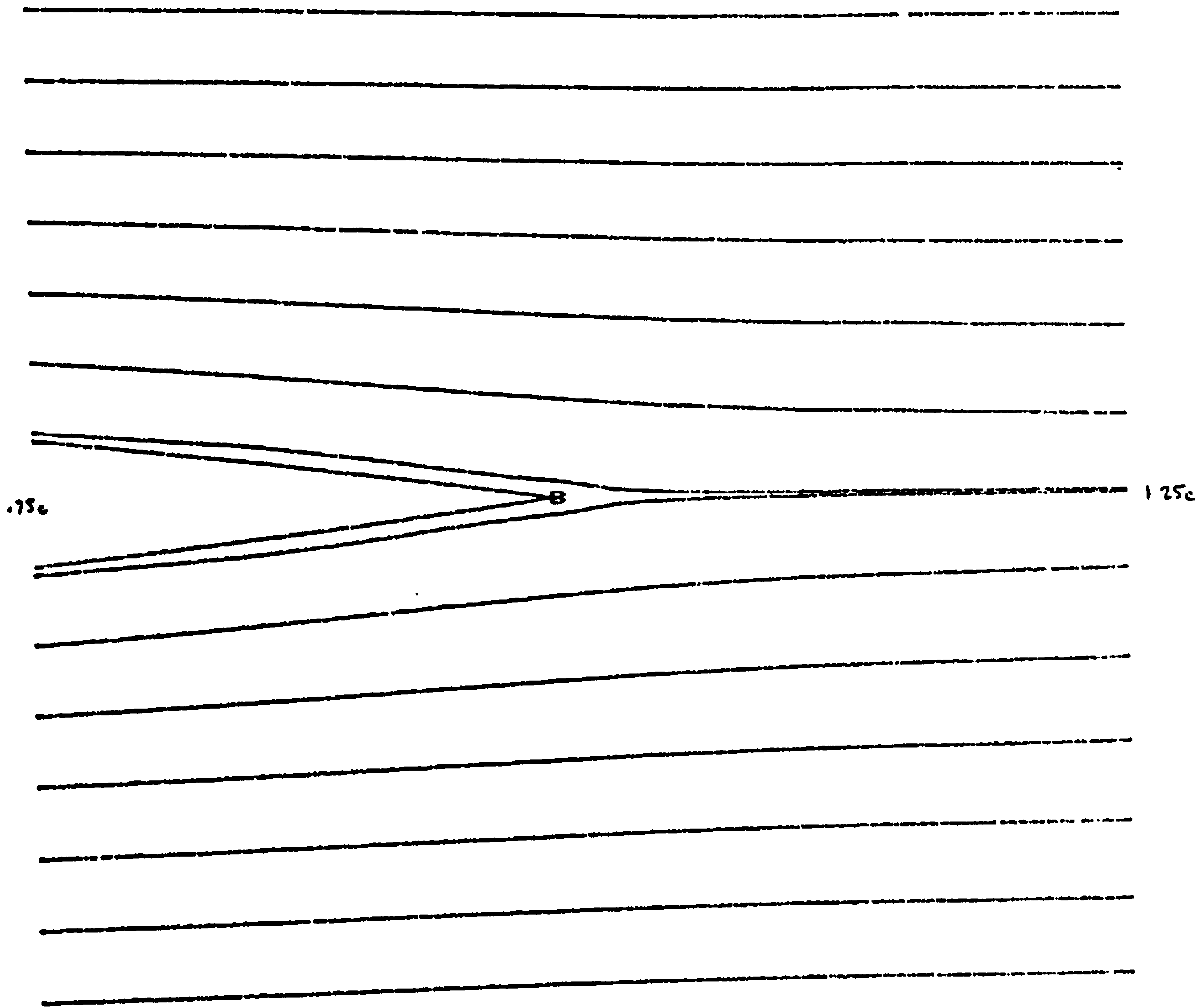


Fig.3.26 Particle paths - Grid D
- some recirculation present.

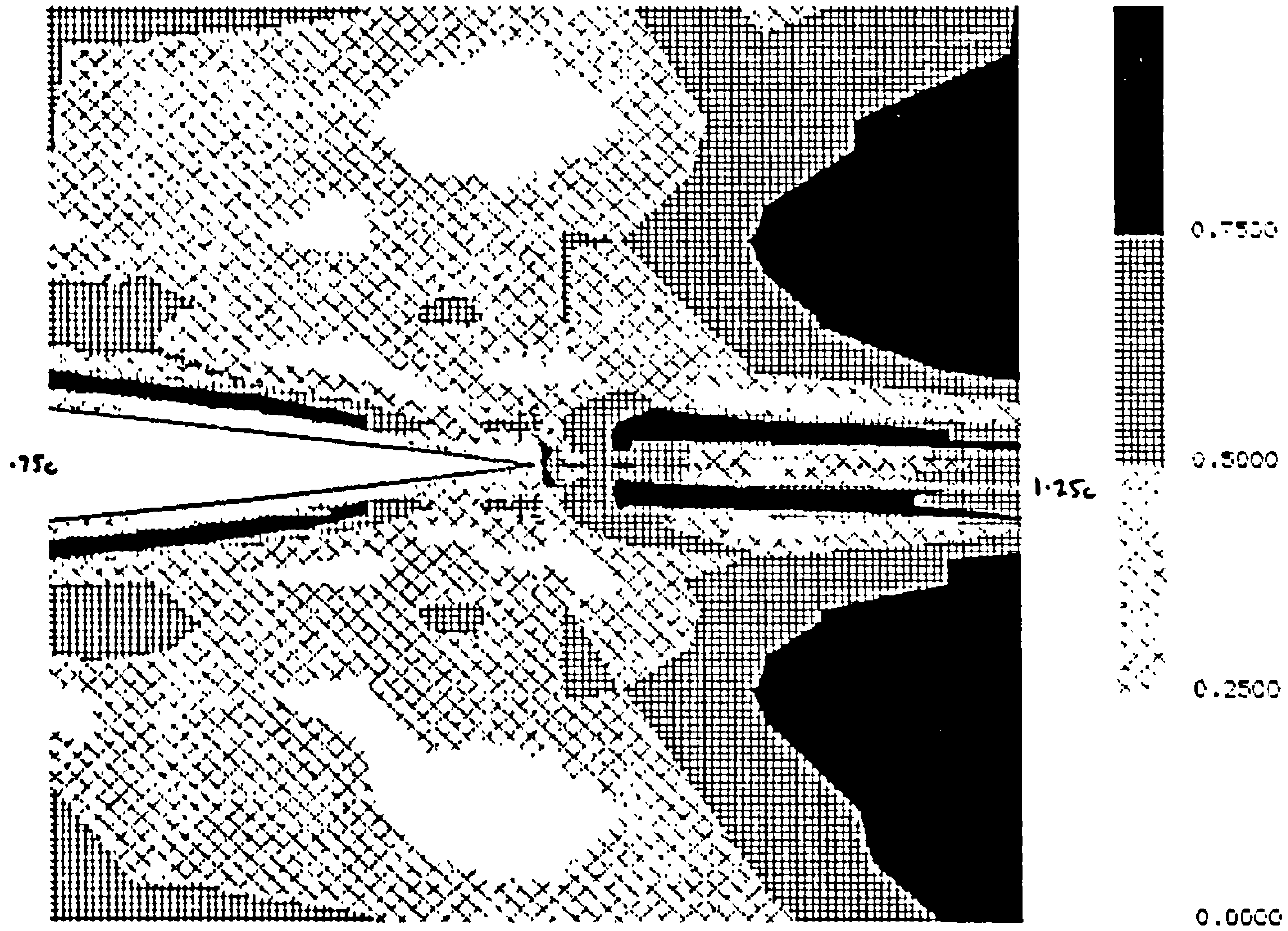


Fig.3.27 Grid D - NDR.

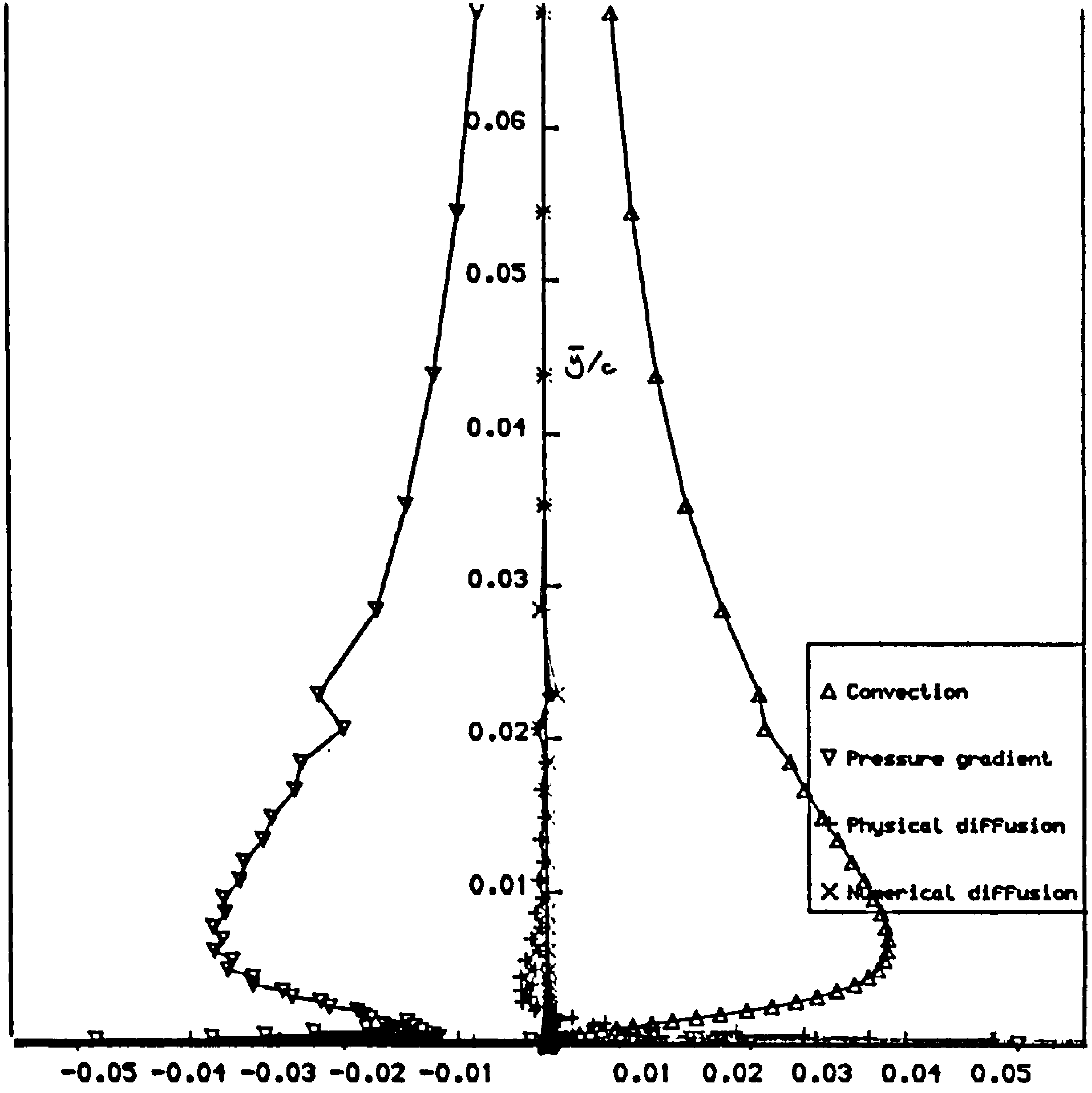


Fig.3.28 Balance of terms in solution - Grid E - station 1.

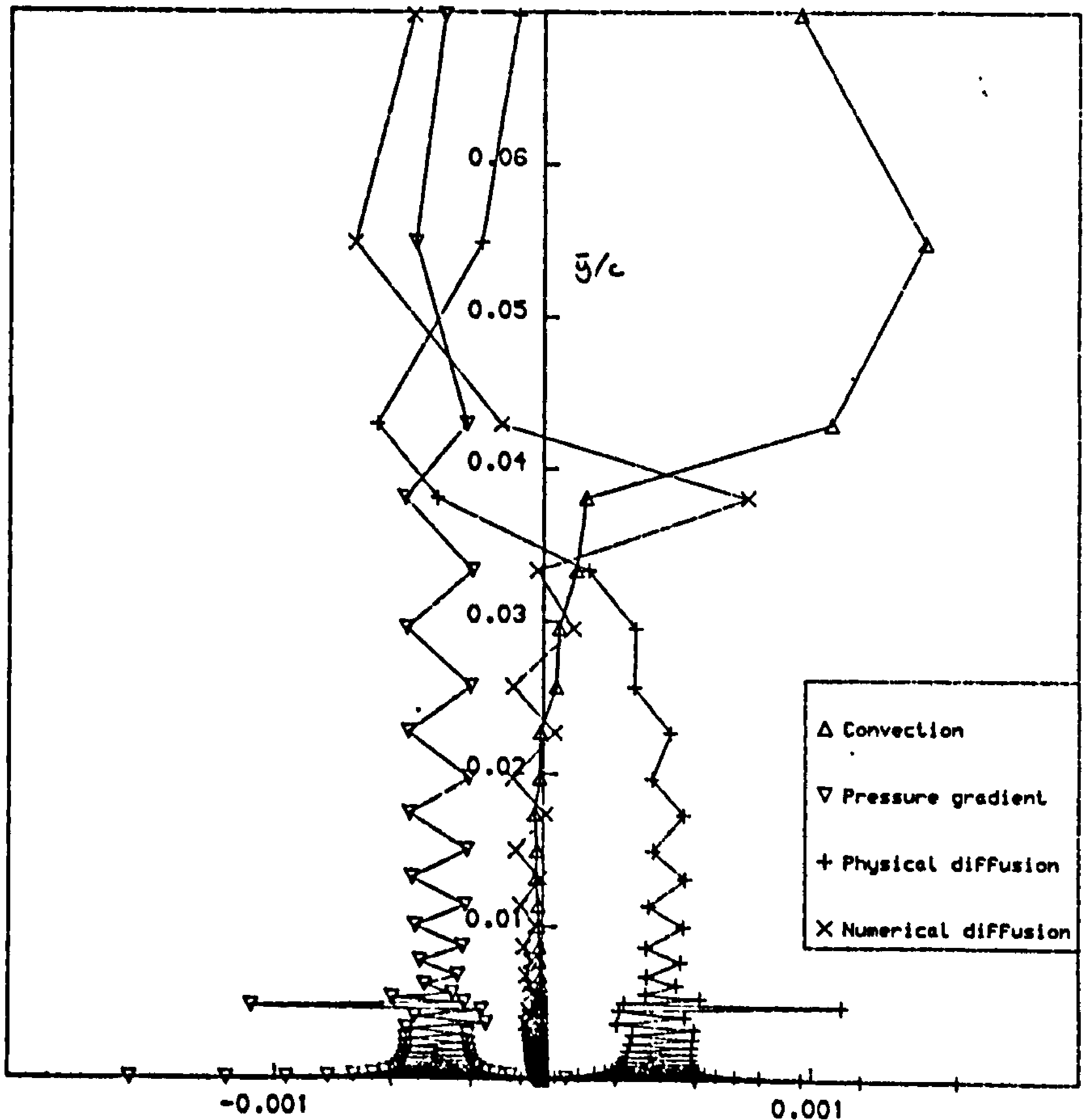


Fig.3.29 Balance of terms in solution - Grid E - station 2.

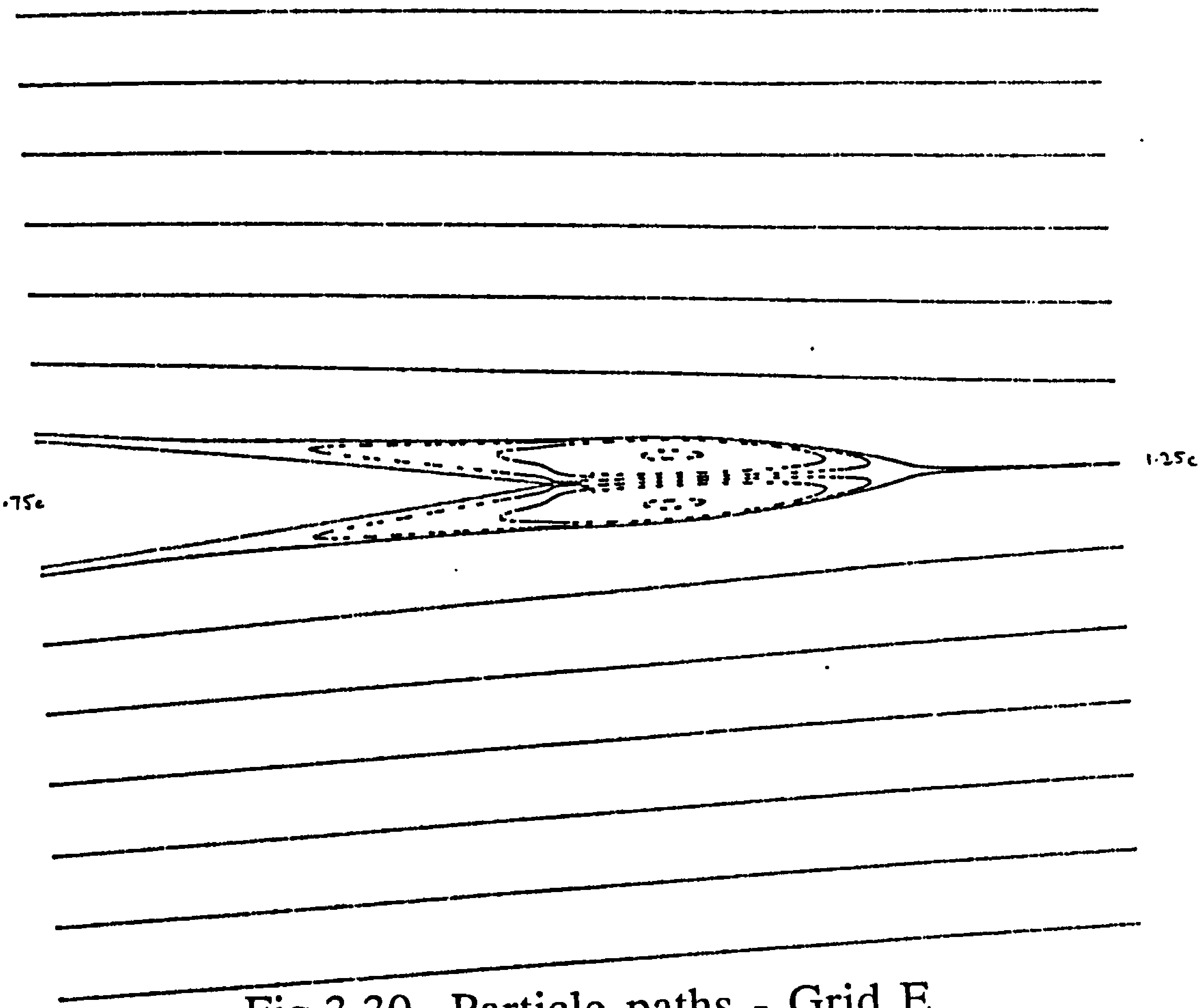


Fig.3.30 Particle paths - Grid E
- recirculation region.

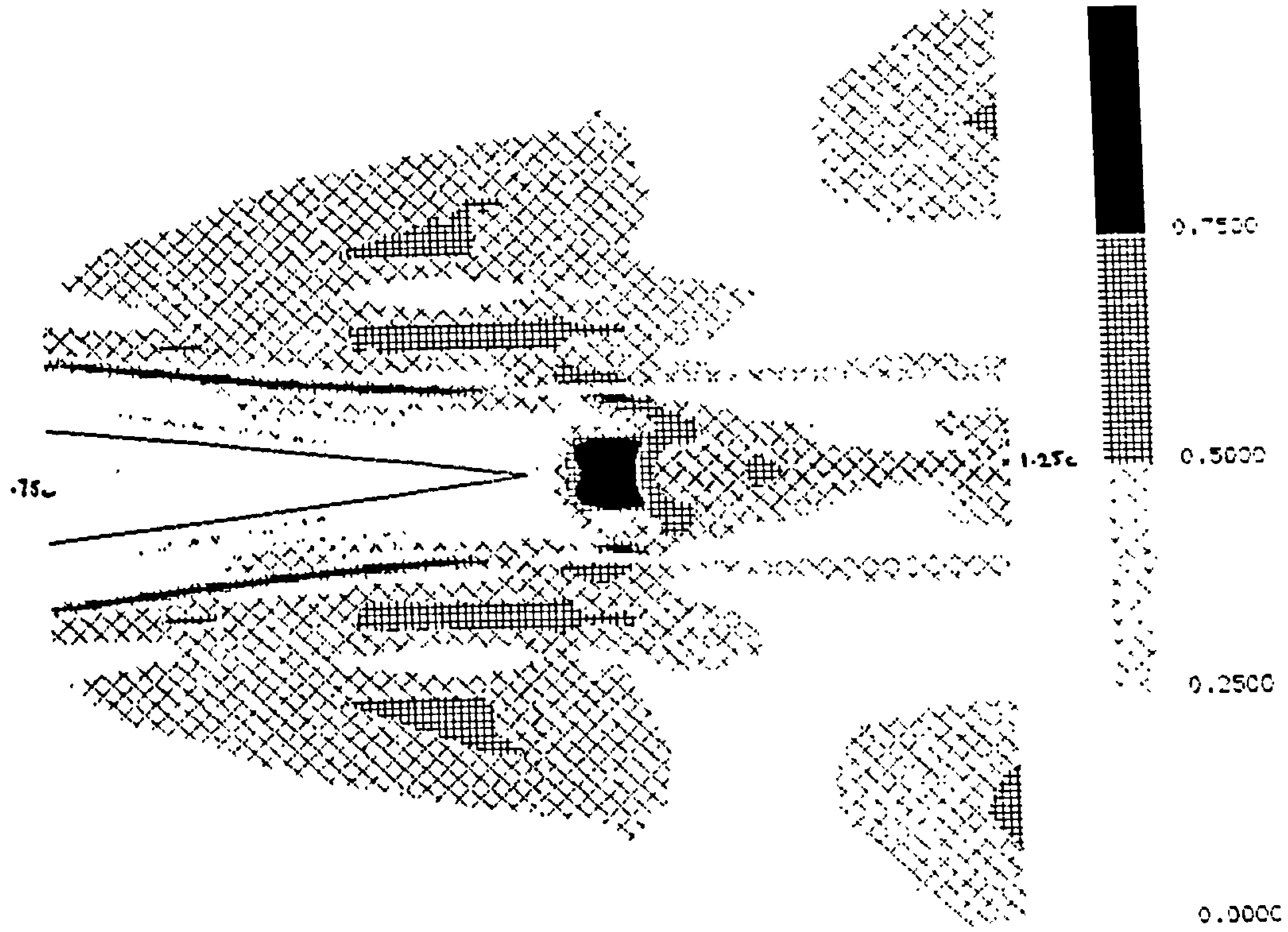


Fig.3.31 Grid E - NDR.

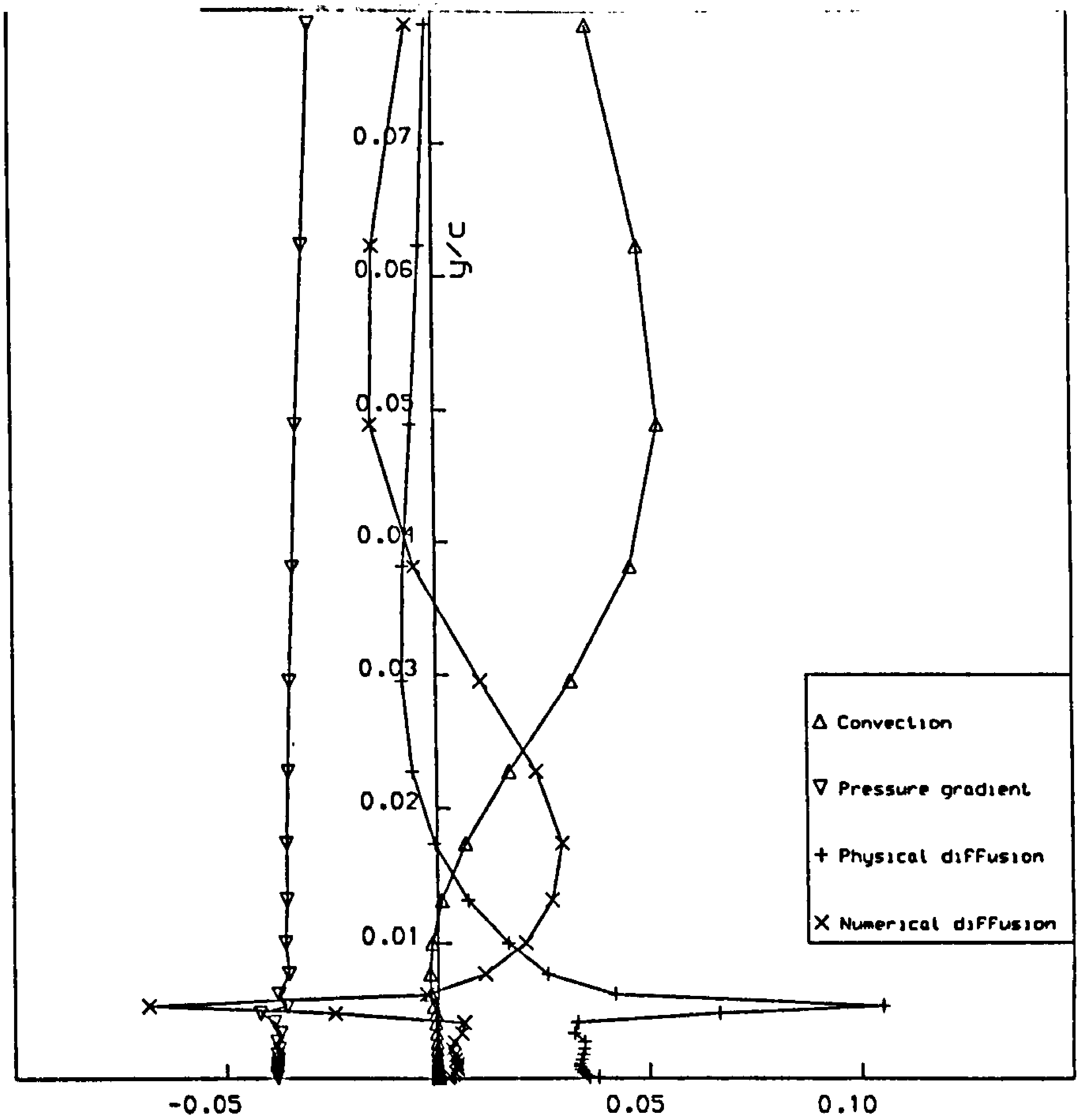


Fig.3.32 Balance of terms in solution - Grid C - station 2 - explicit algorithm.

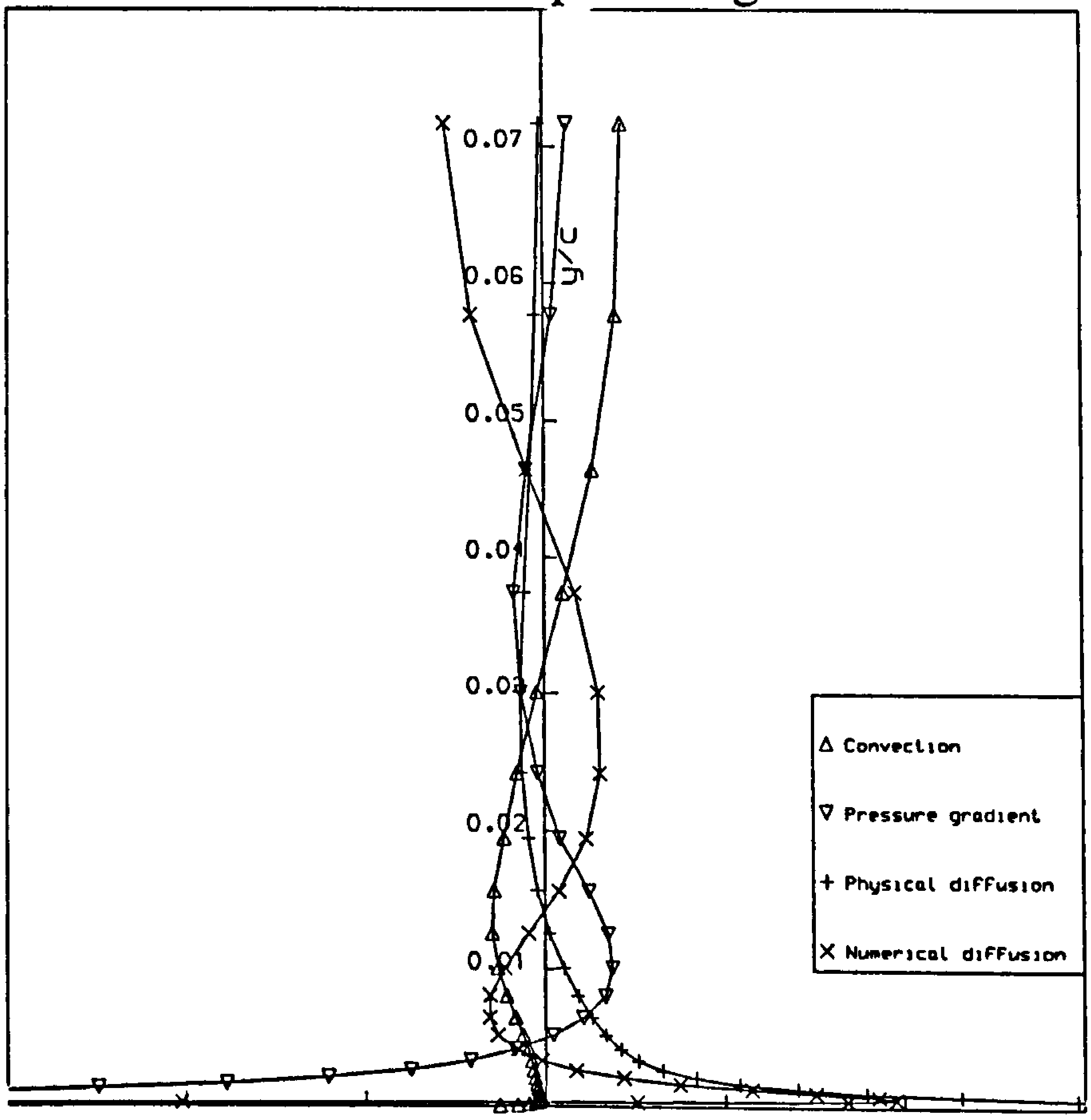


Fig.3.33 Balance of terms in solution - Grid C - station 3 - explicit algorithm.

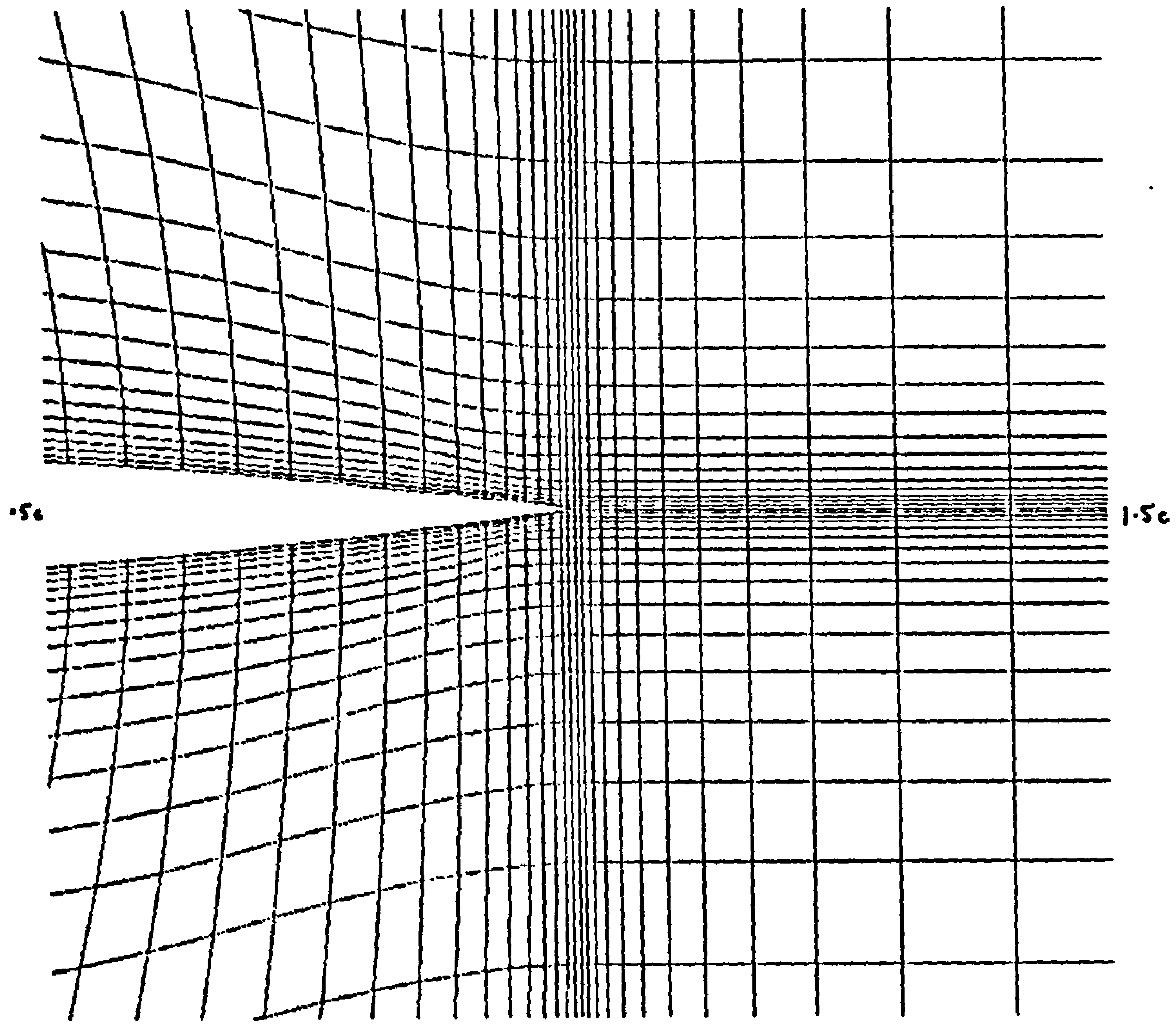


Fig.3.34 Grid for compressible explicit case - trailing-edge region.

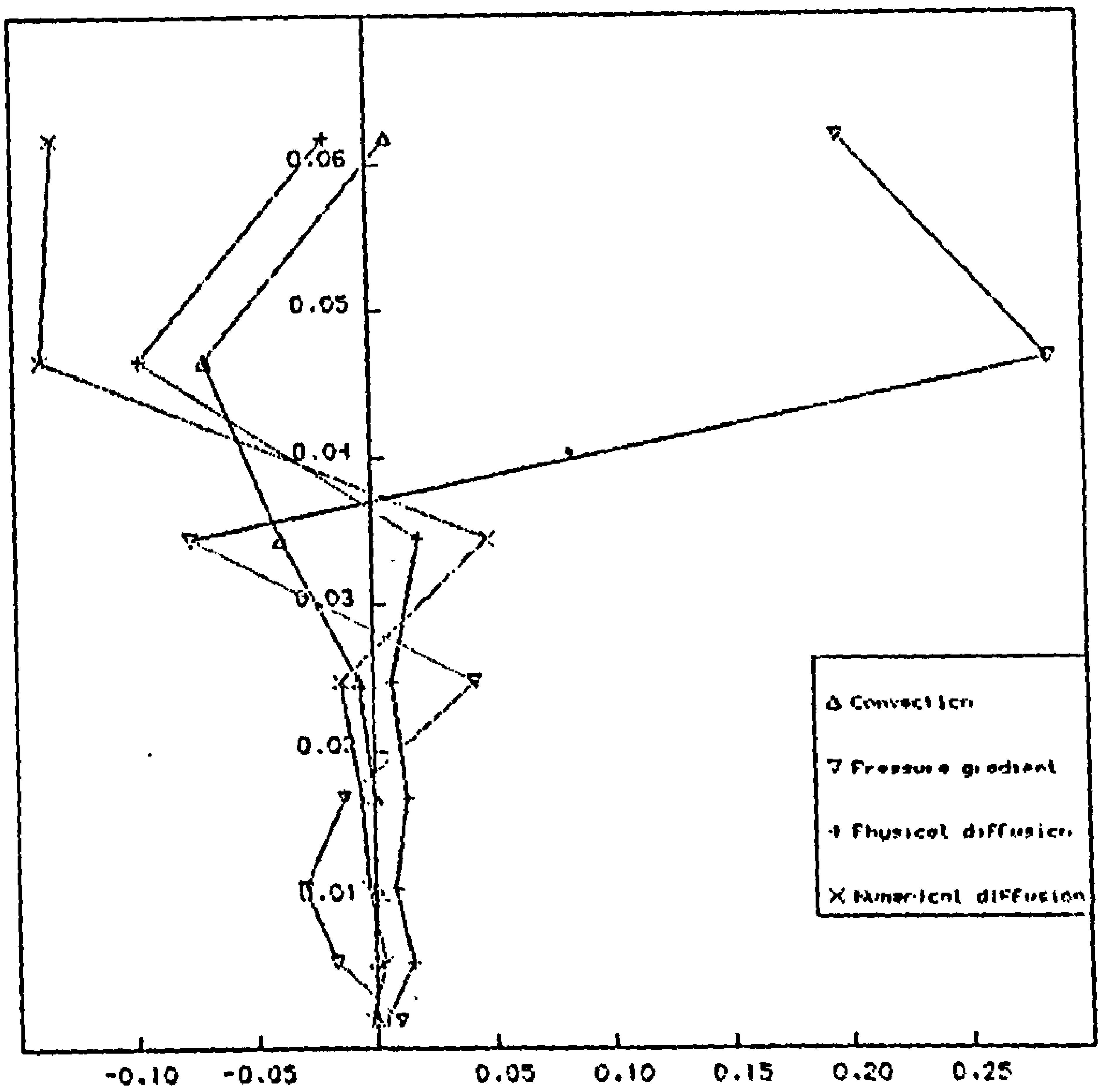


Fig.3.35 Balance of terms in solution - Compressible explicit case - station 2.

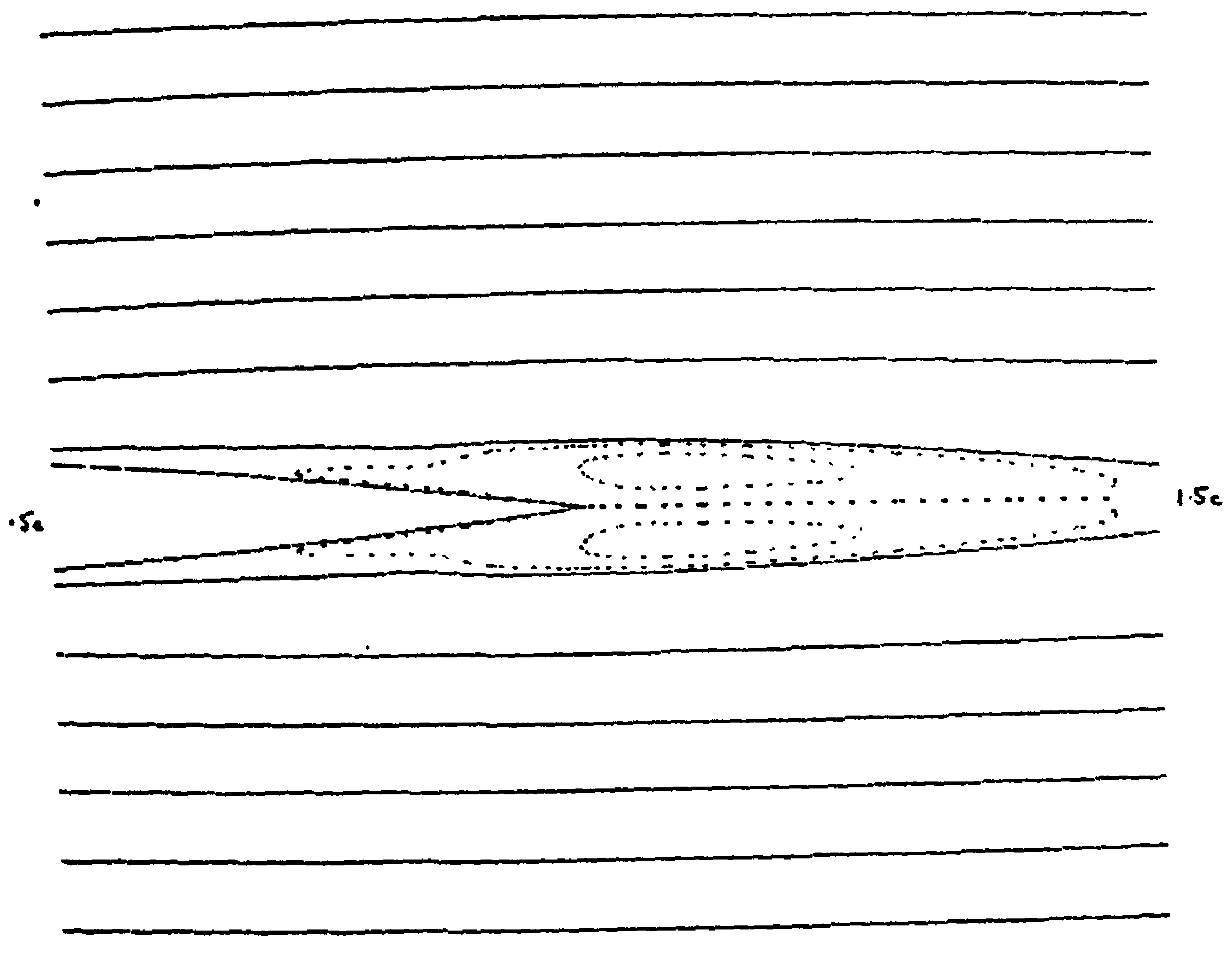


Fig.3.36 Particle paths - Compressible explicit case - recirculation region.

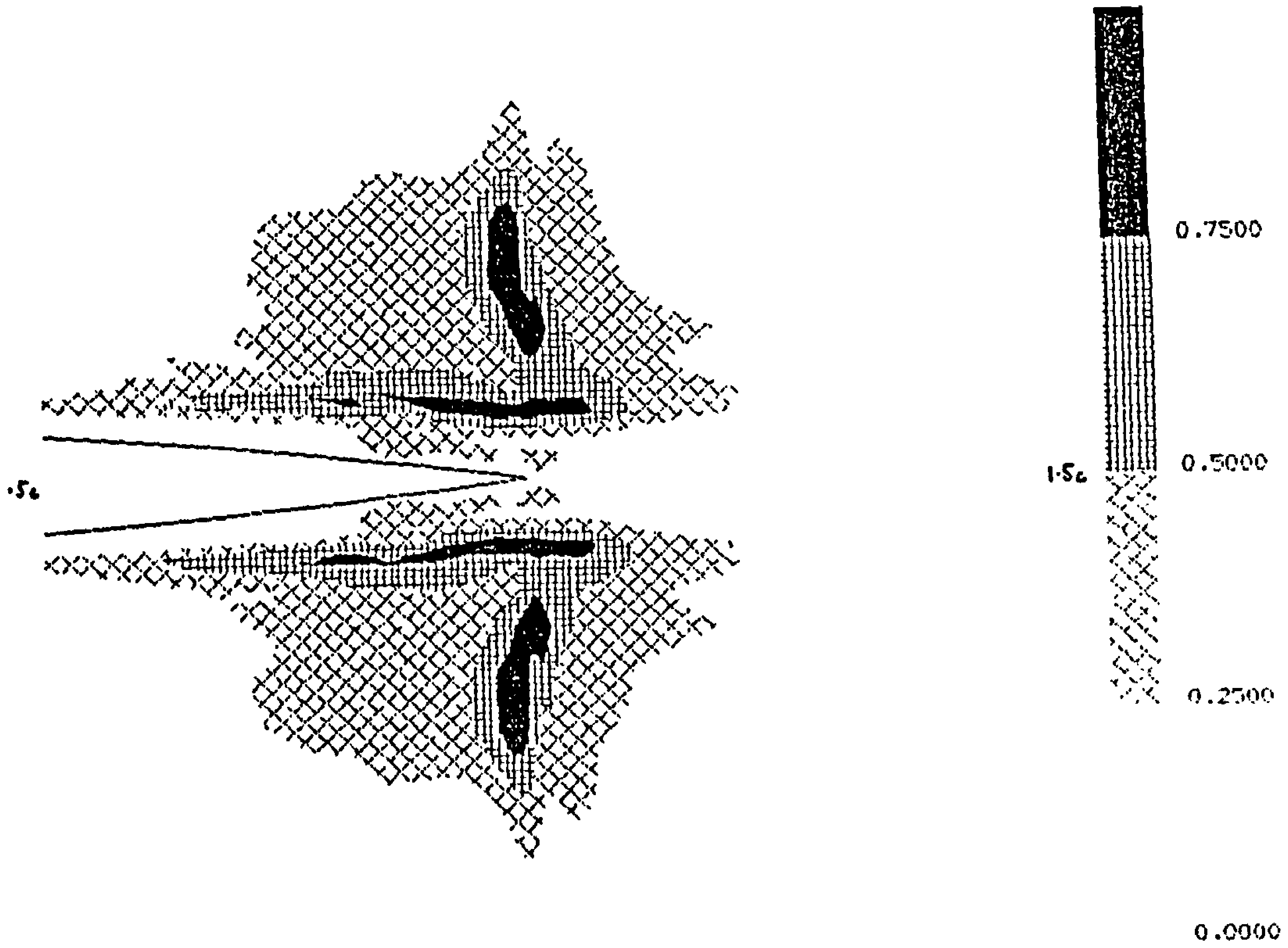


Fig.3.37 Compressible explicit case - NDR.

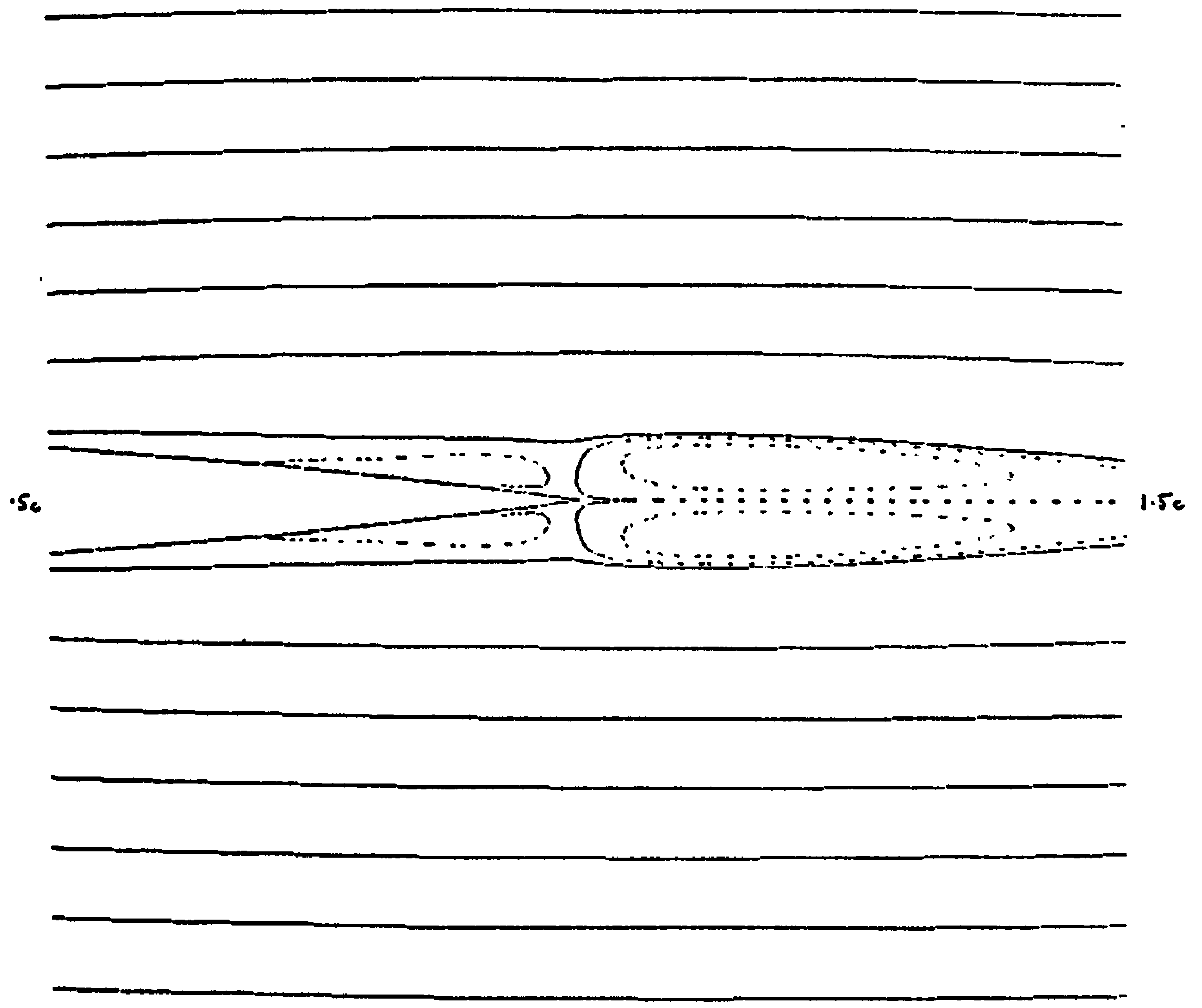


Fig.3.38 Particle paths - Grid B (compressible explicit case) - recirculation regions.

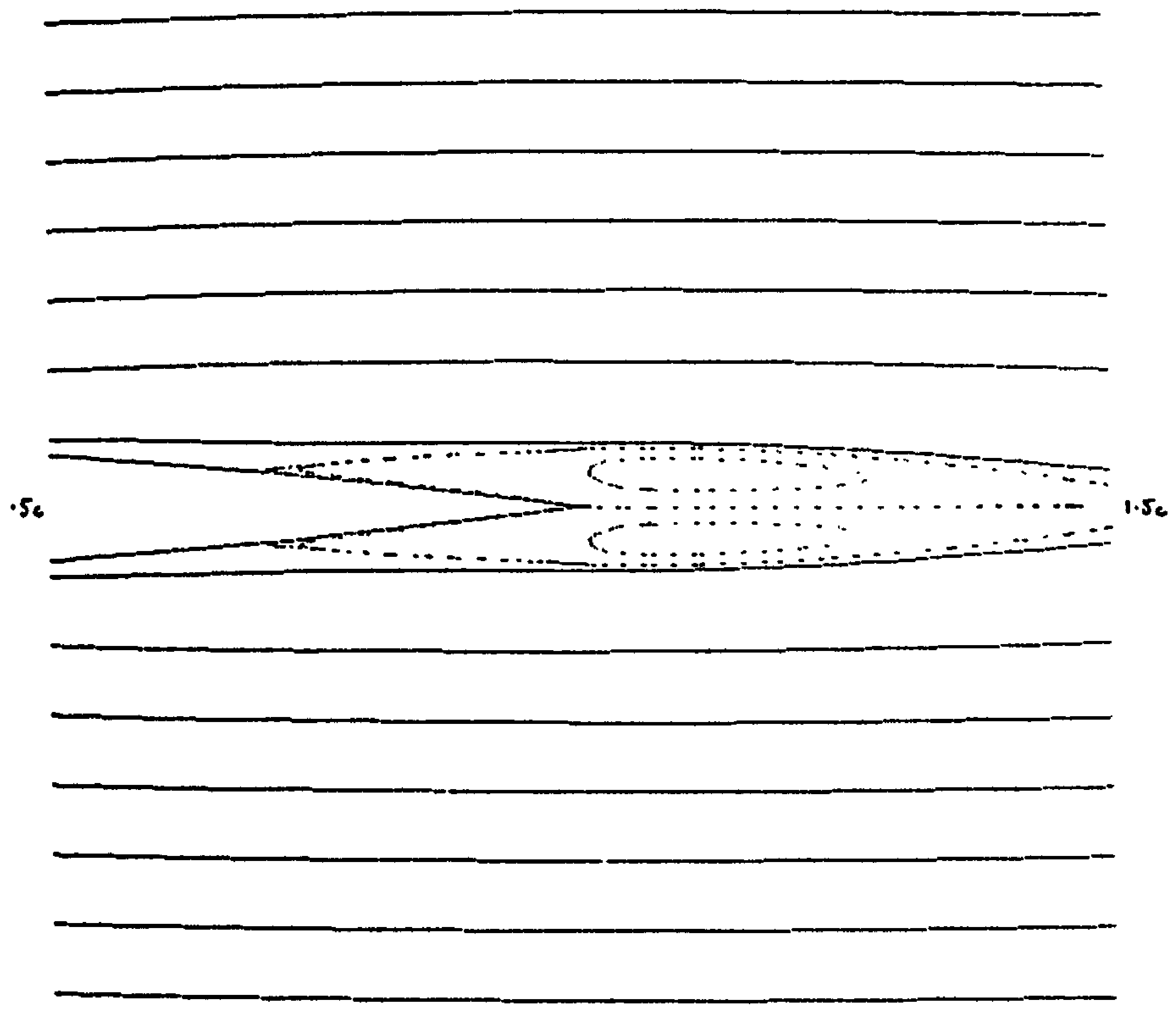


Fig.3.39 Particle paths - Grid C (compressible explicit case) - recirculation region.

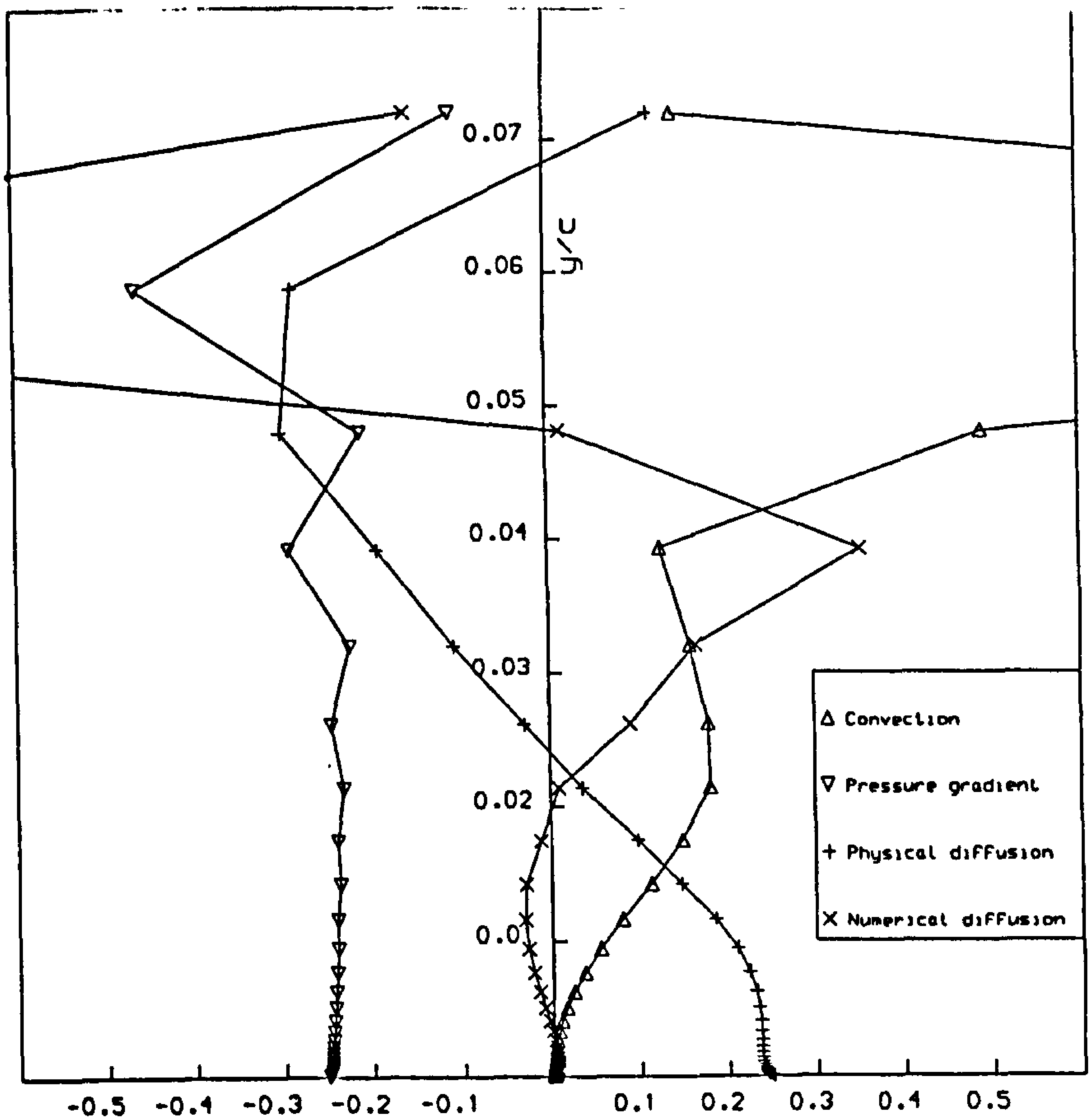


Fig.3.40 Balance of terms in solution - Grid B (compressible explicit case) - station 2.

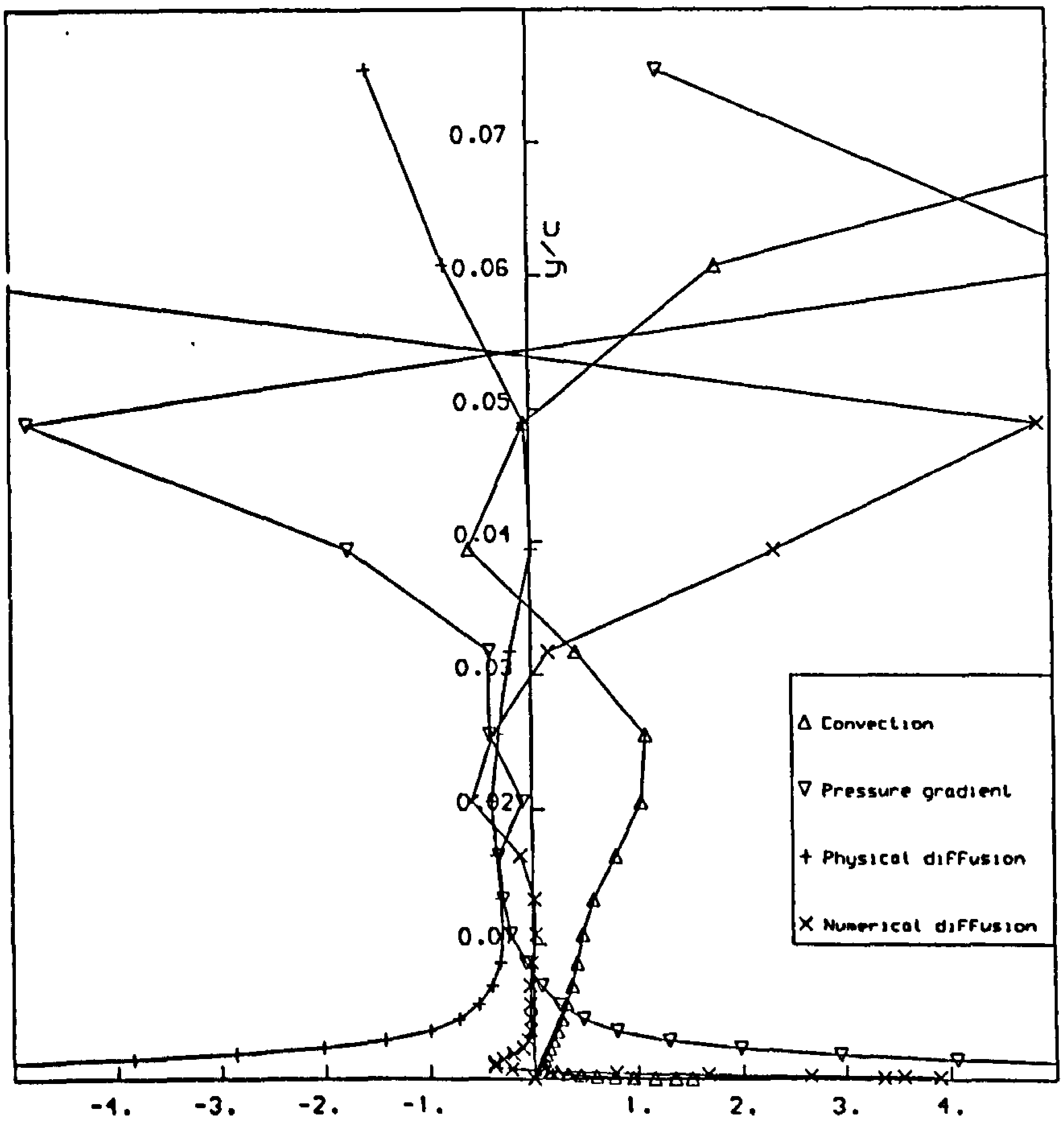


Fig.3.41 Balance of terms in solution - Grid B (compressible explicit case) - station 3.

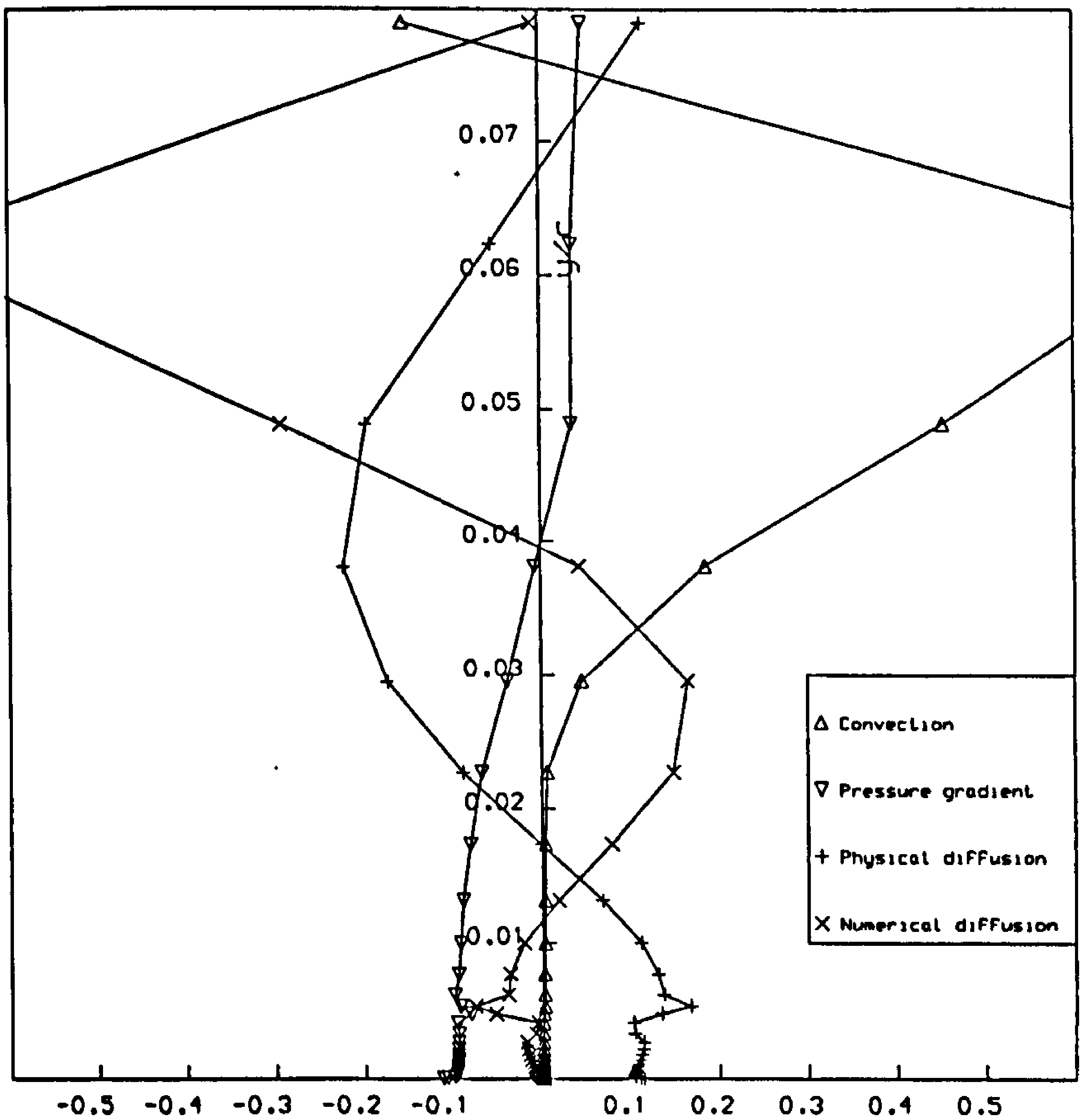


Fig.3.42 Balance of terms in solution - Grid C (compressible explicit case) - station 2.

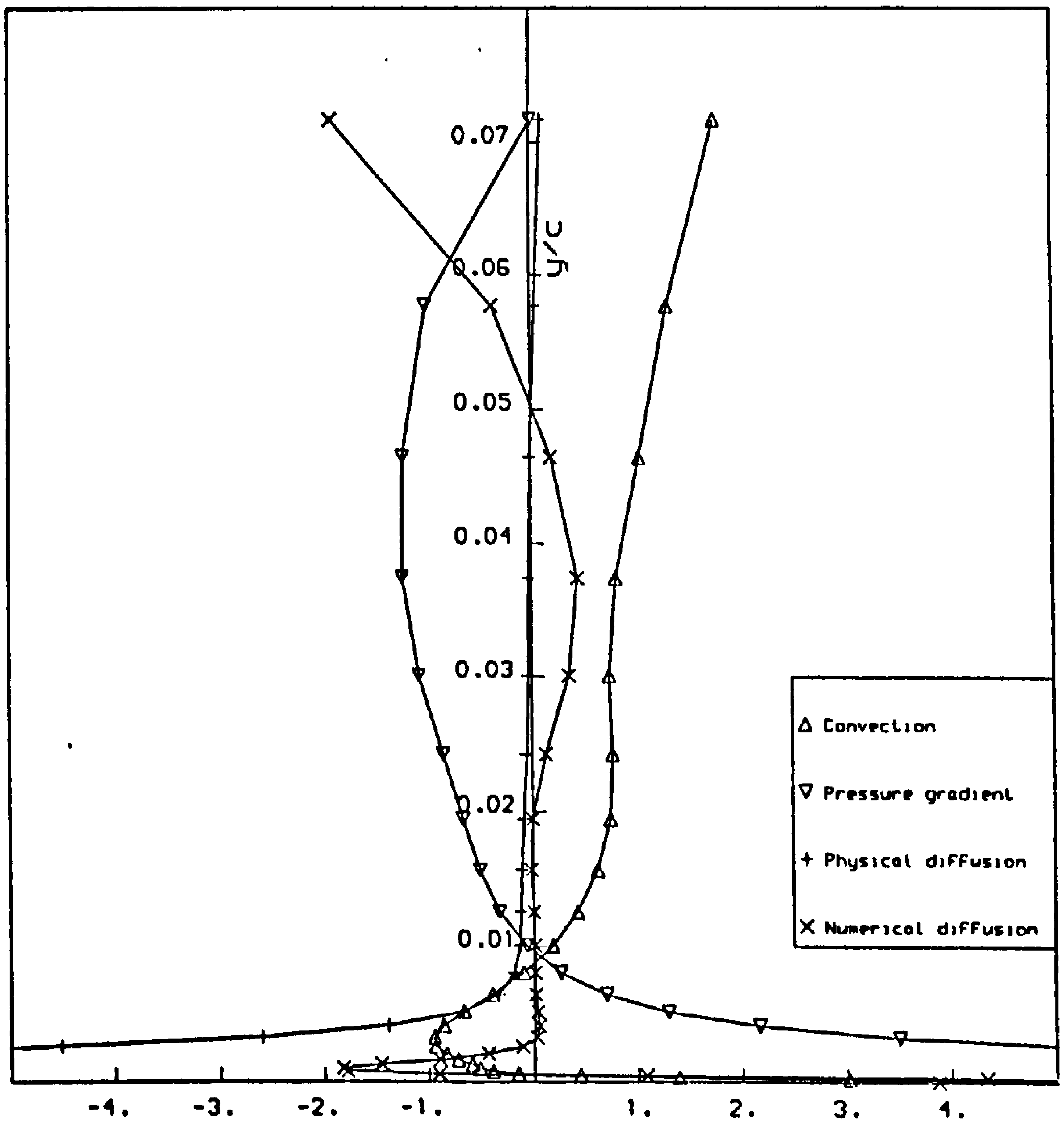


Fig.3.43 Balance of terms in solution - Grid C (compressible explicit case) - station 3.

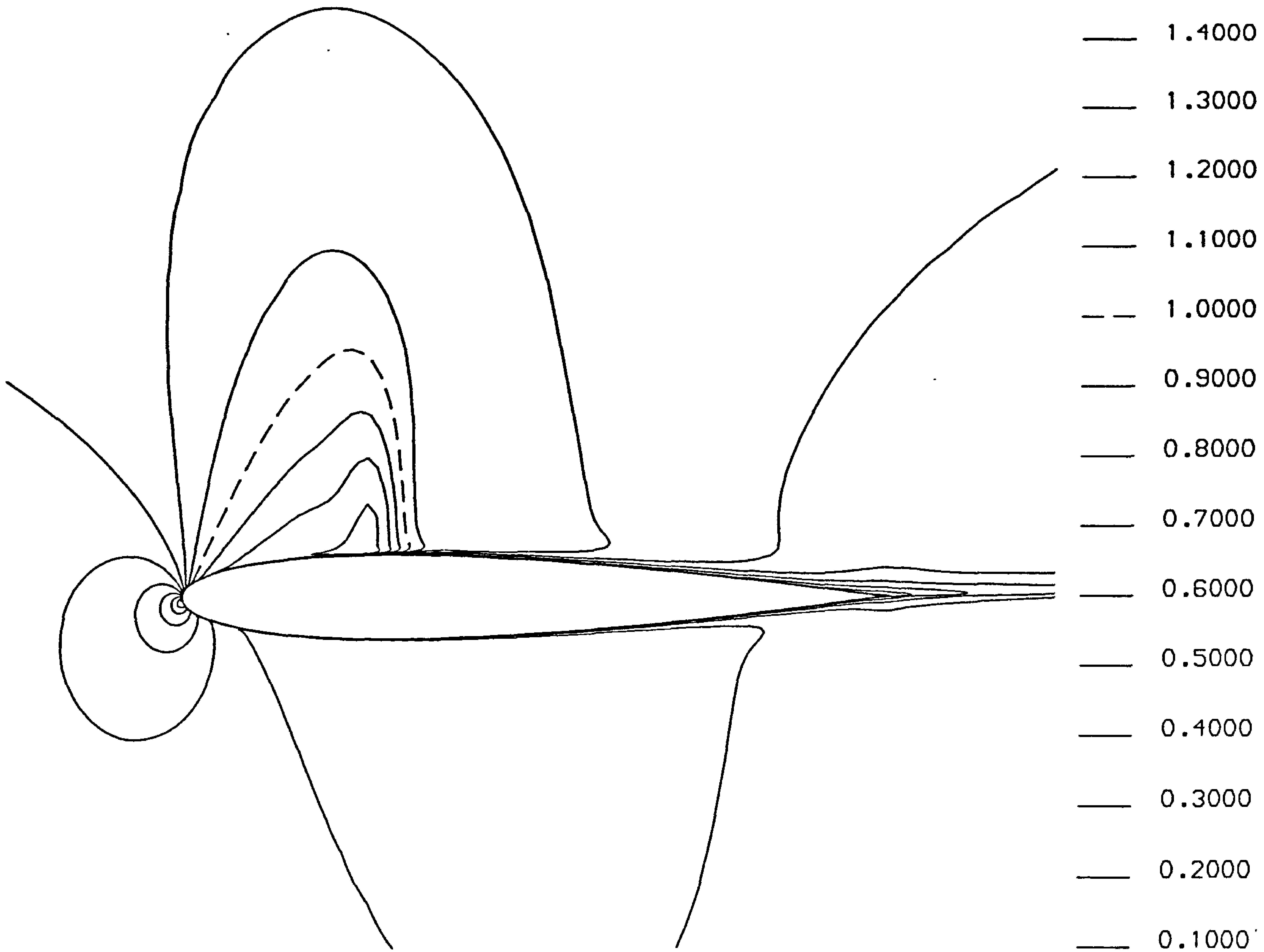


Fig.3.44 Mach number contours - Turbulent case - explicit algorithm.

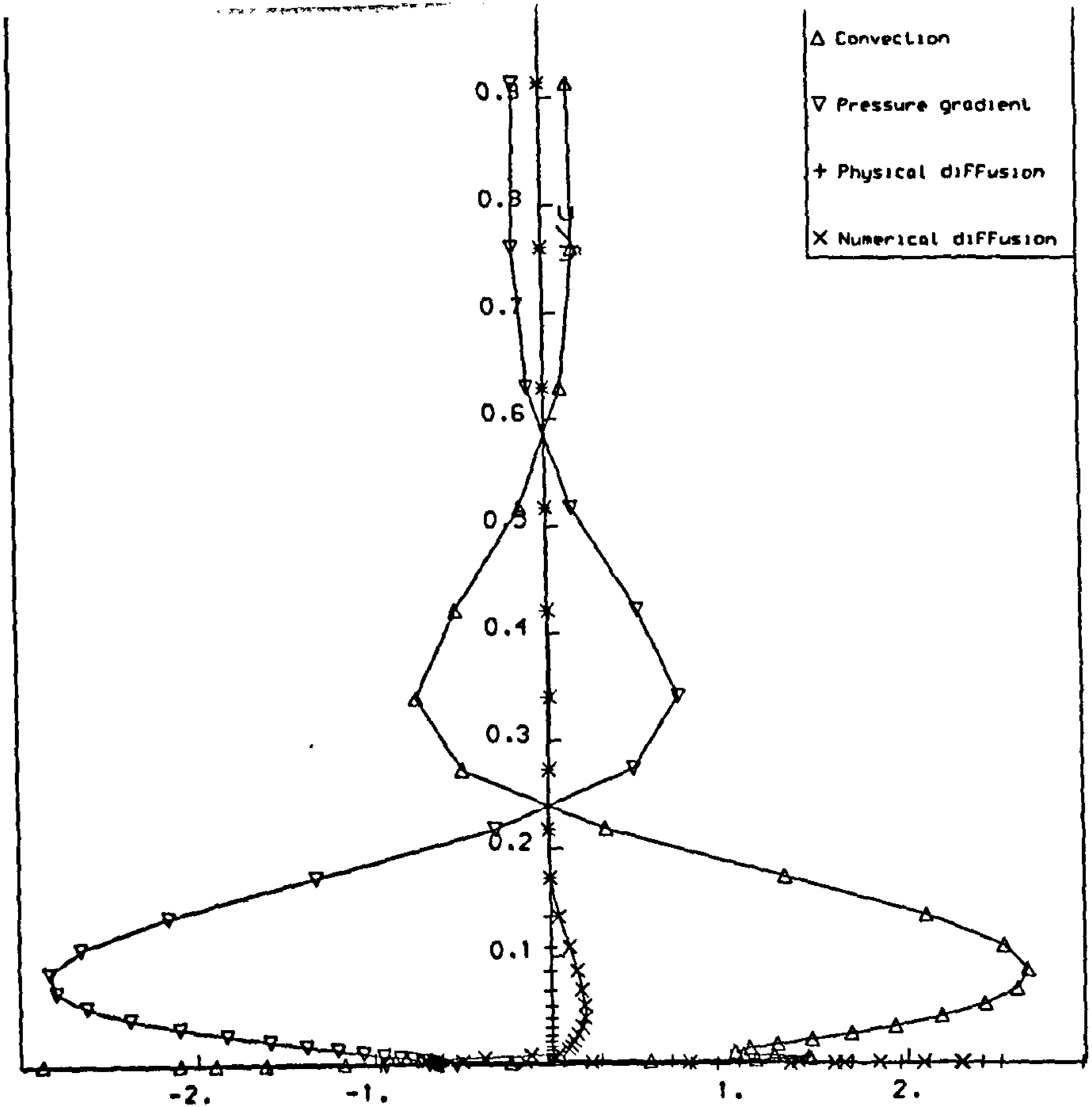


Fig.3.45 x/c = 0.25 - Turbulent case - explicit algorithm.

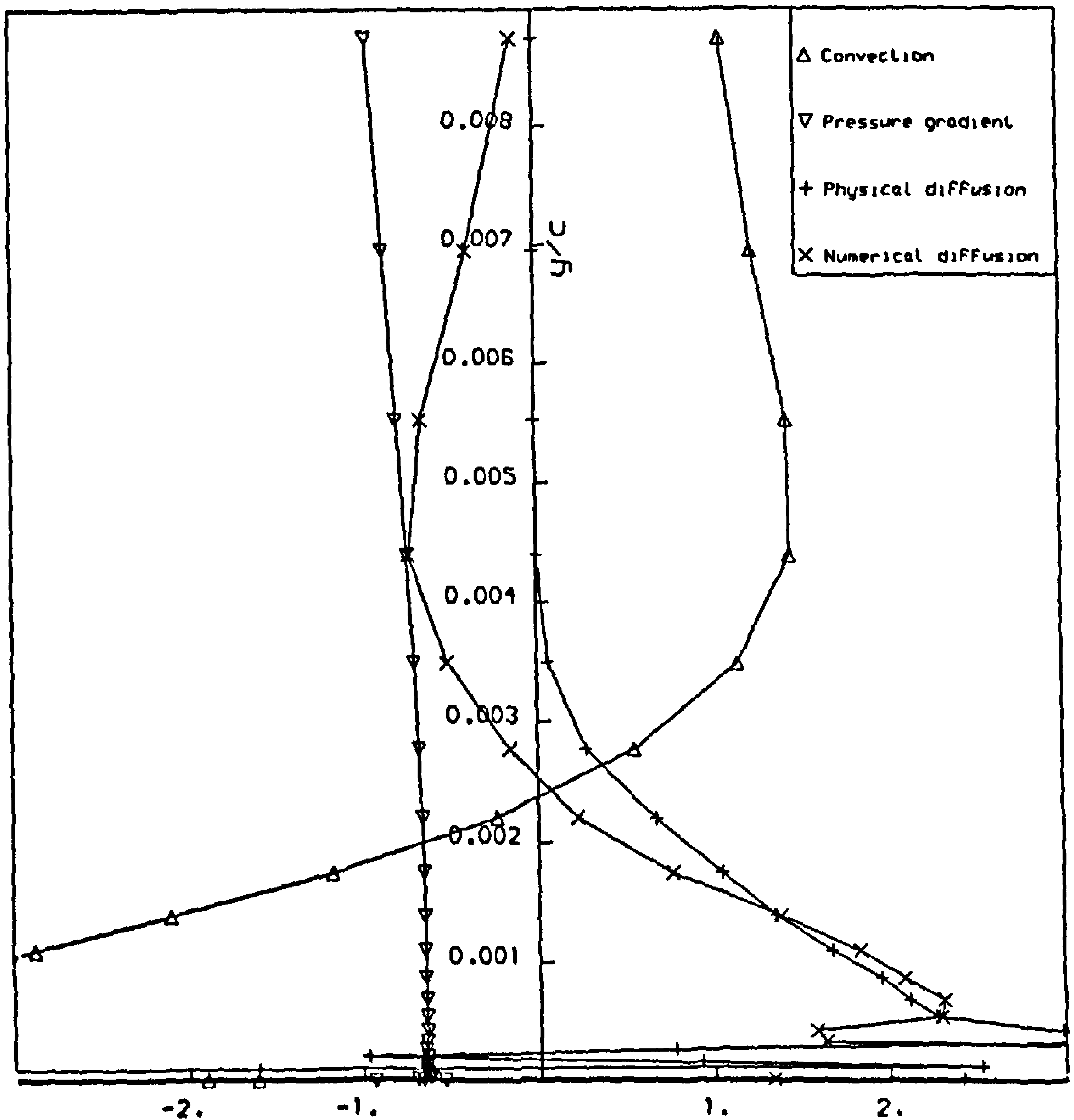


Fig.3.46 x/c = 0.25 - Turbulent case - explicit algorithm, detail close to surface.

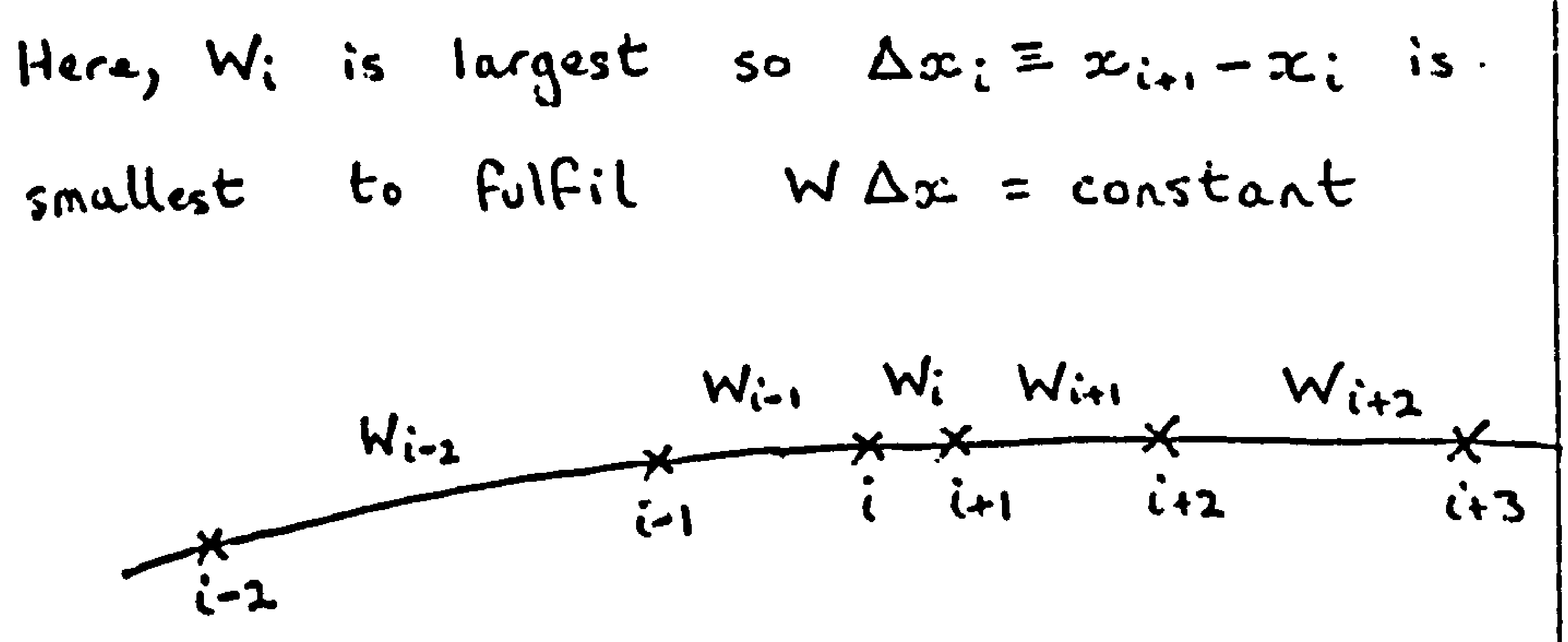

Here, W_i is largest so $\Delta x_i \equiv x_{i+1} - x_i$ is smallest to fulfil $W \Delta x$ = constant

Fig.4.1 Equidistribution principle in 1D.

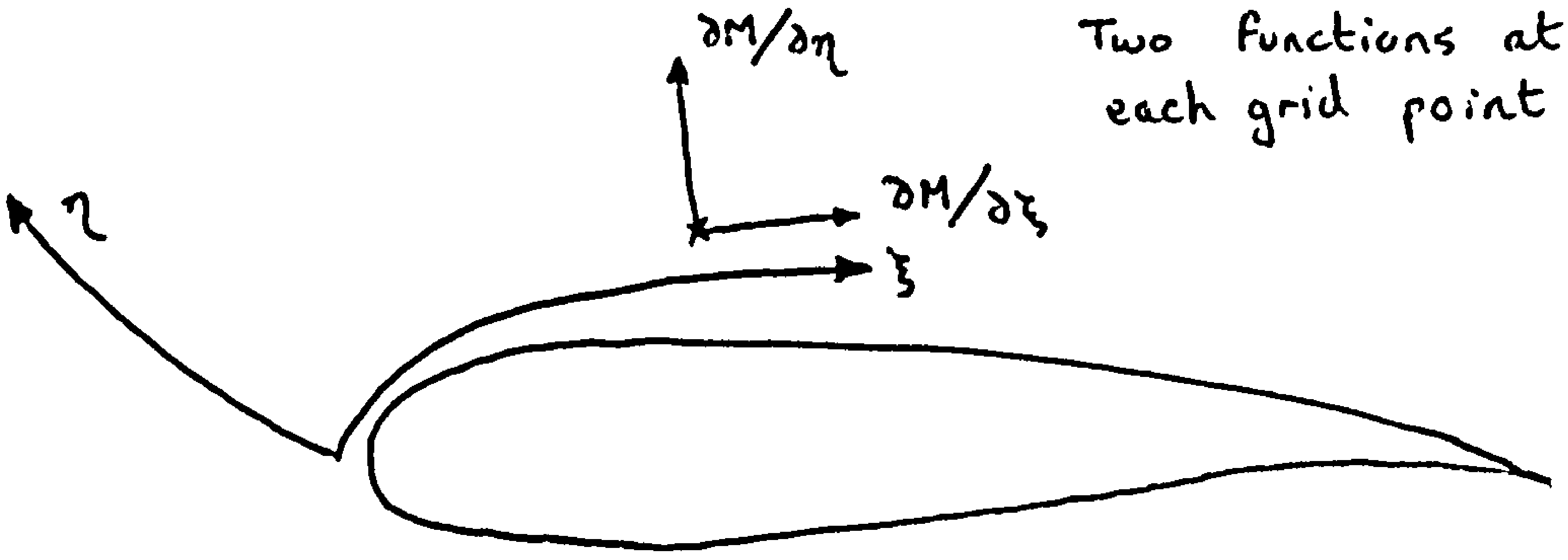
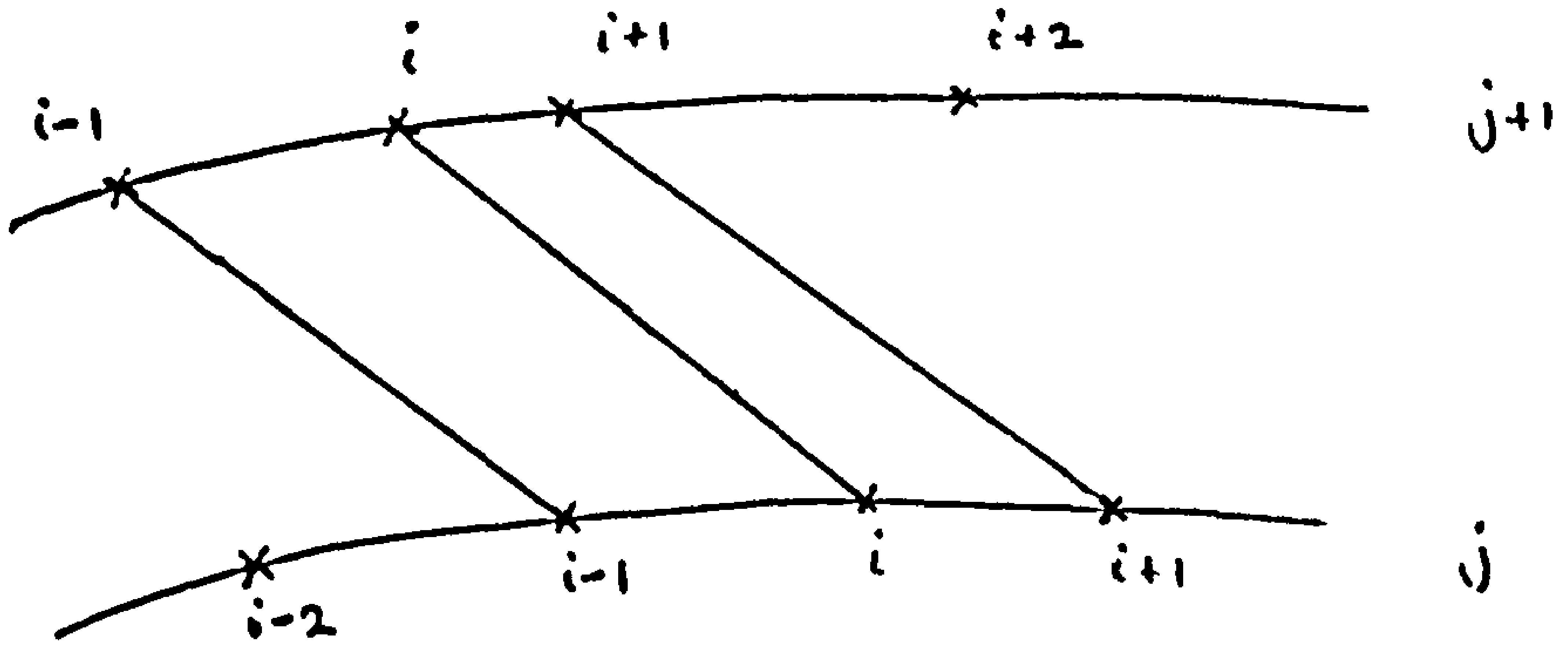
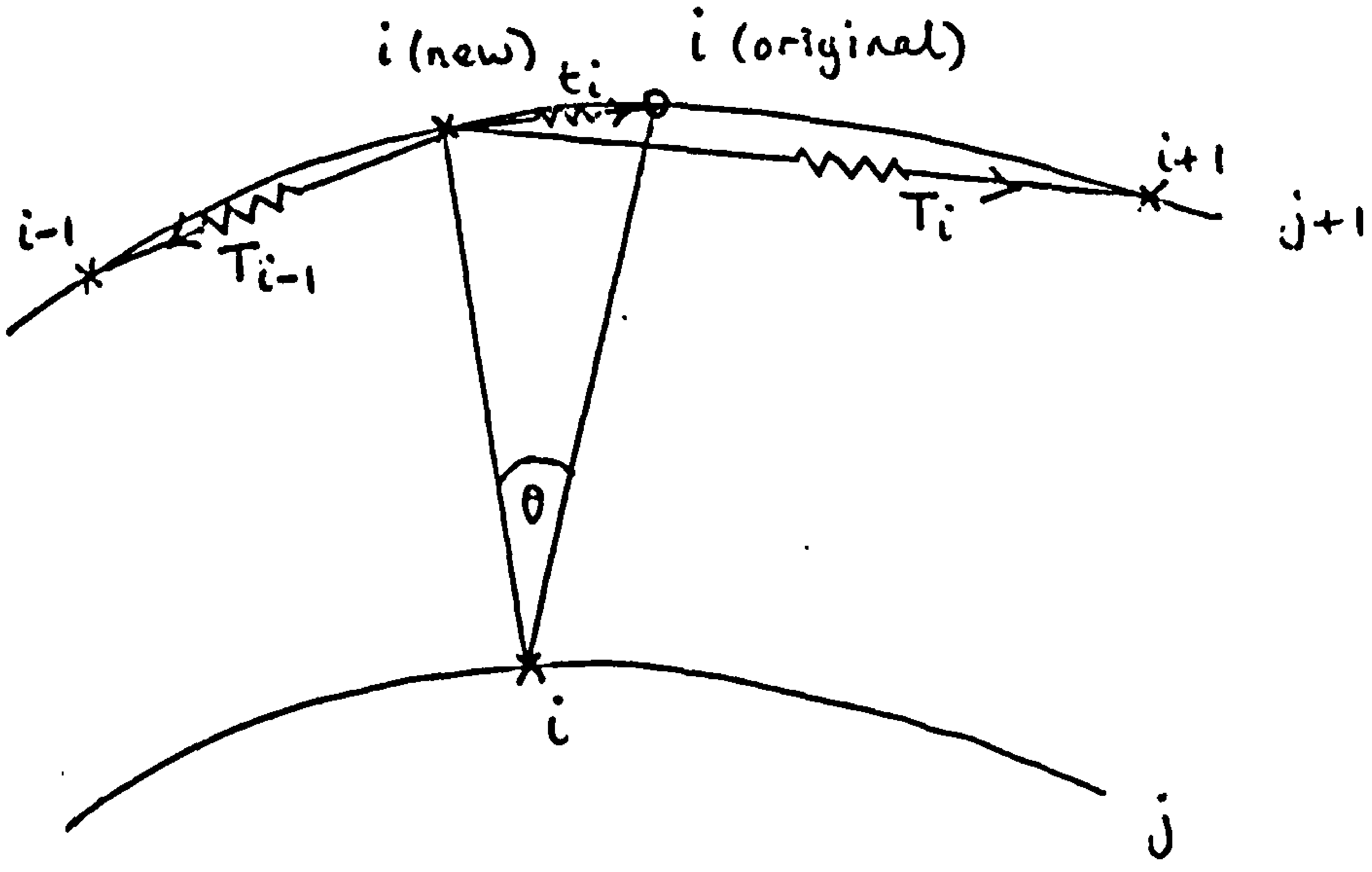

Two functions at each grid point

Fig.4.2 Weighting functions for grid adaptation.



Independent adaptation could lead to such a situation.

Fig.4.3 Spring analogy in 2D.



t_i is related to θ, preventing node i, j+1 moving too far towards i-1, j+1

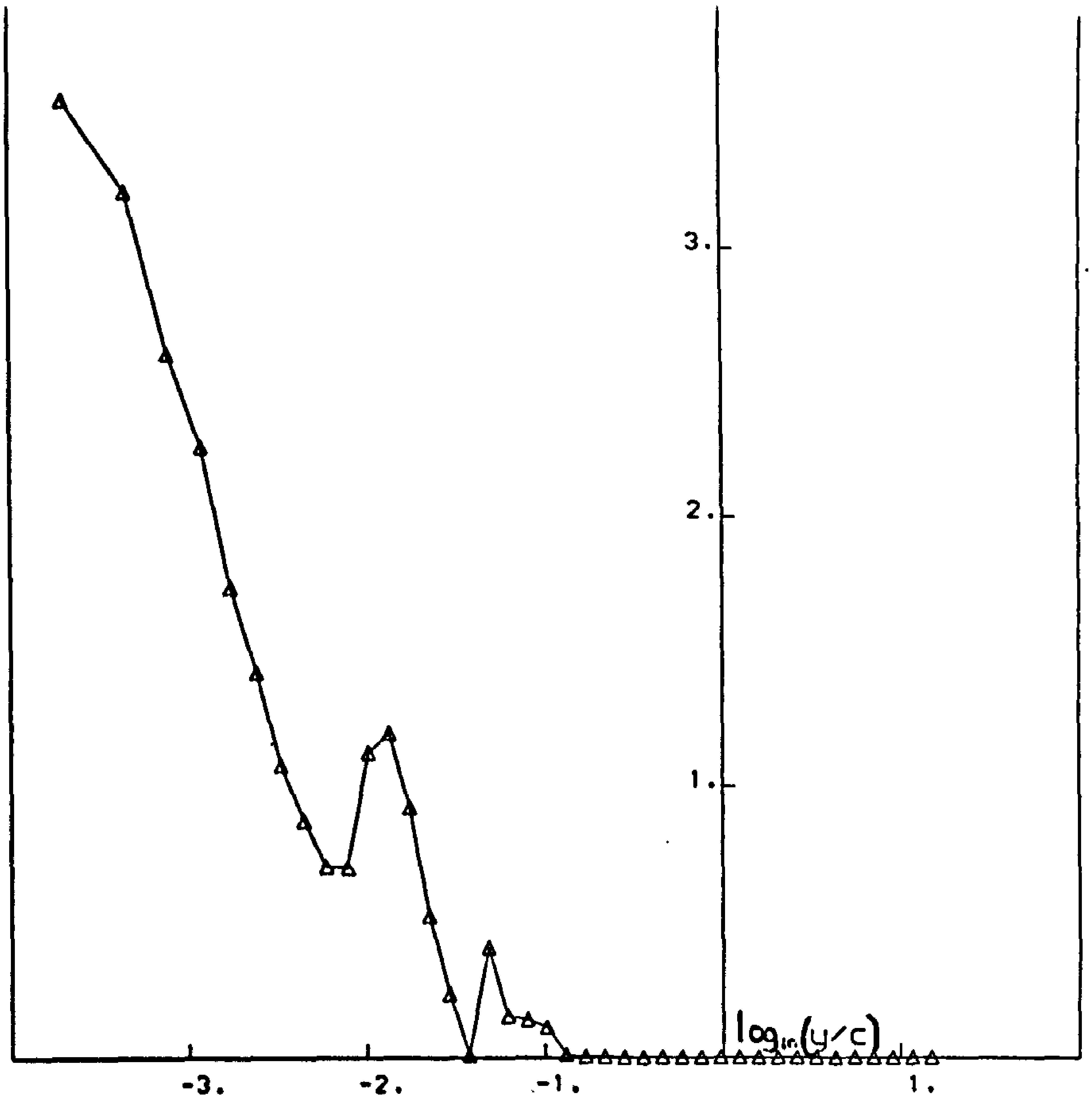


Fig.4.4 Physical diffusion / max(Convection, Pressure gradient) - station 3, Case A.

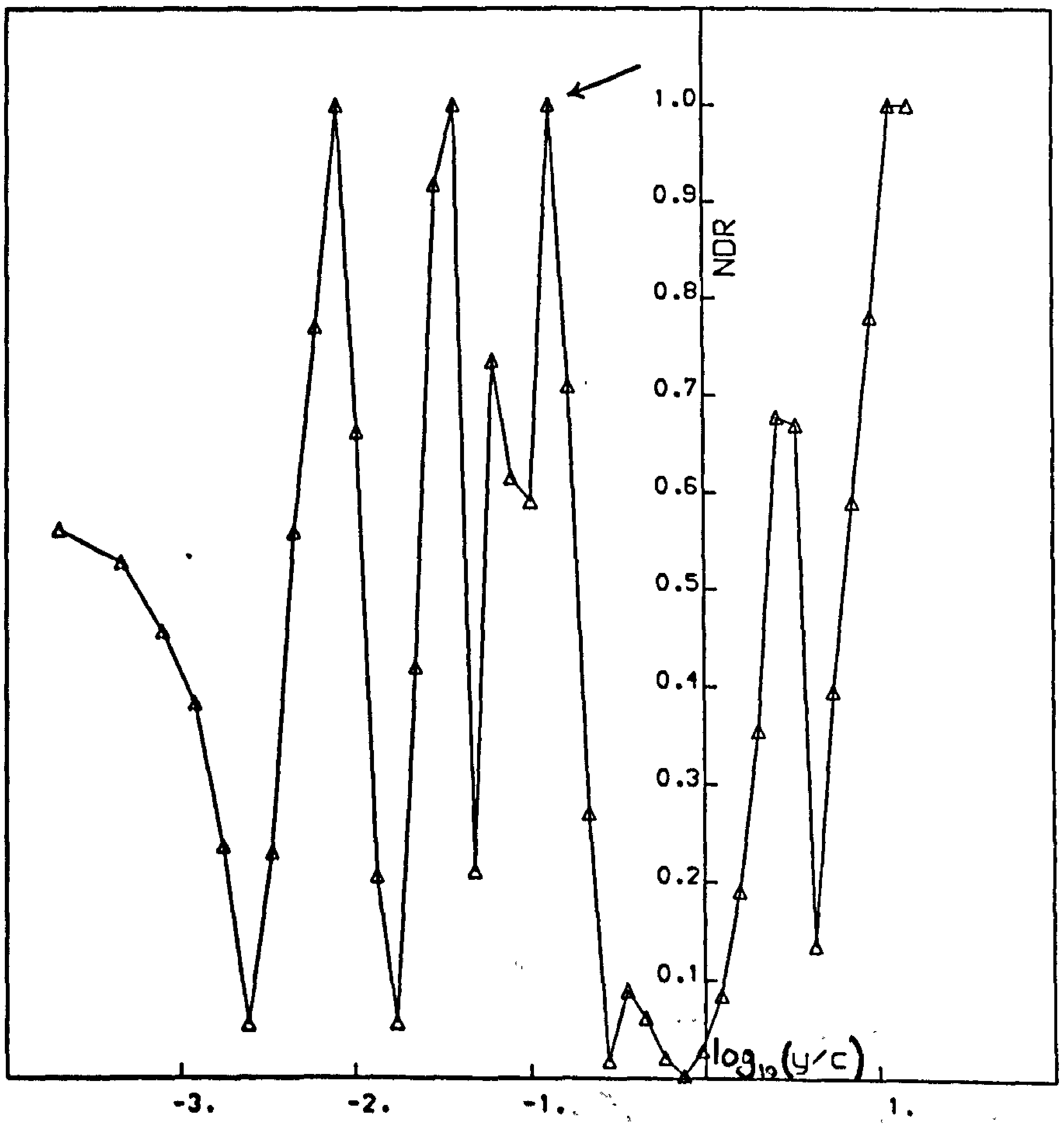


Fig.4.5 NDR - station 3, Case A.

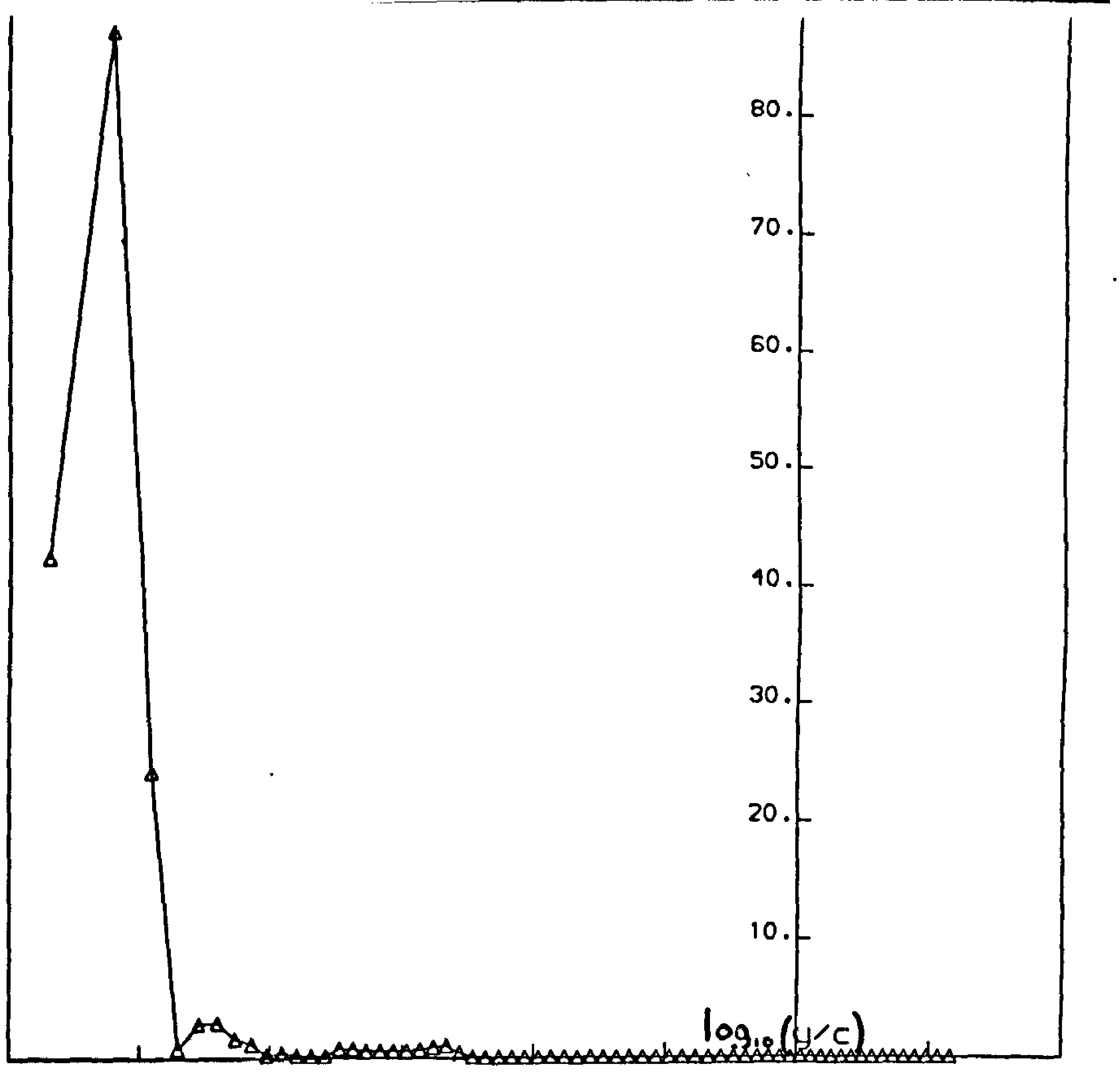


Fig.4.6 Physical diffusion / max(Convection, Pressure gradient) - x/c = 0.25, Turbulent case - explicit algorithm.

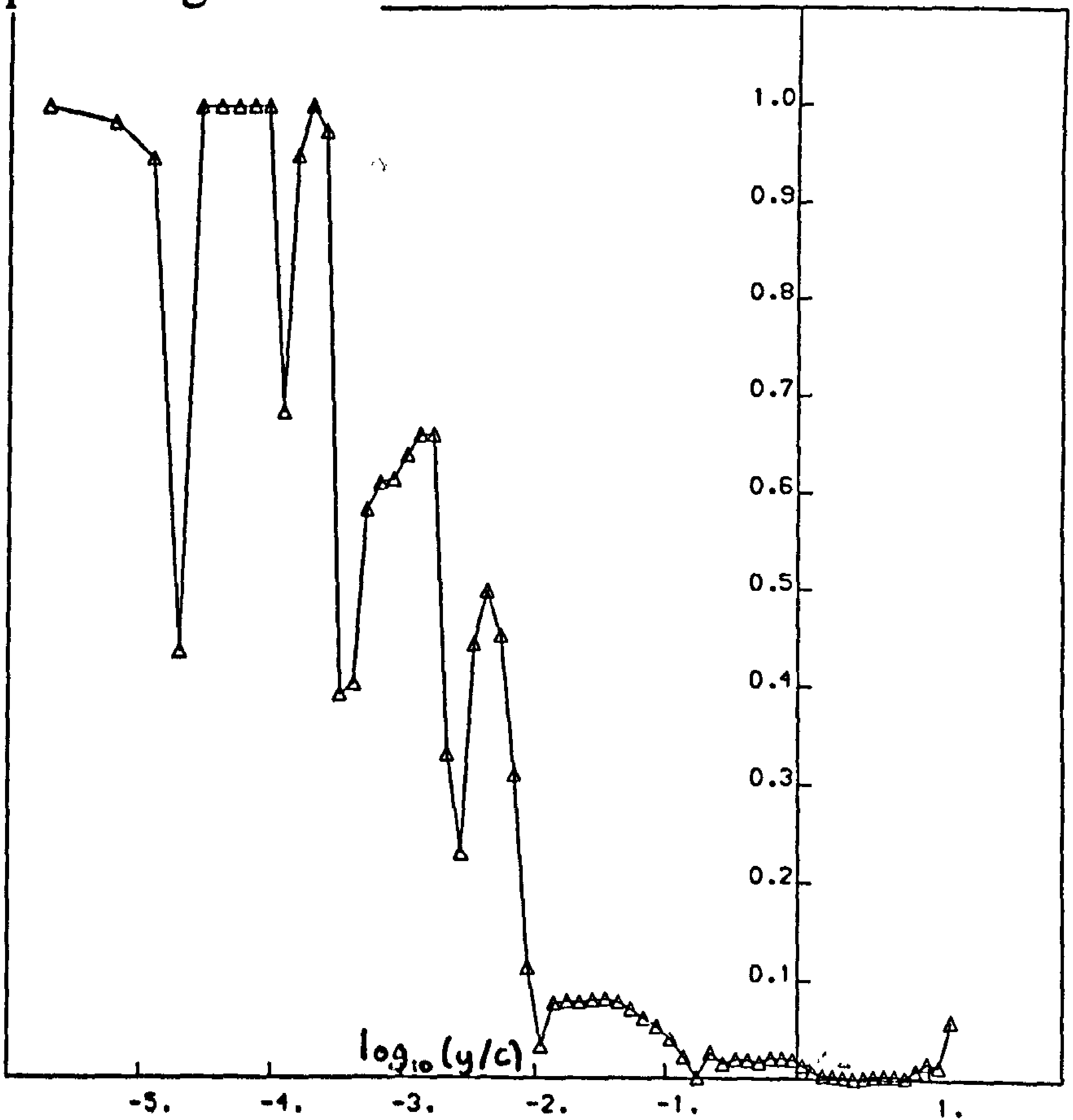


Fig.4.7 NDR - x/c = 0.25, Turbulent case - explicit algorithm.

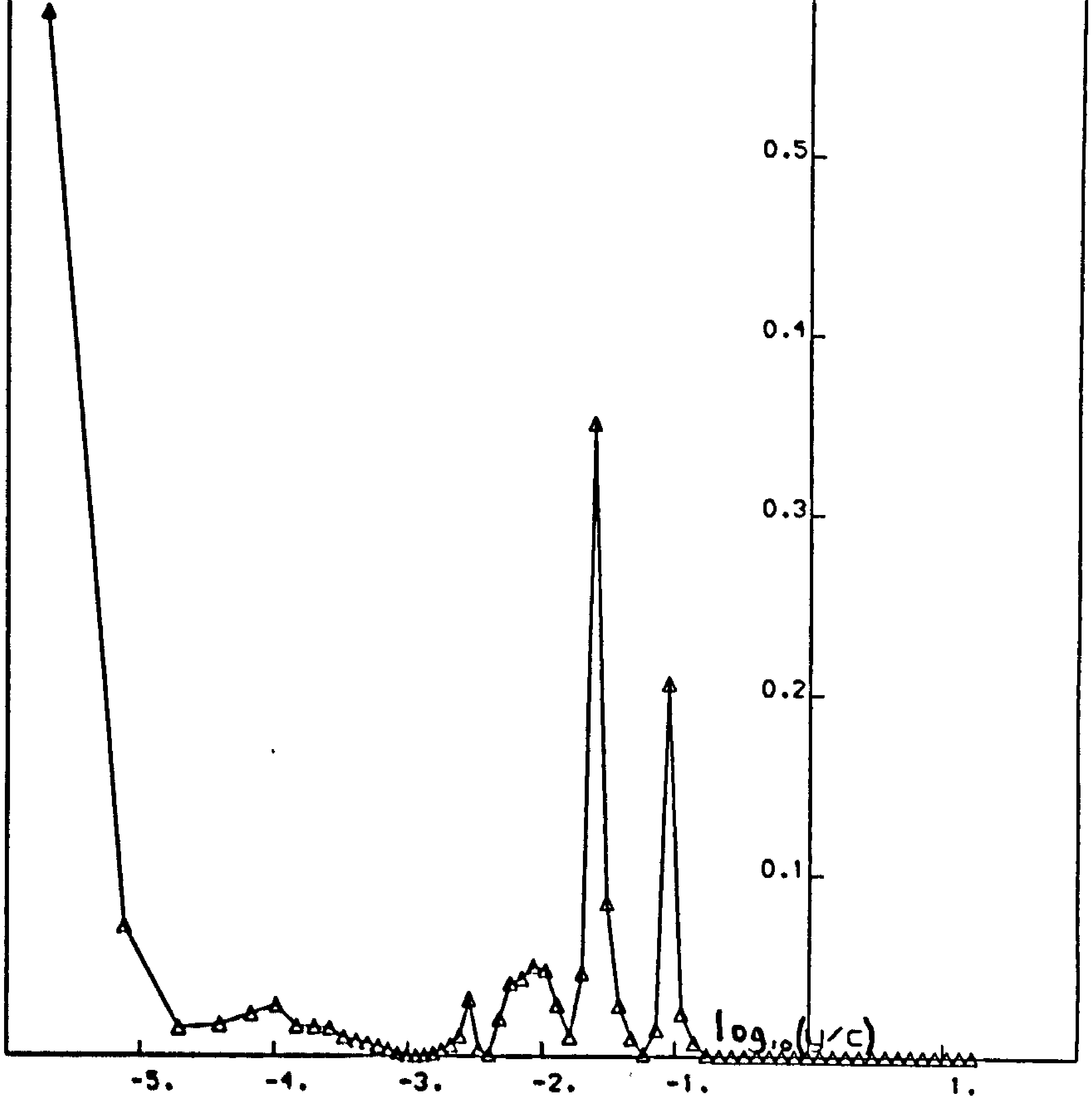


Fig.4.8 Physical diffusion / max(Convection, Pressure gradient) - x/c = 5.0, Turbulent case - explicit algorithm.

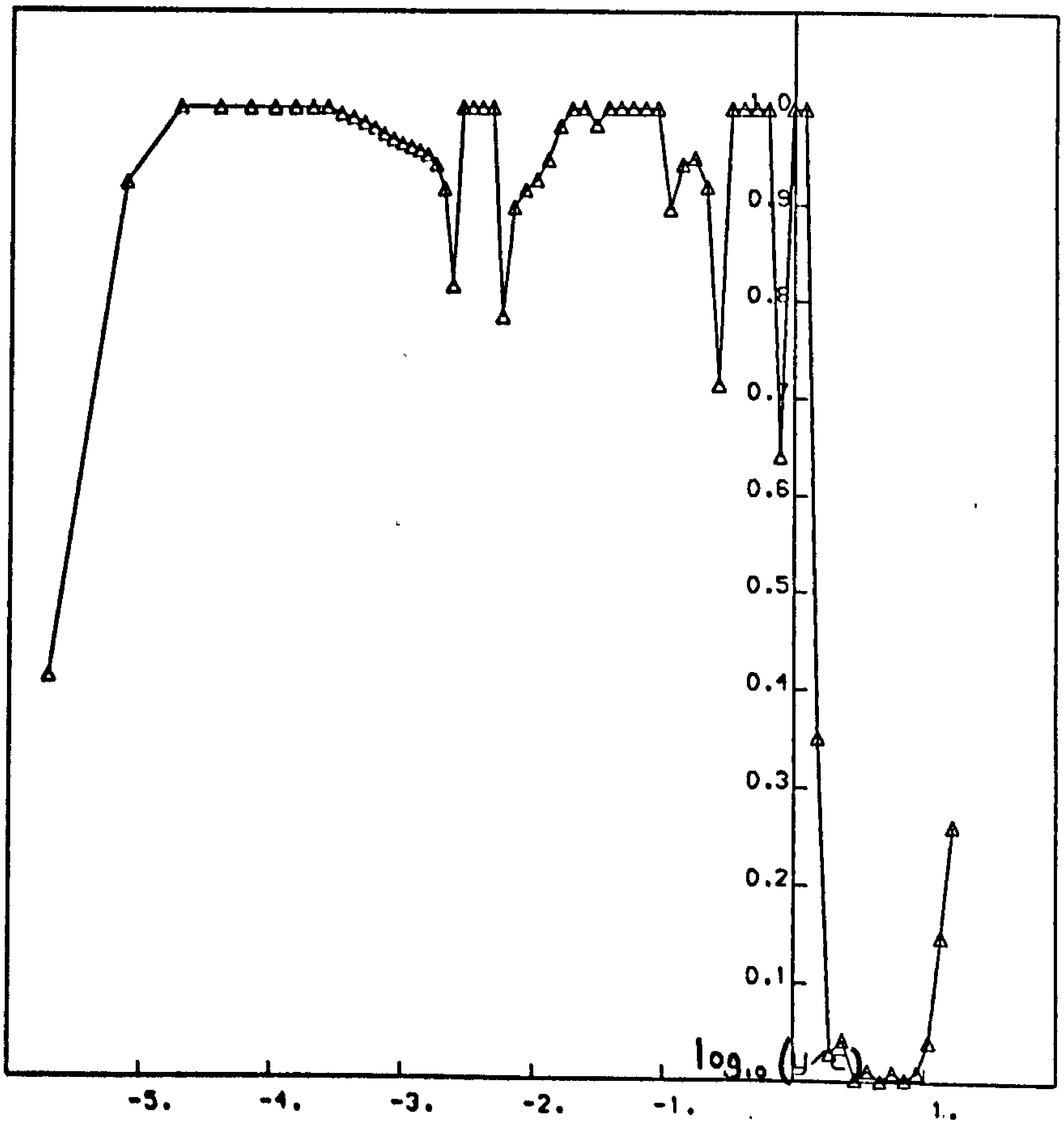


Fig.4.9 NDR - x/c = 5.0, Turbulent case - explicit algorithm.

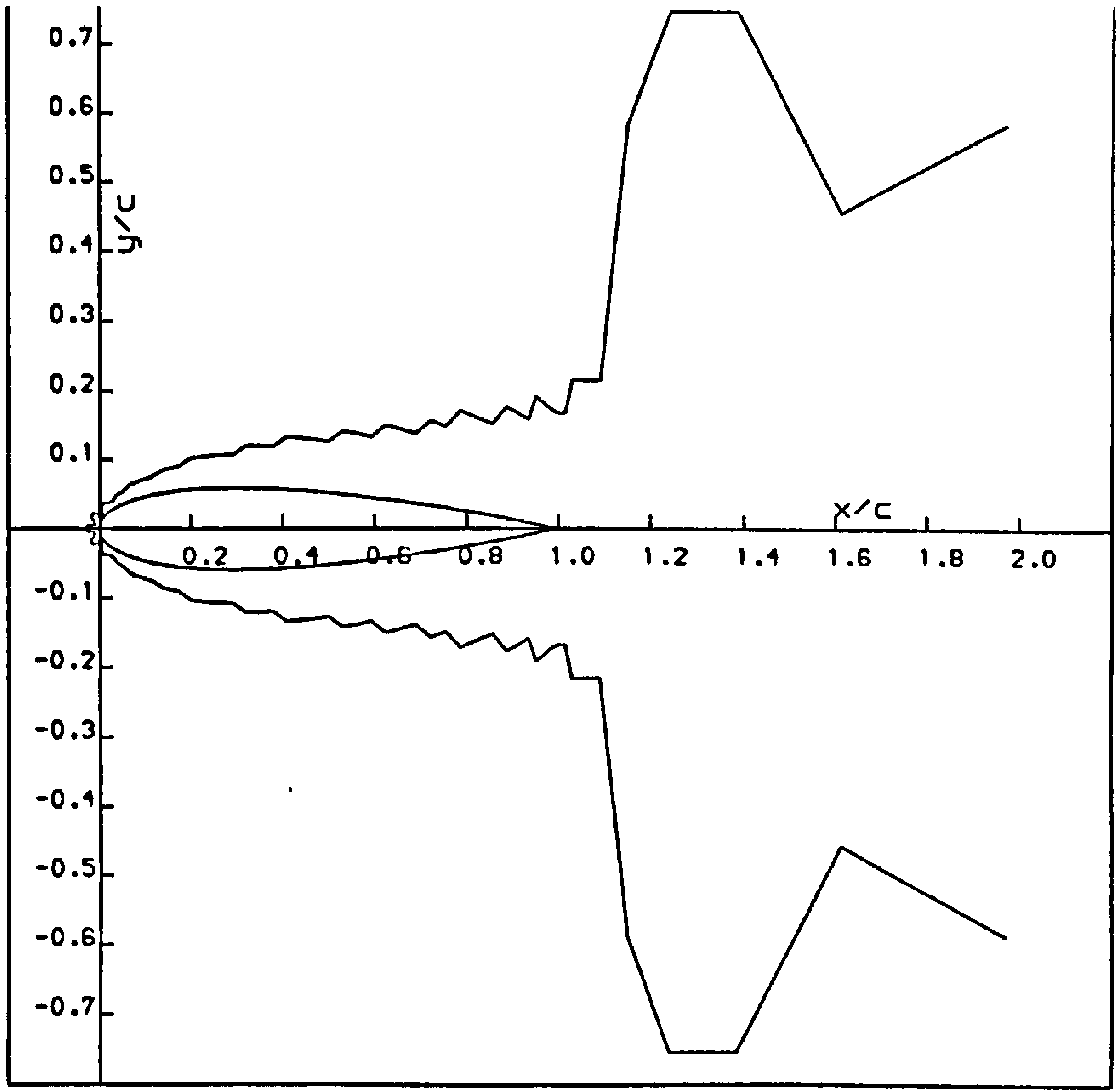


Fig.4.10 Contour for 10% NDR cutoff - Case A.

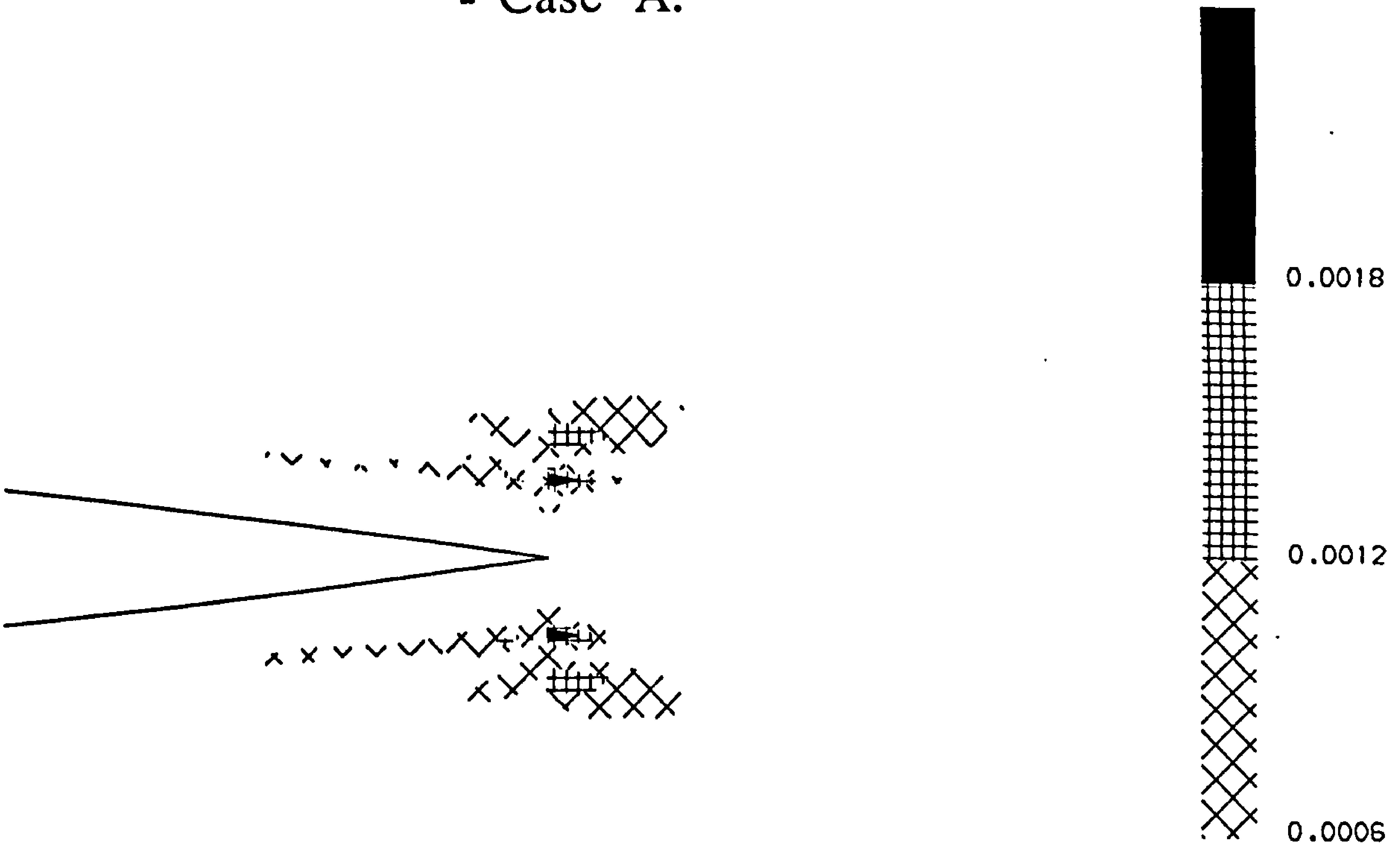


Fig.4.11 Original NDI - trailing-edge region (Grid B).

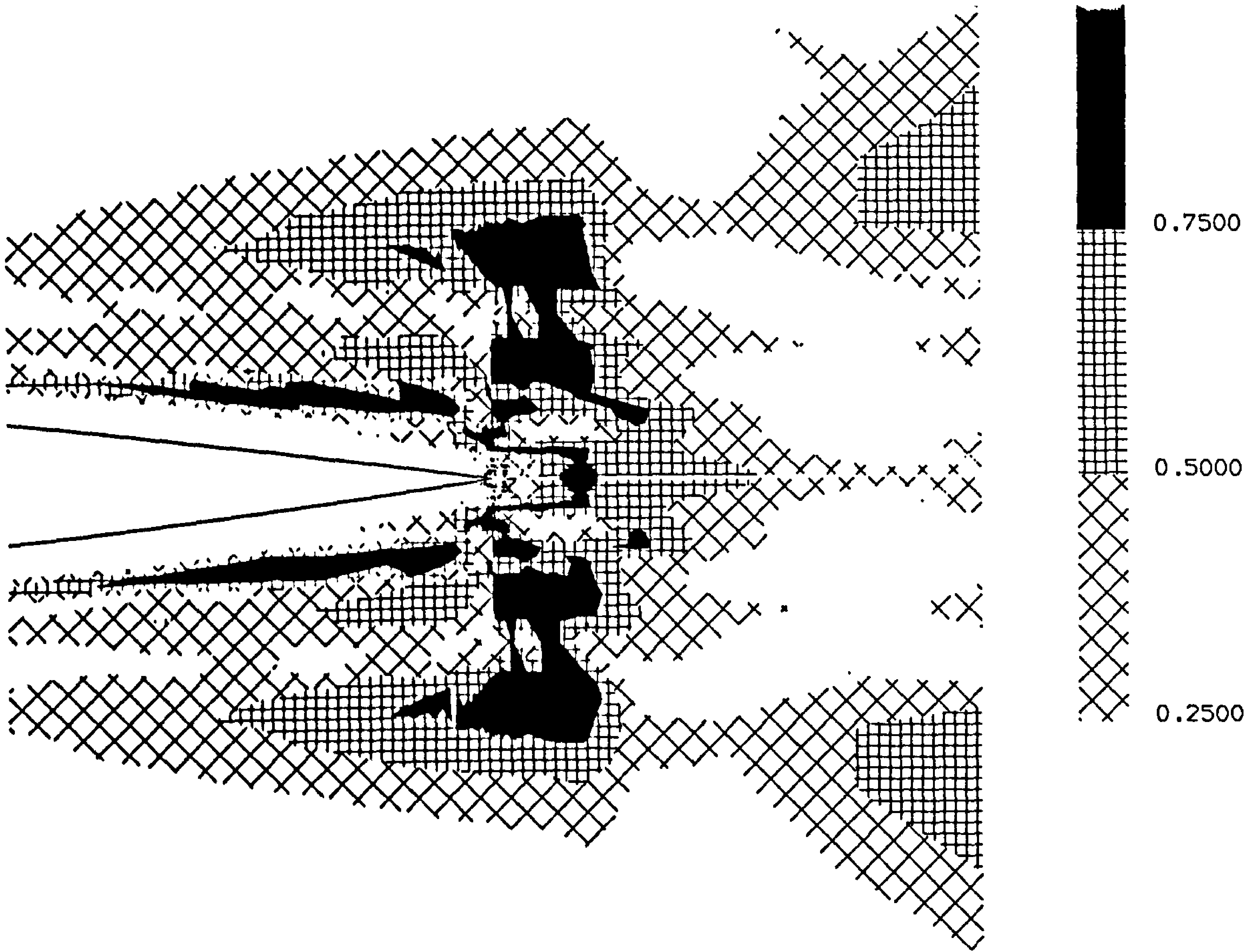


Fig.4.12 Original NDR - trailing-edge region (Grid B).

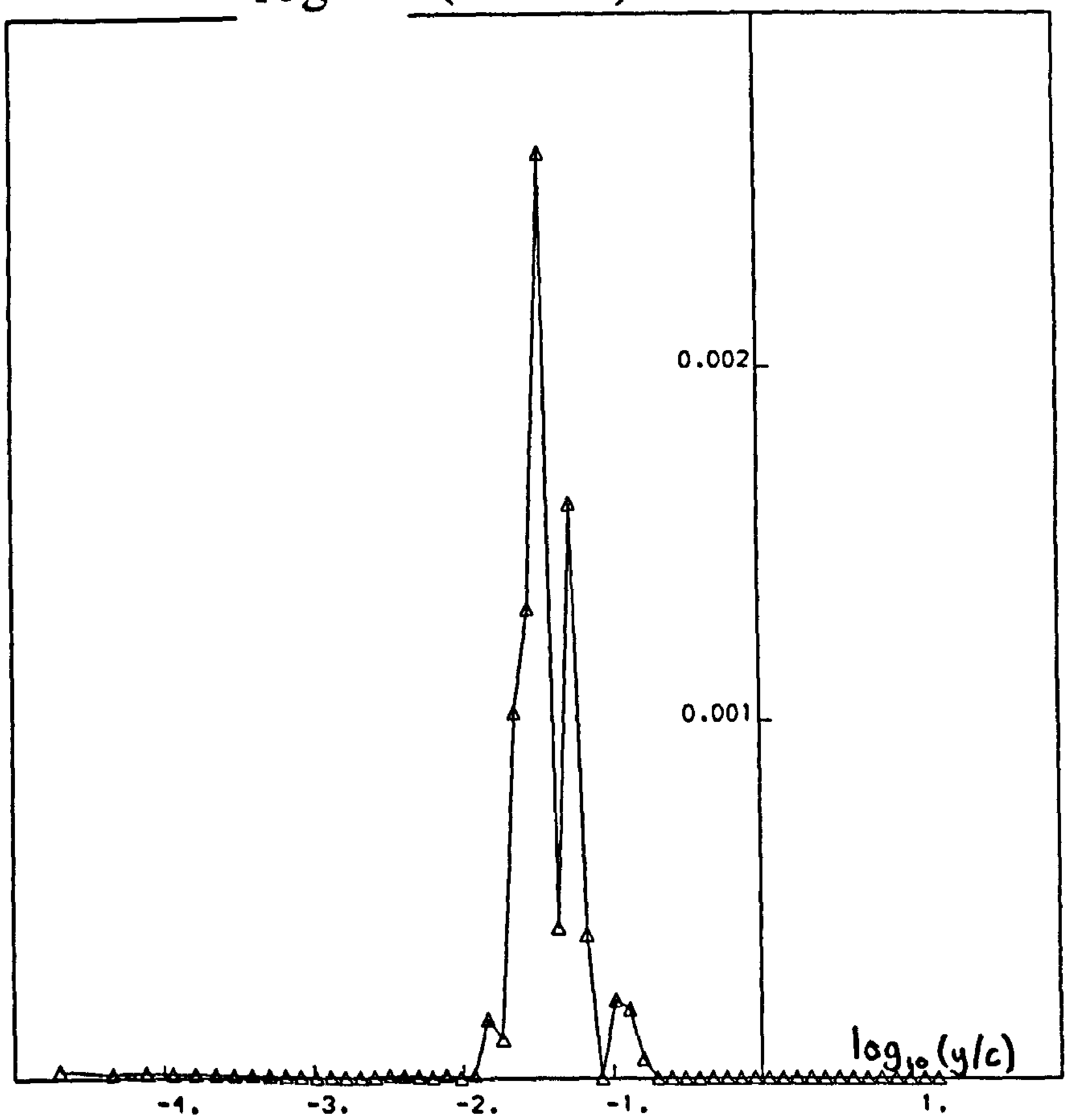


Fig.4.13 Original NDI - station 3 (Grid B).

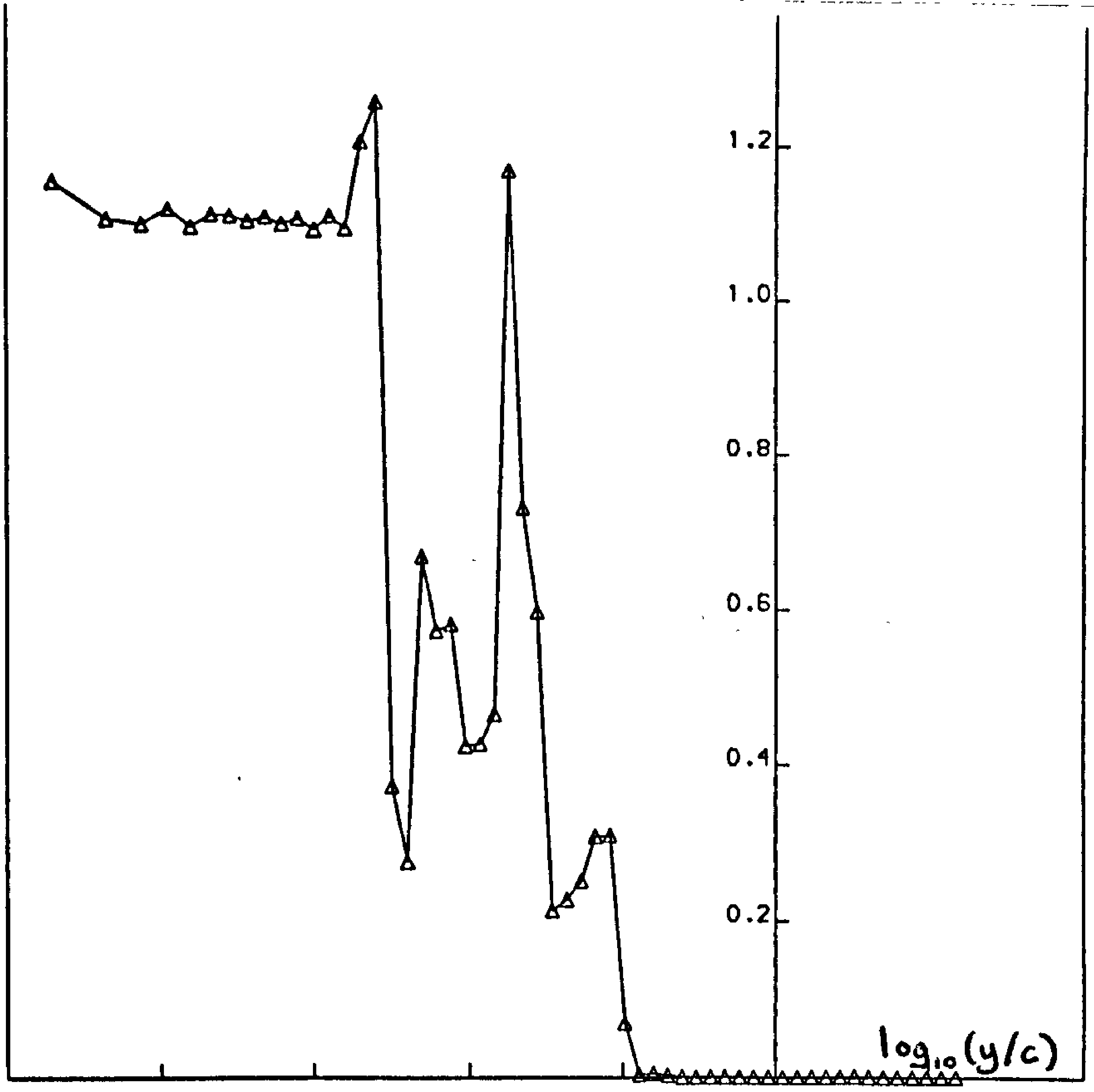


Fig.4.14 Physical diffusion / max(Convection, Pressure gradient) - station 3 (Grid B).

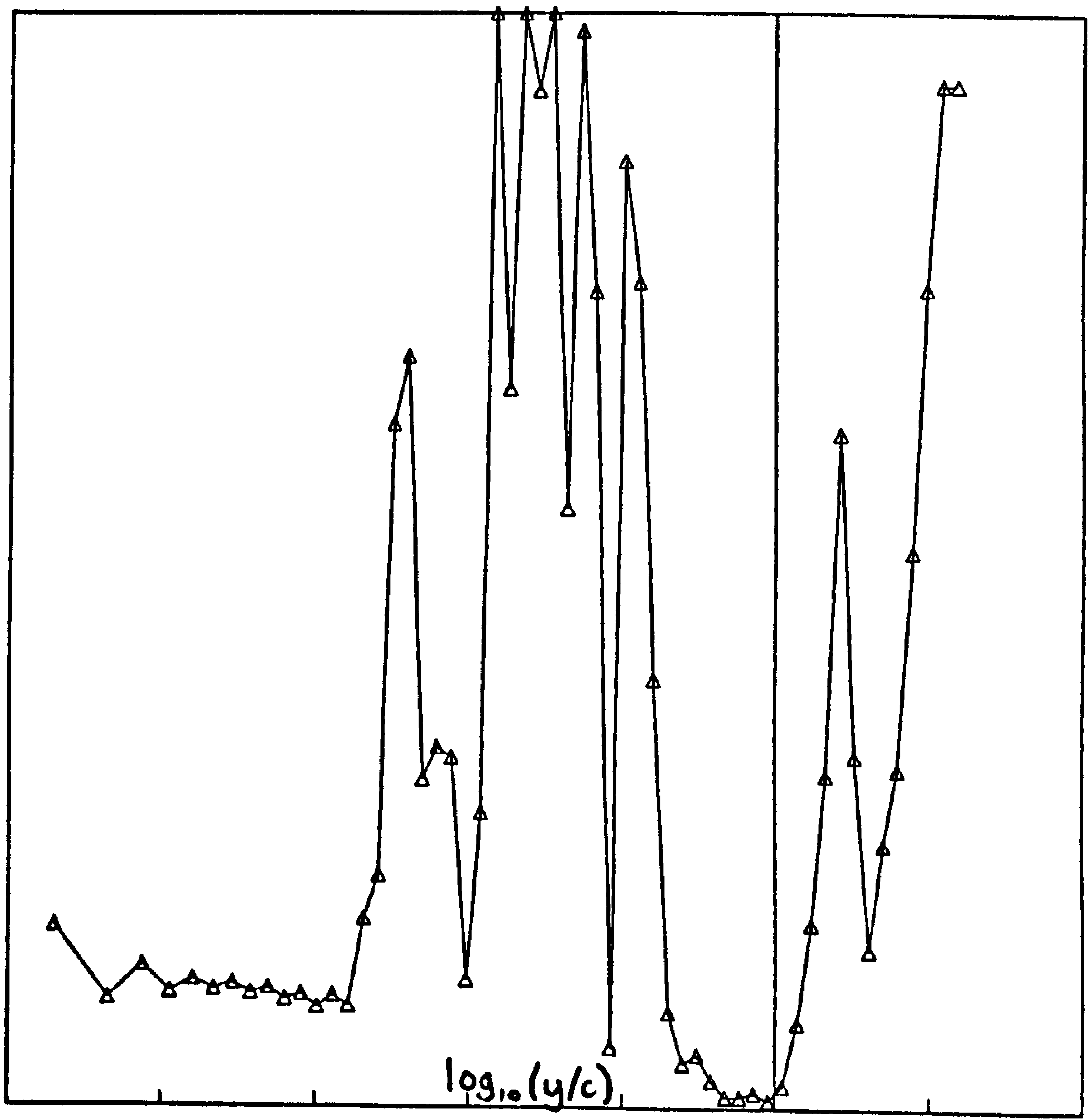


Fig.4.15 Original NDR - station 3 (Grid B).

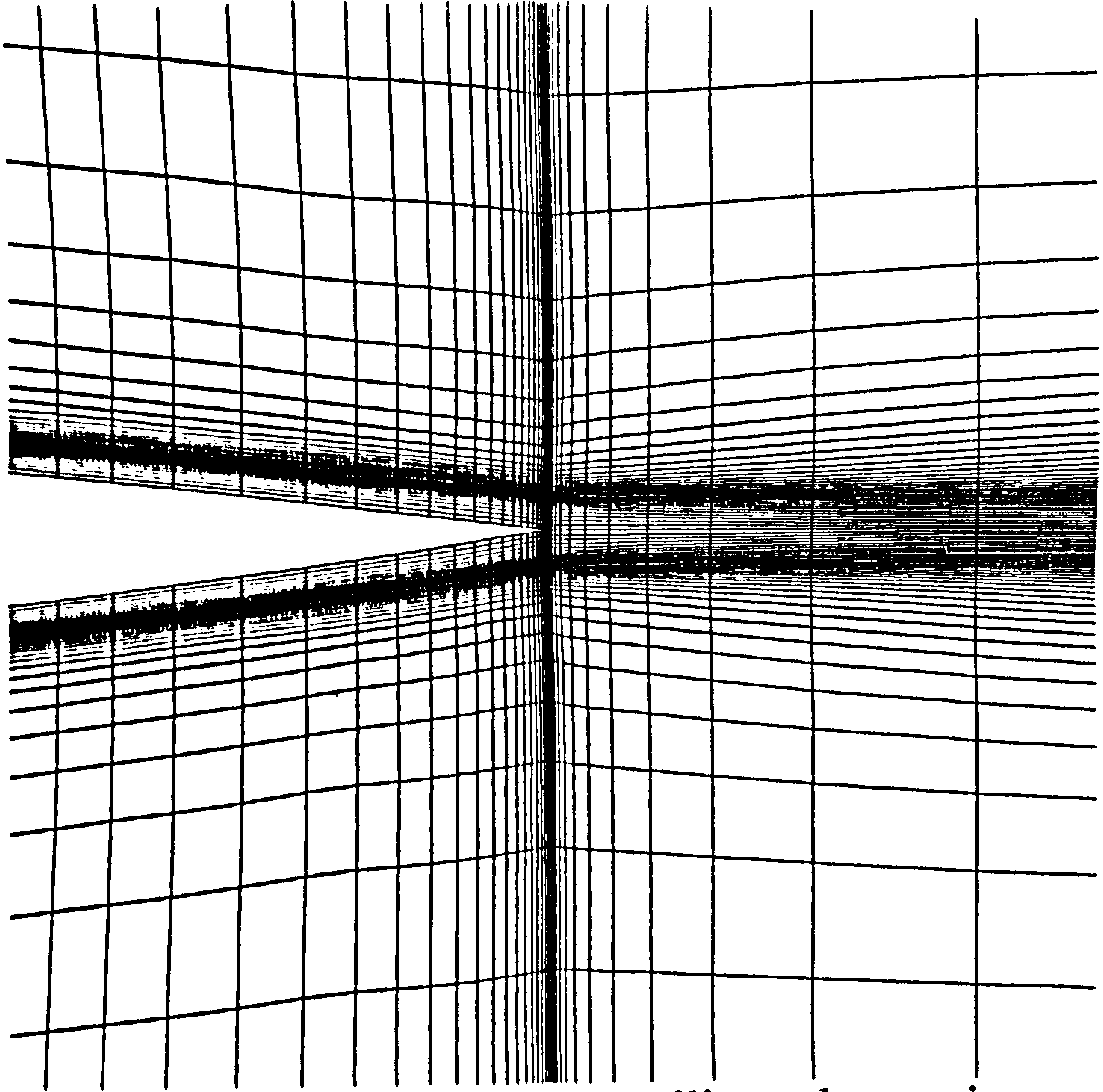


Fig.4.16 Adapted grid - trailing-edge region.

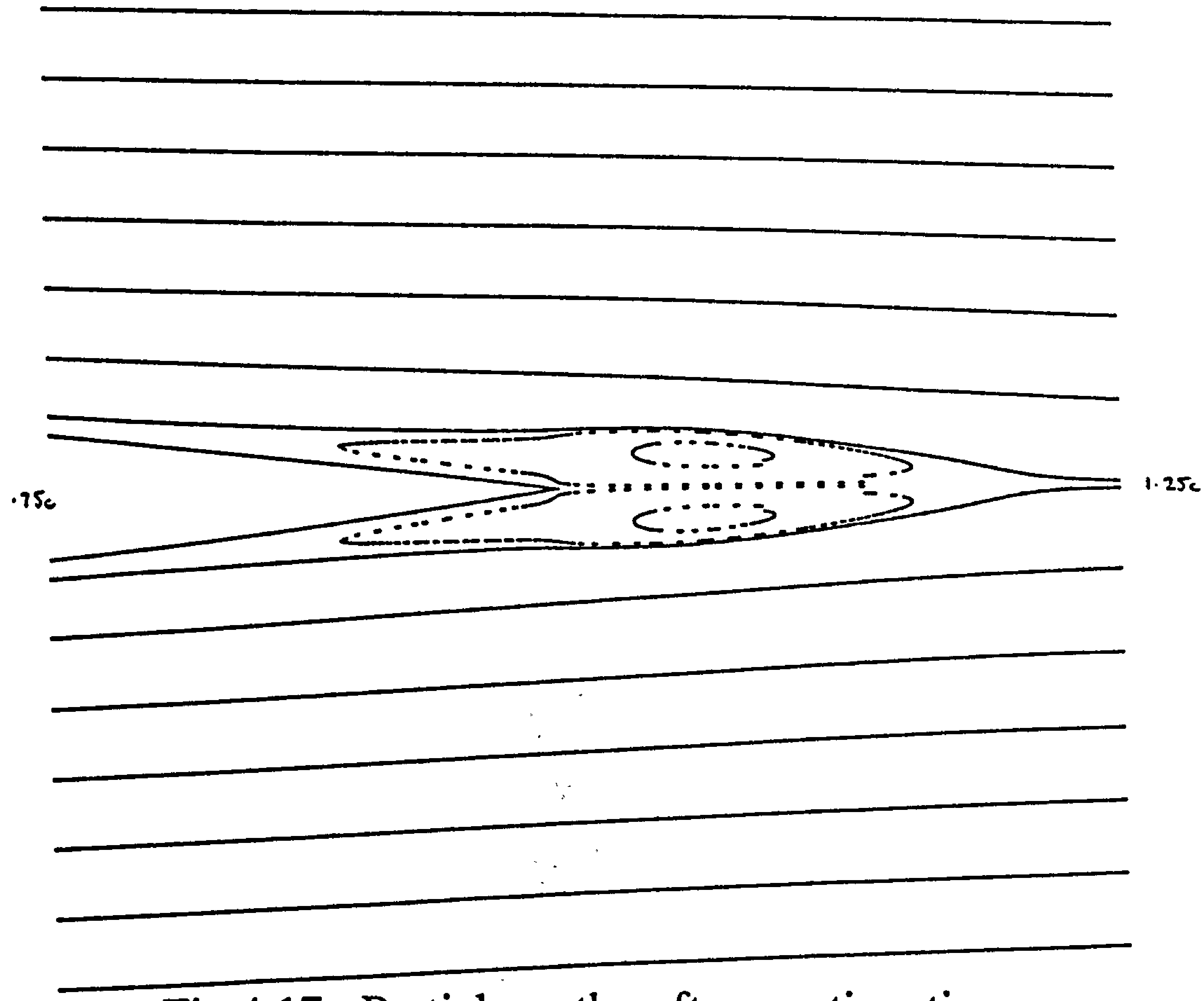


Fig.4.17 Particle paths after continuation run on adapted grid.

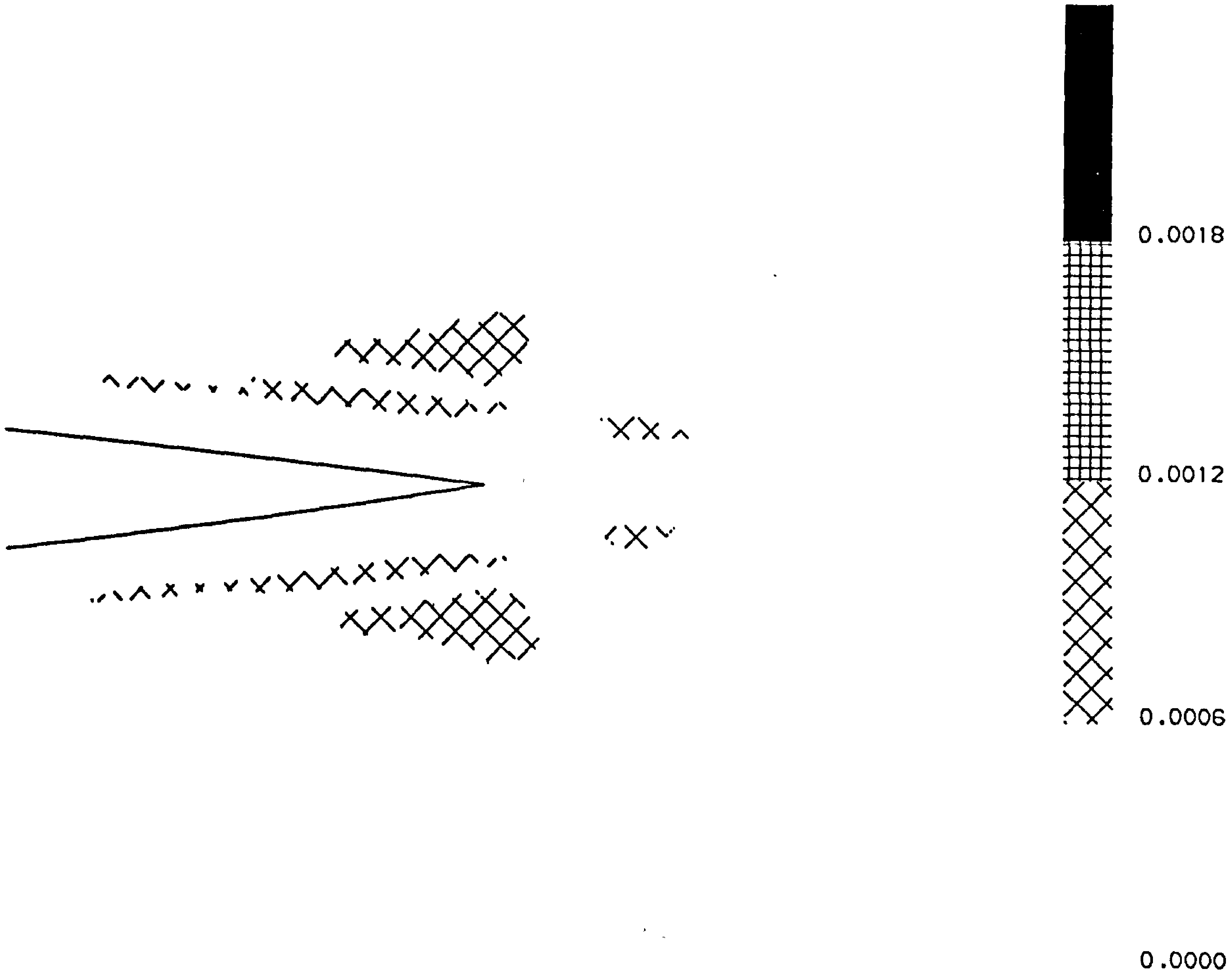


Fig.4.18 NDI after calculation on adapted grid.

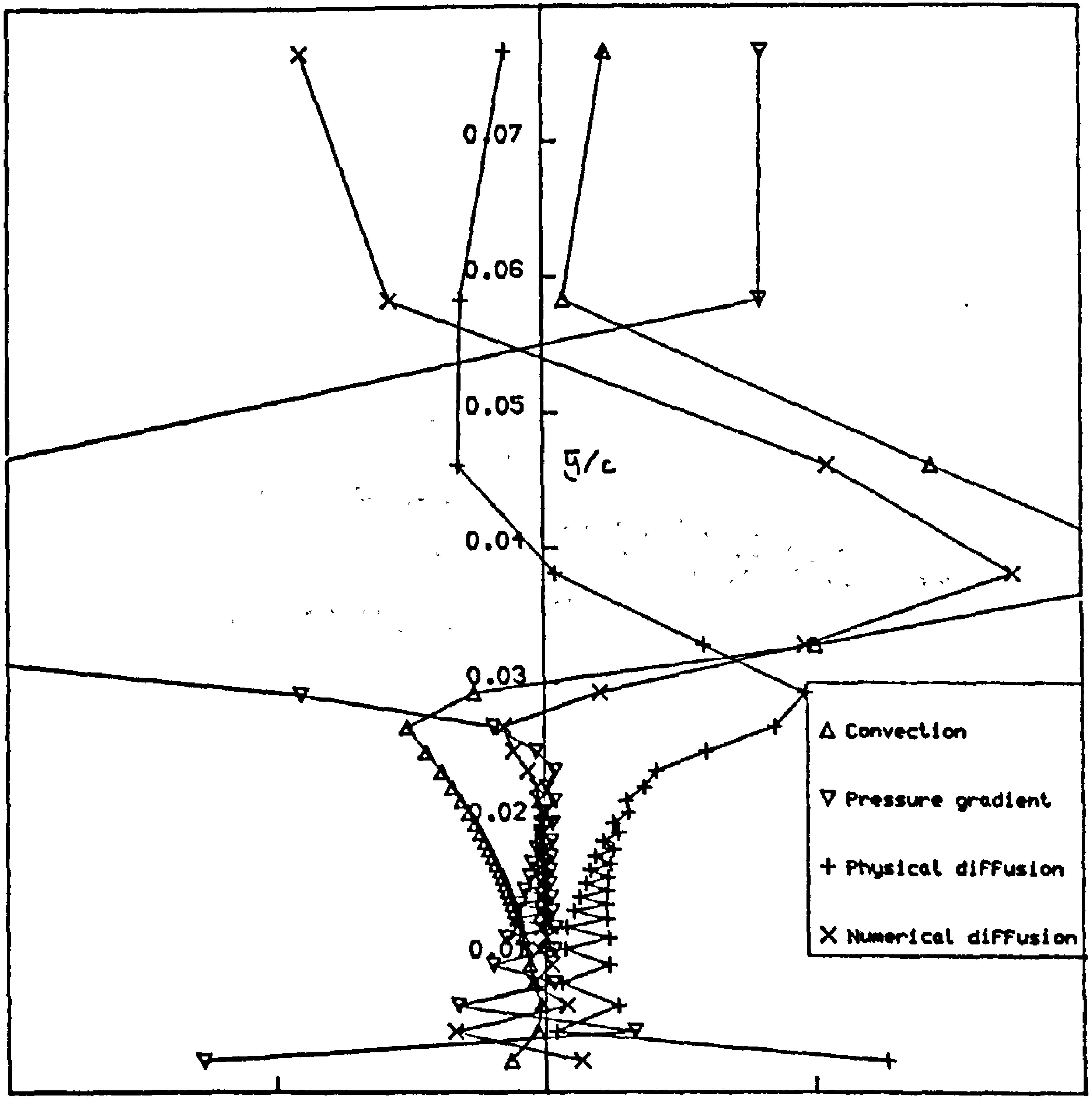


Fig.4.19 Balance of terms in solution - Adapted solution - station 3.

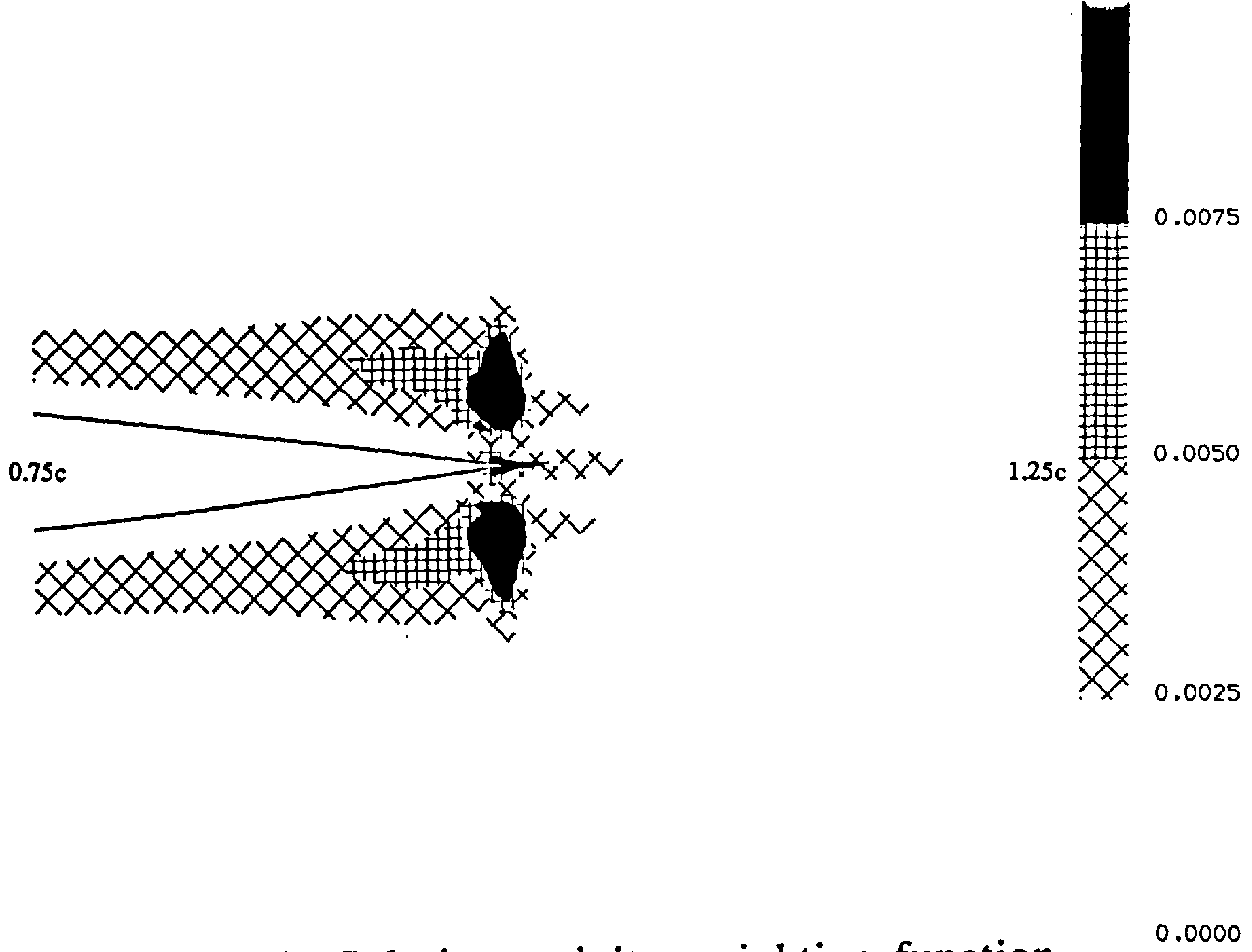


Fig.4.20a Solution-activity weighting function (streamwise direction) - Grid A.

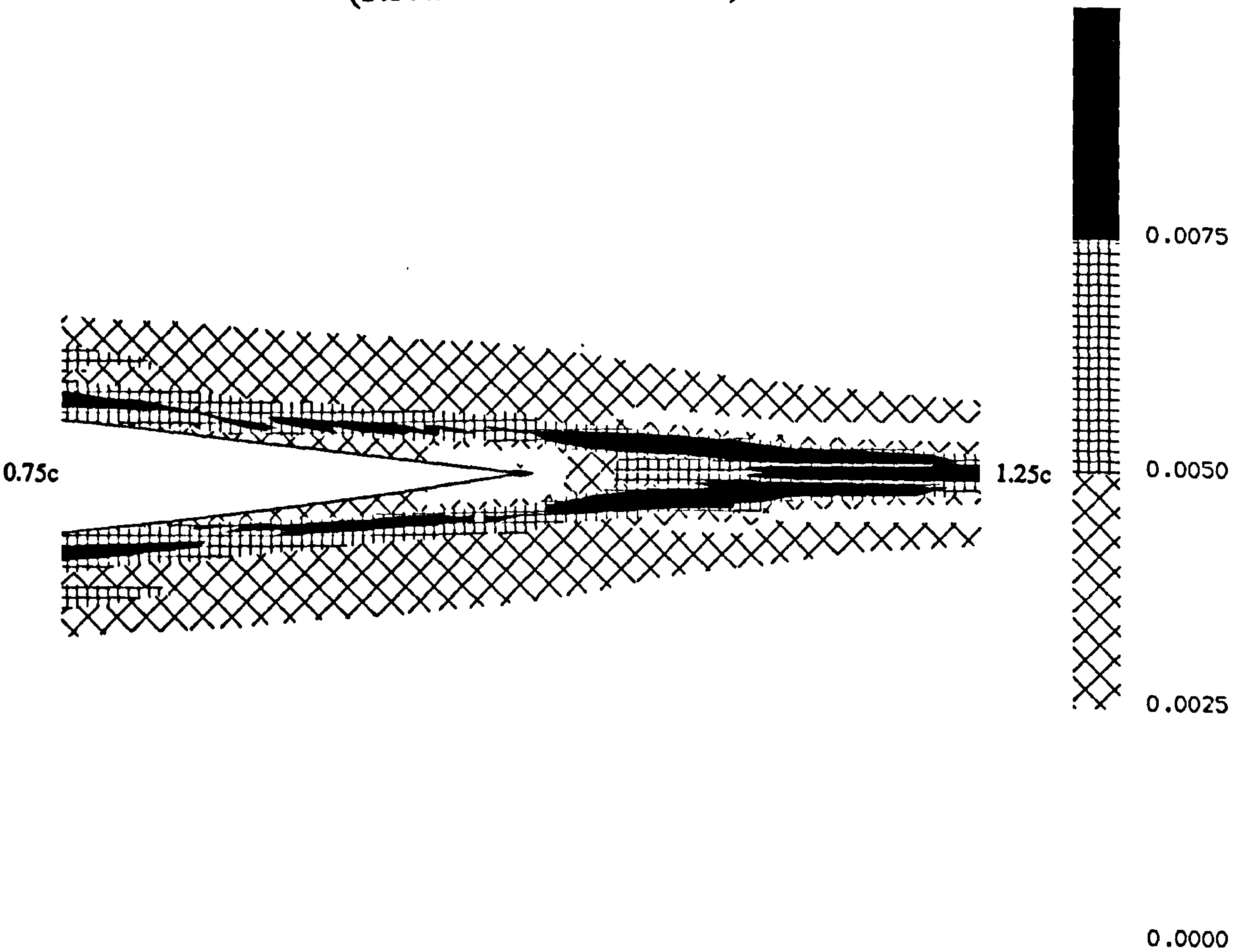


Fig.4.20b Solution-activity weighting function (normal direction) - Grid A.

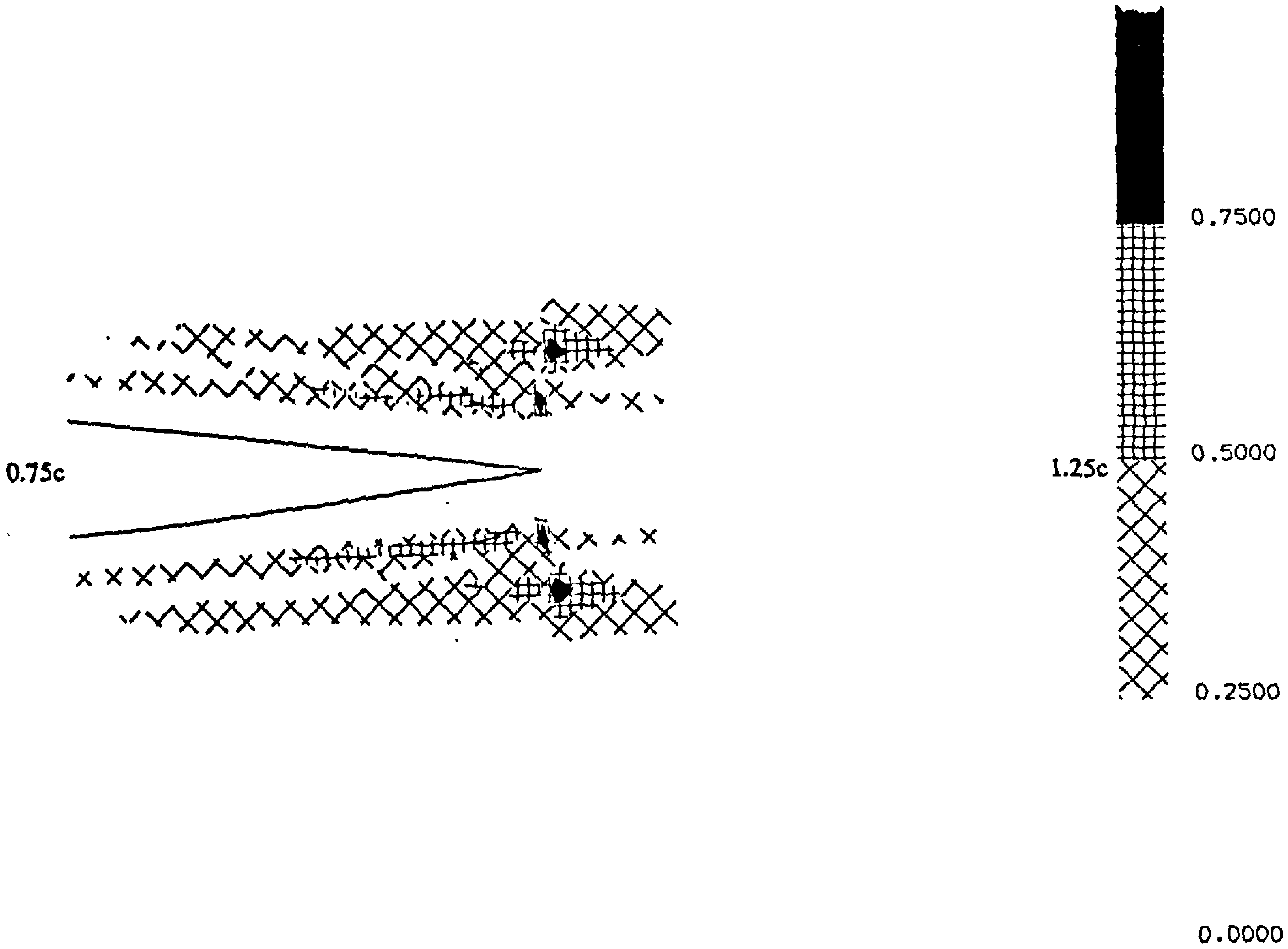


Fig.4.21 NDI - Grid A.

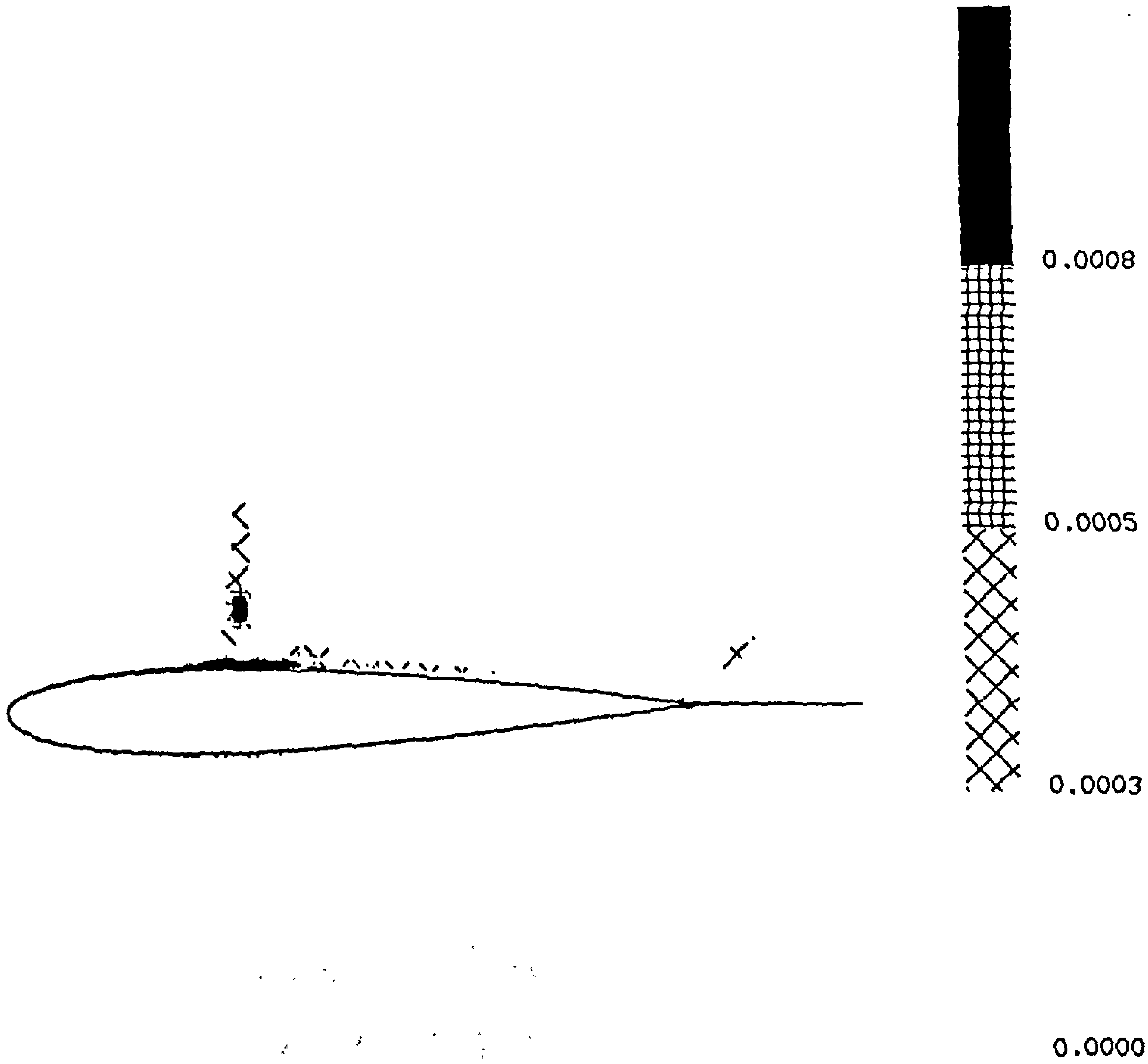


Fig.4.22 NDI - Turbulent case (5°)
- explicit algorithm.

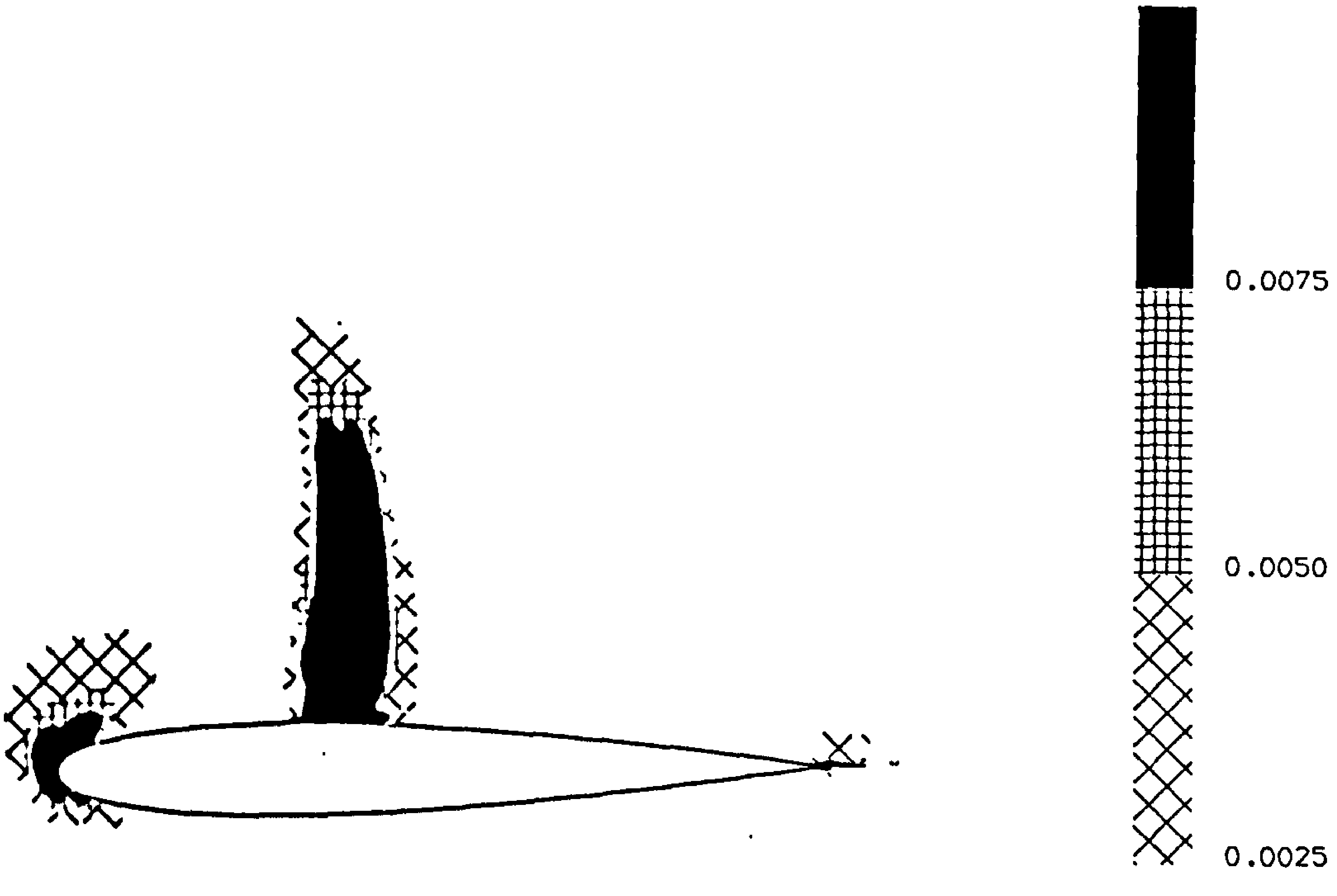


Fig.4.23 Solution-activity weighting function - Turbulent case (5°) - explicit algorithm.

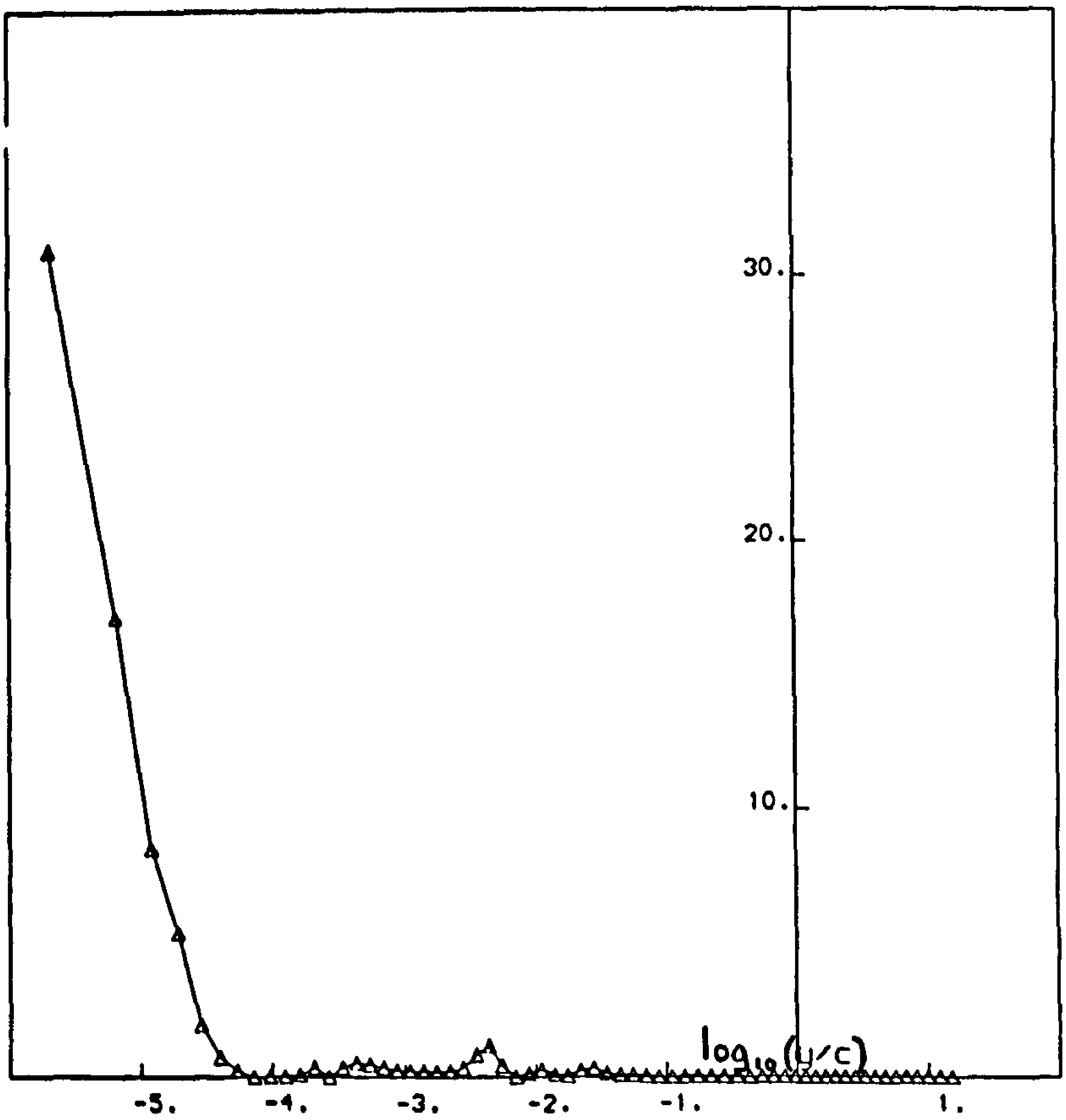


Fig.4.24 NDI - x/c = 0.45, Turbulent case (5°) - explicit algorithm.

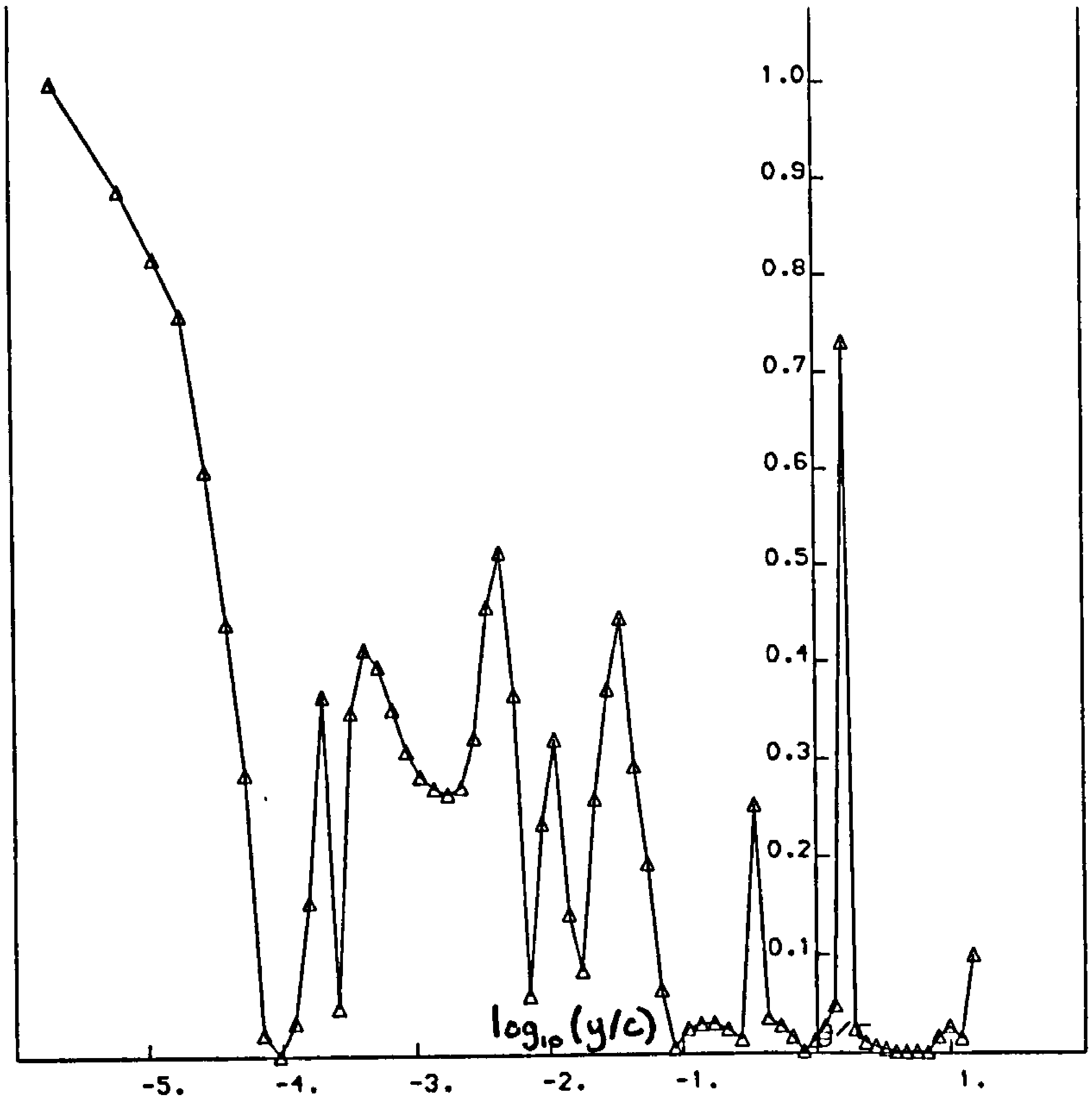


Fig.4.25 NDR - x/c = 0.45, Turbulent case (5°)
- explicit algorithm.

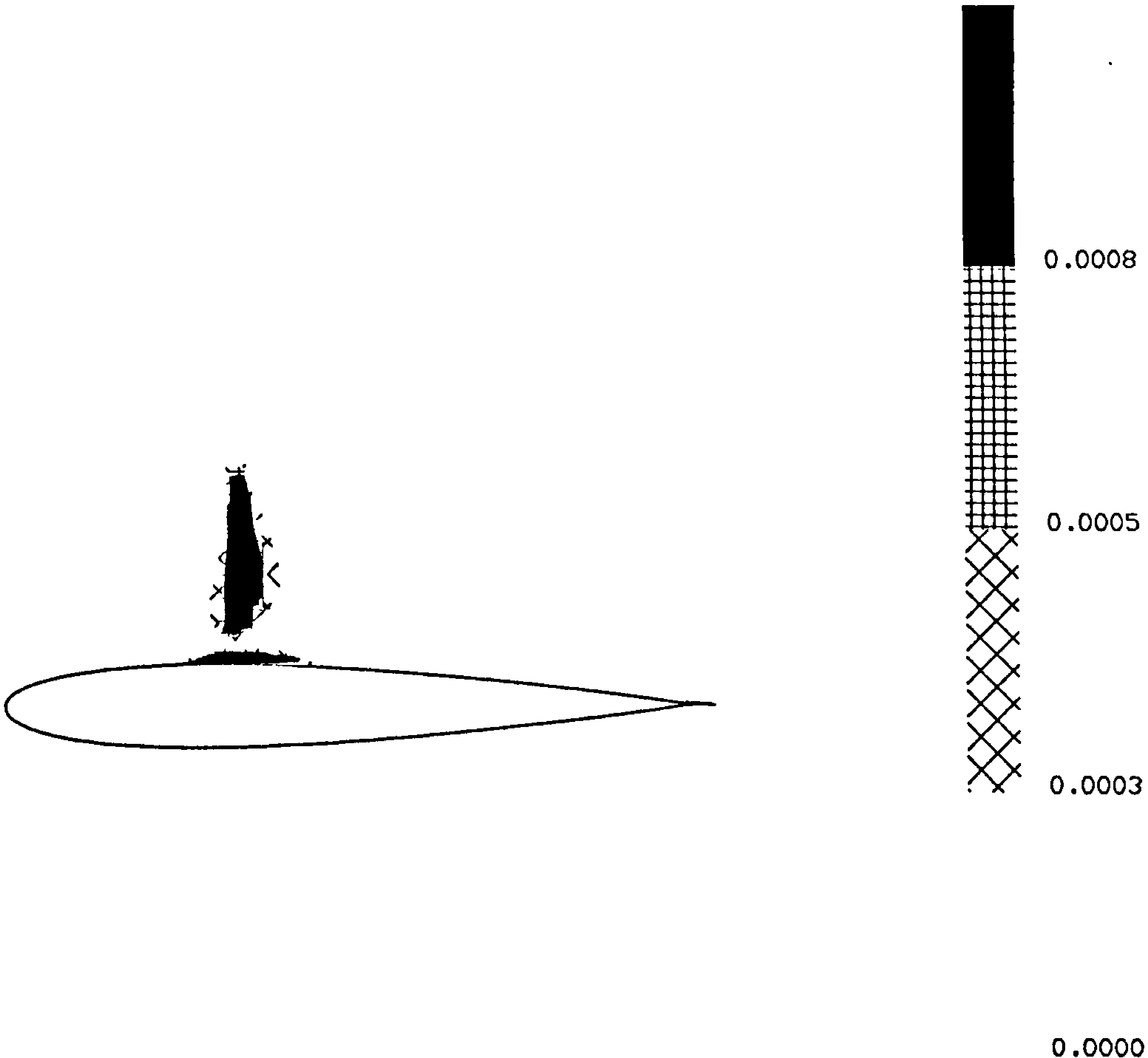
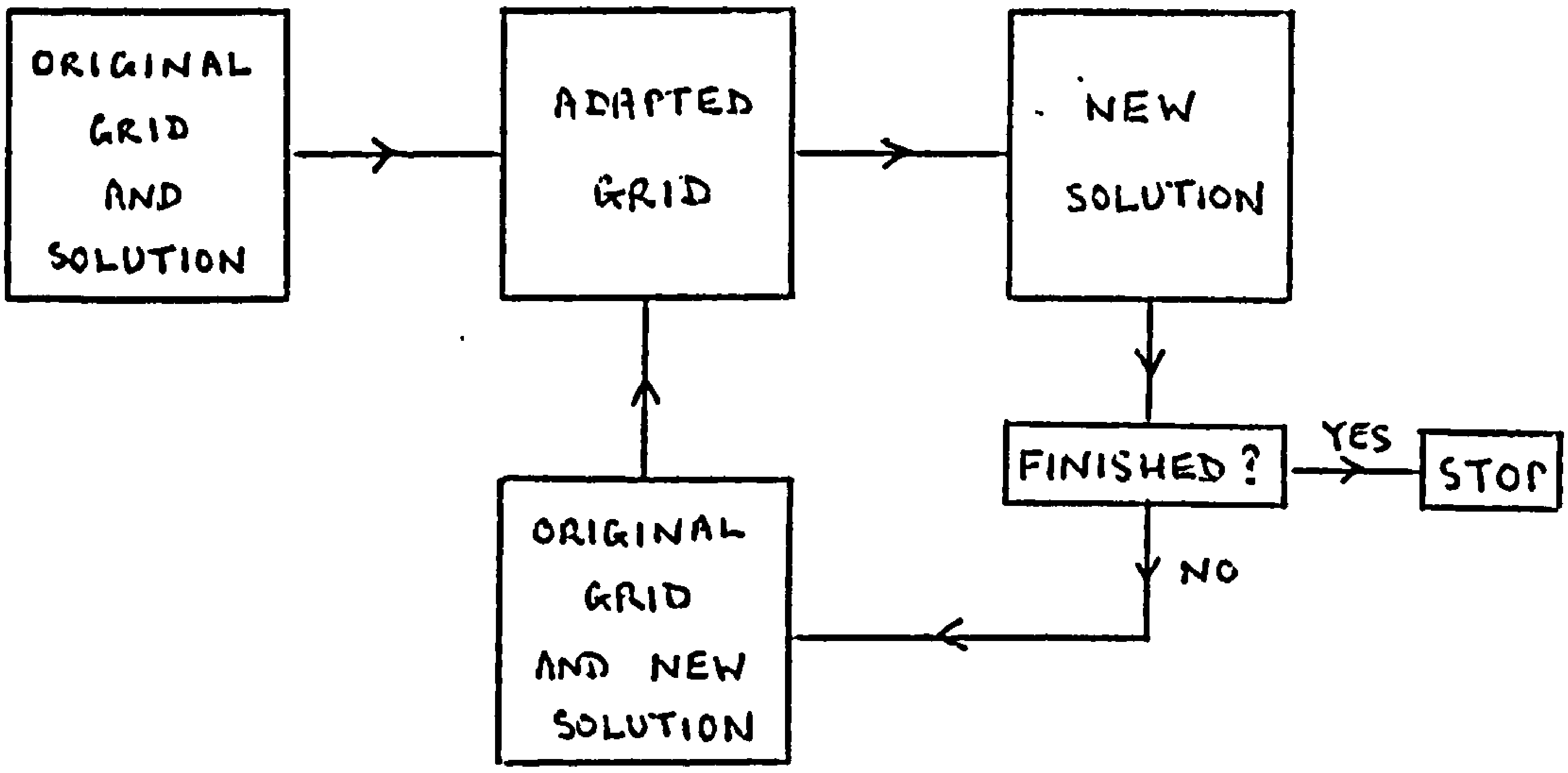
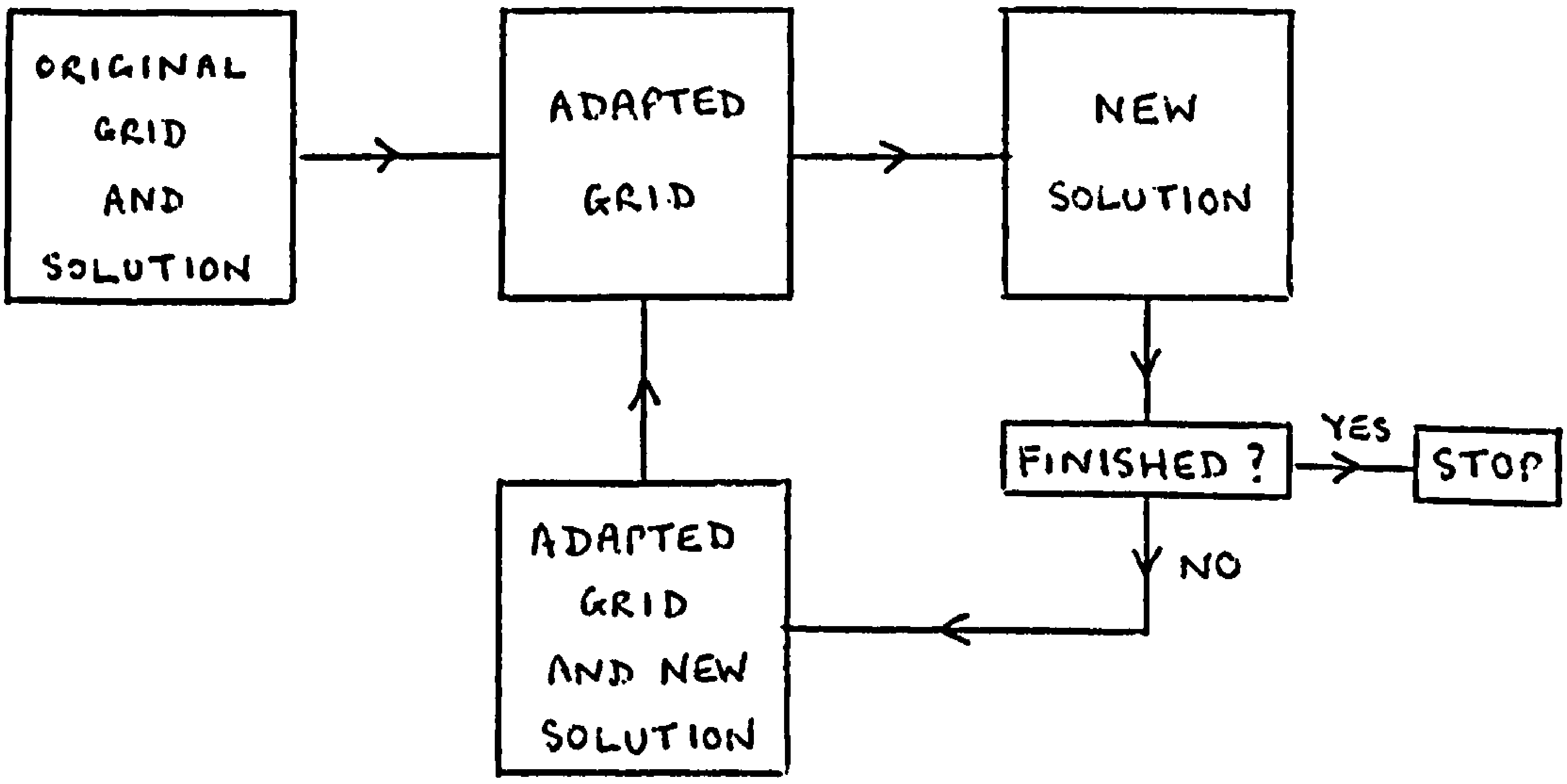


Fig.4.26 NDFξ - Turbulent case (5°)
- explicit algorithm.



NORMAL METHOD

Fig.5.1 Alternative grid adaptation strategies.



ALTERNATIVE METHOD

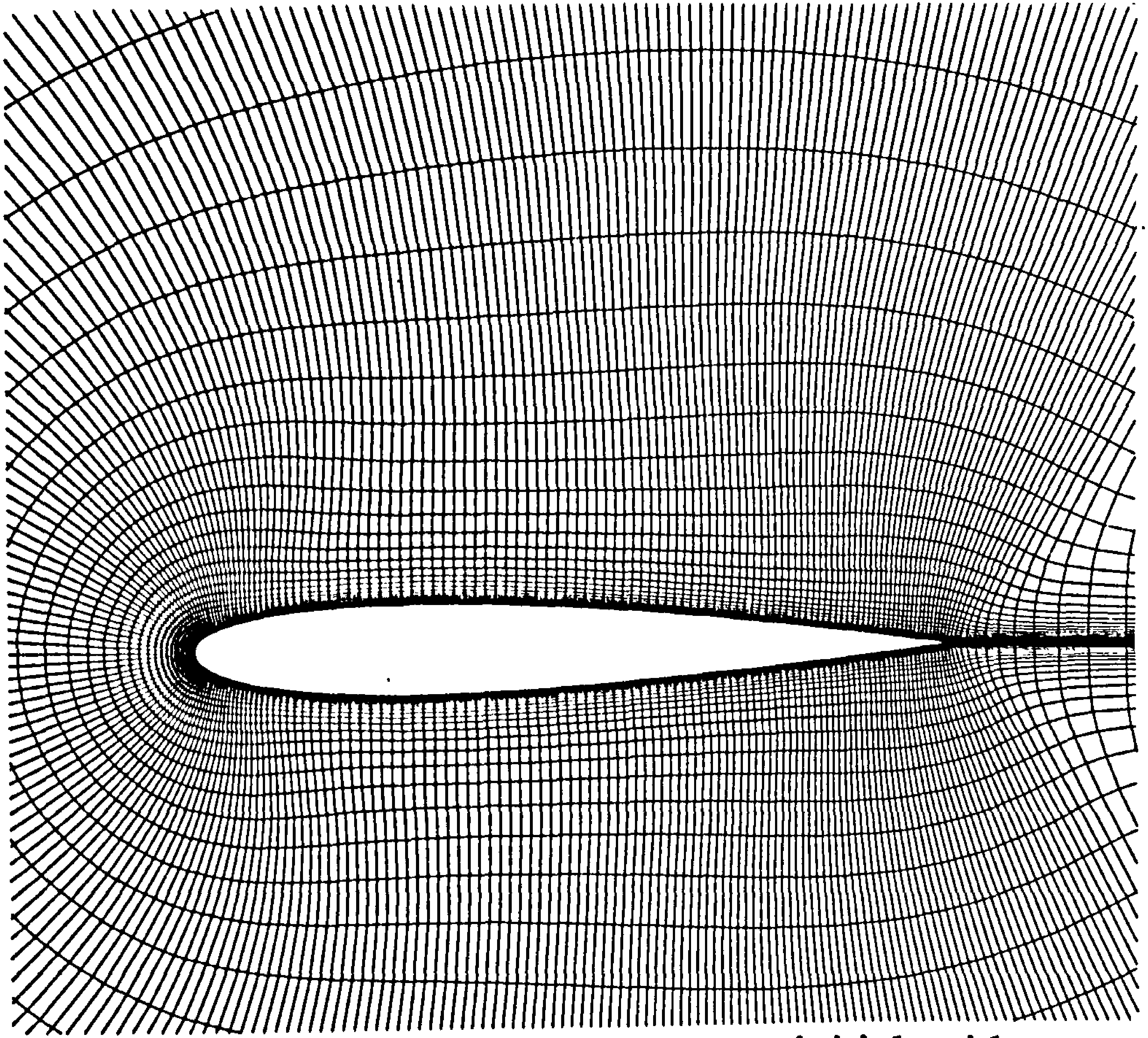


Fig.5.2 Turbulent (3°) case - initial grid.

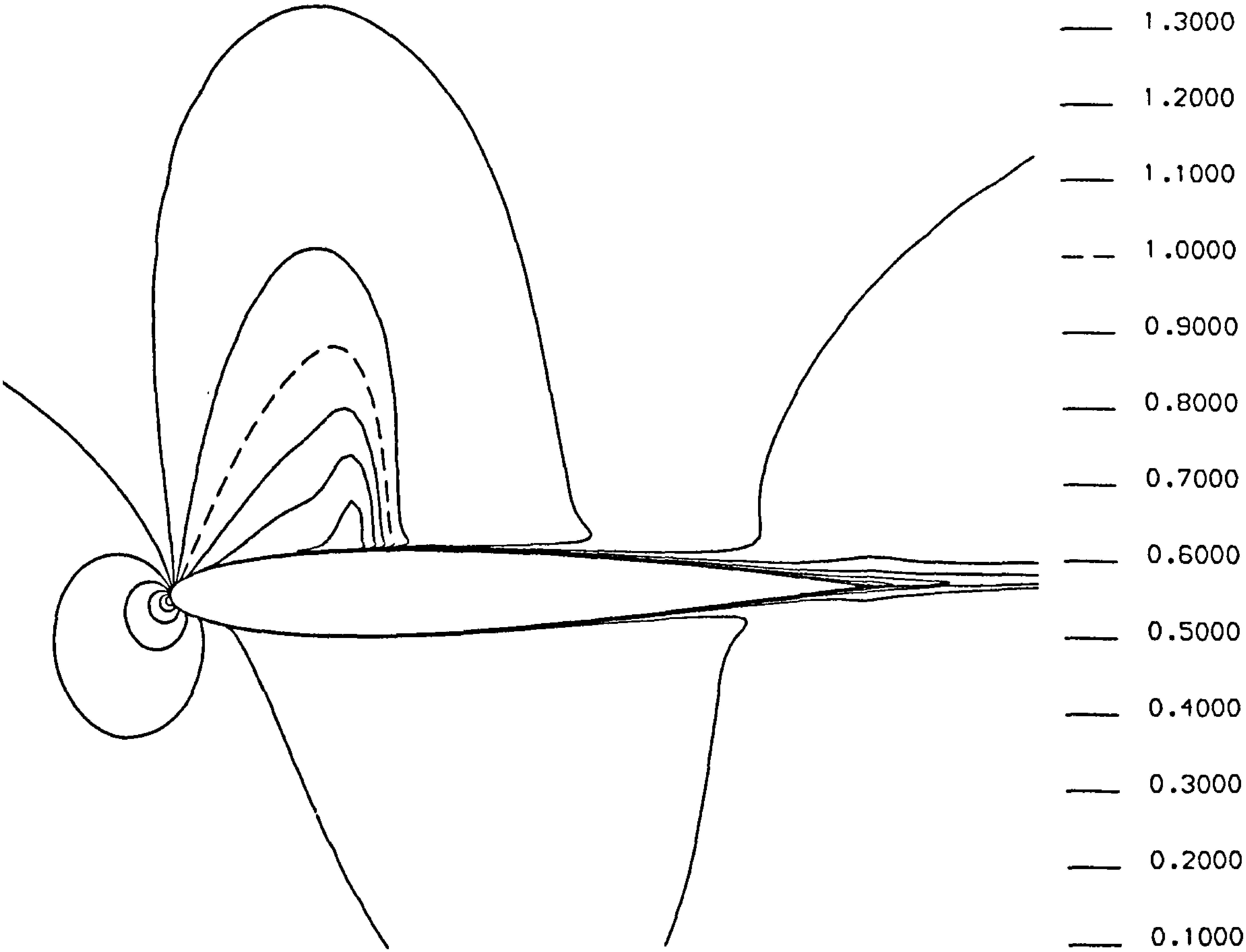


Fig.5.3 Turbulent (3°) case - initial Mach number contours.

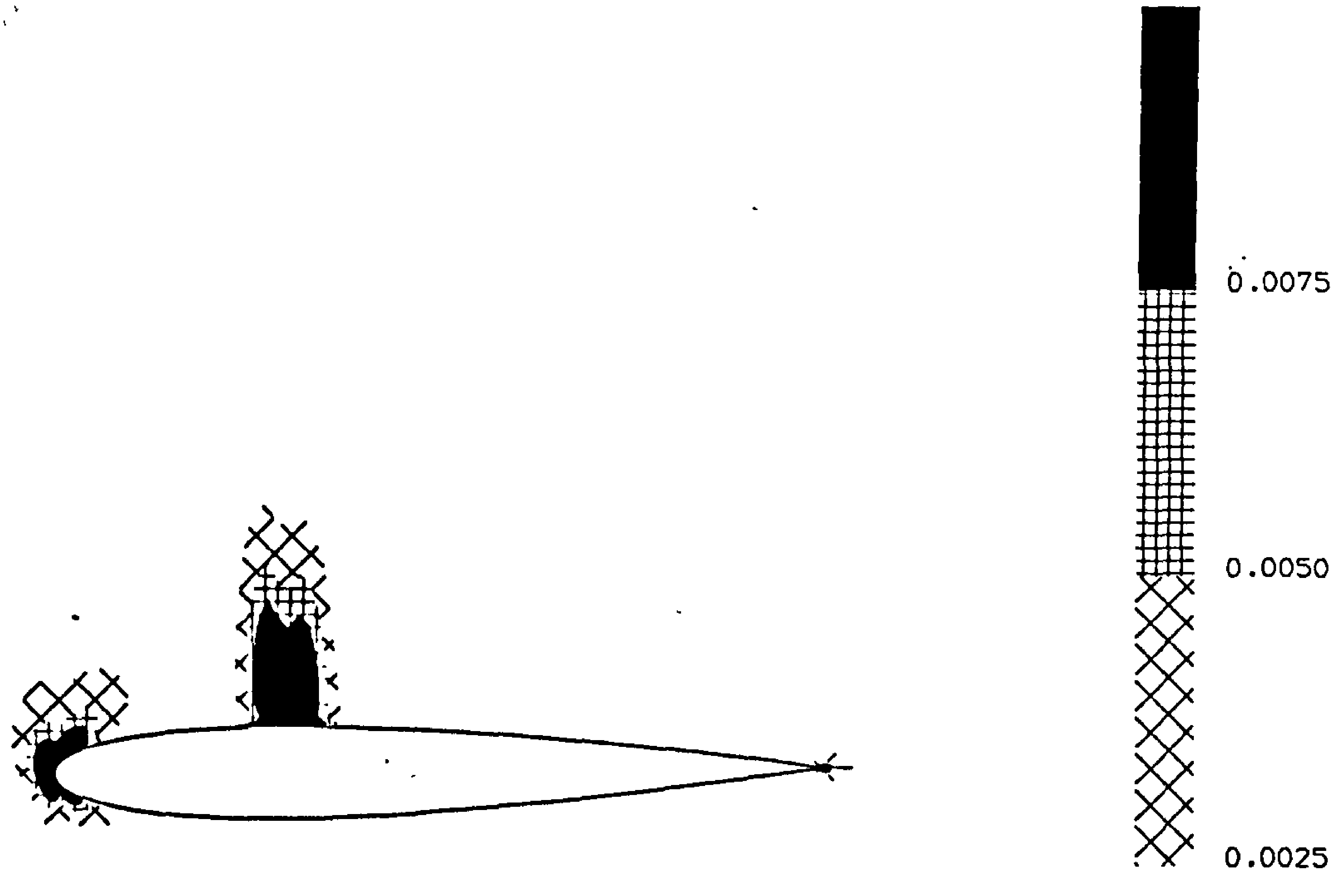


Fig.5.4 Turbulent (3°) case - DMDP (solution activity) weighting function on initial grid.

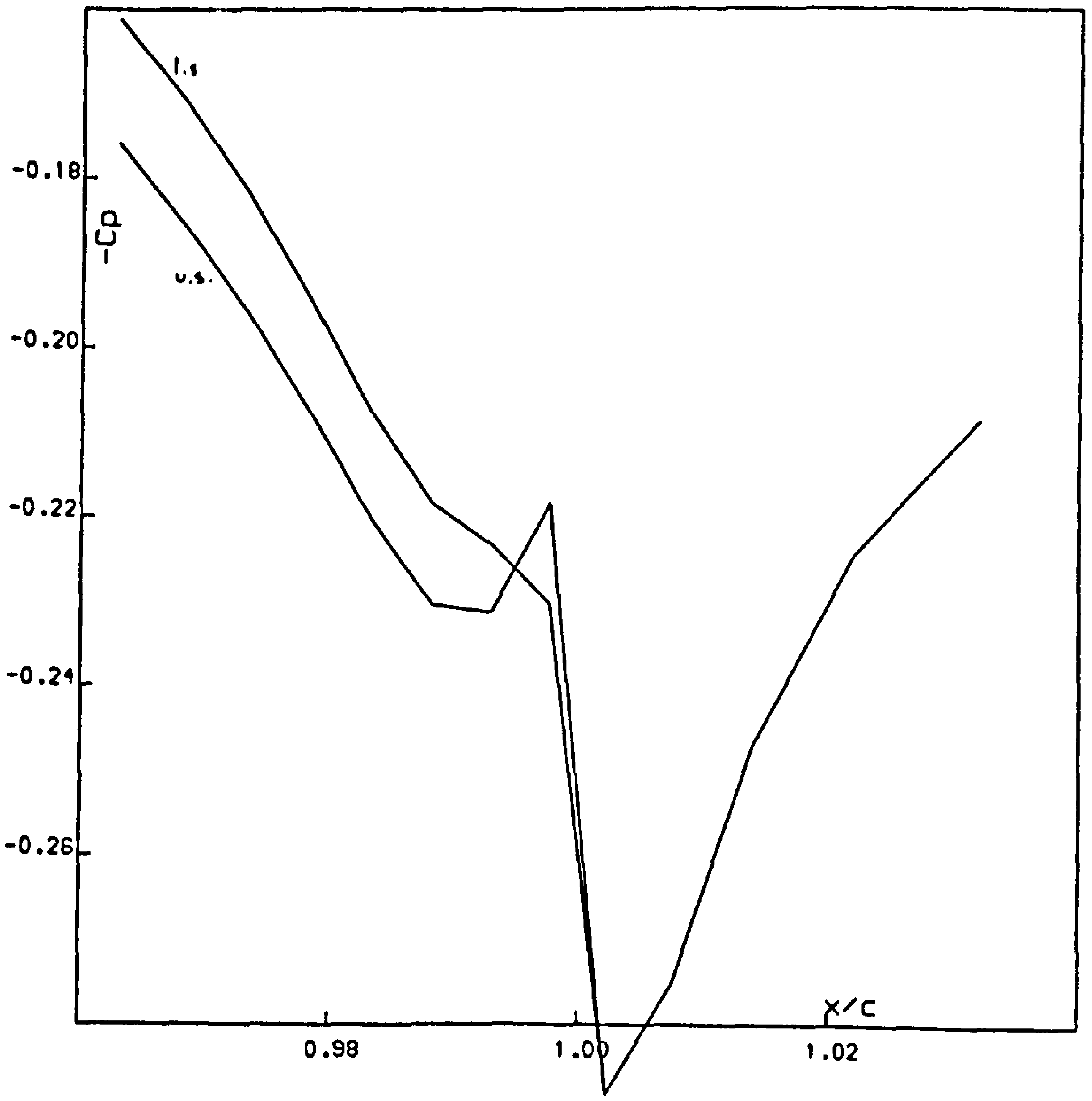


Fig.5.5 Turbulent (3°) case - C_p plot for calculation on initial grid.

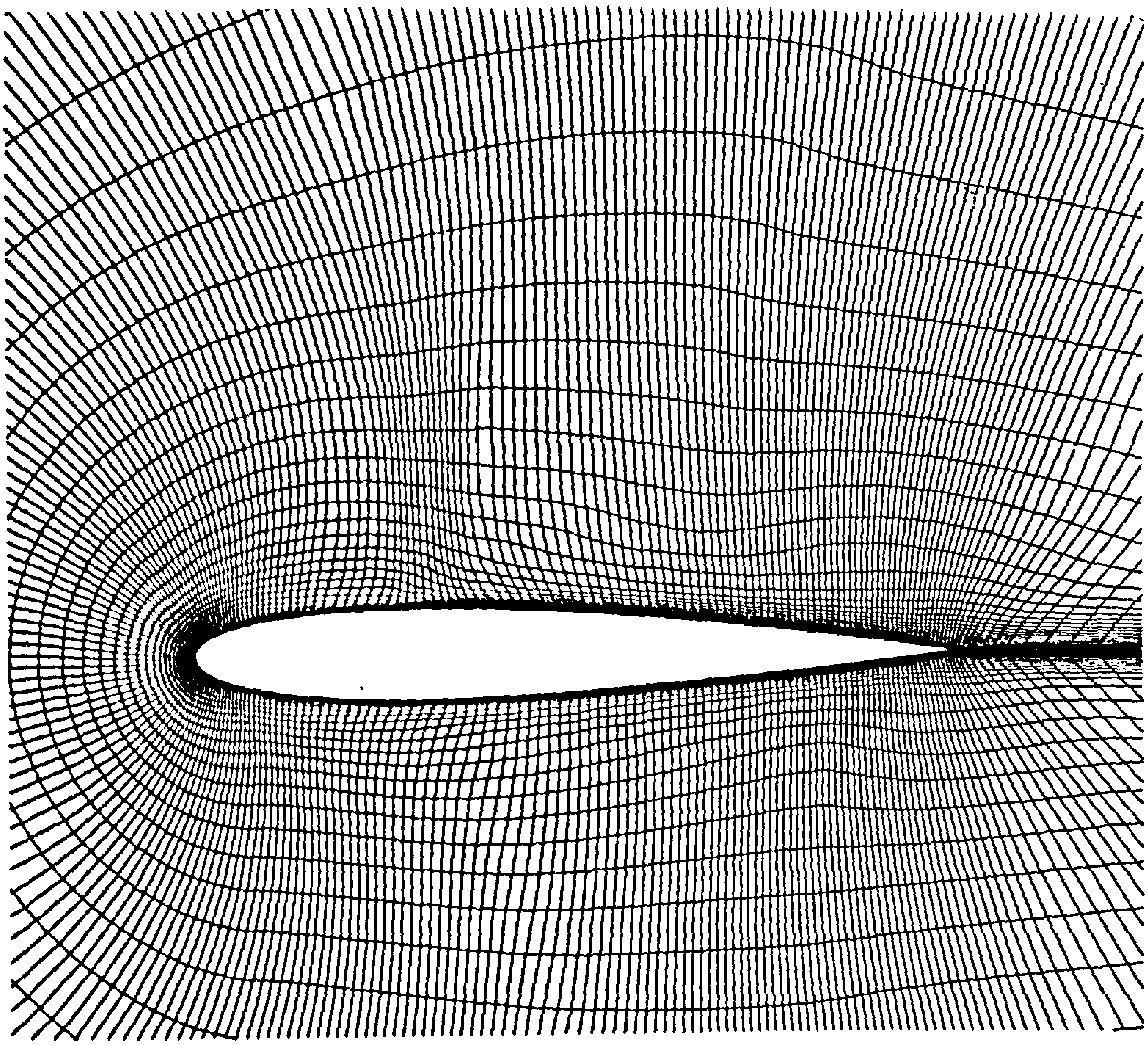


Fig.5.6 Turbulent (3°) case - Grid after first adaptation to DMDP.

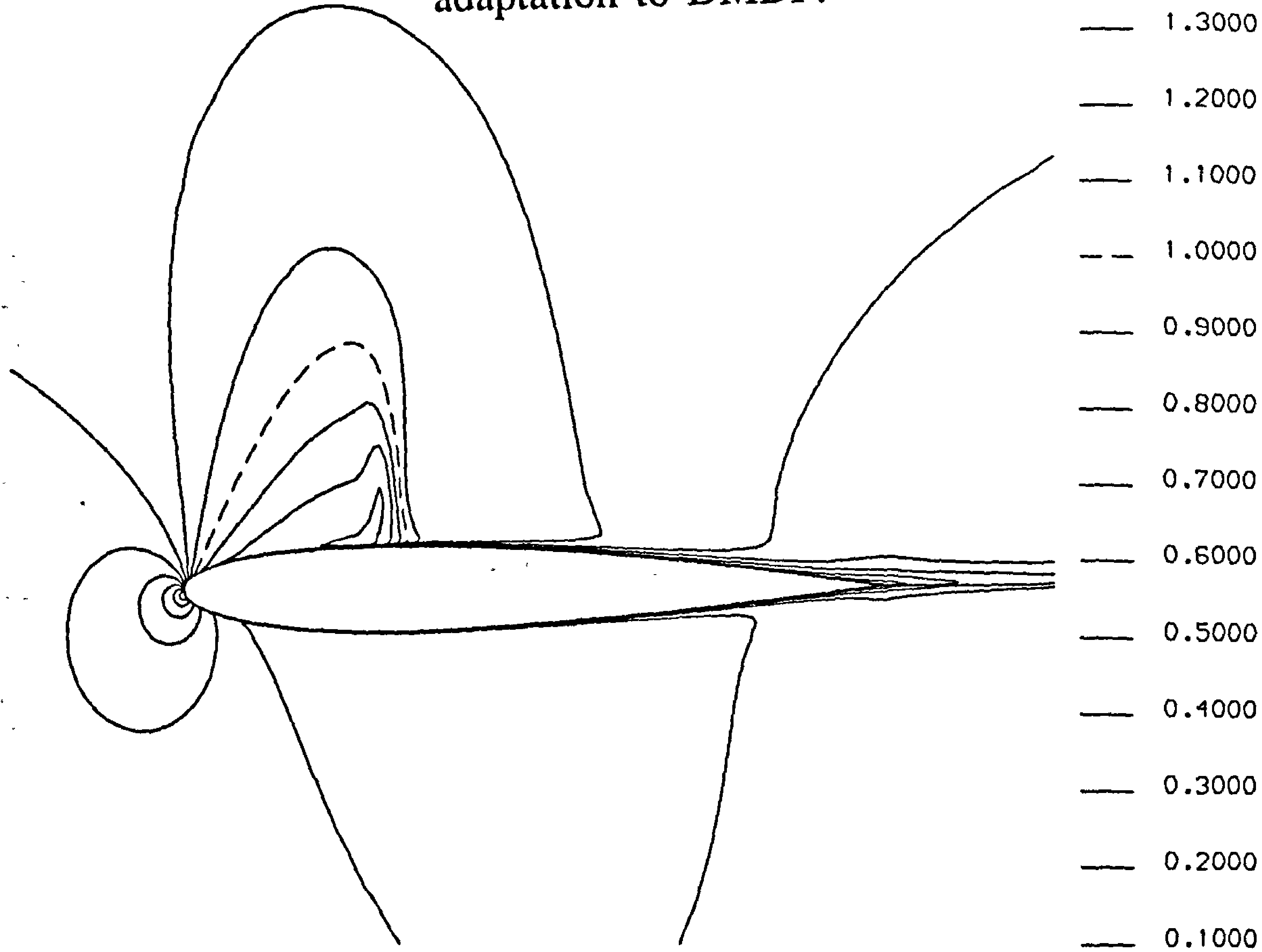


Fig.5.7 Turbulent (3°) case - Mach number contours after first adaptation to DMDP.

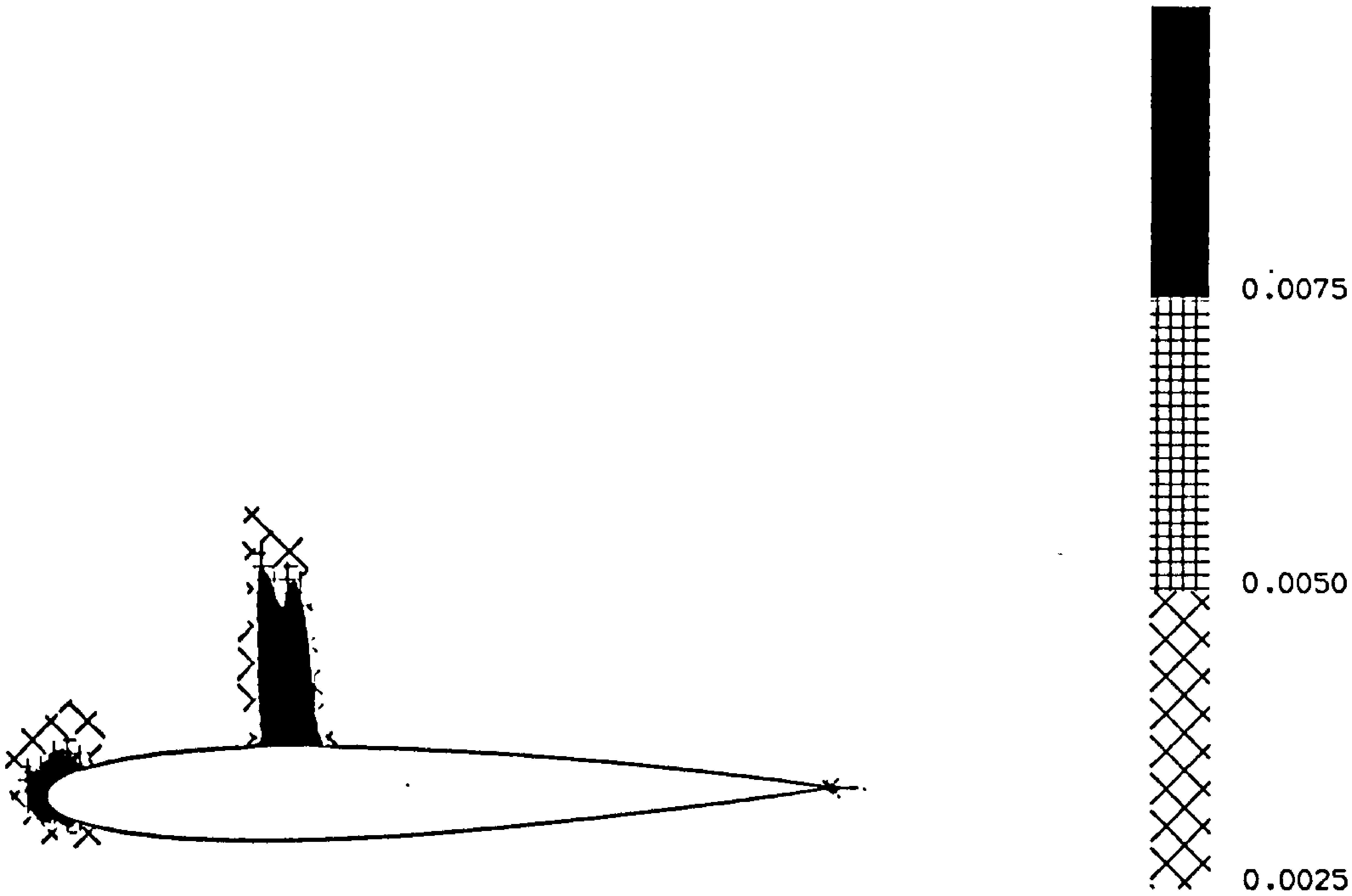


Fig.5.8 Turbulent (3°) case - DMDP weighting function on first adapted grid.

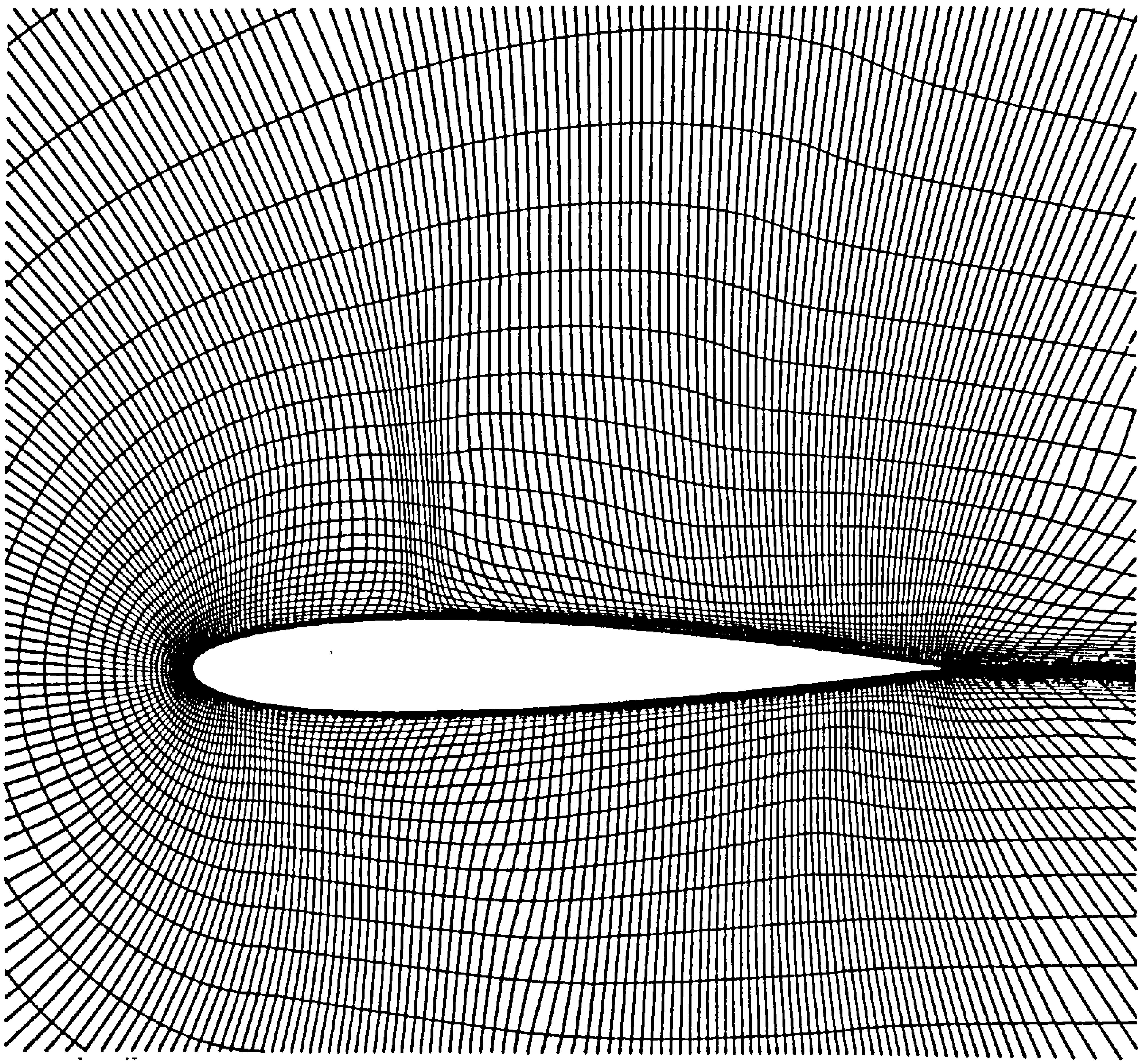


Fig.5.9 Turbulent (3°) case - Grid after second adaptation to DMDP.

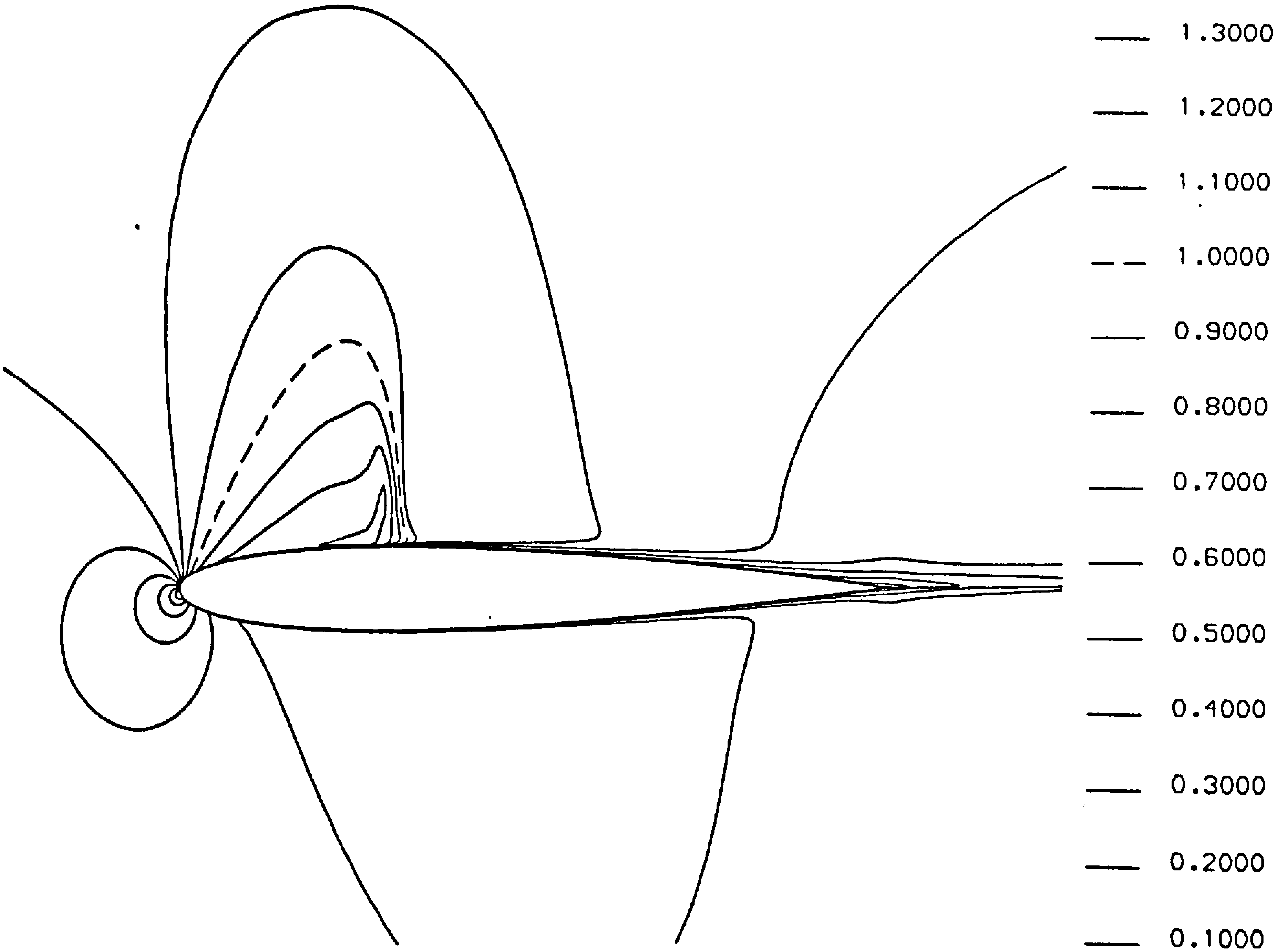


Fig.5.10 Turbulent (3°) case - Mach number contours after second adaptation to DMDP.

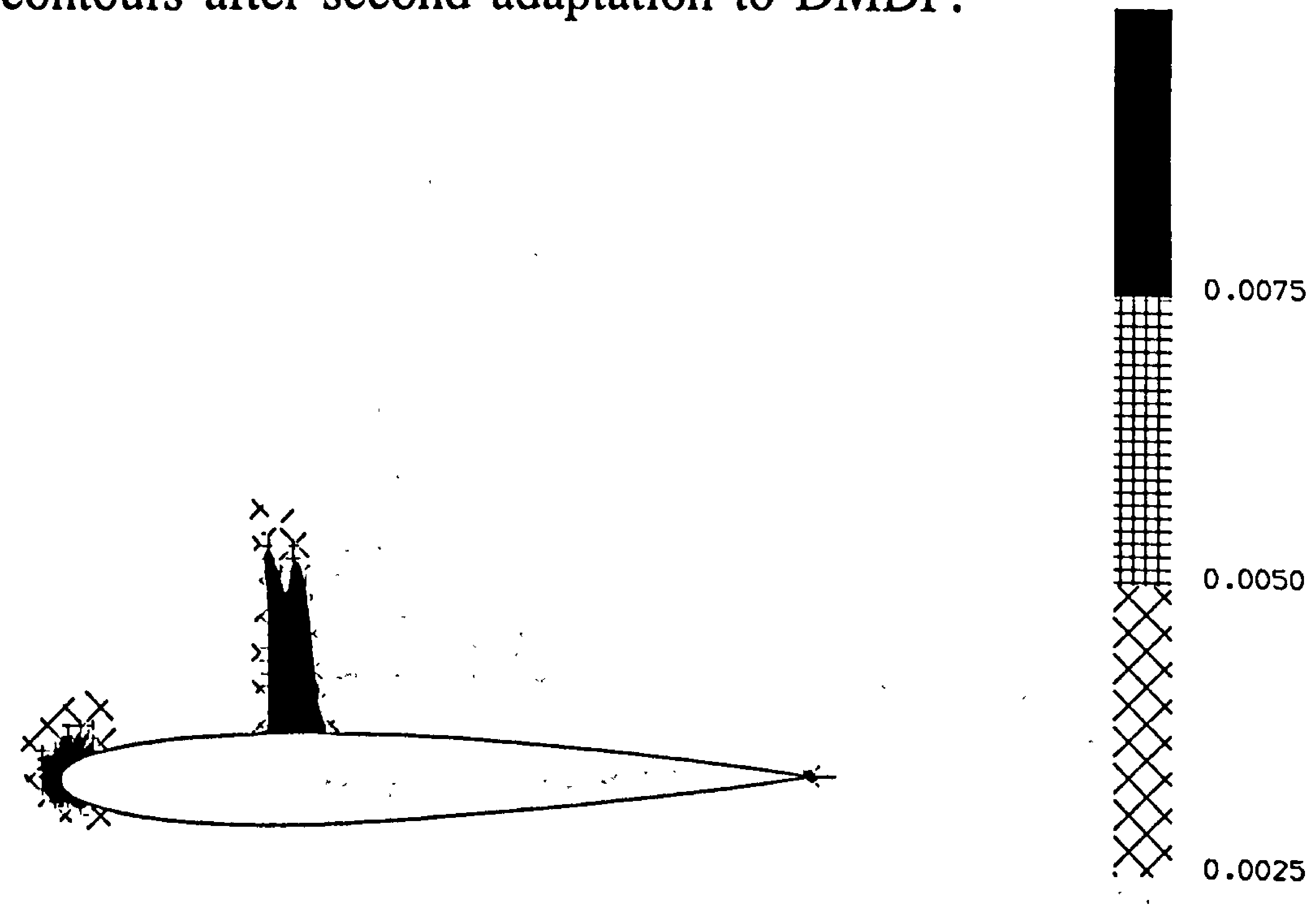


Fig.5.11 Turbulent (3°) case - DMDP weighting function on second adapted grid.

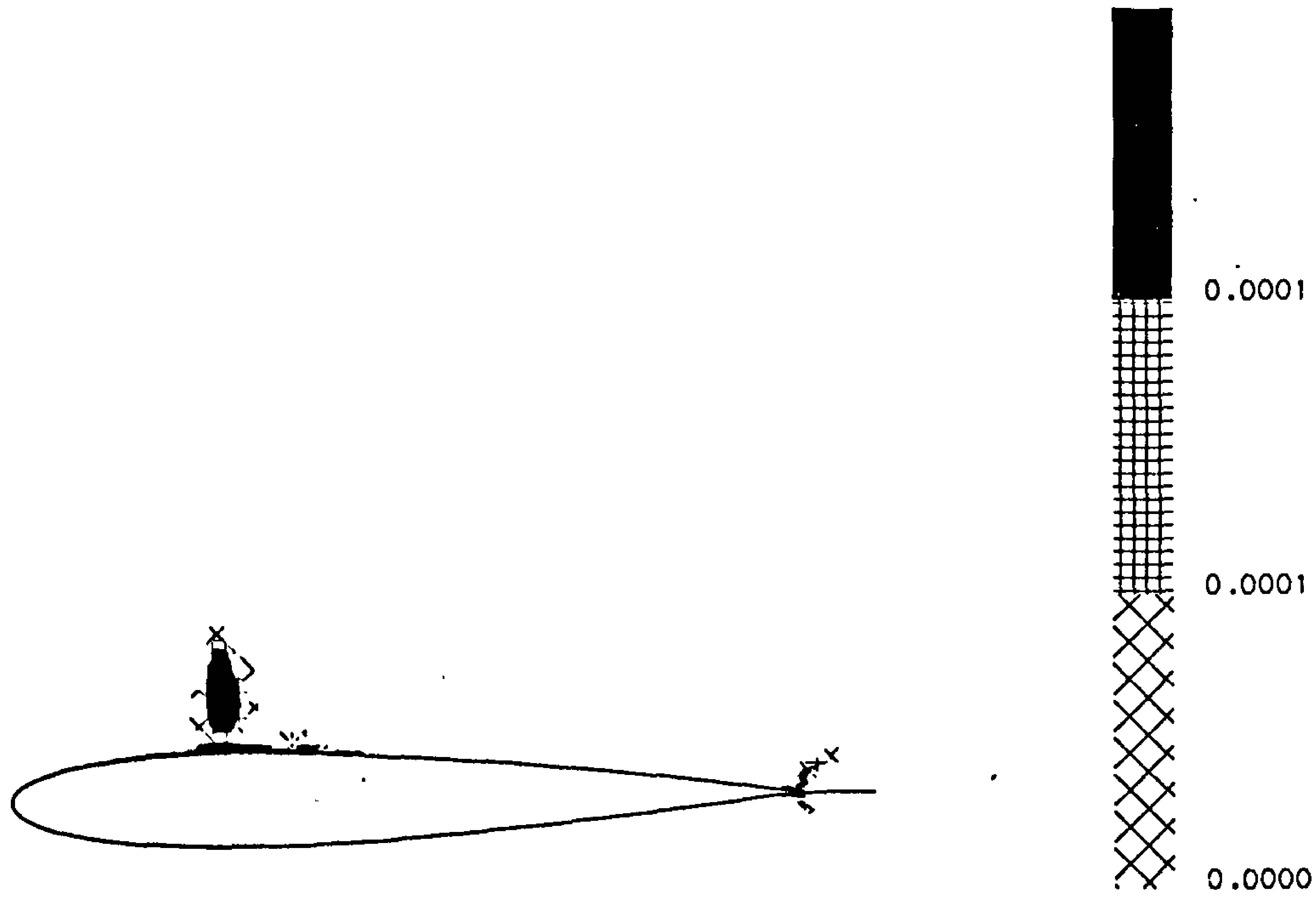


Fig.5.12 Turbulent (3°) case - NDFξ on initial grid.

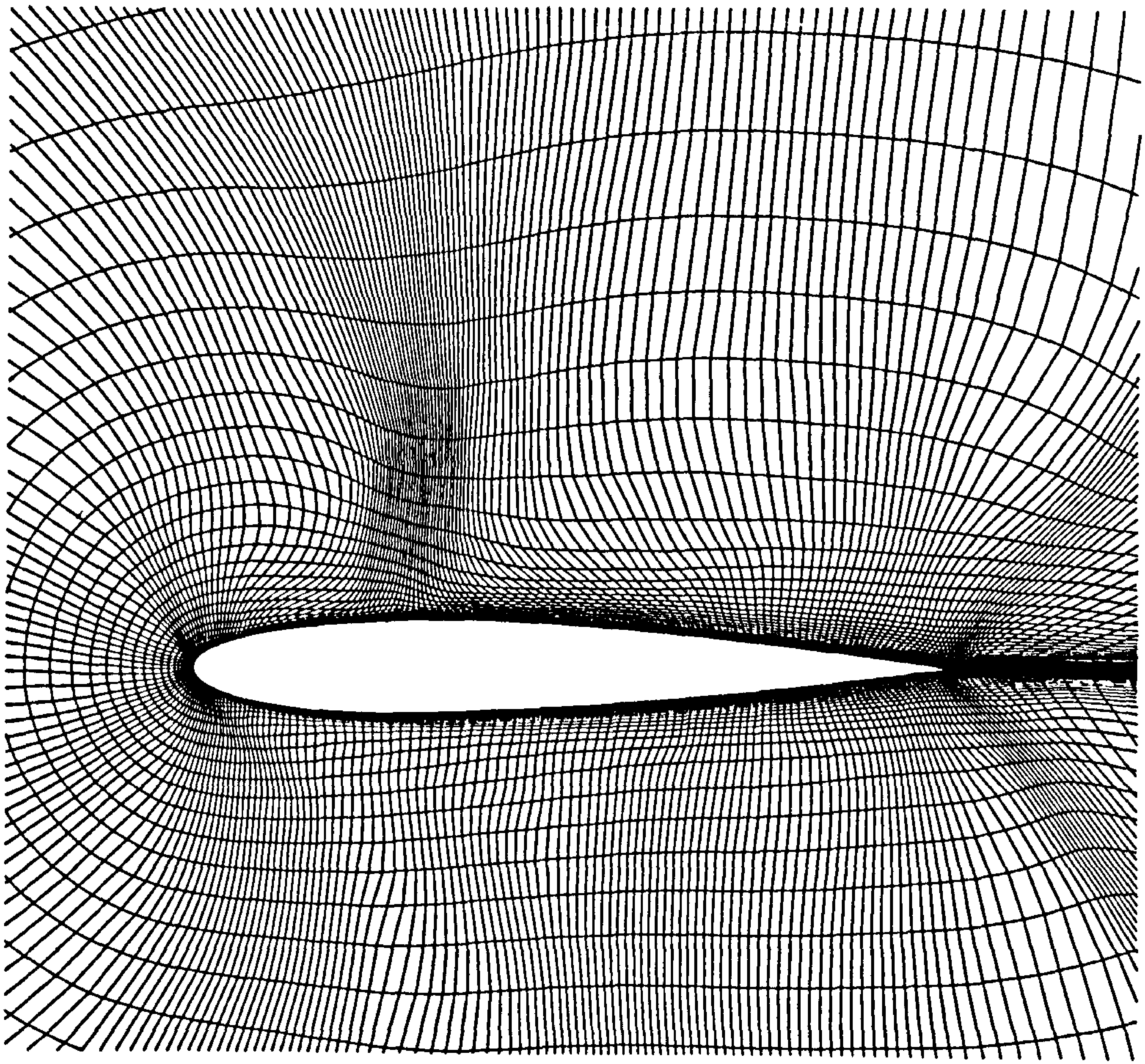


Fig.5.13 Turbulent (3°) case - Grid after adaptation to NDF - λ_E = 1.0.

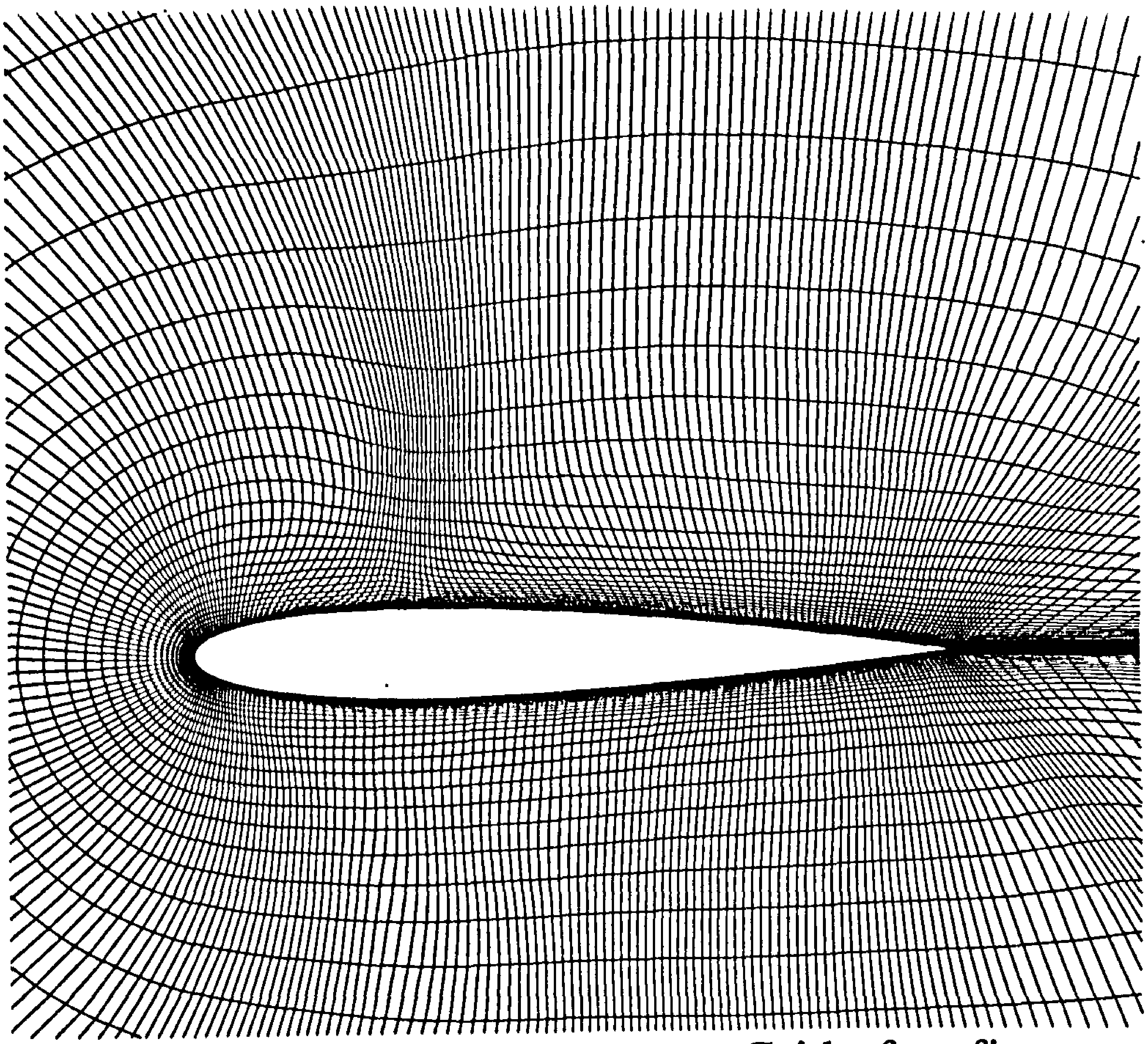


Fig.5.14 Turbulent (3°) case - Grid after first adaptation to NDF - $\lambda_E = 0.5$.

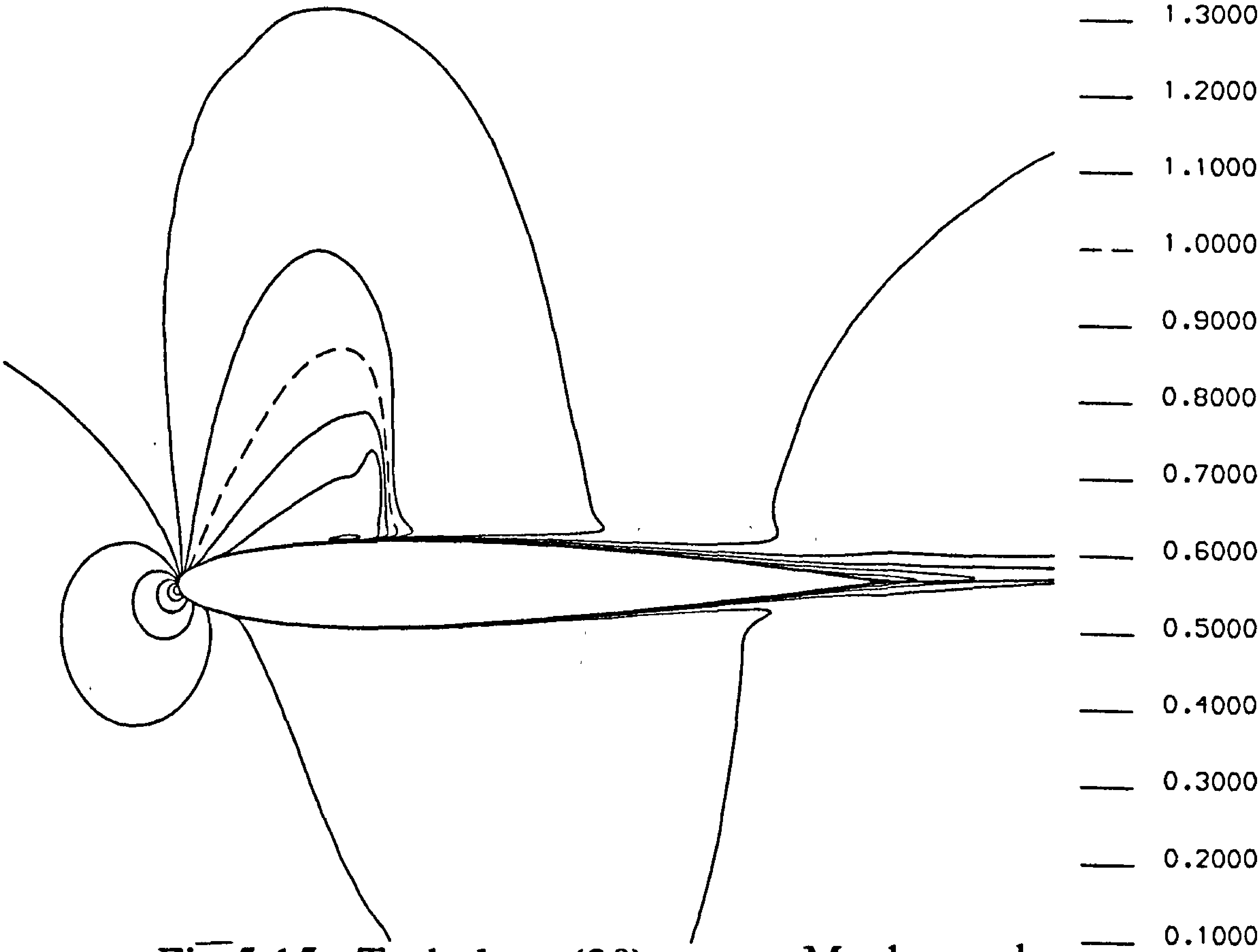


Fig.5.15 Turbulent (3°) case - Mach number contours after first adaptation to NDF.

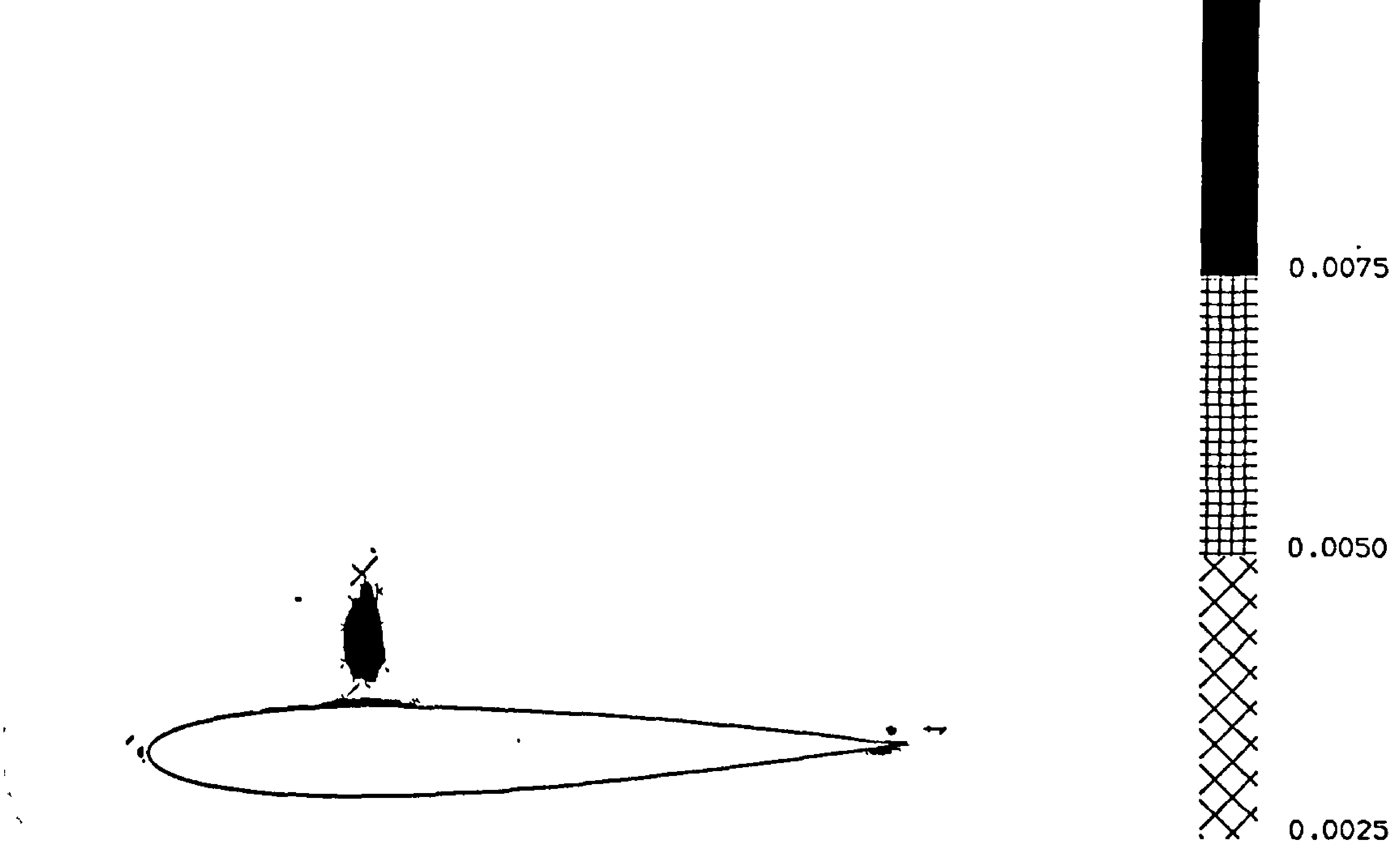


Fig.5.16 Turbulent (3°) case - NDFξ on first adapted grid.

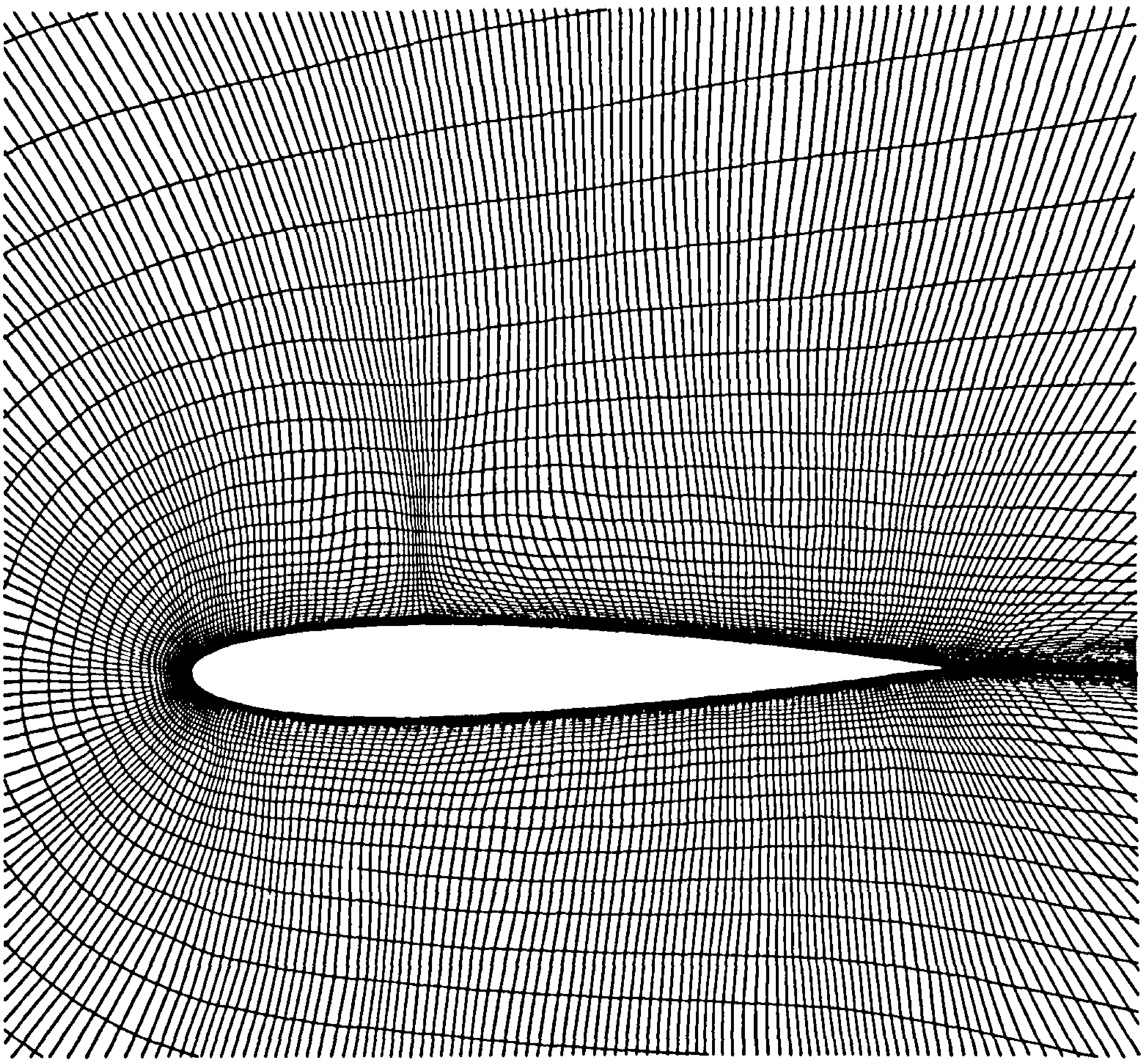


Fig.5.17 Turbulent (3°) case - Grid after fourth adaptation to NDF.

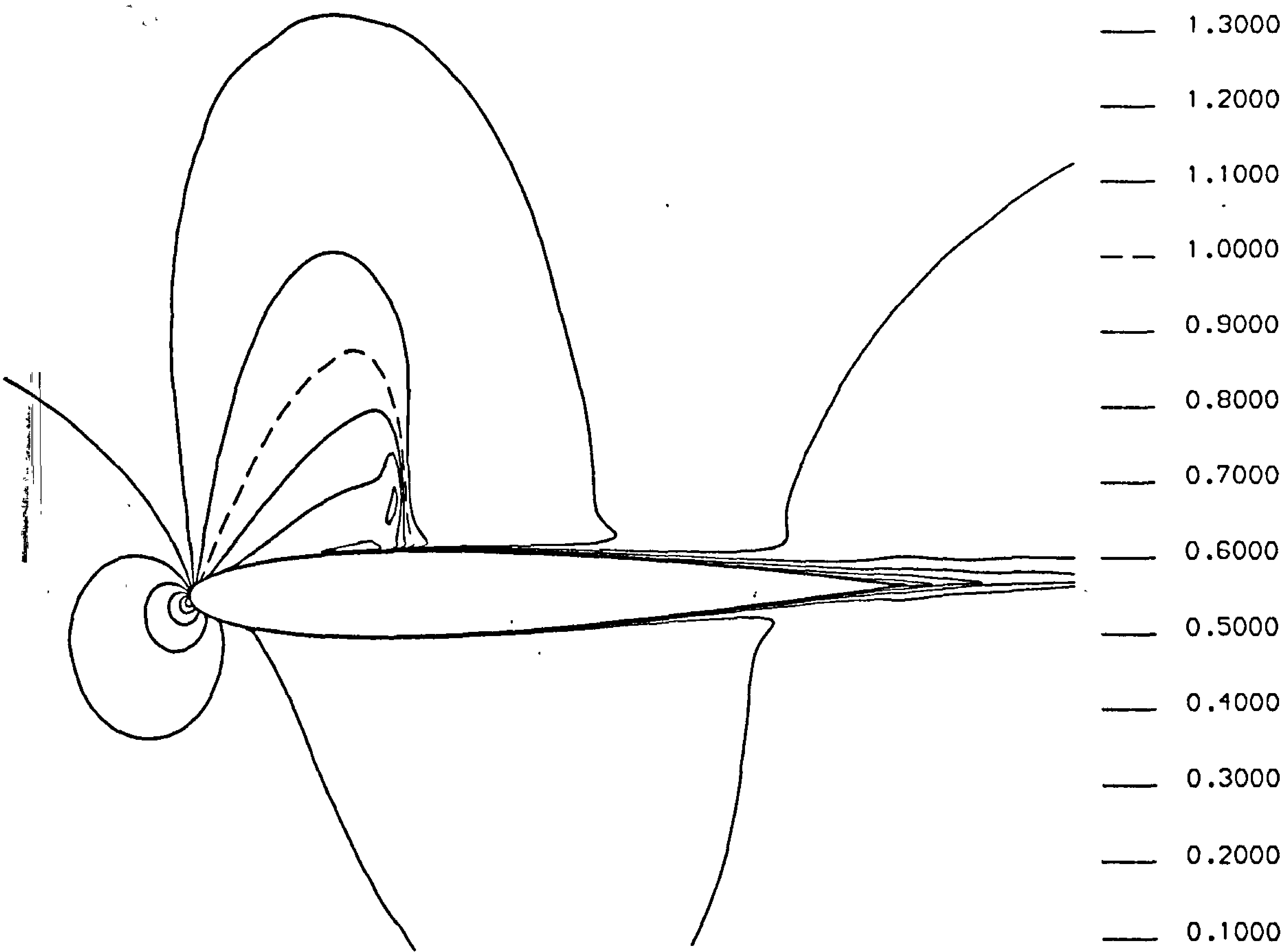


Fig.5.18 Turbulent (3°) case - Mach number contours after fourth adaptation to NDF.

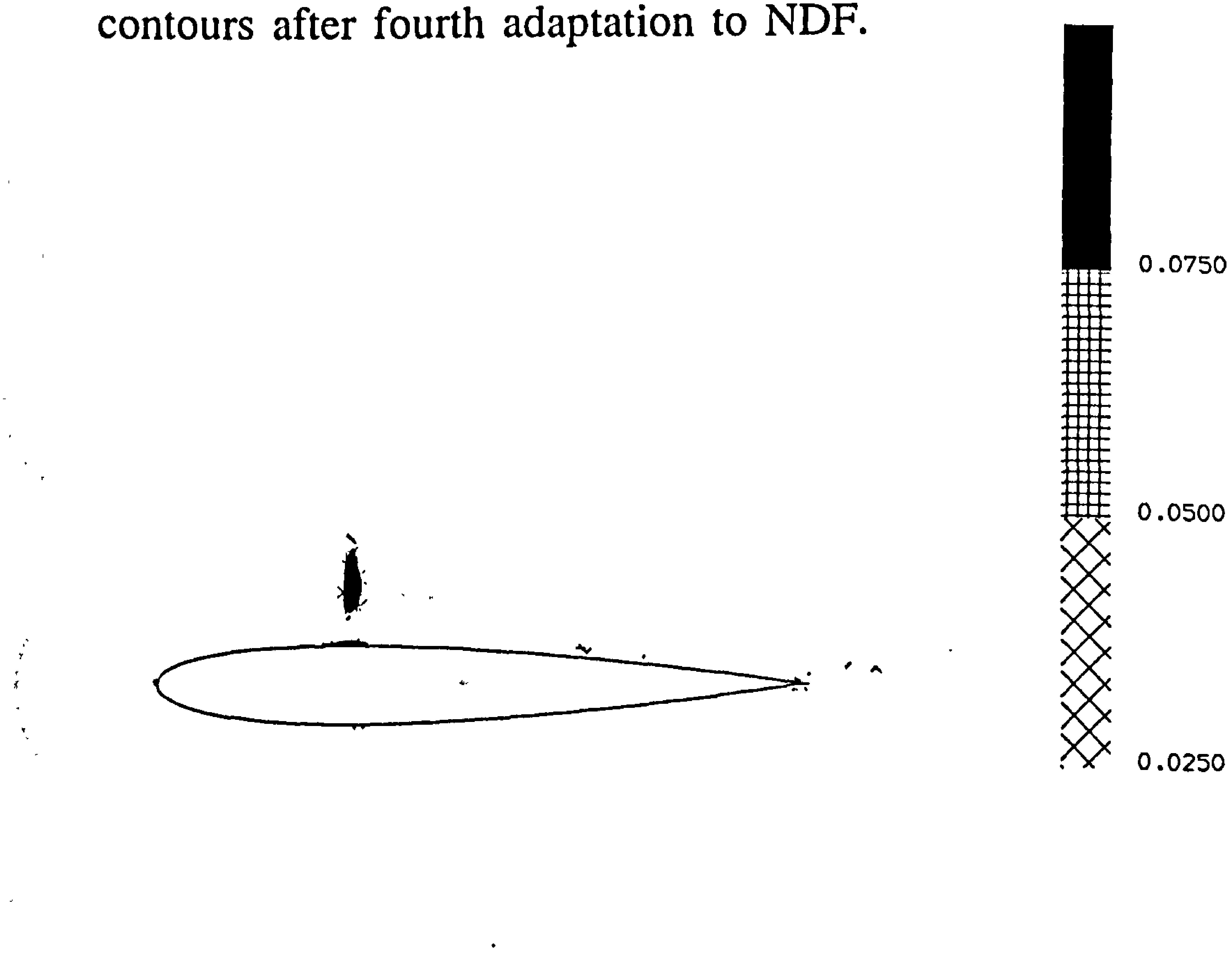


Fig.5.19 Turbulent (3°) case - NDFξ on fourth adapted grid.

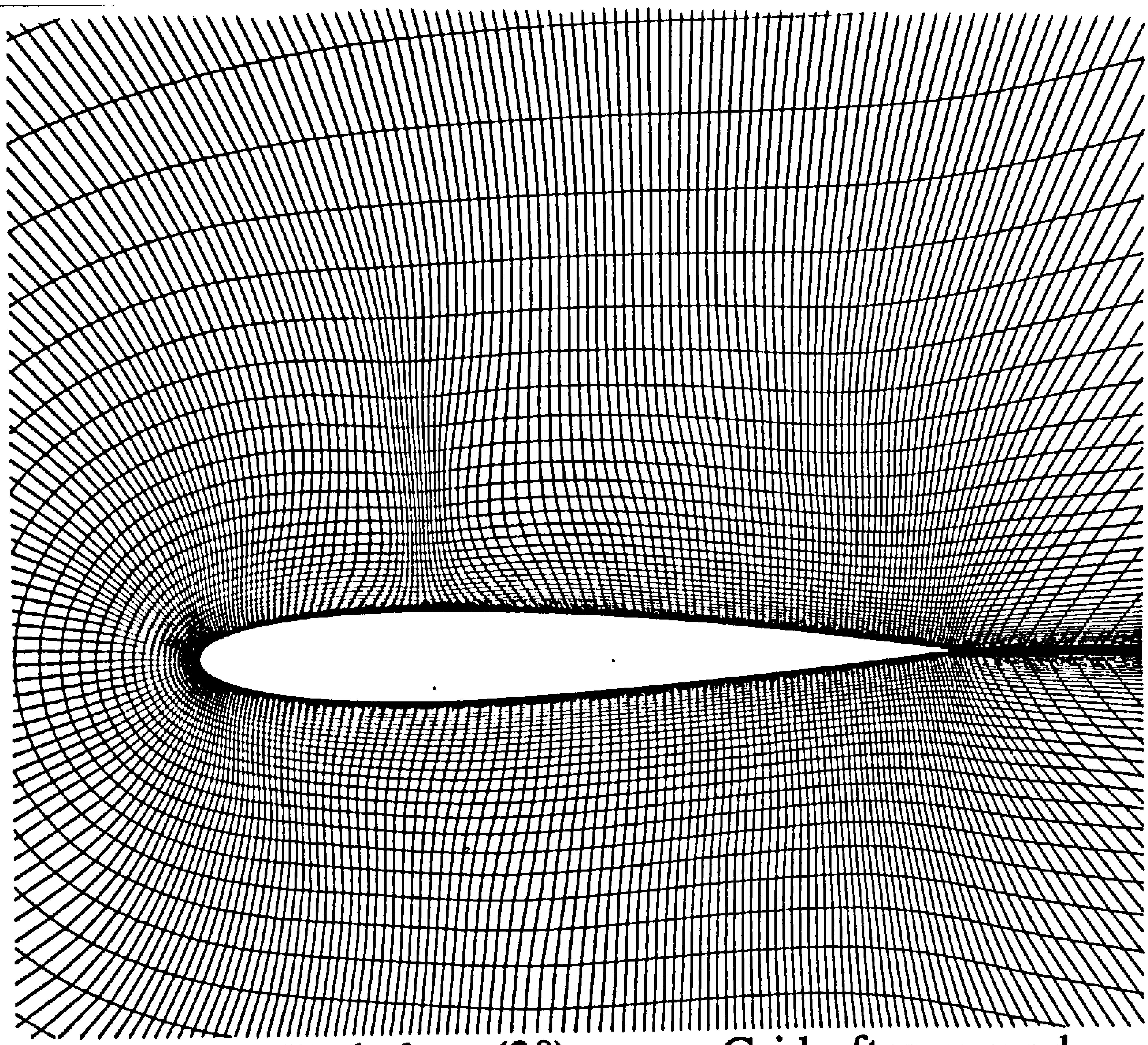


Fig.5.20 Turbulent (3°) case - Grid after second adaptation to COMB (combined measure).

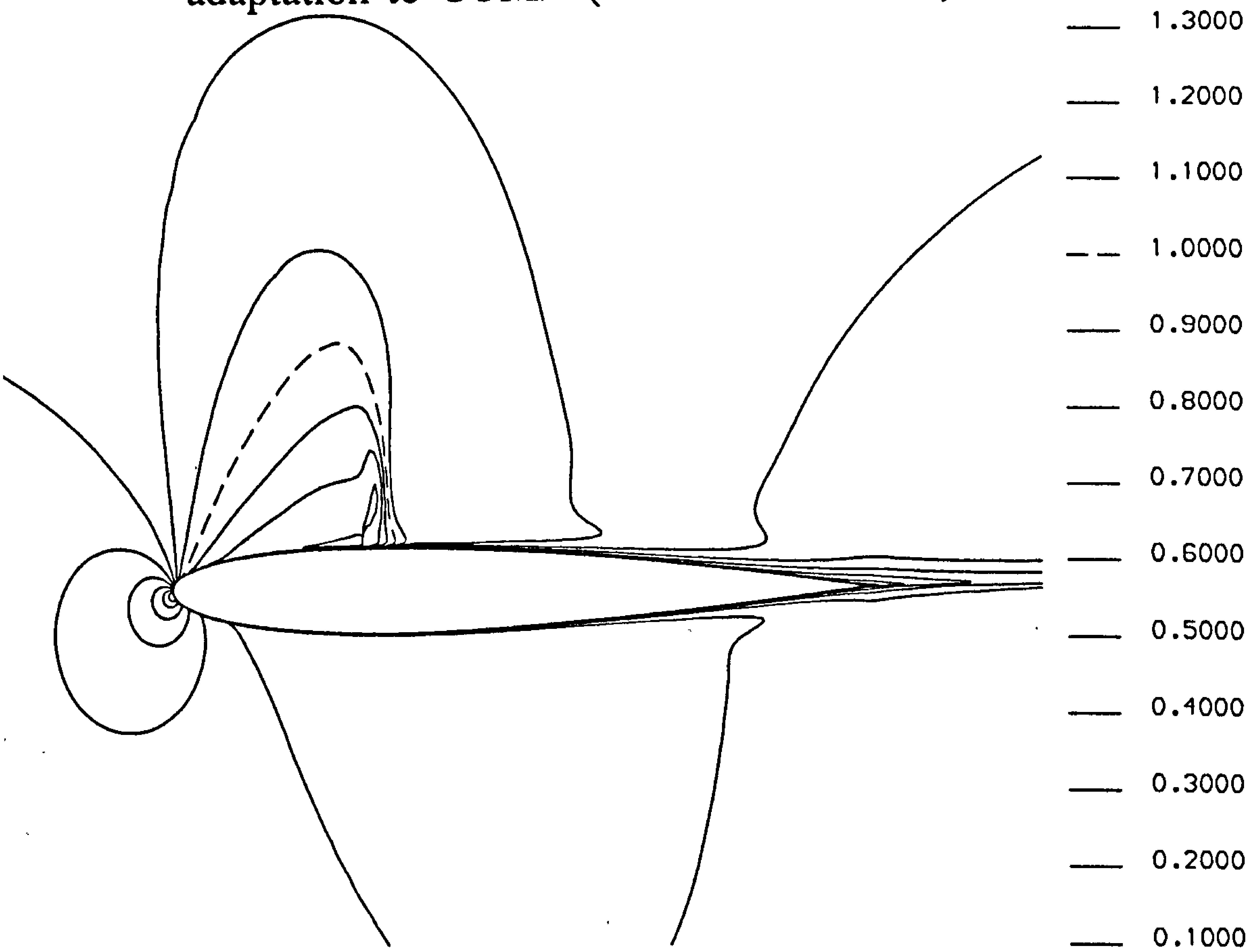


Fig.5.21 Turbulent (3°) case - Mach number contours after second adaptation to COMB.

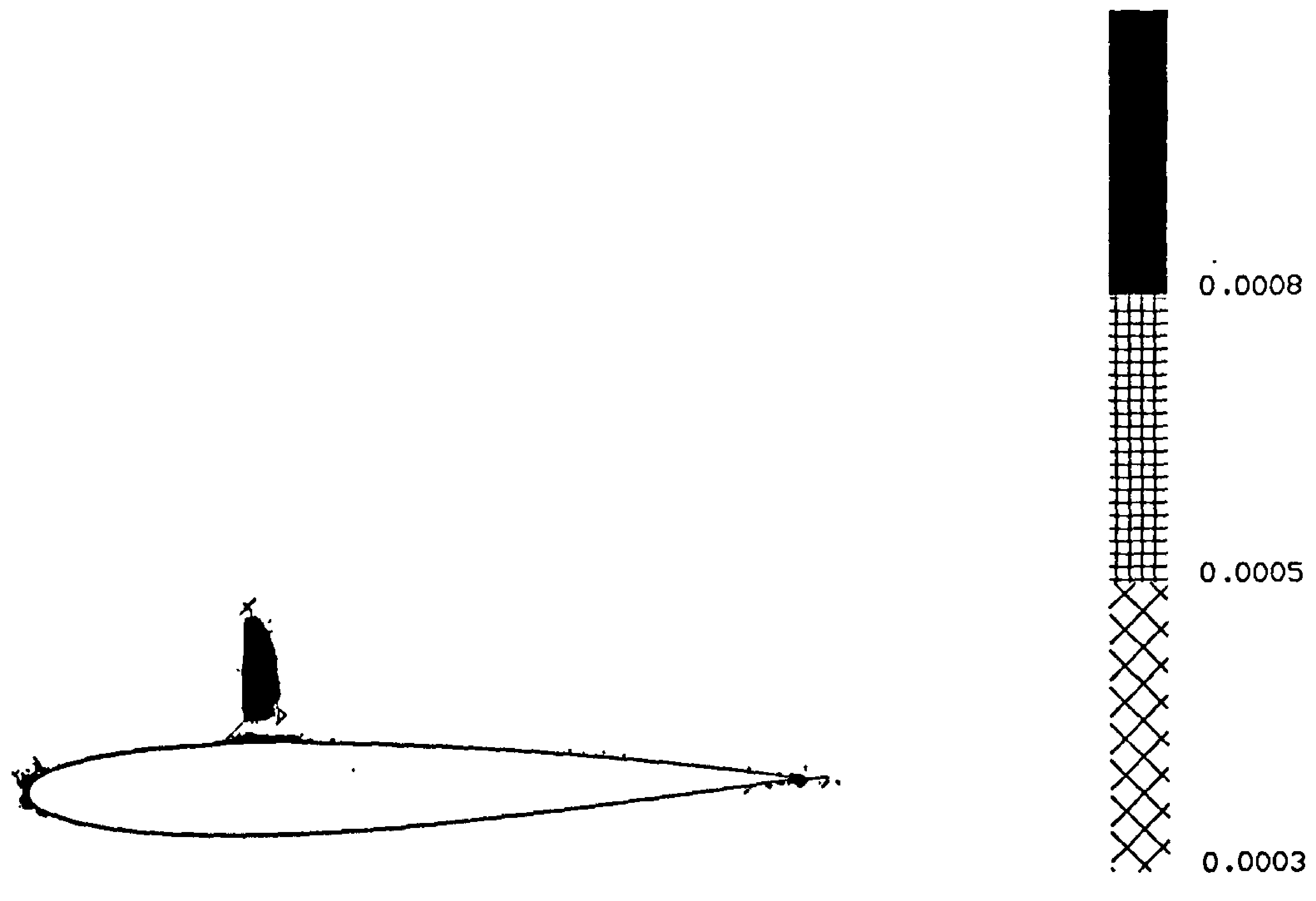


Fig.5.22 Turbulent (3°) case - COMB weighting function on second adapted grid.

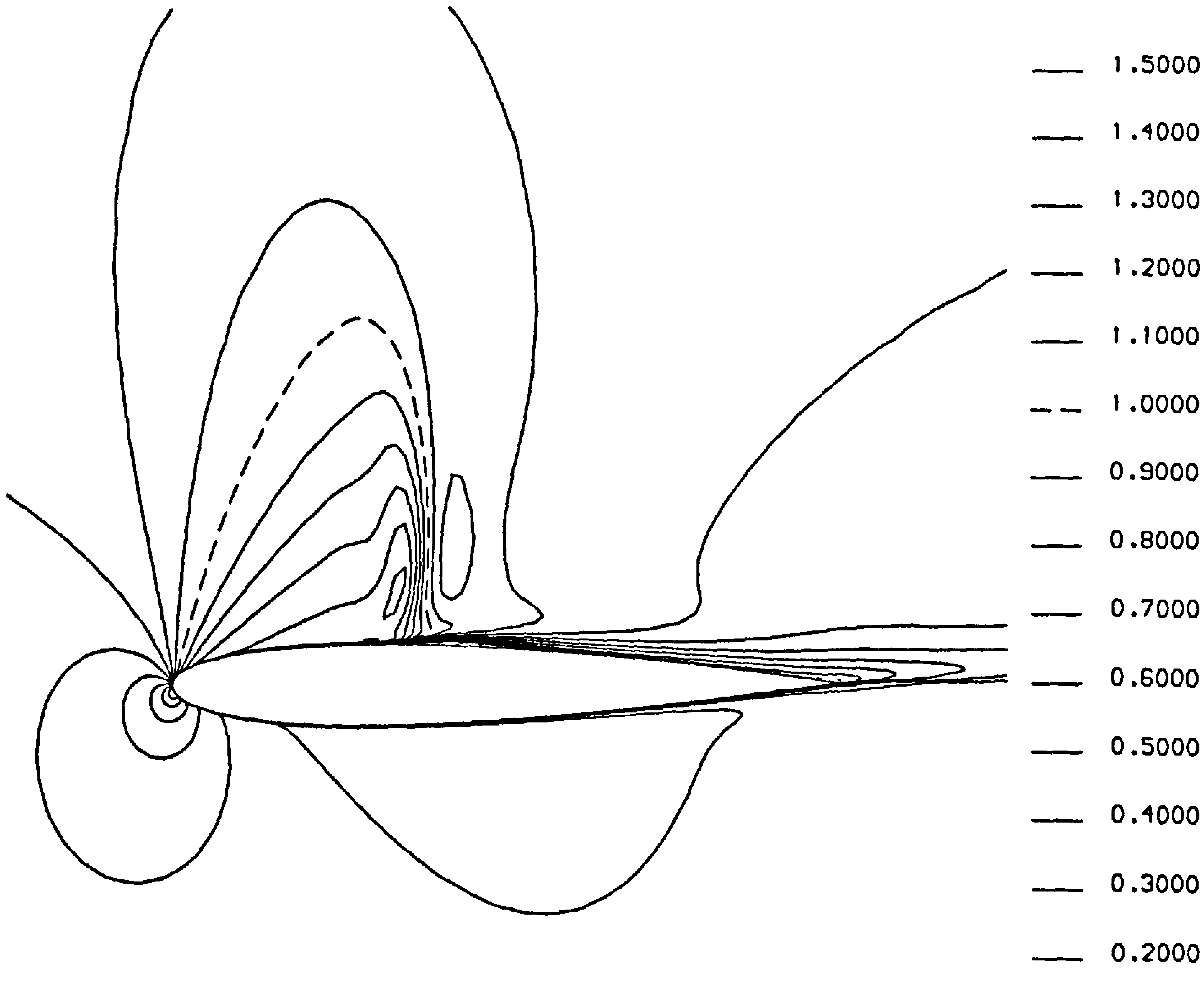


Fig.5.23 Turbulent (5°) case - Mach number contours on grid c5 (initial).

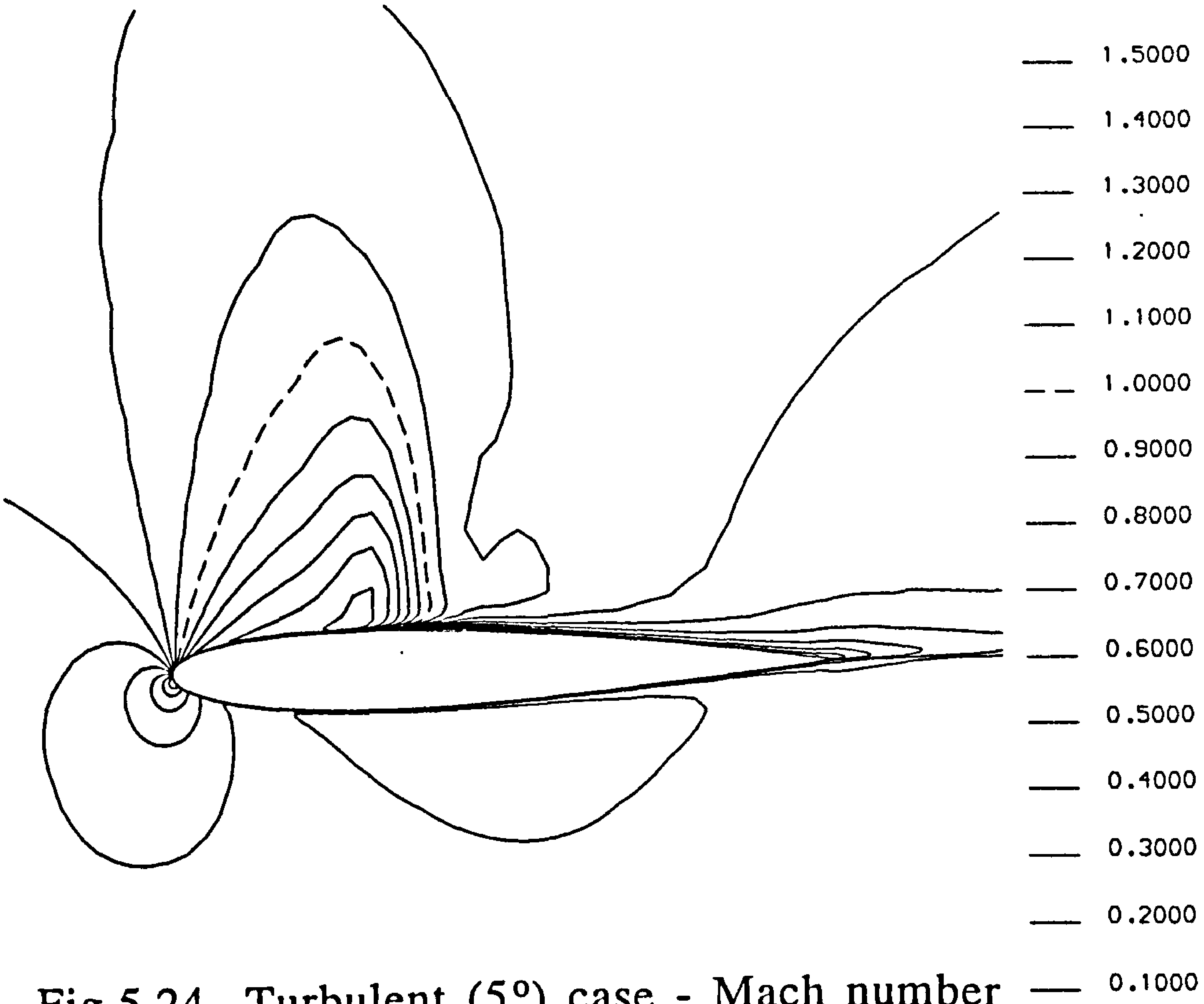


Fig.5.24 Turbulent (5°) case - Mach number contours on grid c53.

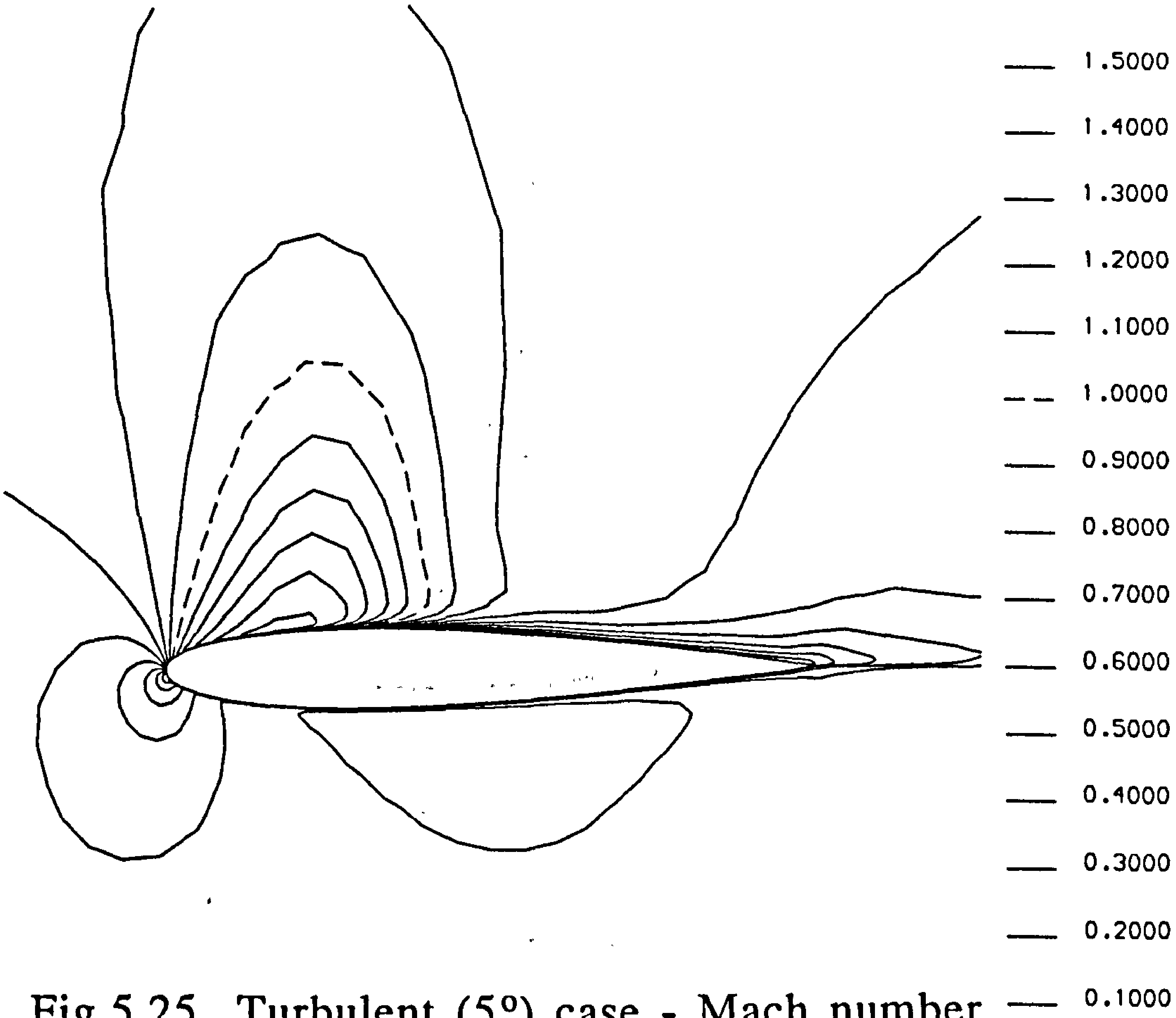


Fig.5.25 Turbulent (5°) case - Mach number contours on grid c54.

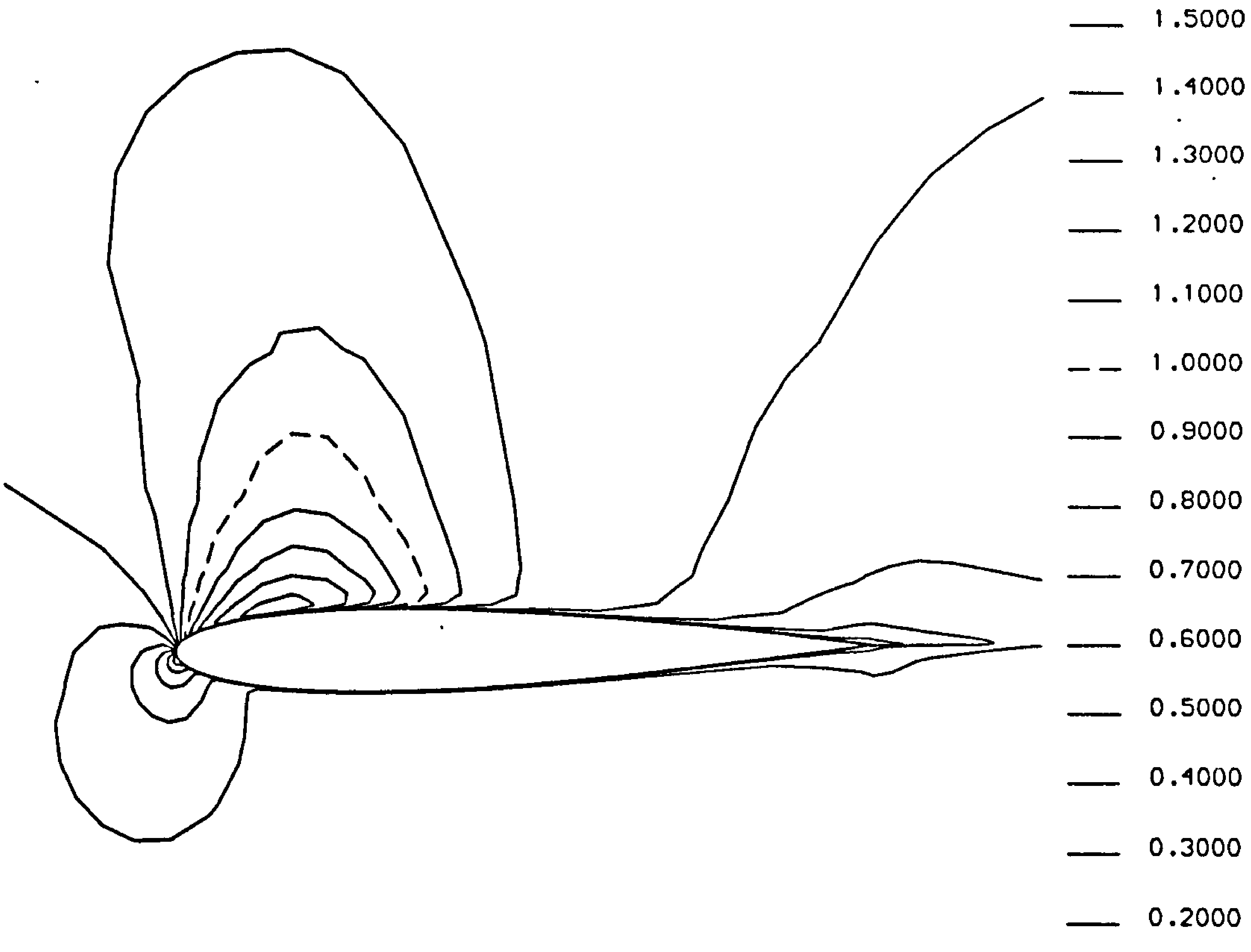


Fig.5.26 Turbulent (5°) case - Mach number contours on grid c55.

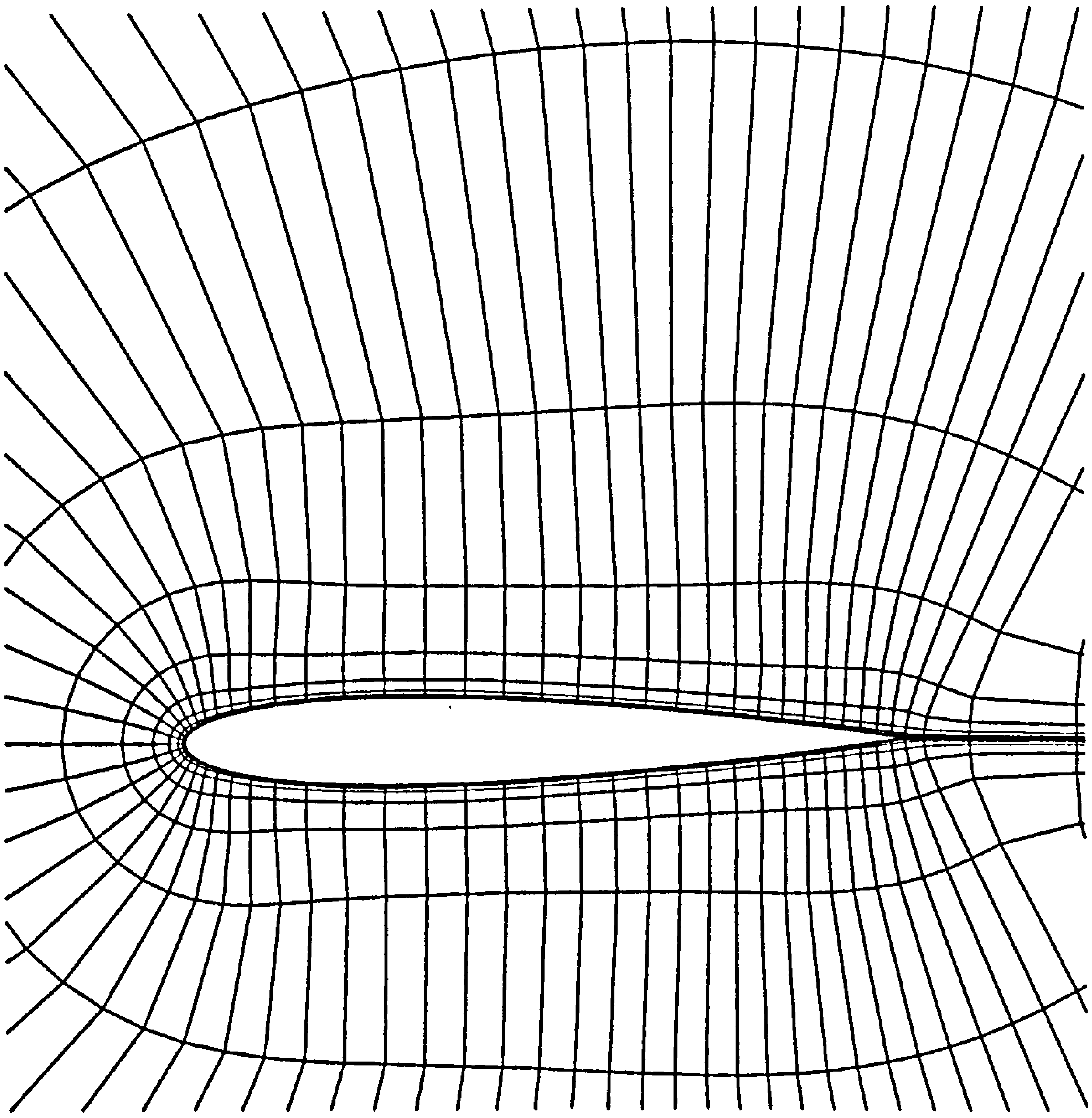


Fig.5.27 Grid c55.

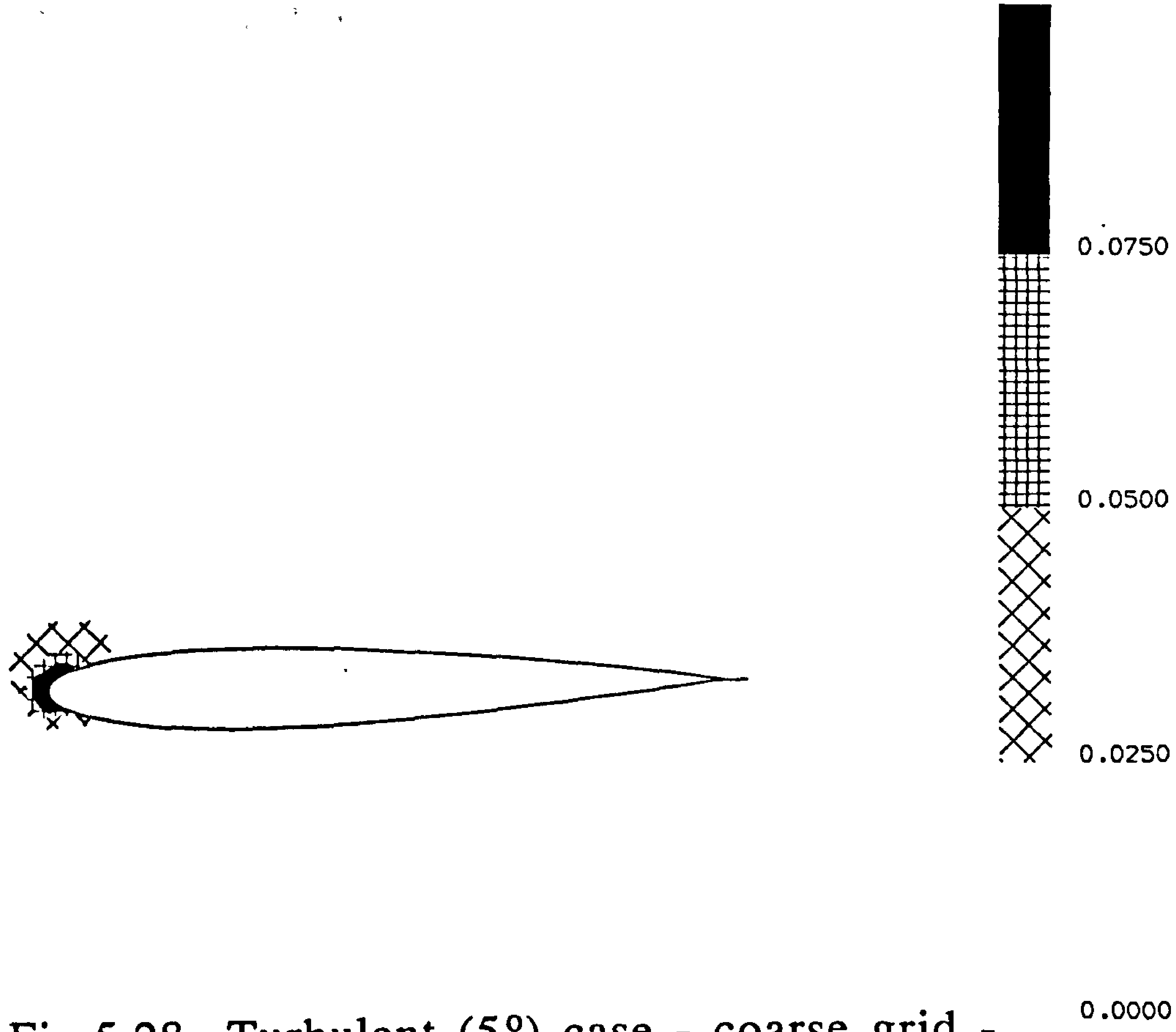


Fig.5.28 Turbulent (5°) case - coarse grid - initial DMDP weighting function.

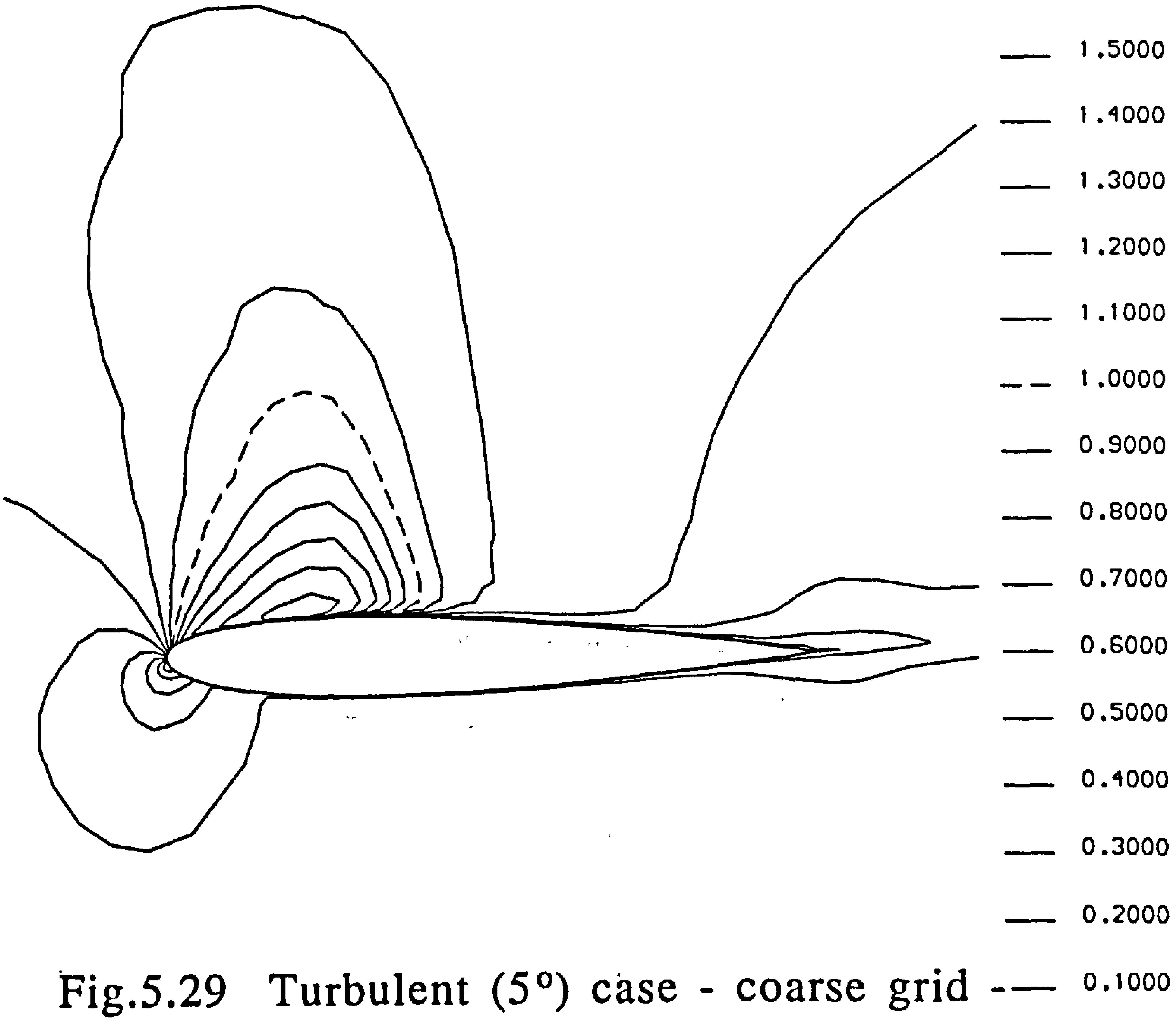


Fig.5.29 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to DMDP.

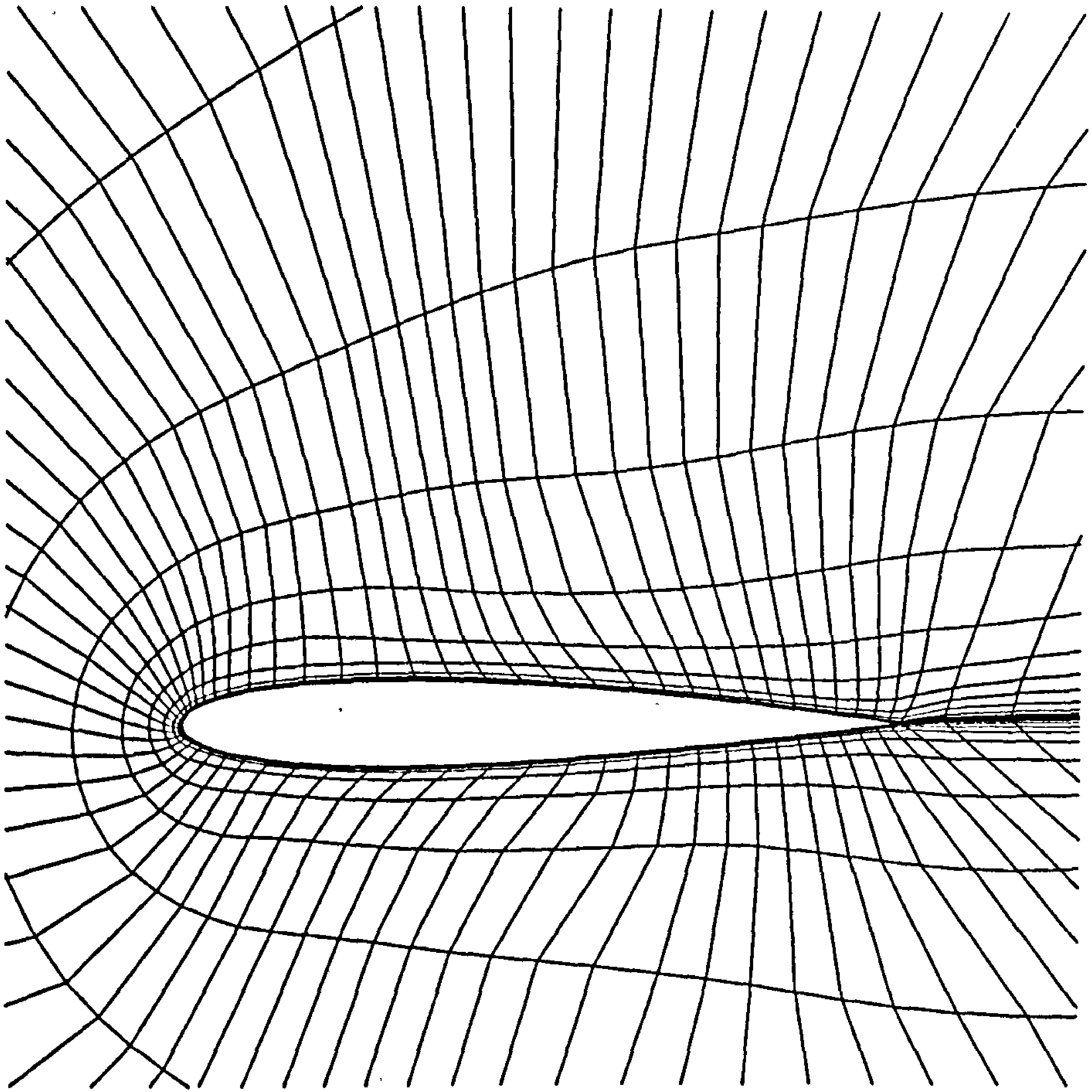


Fig.5.30 Turbulent (5°) case - coarse grid - grid after fourth adaptation to DMDP.

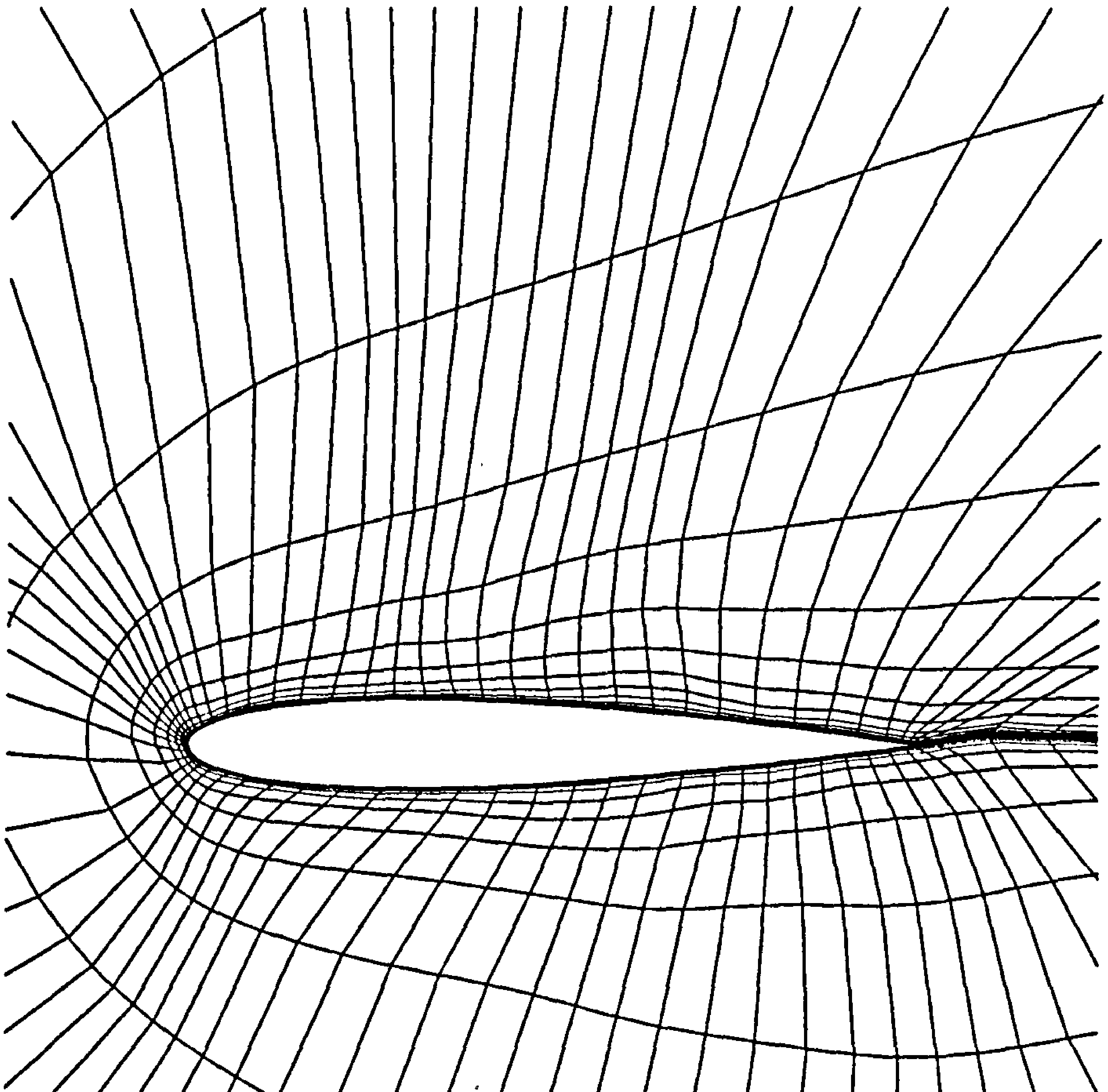


Fig.5.31 Turbulent (5°) case - coarse grid - grid after fourth adaptation to NDF - Strategy A.

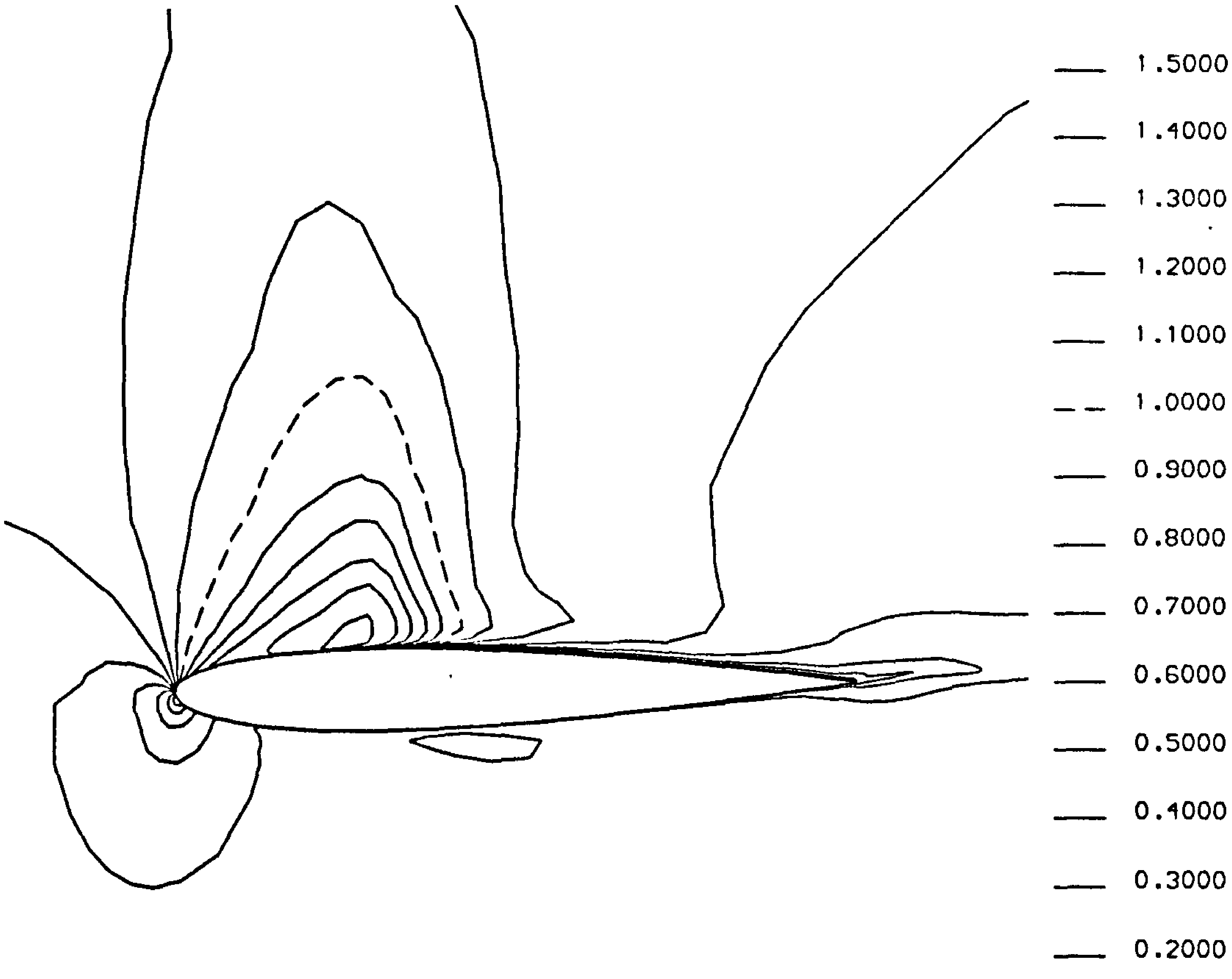


Fig.5.32 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to NDF - Strategy A.

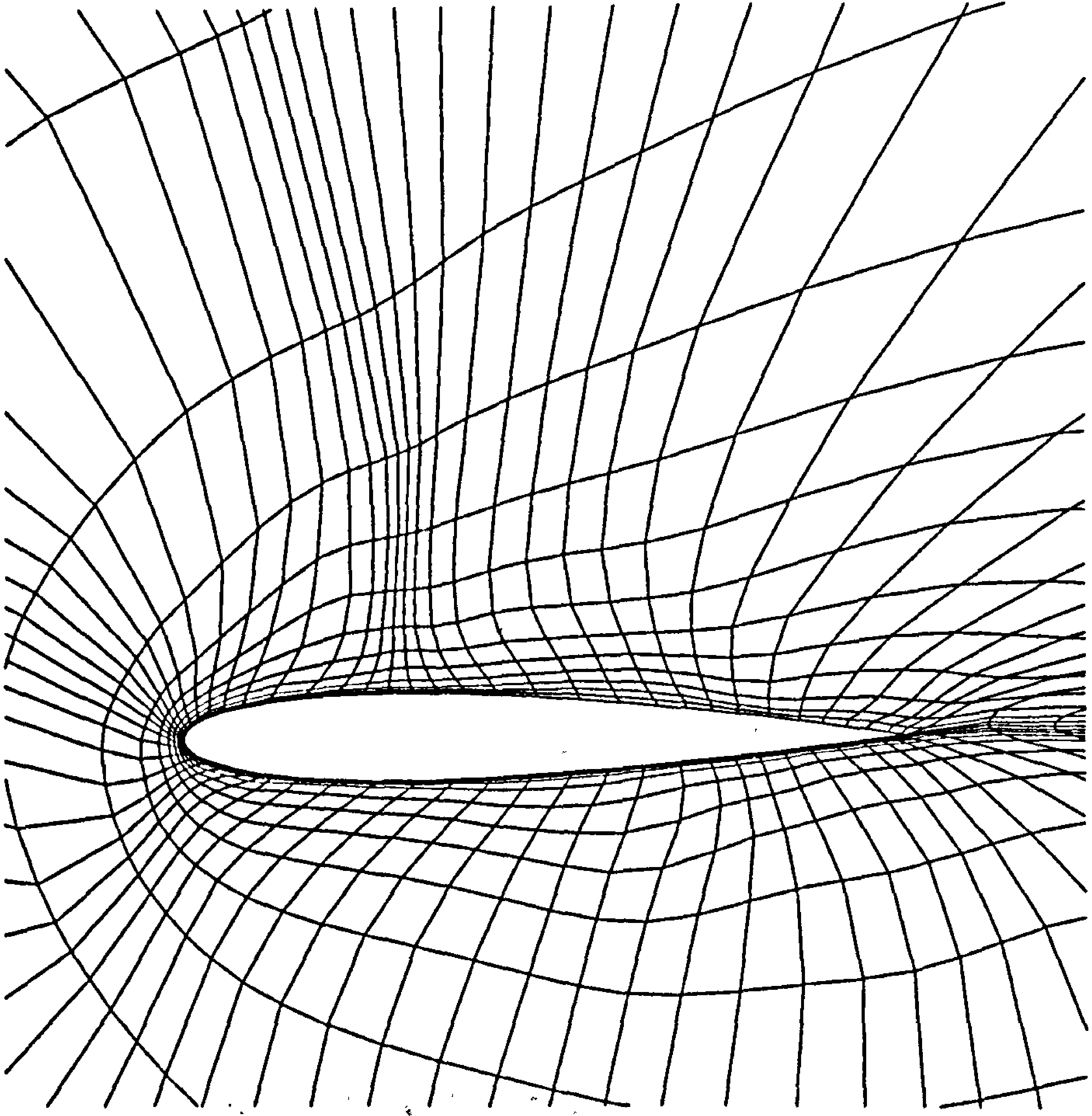


Fig.5.33 Turbulent (5°) case - coarse grid - grid after fourth adaptation to NDF - Strategy B.

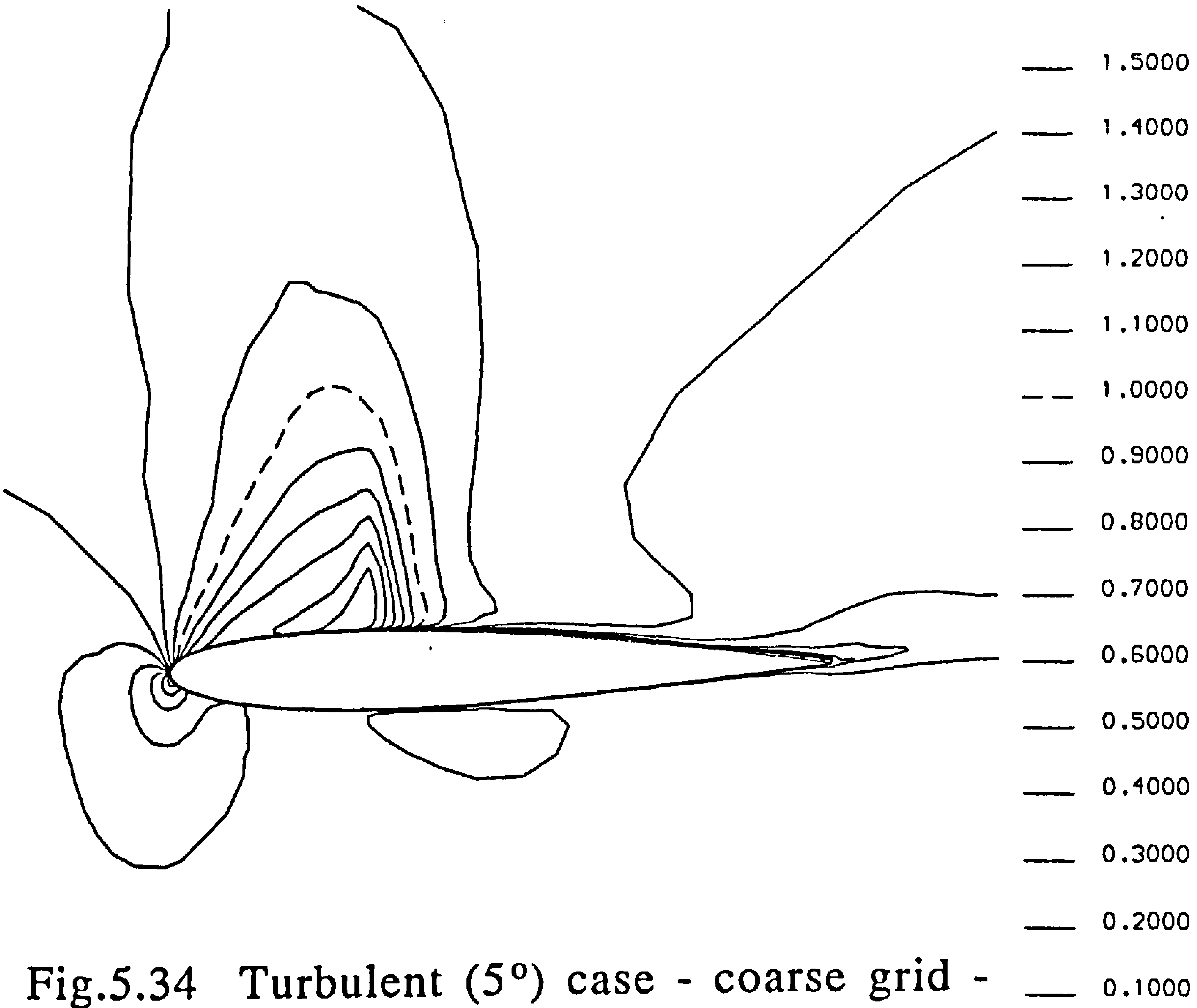


Fig.5.34 Turbulent (5°) case - coarse grid - Mach number contours after fourth adaptation to NDF - Strategy B.

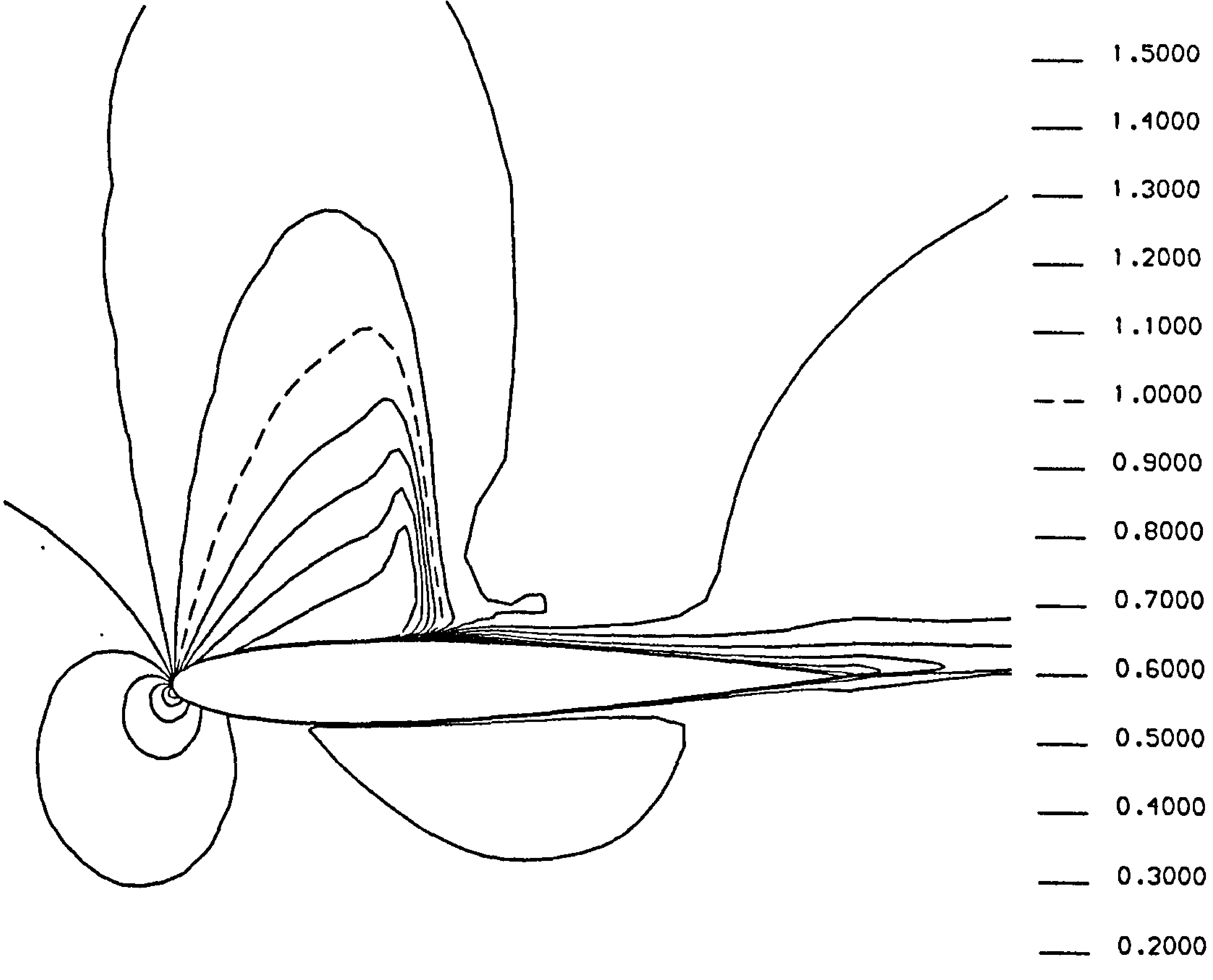


Fig.5.35 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to DMDP.

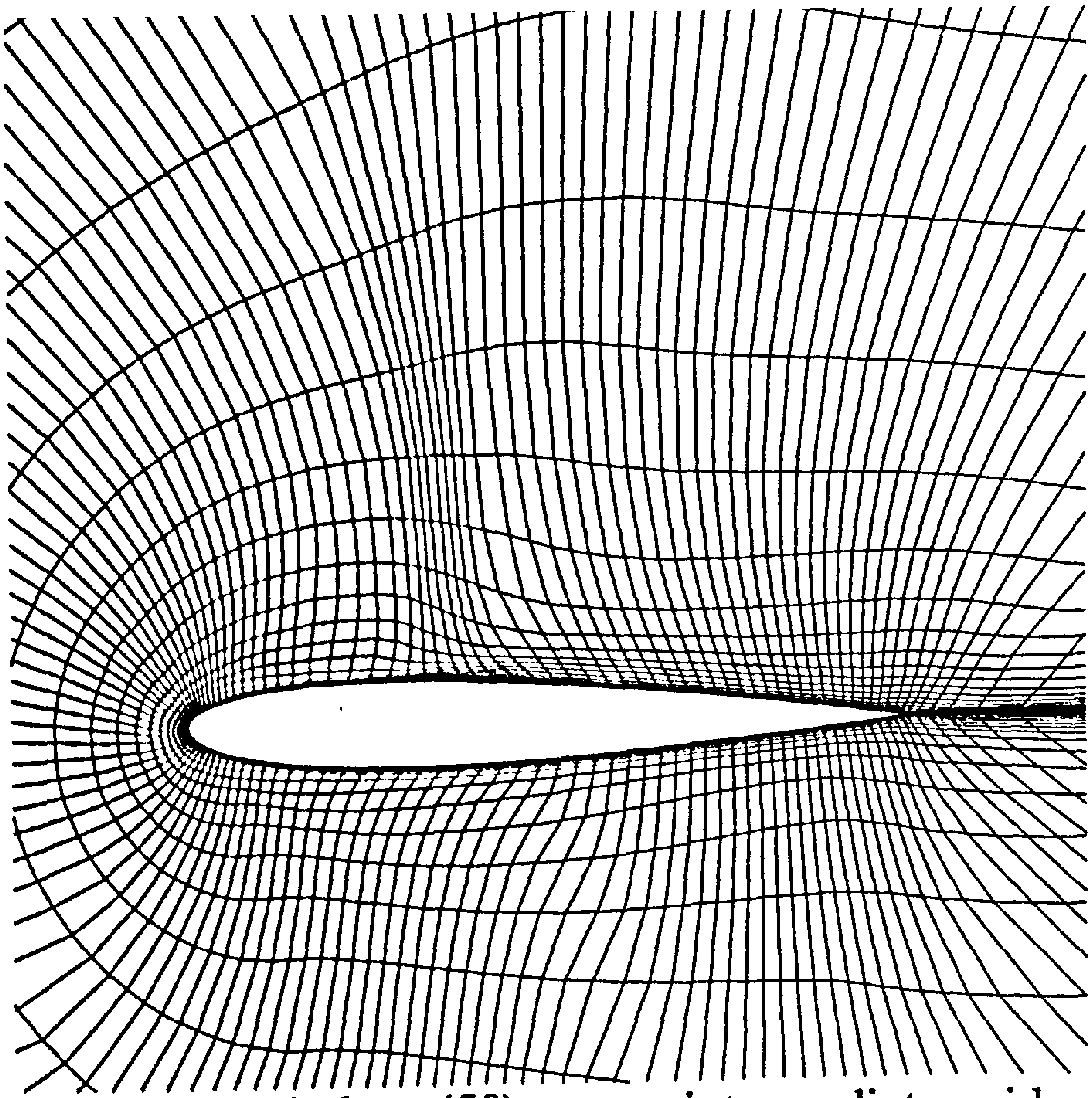


Fig.5.36 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to DMDP.

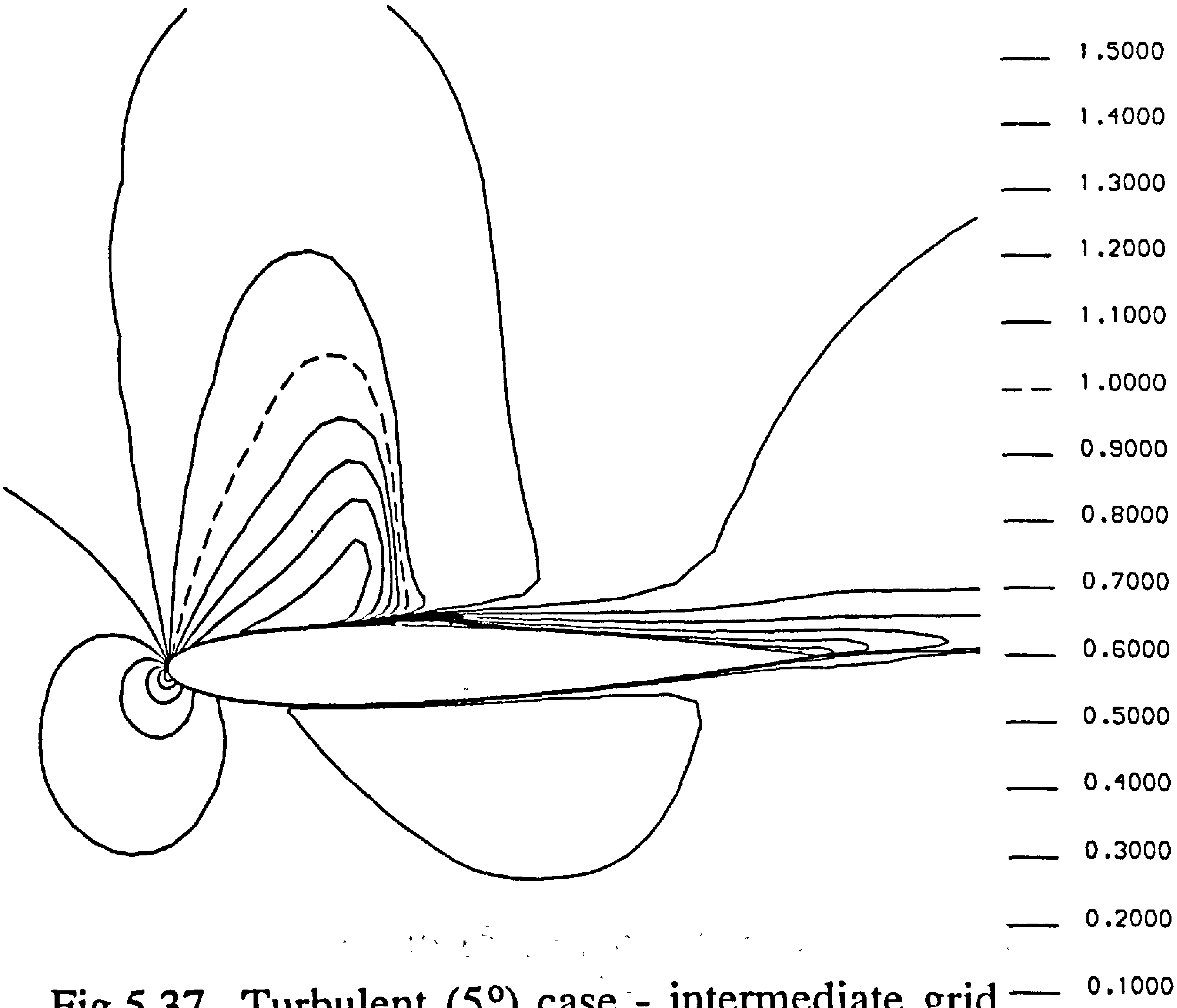


Fig.5.37 Turbulent (5°) case - intermediate grid - Mach number contours after fifth adaptation to DMDP.

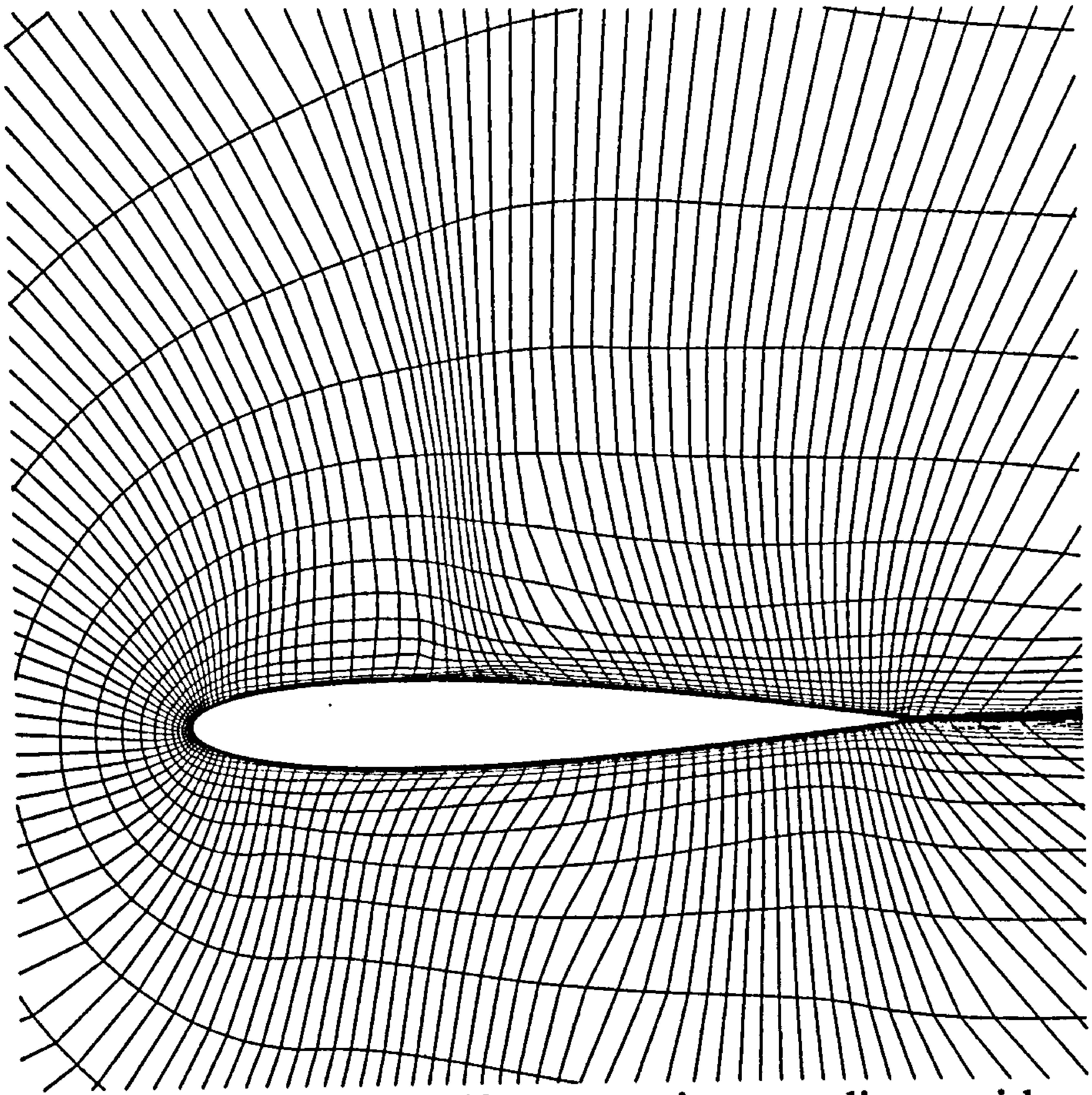


Fig.5.38 Turbulent (5°) case - intermediate grid - grid after fifth adaptation to DMDP.

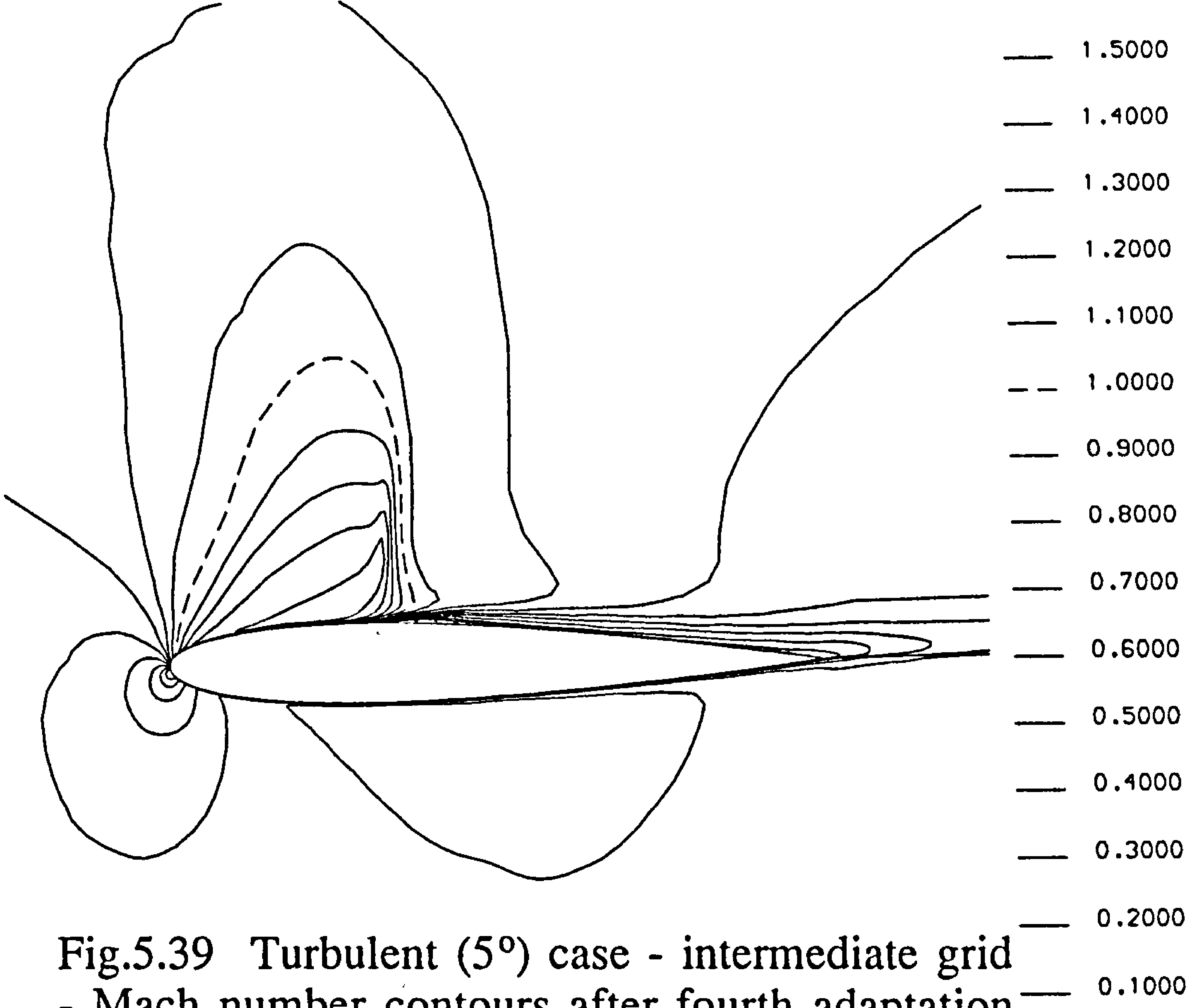


Fig.5.39 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to NDF - Strategy A.

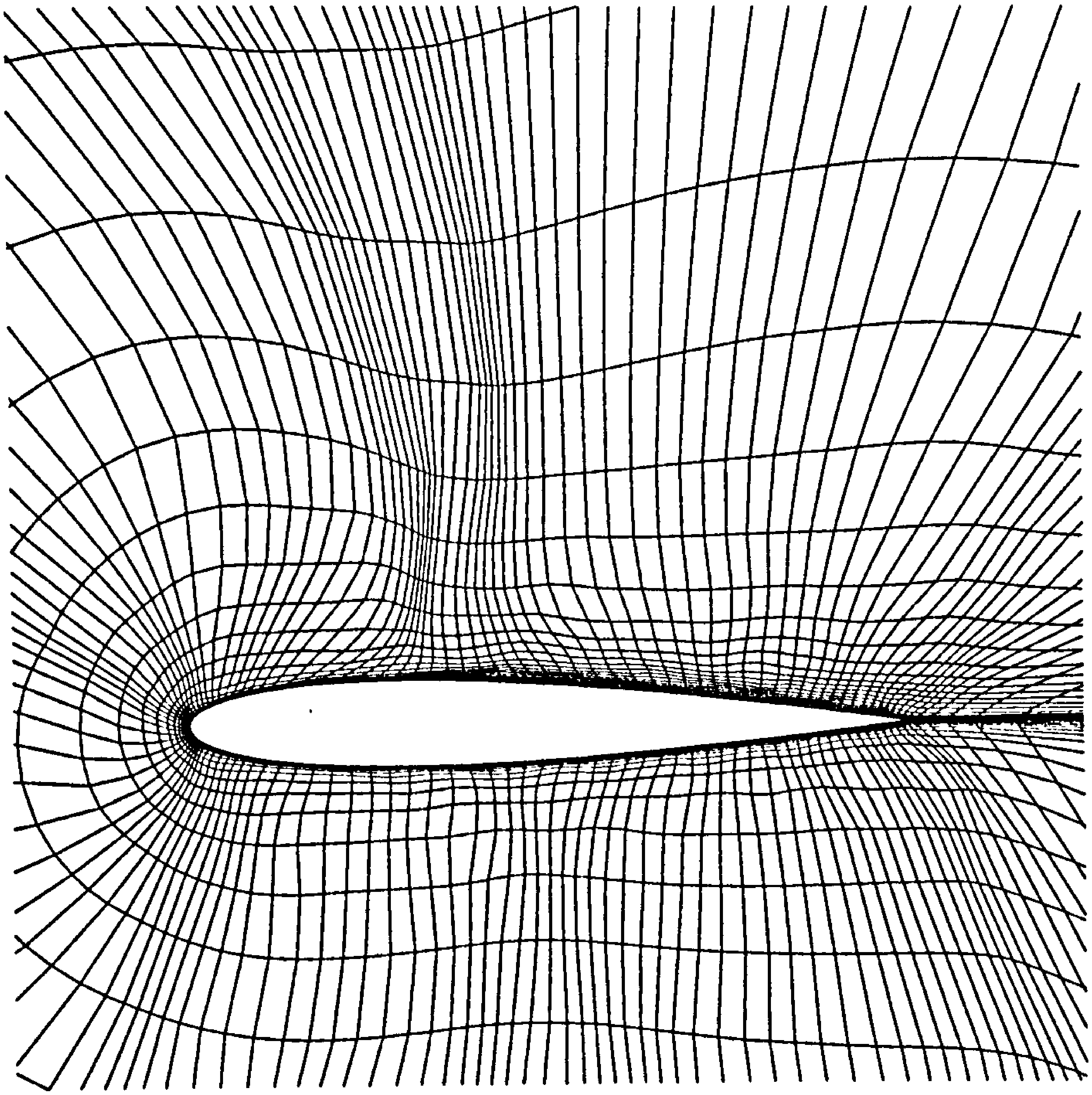


Fig.5.40 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to NDF - Strategy A.

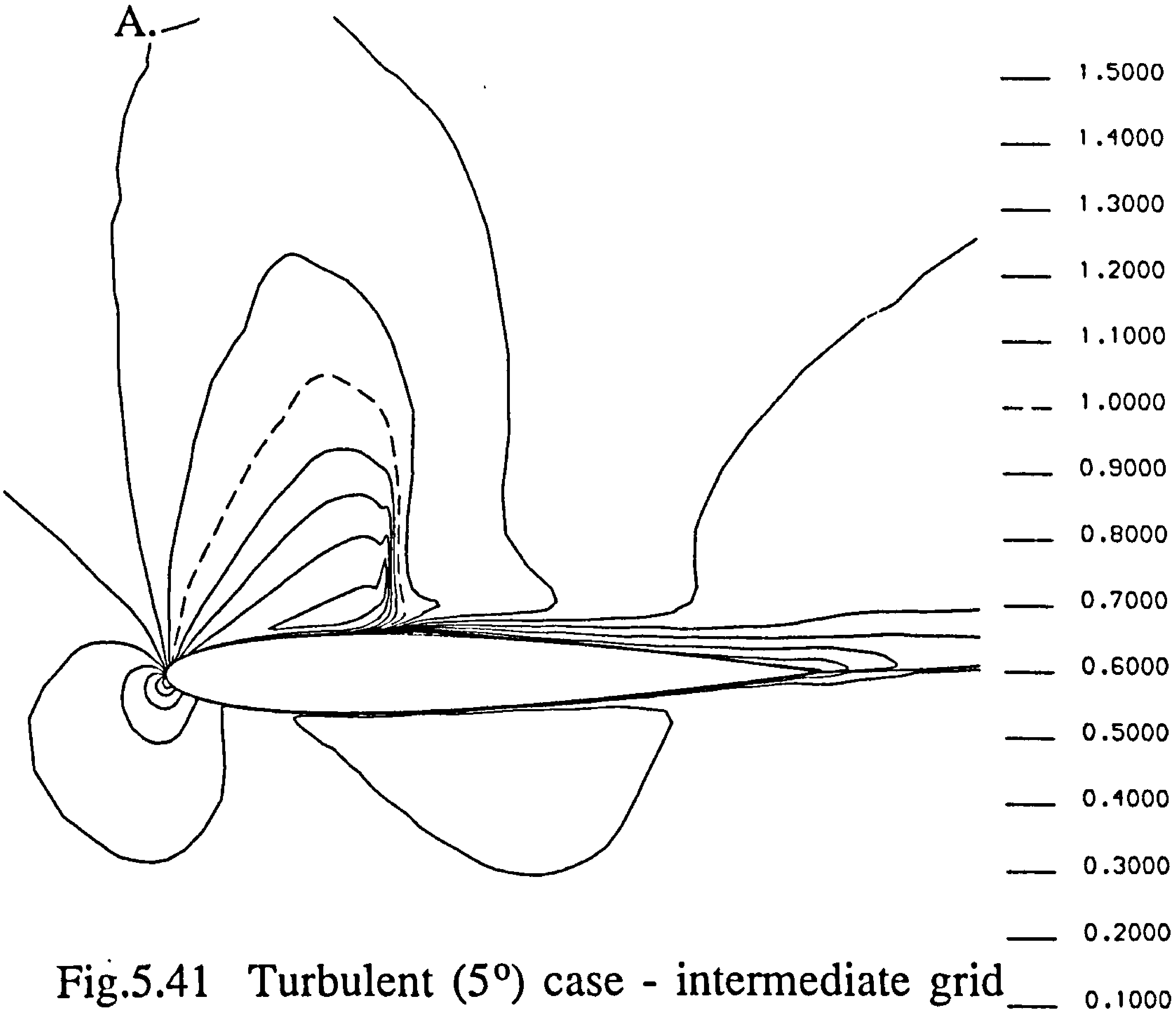


Fig.5.41 Turbulent (5°) case - intermediate grid - Mach number contours after fourth adaptation to NDF - Strategy B.

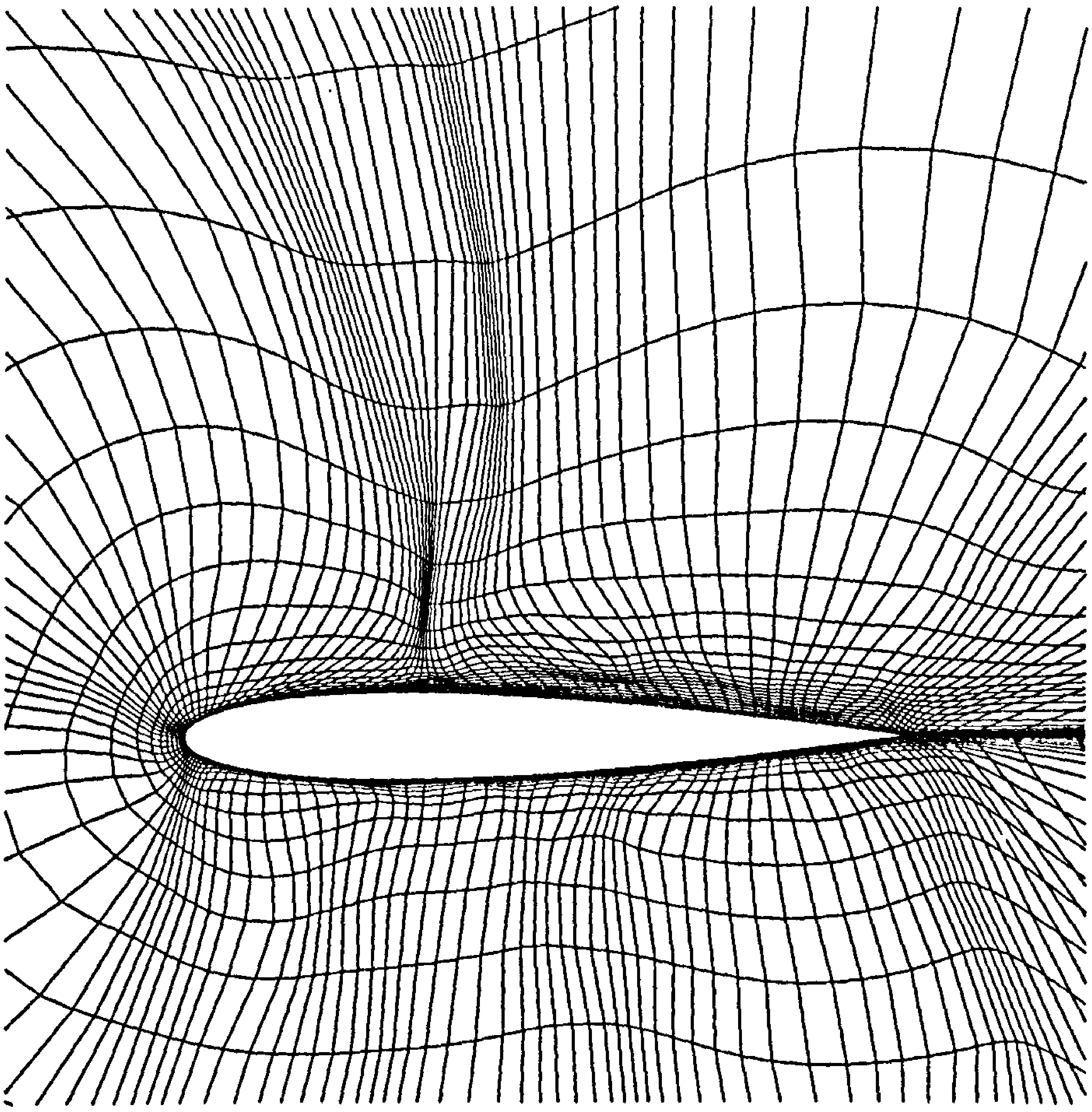


Fig.5.42 Turbulent (5°) case - intermediate grid - grid after fourth adaptation to NDF - Strategy B.

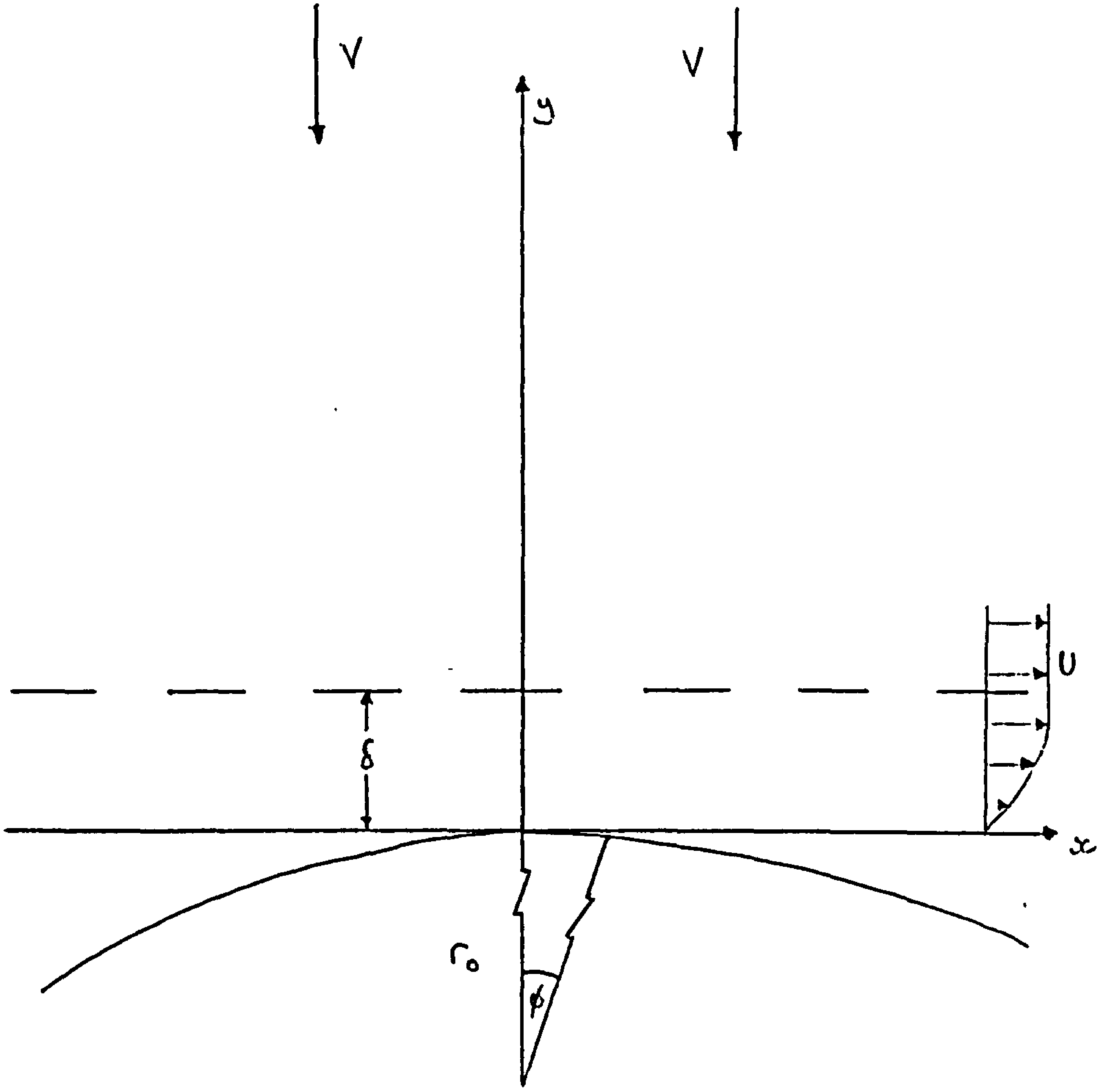
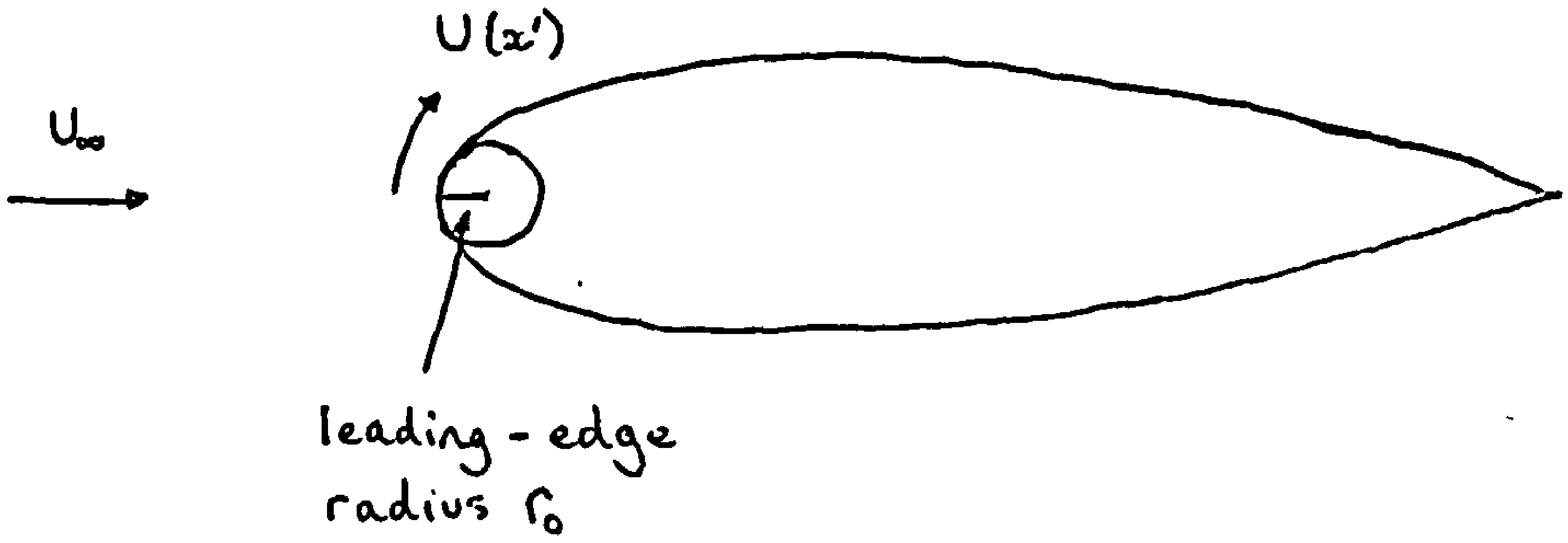


Fig.C.1 Stagnation point flow - quantities used in Appendix C.



Appendices

Appendix A - The Baldwin-Lomax Algebraic Turbulence Model

The Baldwin-Lomax[26] model gives the turbulent viscosity as:

$$\mu_t = \begin{cases} (\mu_t)_{inner} & y<y_{crossover} \\ (\mu_t)_{outer} & y_{crossover}<y \end{cases}$$

where y is the dimensionless distance from the wall and $y_{crossover}$ is the minimum value of y at which $(\mu_t)_{inner} = (\mu_t)_{outer}$. In the inner region,

$$(\mu_t)_{inner} = \rho\, l^2\, |\omega|$$

where

$$l = k\, y \left[1 - \exp(-y^+/A^+)\right]$$

is the length scale in the inner region, k is a model constant, $|\omega|$ is the dimensionless magnitude of the vorticity vector and y^+ is the dimensionless distance from the wall in wall units :

$$y^+ = \frac{(\rho_w\, \tau_w)^{1/2}\, y}{\mu_w}\,.$$

In the outer region,

$$(\mu_t)_{outer} = \rho\, K\, C_{cp}\, F_{wake}\, F_{Kleb}$$

where K and C_{cp} are model constants, and F_{wake} is

$$F_{wake} = \min \left\{ y_{max}\, F_{max} \text{ or } C_{wk}\, y_{max}\, U^2_{Diff}/F_{max} \right.$$

and y_{max} and F_{max} are the location and value of the maximum of the function

$$F = y\, |\omega| \left\{ 1 - \exp(-y^+/A) \right\}$$

In wake regions, the exponential term in the above equation is set to zero. $U_{Diff} = U_{max} - U_{min}$ where U_{max} is the maximum magnitude of the velocity in the profile and U_{min} is set to zero except in wakes. F_{Kleb} is given by:

$$F_{Kleb} = \left[1 + 5.5 \left\{ \frac{C_{Kleb} y}{y_{max}} \right\}^{6} \right]^{-1} .$$

Transition to turbulence is modelled by setting the eddy viscosity to zero if the computed value is below a certain level. The criterion used is:

$$\mu_t = 0 \qquad \text{if} \qquad \mu_{max\ in\ profile} < C_{mutm} .$$

The model constants normally used are:

$$A^+ = 26, \quad C_{cp} = 1.6, \quad C_{Kleb} = 0.3, \quad C_{wk} = 1.0,$$
$$k = .4, \quad K = .0168, \quad C_{mutm} = 14.$$

Appendix B - The Out-of-Balance Residual Technique for Indirectly-Added Numerical Diffusion Schemes

The following is an abbreviated version of the explanation given in ref.36. Consider the convective transport of a surrogate variable ϕ, and assume, for simplicity, that the velocity U in the x-direction is constant. Central differencing gives:

$$\rho U\frac{\partial\phi}{\partial x}\bigg|_i \approx \rho\frac{U}{2\delta x}(\phi_{i+1}-\phi_{i-1}) \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad \text{(B.1)}$$

and upwind differencing gives:

$$\rho U\frac{\partial\phi}{\partial x}\bigg|_i \approx \rho\frac{U}{\delta x}(\phi_{i}-\phi_{i-1}) \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad \text{(B.2)}$$

assuming equal grid spacing δx for simplicity.

Taylor-series expansions of ϕ about x_i give the following:

$$\begin{aligned}\rho\frac{U}{2\delta x}(\phi_{i+1}-\phi_{i-1}) = \rho U \bigg\{ & \frac{d\phi}{dx}\bigg|_i + \frac{1}{3!}\,\frac{d^3\phi}{dx^3}\bigg|_i(\delta x)^2 + \ldots. \\ & + \frac{1}{\gamma!}\,\frac{d^\gamma\phi}{dx^\gamma}\bigg|_i(\delta x)^{\gamma-1} + \ldots \bigg\} \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad \text{(B.3)}\end{aligned}$$

where $\gamma = 2n+1$, integer n; and

$$\begin{aligned}\rho\frac{U}{\delta x}(\phi_{i}-\phi_{i-1}) = \rho U \bigg\{ & \frac{d\phi}{dx}\bigg|_i - \underline{\frac{1}{2!}\,\frac{d^2\phi}{dx^2}\bigg|_i\delta x} + \ldots. \\ & + \frac{(-1)^n d^n\phi}{n!\ dx^n}\bigg|_i(\delta x)^{n-1} + \ldots \bigg\}. \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad \text{(B.4)}\end{aligned}$$

Upwind differencing gives rise to a leading truncation term (underlined in equation B.4) which resembles a diffusion term and is normally referred to as numerical diffusion. The local magnitude of the numerical diffusion can be assessed, following ref.75. Let $\Phi^*(x,y)$ be a converged solution of the finite-difference equations and consider a location at which upwind differencing has been used. Then Φ^* satisfies the finite-difference analogue of the differential equation:

$$\rho U\frac{\partial\phi}{\partial x}\bigg|_i = \rho\frac{U}{\delta x}(\Phi_i^*-\Phi_{i-1}^*)\ . \tag{B.5}$$

Now consider the case in which central differencing has been used so that:

$$\rho U\frac{\partial\phi}{\partial x}\bigg|_i = \rho\frac{U}{2\delta x}(\Phi_{i+1}^*-\Phi_{i-1}^*)\ . \tag{B.6}$$

If the left hand side of (B.5) is replaced by (B.6) the terms in the finite-difference analogue of the differential equation satisfy an inequality in which the difference between the two sides is an "out-of-balance residual" term T, given by:

$$T = \rho U\left[\frac{1}{2!}\frac{d^2\Phi^*}{dx^2}\bigg|_i\delta x + \ldots\ + \frac{1}{\beta!}\frac{d^\beta\Phi^*}{dx^\beta}\bigg|_i(\delta x)^{\beta-1} + \ldots\ \right] \tag{B.7}$$

where $\beta=2n$, integer n. In cases where upwind differencing has been used in part, or all, of the computation domain, the relative importance of numerical diffusion locally can be examined, on a point-by-point basis, by evaluating the term T and comparing it with the other terms (convection, pressure gradient, physical diffusion) in a similar manner to the work described in Chapter 3 et seq.

Appendix C - Calculation of Theoretical Boundary Layer Thickness at a Leading-Edge Stagnation Point

From the (exact) theory for plane, two-dimensional, stagnation-point flow[64], the boundary-layer thickness there is given by

$$\delta = \eta_\delta \left[\frac{\nu}{a}\right]^{½}$$

where η is a function of the normal distance y, ν is the kinematic viscosity, and a is the constant for the frictionless potential flow $U = ax$, $V = -ay$. The situation is shown in the top part of fig.C.1. From the work of Hiemenz[76] and Howarth[77], $\eta_\delta \approx 2.4$.

Assume that the above theory applies in the close vicinity of the leading edge of an aerofoil as suggested in fig.C.1. From the potential flow around a cylinder, $U(x') = 2U_\infty \sin\phi$, where x' is distance along the surface from the stagnation point. For small ϕ, referring to fig.C.1, $x \approx R_o \sin\phi$. Putting

$$U(x) = ax \approx U(x') = 2U_\infty \sin\phi \approx \frac{2U_\infty x}{R_o}$$

gives $a \approx 2U_\infty/R_o$. From the above and the definition of the Reynolds number based on chord as $R_c = U_\infty c/\nu$, the boundary-layer thickness at the leading edge is given by

$$\delta \approx 2.4\left[\frac{R_o \nu}{2U_\infty}\right]^{½} = 2.4\left[\frac{R_o c}{2R_c}\right]^{½}.$$

For the NACA 0012 aerofoil the leading-edge radius $R_o = 0.0158c$. In the case considered in the text, $R_c = 10000$. The estimate for the boundary-layer thickness is

$$\delta/c \approx 2.4\,(0.0158/20000)^{½} = 0.00213.$$