The First International Conference on

NUMERICAL GRID GENERATION IN COMPUTATIONAL FLUID DYNAMICS

LANDSHUT, W. GERMANY
14-17th July, 1986

TITLE OF PAPER

Author's Name Here

COMPOSITE COMPUTATIONAL GRID GENERATION USING OPTIMIZATION

Stephen R. Kennon
Graduate Research Assistant
Dept. of Aerospace Engineering and Engineering Mechanics
University of Texas at Austin

George S. Dulikravich
Associate Professor
Dept. of Aerospace Engineering
Pennsylvania State University

ABSTRACT

A method is described for generating composite computational grids using principles of nonlinear optimization. The method is an extension of a technique developed by the authors for generation of single grids to the case of generating composite (patched or zonal) grids for complex configurations. The properties of the resulting composite grids include maximal local grid orthogonality and smoothness in addition to having a user-specified degree of clustering towards any or all of the grid boundaries. Three types of boundary conditions can be enforced on interfaces between different zones of the composite grid. Examples of zonal grids for a turbine cascade show the usefulness of this method for realistic turbomachinery flow computations.

Last line here

INTRODUCTION

Most finite-difference or finite-volume schemes for the solution of partial differential equations require a boundary conforming computational grid in the physical plane that can be mapped to a rectangular computational domain. In addition, the grid should be as smooth and orthogonal as possible for the schemes to produce accurate solutions. These requirements are often nearly impossible to meet with standard grid generation schemes that are based on mapping a single rectangular computational domain to the physical domain. Thus, various researchers have looked at the possibility of dividing the physical domain into several blocks (also called patches or zones) which are each associated with a rectangular computational domain. This allows the discretization of very arbitrary domains, at the expense of increased user input to define the subdivision of the physical domain. This zonal grid generation approach is the method adopted in this paper.

The method of Kennon and Dulikravich[1] for generating computational grids using principles of nonlinear optimization is here extended to the case of generating composite grids with user specified clustering towards any or all grid boundaries. The iterative optimization procedure is performed on each zone of the grid in turn, and then various boundary conditions are imposed on the inter-zonal interfaces. The conditions that can be enforced include: a) grid points are fixed; b) grid points are allowed to float according to the requirements of an optimal grid; c) grid points are allowed to float along a specified line (or plane in the three-dimensional case), d) grid points are allowed to float along a specified curve (or surface in the three-dimensional case) defined by a cubic spline through the initial boundary points. By combining grid zones and these conditions between the zones, very complex domains can be successfully discretized and good quality grids achieved. We present the method for the case of generating two-dimensional grids although the method can be extended to the problem of generating three-dimensional[2] zonal grids.

ANALYSIS

We begin by assuming that the physical domain is divided initially into a number of zones, each of which are mappable to a rectangular computational domain. Within each zone, the grid is optimized and information between zones is passed through the conditions imposed on the zonal interfaces.

The grid optimization method within each zone of the grid is similar to the variational method of Brackbill and Saltzman[3] in that certain grid quality measures (smoothness and orthogonality) are minimized. The variational method begins with two functionals that measure the global smoothness and orthogonality of the grid. Next, these functionals are converted to their equivalent differential forms by the use of the elementary cells, which could be formulated as, for example,

$$A_1 = \| \mathbf{r}_{i+1,j} \times \mathbf{r}_{i,j+1} \|$$

The master cell is orthogonal if the curvilinear coordinate directions i = const. and j = const. intersect at P_{ij} at right angles (Fig. 1). Hence, a quantitative measure of the departure from local grid orthogoanlity can be expressed as, for example,

$$ORT_{i,j} = (\mathbf{r}_{i+1,j} \cdot \mathbf{r}_{i,j+1})^2 + (\mathbf{r}_{i,j-1} \cdot \mathbf{r}_{i+1,j})^2 + (\mathbf{r}_{i-1,j} \cdot \mathbf{r}_{i,j-1})^2 + (\mathbf{r}_{i,j+1} \cdot \mathbf{r}_{i-1,j})^2$$

Finally, a global objective function can be formed for each zone as, for example,

$$F = \Sigma_i \Sigma_j \; [\alpha \, ORT_{i,j} + (1-\alpha) SM_{i,j}]$$

where α is a user specifed weighting factor $0 \le \alpha \le 1$. The global objective function can then be minimized using, say, the Fletcher-Reeves conjugate gradient optimization algorithm.

<u>Grid Clustering</u>

To achieve high accuracy solutions, a computational grid needs to be clustered in regions where the numerical solution error is the largest (usually in regions of high flow gradients). To achieve grid clustering, we use a simple form of the spring analogy[4]. Assume that we wish to cluster the grid towards the j=1 grid boundary. We form the grid weighting function defined by

$$W_{i,j} = \| \mathbf{r}_{i,j+1} \|^2 + \varepsilon^2 \| \mathbf{r}_{i,j-1} \|^2$$

where ε is a specified constant > 1. The global cost functional F is modified by adding the weight function W to it:

$$F = \Sigma_i \Sigma_j \; [\alpha \, ORT_{i,j} + (1-\alpha) SM_{i,j} + \beta \, W_{i,j}]$$

where β is a specified constant of O(1). This will have the effect of clustering the grid towards the j=1 boundary in an exponential fashion while maintaining maximal local grid smoothness and orthogonality. Similar weight functions can easily be added to generate adaptive grids that adapt to, say, flow field gradients[4].

RESULTS

A fast computer code for simultaneous optimization of the zones of a composite computational grid was developed. Input to the program consists of the initial grid point coordinates for all the zones, and various boundary condition information for each zonal interface. In addition, the user specifies information related to the desired clustering of the grid. We present a representative example of the capabilities of the method for generating good quality zonal grids. This example consists of the problem of generating a periodic boundary conforming computational grid for a turbine cascade. Turbine blades usually have rounded trailing edges. Thus, one would like to have an O-type grid around the trailing edge to maximize the accuracy of the solution in that vicinity. On the other hand, for a viscous flow calculation, one would like to have, say, a C-type grid around the blade to aid in capturing the viscous wake. In addition, the grid periodicity requirement makes a purely O-type grid extremely sheared. To alleviate this problem, it has been suggested that a zonal O- and C- type grid be used. We generated an initial O-C grid around a typical turbine cascade using a fast algebraic method (figs. 2a-c). This initial grid was very sheared and had regions of overlapped grid cells (fig. 2c). The grid was then optimized as shown in figures 3a-c. Grid points along the interface between the O- and C-type grids and along the 'wake' of the C-type grid were allowed to float and therefore move to optimal positions. In addition, the grid was specified to be clustered towards the blade surface with a clustering factor $\varepsilon = 1.8$ (fig. 3c). Fig. 3b is a view of the trailing edge region showing the singular point at the interface between the O and C grids. Note the overall smoothness and orthogonality of grid in the vicinity of the trailing edge. This grid was generated in 30 iterations on a VAX 11/780 computer. Total execution time was 27.43 seconds for this grid consisting of 830 nodes (O-grid: 48 x 4 cells, C-grid: 64 x 8 cells) which included all the I/O done by the program. This example clearly shows the versatility and efficiency of the method for generating good quality zonal grids. The method can be extended to generate three-dimensional[2] zonal adaptive[4] grids and unstructured, finite-element type grids.

SUMMARY

A method for fast generation of zonal grids has been presented. The technique is based on iteratively optimizing the smoothness and orthogonality of the grid within each zone. Resulting zonal grids have been shown to be of high quality and can be generated very efficiently.

REFERENCES

1. Kennon, S. R. and G. S. Dulikravich, "A Posteriori Optimization of Computational Grids," AIAA Paper 85-0483, Reno, Nevada, USA, January, 1985.

2. Carcaillet, R., S. R. Kennon and G. S. Dulikravich, "Optimization of Three-Dimensional Computational Grids," AIAA Paper 85-4087, Colorado Springs, Colorado, USA, October, 1985.

3. Brackbill, J. U. and J. S. Saltzman, "Adaptive Zoning for Singular Problems in Two Dimensions," Journal of Computational Physics **46** (1982), pp. 342-368.

4. Carcaillet, R., "Generation and Optimization of Flow Adaptive Computational Grids," M. Sc. Thesis, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, Texas, USA, August, 1985.

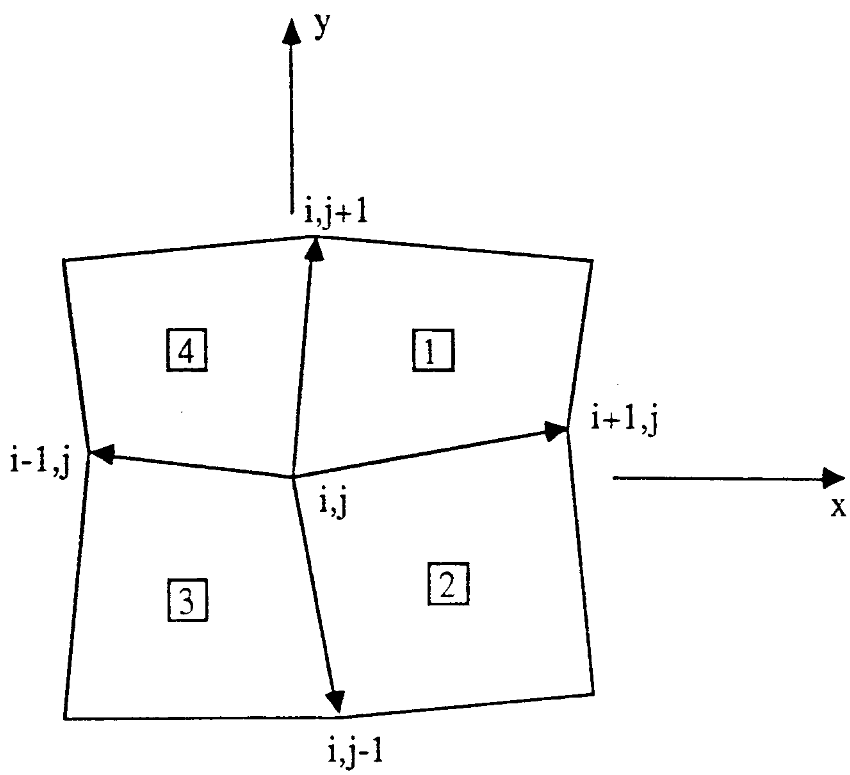

Fig. 1 Master cell composed of four elementary cells

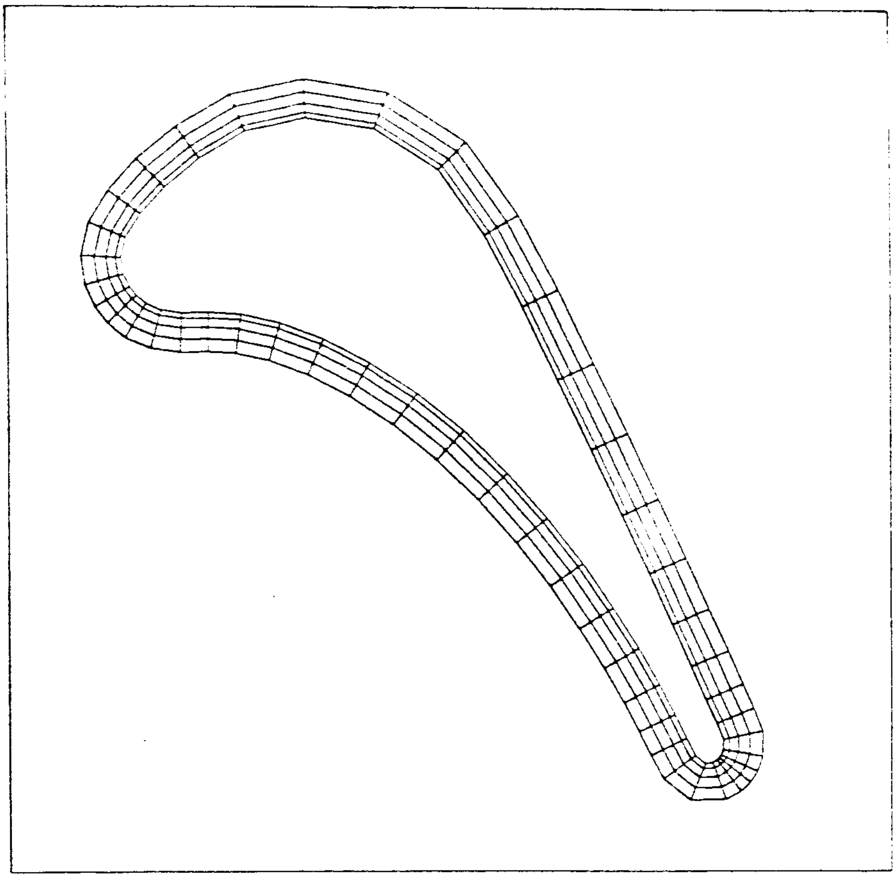

Fig. 2a Initial O-type grid about the turbine blade

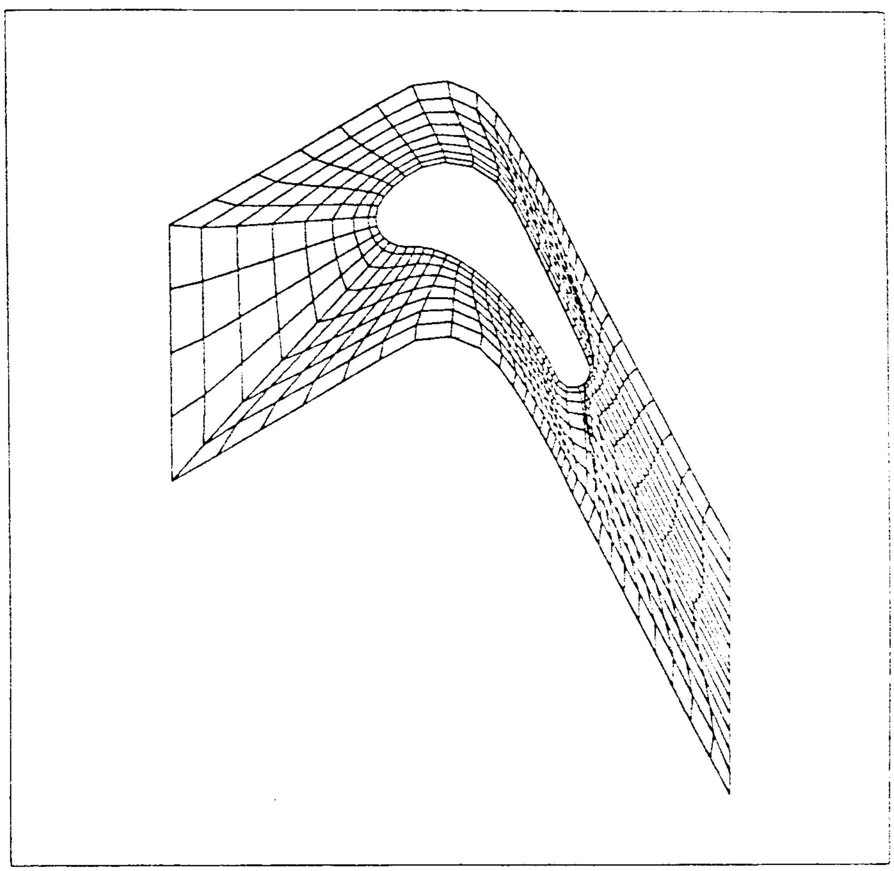

Fig. 2b Initial C-type grid surrounding the O-type grid

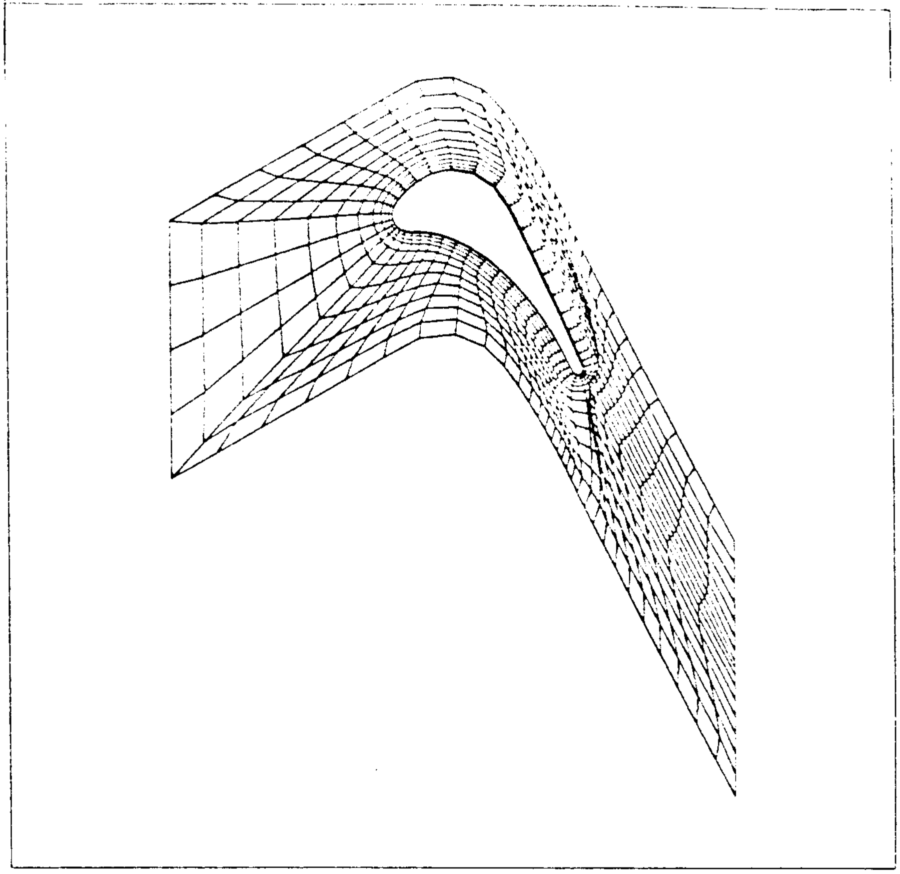

Fig. 2c Initial composite O-C grid

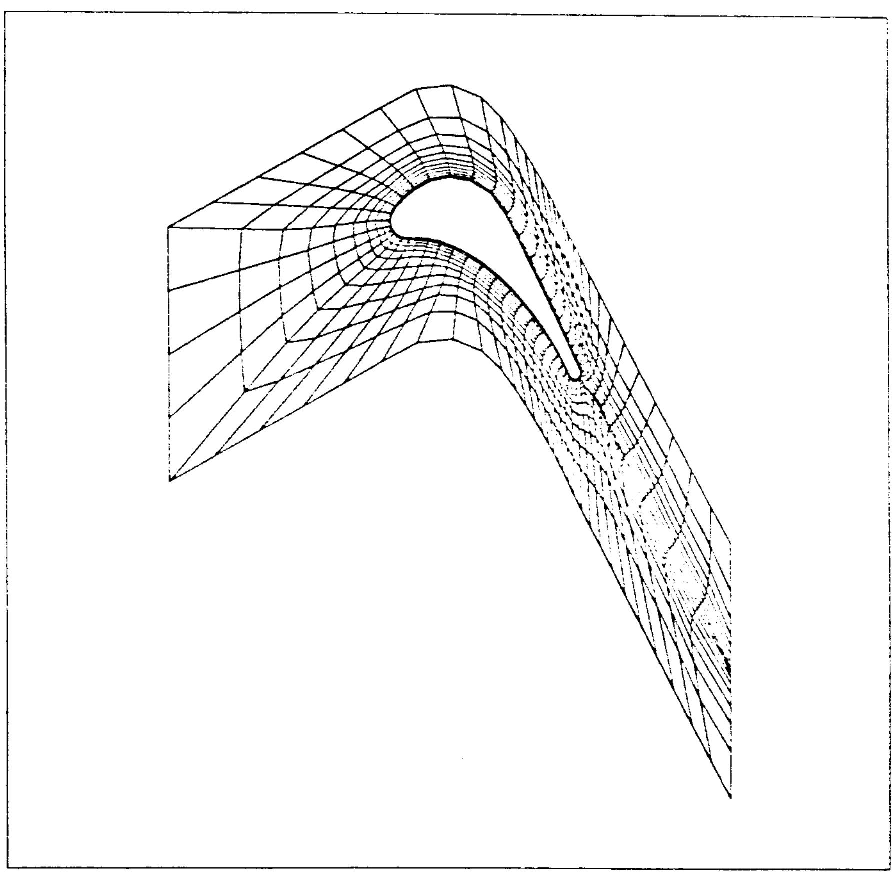

Fig. 3a Optimized composite O-C grid

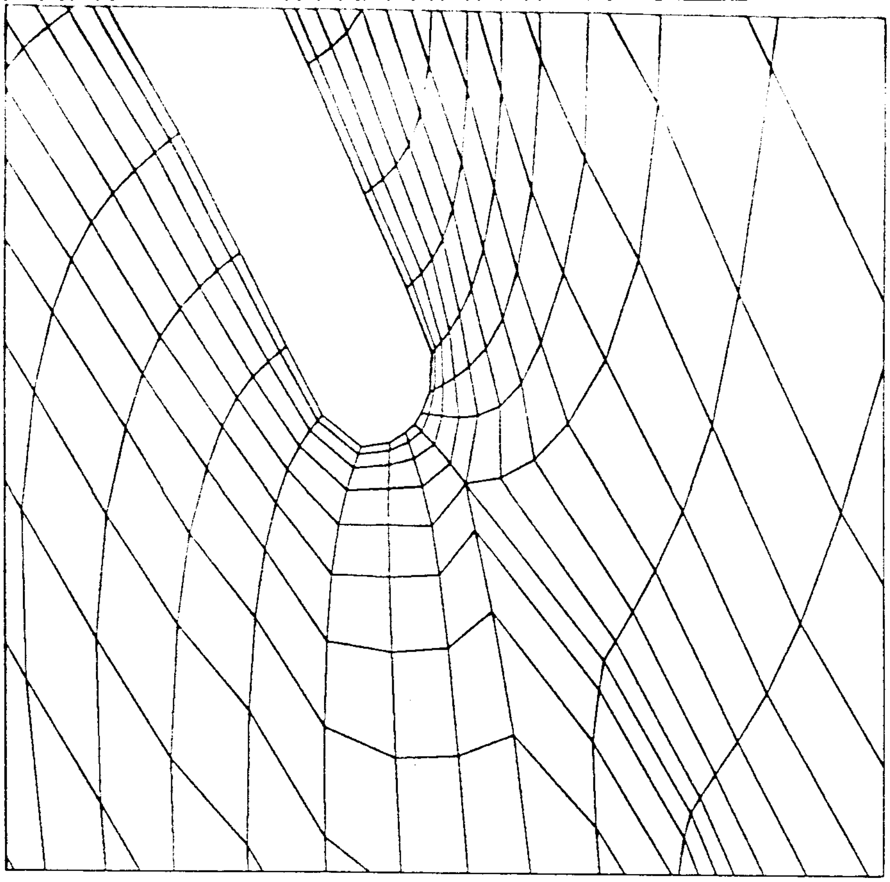

Fig. 3b Optimized grid: trailing edge detail

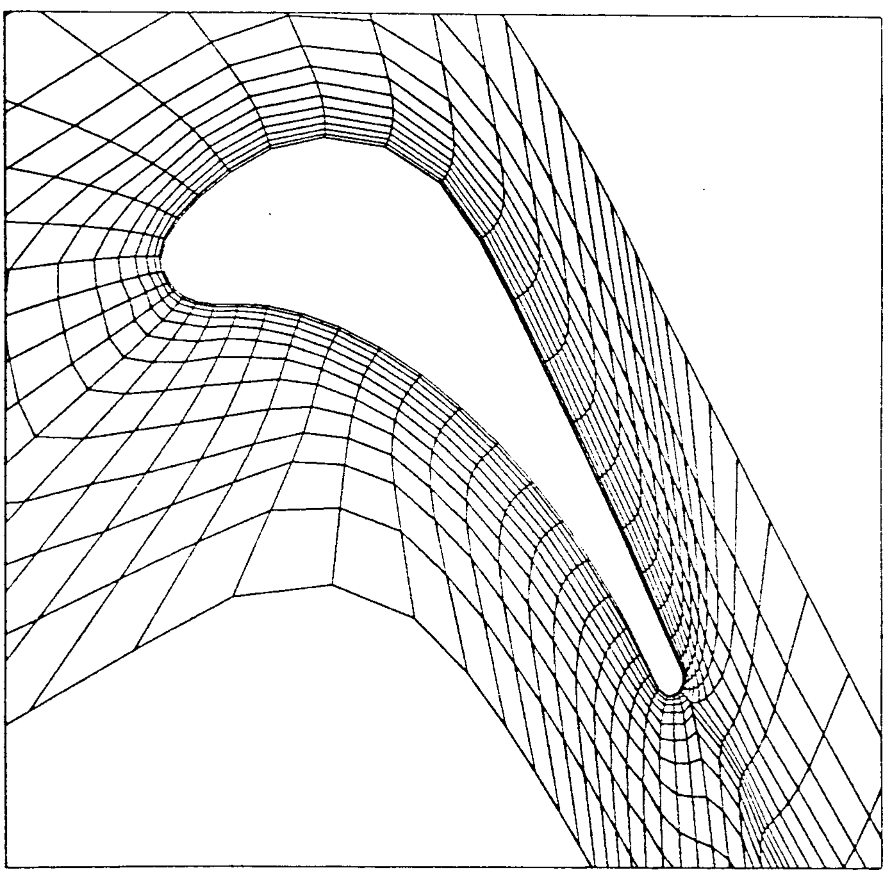

Fig. 3c Optimized grid: detail of turbine blade