Mississippi State University

Department of Aerospace Engineering

Drawer A

Mississippi State, MS 39762

601-325-3623

Improved 3-D Turbomachinery CFD Algorithm

November 1, 1986 - May 15, 1988

June 1988

Final Report

NAS8-36185

J. Mark Janus

David L. Whitfield

Prepared for

Scientific Research Associates, Inc.

Glastonbury, CT 06033

(NASA-CR-179364) IMPROVED 3-D TURBOMACHINERY CFD ALGORITHM Final Report, 1 Nov. 1986 - 15 May 1988 (Mississippi State Univ.) 56 p CSCL 20D

N88-26619

Unclas

G3/34 0154098

ABSTRACT

This report describes the building blocks of a computer algorithm developed for the time-accurate flow analysis of rotating machines. The flow model is a finite volume method utilizing a high resolution approximate Riemann solver for interface flux definitions. This block LU implicit numerical scheme possesses apparent unconditional stability. Multi-block composite gridding is used to orderly partition the field into a specified arrangement. Block interfaces, including dynamic interfaces, are treated such as to mimic interior block communication. Special attention is given to the reduction of in-core memory requirements by placing the burden on secondary storage media. Broad applicability is implied, although the results presented are restricted to that of an even blade count configuration. Several other configurations are presently under investigation, the results of which will appear in subsequent publications.

INTRODUCTION

Current interest in the design of highly efficient turbomachinery has spurred the development of computer algorithms capable of accurately analyzing the flowfields of rotating geometries. These new algorithms can be used to complement the present-day algorithms which are based primarily on empiricism. In addition, the design process can be accelerated with the help of computational fluid dynamics (CFD) as a preliminary geometry analyzer. CFD is progressing to the stage where routine initial geometry flow analysis of relatively complex rotating machines will be performed to reduce the overall cost of project analysis. This is being accomplished by the recent, significant improvements in both computer hardware and software. This report is concerned with the software aspect of computational analysis.

At present, there exists a need for a robust computational algorithm to accurately simulate the fluid flow within a three-dimensional rotating geometrical environment. Of key interest will be computational efficiency, solution accuracy, and logical simplicity. Previously, the focus of this project was on the computational efficiency of the algorithm.[1] This was primarily due to a realization that the flowfield in question would be so computationally exhaustive that even state-of-the-art computational resources would require excessive turn-around time. Although algorithm efficiency is still a premier issue, it was evident at the onset of the continuation of this project that the present level of efficiency must be accepted for the time being. The time had come to take the numerical algorithm and incorporate it into a code whose framework centered around turbomachinery.

This report outlines the mathematical background material and the fundamental code logic necessary for the development of a three-dimensional computer code suitable for the time-accurate analysis of turbomachinery. A detailed report[2] on this subject matter is currently in preparation and not available at the time of this printing.

In the next section a brief description is given of the mathematical equations (tools) used in this effort to develop a flowfield model. The third section outlines the procedure to build a coherent mathematical model within which each equation is exploited for its beneficial attributes. The third section also contains a description of the numerical scheme developed to solve these modeling equations. The fourth section presents the logical approach developed for simulating a dynamic cylindrical environment in addition to outlining specific routines developed to optimize (tailor) the computer code for use on a vector processor, namely a Cray X-MP. The results of a complex rotating configuration are presented in section five, demonstrating the utility of the turbomachinery computer algorithm. The final section is reserved for the conclusions of this entire endeavor along with future plans to enhance the generality and solution accuracy of the code.

MODELING MATHEMATICS

Conservative Integral Model

The most fundamental form of mathematical model describing the behavior of a continuum fluid can be derived using a control volume approach in an Eulerian reference frame. For computational brevity an assumption of a nonconducting, inviscid, perfect gas with no body forces will be made here, yielding

$$\frac{\partial}{\partial t}\int_{\mathbf{V}} \rho d\mathbf{V} = -\int_{\mathbf{S}} \rho\vec{V}\cdot\vec{n}d\mathbf{S}, \tag{1a}$$

$$\frac{\partial}{\partial t}\int_{\mathbf{V}} \rho\vec{V}d\mathbf{V} = -\int_{\mathbf{S}} (\rho\vec{V}\cdot\vec{n}d\mathbf{S})\vec{V} - \int_{\mathbf{S}} p\vec{n}d\mathbf{S}, \tag{1b}$$

$$\frac{\partial}{\partial t}\int_{\mathbf{V}} \rho(\tilde{e} + \frac{|\vec{V}|^2}{2})d\mathbf{V} = -\int_{\mathbf{S}} \rho(\tilde{e} + \frac{|\mathbf{V}|^2}{2})\vec{V}\cdot\vec{n}d\mathbf{S} - \int_{\mathbf{S}} p\vec{V}\cdot\vec{n}d\mathbf{S}, \tag{1c}$$

$$e = \rho(\tilde{e} + \frac{|\vec{V}|^2}{2}) = \frac{p}{\gamma-1} + \frac{\rho|\vec{V}|^2}{2}, \qquad \gamma = \frac{C_p}{C_v}, \tag{1d}$$

where $\mathbf{V}$ is a control volume bounded by surface $\mathbf{S}$.

Using divergence theory and additional vector notation, Eqs. (1) can be cast in the following condensed form.

$$\frac{\partial}{\partial t}\int_{\mathbf{V}} \vec{q}d\mathbf{V} + \int_{\mathbf{V}} \vec{\nabla}\cdot\vec{f}d\mathbf{V} = \vec{0}, \tag{2}$$

where in a Cartesian reference frame

$$\vec{q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \quad \vec{f} = \begin{bmatrix} \rho u & \rho v & \rho w \\ \rho u^2+p & \rho uv & \rho uw \\ \rho uv & \rho v^2+p & \rho vw \\ \rho uw & \rho vw & \rho w^2+p \\ (e+p)u & (e+p)v & (e+p)w \end{bmatrix}.$$

For use within the context of this report (non-constant **V**), the time differentiation appearing in Eq. (2) must be performed with the aid of Leibnitz's Rule. With the aforementioned assumptions, Eq. (2) is commonly referred to as the Euler equations.

Conservative Differential Form

Equation (2) holds for an arbitrary fixed control volume **V**, consequently the integrand satisfies

$$\frac{\partial \vec{q}}{\partial t} + \vec{\nabla} \cdot \vec{f} = \vec{0}. \tag{3}$$

This equation, referred to as the differential conservation law form of the Euler equations, will prove to be very beneficial in matching the numerics with the physics being modeled.

For generality it is desirable to transform the equations from a Cartesian reference frame to a body fitted curvilinear reference frame. The general curvilinear axes are defined as follows

$$\begin{aligned} \xi &= \xi(x,y,z,t), \\ \eta &= \eta(x,y,z,t), \\ \zeta &= \zeta(x,y,z,t), \\ \tau &= t. \end{aligned} \tag{4}$$

Note, the obvious implication that a body fitted reference frame for a dynamic geometry is inherently time dependent.

The details of the transformation are readily available,[2,3,4] hence they are not repeated here. Transforming the Cartesian equations to the more general set of curvilinear equations (dropping vector symbols) yields

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0, \tag{5}$$

where

$$Q = J \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad F = J \begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ \rho w U + \xi_z p \\ U(e+p) - \xi_t p \end{bmatrix},$$

$$G = J \begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ \rho w V + \eta_z p \\ V(e+p) - \eta_t p \end{bmatrix}, \qquad H = J \begin{bmatrix} \rho W \\ \rho u W + \zeta_x p \\ \rho v W + \zeta_y p \\ \rho w W + \zeta_z p \\ W(e+p) - \zeta_t p \end{bmatrix},$$

with the contravariant velocities

$$U = \xi_x u + \xi_y v + \xi_z w + \xi_t,$$

$$V = \eta_x u + \eta_y v + \eta_z w + \eta_t,$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w + \zeta_t,$$

the Jacobian of the inverse transformation, i.e. $\partial(x,y,z)/\partial(\xi,\eta,\zeta)$

$$J = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi(x_\eta x_\zeta - z_\eta x_\zeta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta),$$

and the metric quantities

$$\xi_x = J^{-1}(y_\eta z_\zeta - z_\eta y_\zeta), \qquad \xi_z = J^{-1}(x_\eta y_\zeta - y_\eta x_\zeta),$$

$$\xi_y = J^{-1}(z_\eta x_\zeta - x_\eta z_\zeta), \qquad \xi_t = -x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z,$$

$$\eta_x = J^{-1}(z_\xi y_\zeta - y_\xi z_\zeta), \qquad \eta_z = J^{-1}(x_\zeta y_\xi - y_\zeta x_\xi),$$

$$\eta_y = J^{-1}(x_\xi z_\zeta - z_\xi x_\zeta), \qquad \eta_t = -x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z,$$

$$\zeta_x = J^{-1}(y_\xi z_\eta - z_\xi y_\eta), \qquad \zeta_z = J^{-1}(x_\xi y_\eta - y_\xi x_\eta),$$

$$\zeta_y = J^{-1}(x_\eta z_\xi - z_\eta x_\xi), \qquad \zeta_t = -x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z.$$

Quasilinear Form

A mathematical analysis of the differential form can yield significant insight into the physics of the flowfield in question. Guided by this insight, specific numerical techniques can be employed to more accurately model the flowfield. This analysis begins with the derivation of yet another form of the equations. The final form necessary in the development of this mathematical model is derived from the differential form.

Consider differentiating the vector-valued flux functions F, G, and H (homogeneous functions of degree one in Q) appearing in Eq. (5) with respect to the dependent variable vector Q. The resulting flux Jacobians A, B, and C, are written

$$A = \frac{\partial F}{\partial Q}, \quad B = \frac{\partial G}{\partial Q}, \quad C = \frac{\partial H}{\partial Q} .$$

Substituting these expressions in Eq. (5) yields the quasilinear form of the Euler equations. It should be noted that the quasilinear form, given below, is not in conservation law form.

$$\frac{\partial Q}{\partial \tau} + A \frac{\partial Q}{\partial \xi} + B \frac{\partial Q}{\partial \eta} + C \frac{\partial Q}{\partial \zeta} = 0. \tag{6}$$

Riemann Problem

Essential to the investigation of compressible transonic fluids is the study of discontinuous solutions. To complete the development of the mathematical model, it is necessary to introduce the concept of a particular type of initial-value problem for a hyperbolic system of conservation laws. To begin the discussion, a digression to one-dimension is in order. Consider for example, the one-dimensional hyperbolic system

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0, \tag{7}$$

where the initial data are

$$u(x,0) = \begin{cases} u_L & \text{for } -\infty < x \leq 0 \\ \\ u_R & \text{for } 0 \leq x < \infty \end{cases},$$

and $u(x,t)$ is an **m** component column vector and $f(u)$ is an **m** component vector-valued flux function. The preceding example of the simplest kind of discontinuous data (two semi-infinite regions of constant value, separated by a point of discontinuity) poses what is referred to as a

Riemann problem. Riemann problems present themselves in varying degrees of difficulty, depending on their linearity or lack thereof and on the degrees-of-freedom (**m**) they represent. For a detailed study of the Riemann problem and the mathematical theory of shocks (discontinuities) the reader is referred to [5].

NUMERICAL FORMULAS

Discretized Integral Form

The ultimate goal in computational fluids is to minimize the approximations to the most fundamental modeling equations, presumably to salvage most of the physics, while attaining modest execution times on the available equipment. Hence the approach used here is founded on the discretization of the integral conservation law form of the Euler equations, Eq. (3). This formulation, commonly referred to as a finite volume method, yields the following discretized integral expression for a three-dimensional computational space with finite volume (cell) centers denoted i,j,k:

$$\frac{\Delta Q}{\Delta \tau} + \frac{\delta_i F}{\Delta \xi} + \frac{\delta_j G}{\Delta \eta} + \frac{\delta_k H}{\Delta \zeta} = 0, \tag{8}$$

where

$$\delta_{(\ )}(\cdot) = (\cdot)_{(\ )+1/2} - (\cdot)_{(\ )-1/2} .$$

In this expression the components of the dependent variable vectors, $Q_{i,j,k}$ represent average cell values. It is therefore evident that some method must be devised to accurately represent the vector-valued flux functions F, G, and H on the bounding surfaces (faces) of the cell. One of the methods used in this study is based on a one-dimensional analysis of the Riemann problem local to each interface, established by the discontinuous nature of the dependent variable vector Q in a finite volume field. To facilitate the introduction of this method, a return to a one-dimensional Cartesian space is in order.

First Order Flux Formula

The discretized integral form of the Euler equations written for one spatial dimension appears as

$$\frac{\Delta q}{\Delta t} + \frac{\delta_i f}{\Delta x} = 0 \tag{9}$$

Godunov[6] proposed a procedure to obtain a global solution to Eq. (9) by solving the set of Riemann problems presented by the interface discontinuities. The Riemann problem local to each interface is costly to solve exactly, due to the necessary iteration. Many investigators[7,8,9,10] have made attempts to lessen this computational expense by approximating the solution of the Riemann problem. In essence these methods yield an approximate solution to the exact equation, and are referred to as approximate Riemann solvers.

In [11], Philip Roe suggests an alternate procedural choice. Roe proposed to obtain the exact solution to an approximate equation. The cleverness of Roe is evidenced by his choice of approximate equation. Consider the quasilinear form of Eq. (9)'s parent conservation law.

$$\frac{\partial q}{\partial t} + \bar{A}(q_L, q_R) \frac{\partial q}{\partial x} = 0 \tag{10}$$

where $\bar{A}(q_L, q_R) = \partial f / \partial q$ is a constant matrix representative of local interface conditions. Matrix $\bar{A}$ is chosen to have the following specific list of properties, which Roe "christened Property U (since it is intended to ensure uniform validity across discontinuities)":

(i) It constitutes a linear mapping from the vector space q to the vector space f.

(ii) As $q_L \to q_R \to q$, $\bar{A}(q_L,q_R) \to \bar{A}(q)$, where $\bar{A} = \frac{\partial f}{\partial q}$.

(iii) For any q_L, q_R, $\bar{A}(q_L,q_R) \cdot (q_R - q_L) = f_R - f_L$.

(iv) The eigenvectors of $\bar{A}$ are linearly independent.

Restricting $\bar{A}$ to the satisfaction of Property U results in a special averaging process for the dependent variables from which $\bar{A}$ is constructed. Referred to as "Roe averaged", the dependent variables are given by the following expressions:

$$\rho = (\rho_L \rho_R)^{1/2}, \tag{11a}$$

$$u = \frac{\rho_L^{1/2} u_L + \rho_R^{1/2} u_R}{\rho_L^{1/2} + \rho_R^{1/2}}, \tag{11b}$$

$$H = \frac{\rho_L^{1/2} H_L + \rho_R^{1/2} H_R}{\rho_L^{1/2} + \rho_R^{1/2}}, \tag{11c}$$

where the total enthalpy, H, is defined

$$H = \frac{1}{\rho}(e + p). \tag{11d}$$

The flux difference can be expressed as

$$df = f_R - f_L = \bar{A} \cdot (q_R - q_L) = \bar{A} \cdot dq \tag{12}$$

where $\bar{A}$ is constructed with "Roe averaged" variables. Armed with the eigensystem of $\bar{A}$ and the knowledge that the interface differential dq is a proportional to the right eigenvectors of $\bar{A}$, the interface flux difference can be written relative to the right eigenvector basis as

$$df = \sum \alpha_j \lambda^{(j)} r^{(j)} = \sum^{+} \alpha_j \lambda^{(j)} r^{(j)} + \sum^{-} \alpha_j \lambda^{(j)} r^{(j)} = df^{+} + df^{-} \qquad (13)$$

Physically, the flux difference is shown to be the composition of a collection of waves. In this equation $r^{(j)}$ is a right eigenvector of $\bar{A}$; α_j is the strength of the jth wave (the jump in the characteristic variable across it); $\lambda^{(j)}$ is an eigenvalue of $\bar{A}$ (the speed of the jth wave), and Σ^{-} and Σ^{+} denote summation over the negative and positive wave speeds, respectively.

The interface flux, see Fig. (1), can be computed from either of the following formula:

$$f_{i+1/2} = f_L + \sum^{-} \alpha_j \lambda^{(j)} r^{(j)} \qquad (14a)$$

$$f_{i+1/2} = f_R - \sum^{+} \alpha_j \lambda^{(j)} r^{(j)} \qquad (14b)$$

$$f_{i+1/2} = \frac{1}{2} \left[f_L + f_R - \sum \alpha_j |\lambda^{(j)}| r^{(j)} \right] \qquad (14c)$$

This first-order interface flux formula developed for the one dimensional equations can be used in a multidimensional space provided the assumption that all waves travel normal to the interfaces is made.

Higher Order (TVD) Flux Formulas

In order to provide solutions of higher spatial accuracy, a family of schemes[12] can be represented by the addition of a corrective flux to the first order flux, Eq.(14), produced by the preceding analysis:

$$f_{i+1/2} = f_{i+1/2} + \frac{(1+\phi)}{4} \left[df^{+}_{i+1/2} - df^{-}_{i+1/2} \right] + \frac{(1-\phi)}{4} \left[df^{+}_{i-1/2} - df^{-}_{i+3/2} \right] \qquad (15)$$

The principle part of the truncation error for this flux formula is found to be

$$TE = -\frac{(\phi - 1/3)}{4}(\Delta x)^2 \frac{\partial^3}{\partial x^3} f(q) \tag{16}$$

The details (common names, order of accuracy, etc.) of the members of this family can be found in [13,14]. Two members, third-order ($\phi = 1/3$) and fully upwind second order ($\phi = -1$) were, by choice, exclusively used in this project.

The discussion of a higher order scheme inherently involves a method used to control spurious oscillations, i.e. dispersive errors. The method, actually methods, used in this project concern limiting components of the interface flux to produce total variation diminishing (TVD) schemes, i.e. non-oscillatory schemes. The following formulas have theoretical development as TVD schemes only in scalar nonlinear equations and systems of linear equations in one-dimension. One of the "limiters" used here, referred to as a minmod limiter, yields the following expressions for the corrective flux terms:

$$\begin{aligned} df^+_{i+1/2} &= \sum^+ L_j(1,-1) r^{(j)}_{i+1/2}, & df^-_{i+1/2} &= \sum^- L_j(1,3) r^{(j)}_{i+1/2}, \\ df^+_{i-1/2} &= \sum^+ L_j(-1,1) r^{(j)}_{i+1/2}, & df^-_{i+3/2} &= \sum^- L_j(3,1) r^{(j)}_{i+1/2}, \end{aligned} \tag{17}$$

where

$$L_j(\ell,n) = \text{minmod}(\sigma^{(j)}_{i+\ell/2}, \beta\sigma^{(j)}_{i+n/2}), \tag{18}$$

with $\sigma^{(j)}$, a parameter proportional to the change in characteristic variables across nearby interfaces, defined by

$$\sigma^{(j)}_{i+\frac{(\cdot)}{2}} = \lambda^{(j)}_{i+\frac{1}{2}}\, \alpha_{j,i+\frac{(\cdot)}{2}} = \lambda^{(j)}_{i+\frac{1}{2}}\, \ell^{(j)}_{i+\frac{1}{2}}\, dq_{i+\frac{(\cdot)}{2}}, \tag{19}$$

where $\ell^{(j)}$ is a left eigenvector of $\bar{A}$. The minmod limiter is then defined

$$\text{minmod}[x,y] = \text{sign}(x)\, \max\{0,\, \min[|x|,\, y\, \text{sign}(x)]\} \tag{20a}$$

and the parameter β is a "compression" parameter given by

$$1 < \beta \leq \frac{3-\phi}{1-\phi} \tag{20b}$$

For this project the maximum β was used in all cases.

Another limiter used here, this one credited to Roe[15], is called Superbee. It is defined

$$L_j(\ell,n) = \text{cmplim}(\sigma^{(j)}_{i+\frac{\ell}{2}},\, \sigma^{(j)}_{i+\frac{n}{2}}), \tag{21}$$

where

$$\text{cmplim}[x,y] = \text{sign}(x)\, \max\{0,\, \min[|x|,\, \beta y\, \text{sign}(x)],\, \min[\beta|x|,\, y\, \text{sign}(x)]\} \tag{22}$$

and the compression parameter β, not defined by Eq.(18b), is taken here to be 2.

Approximately Factored Implicit Scheme

Up to this point no mention has been made as to what time level the numerical interface fluxes appearing in Eq. (8) are evaluated. The underlying theory of the approximate Riemann solver presented thus far is based on explicit concepts which result in an unattractive, rather

stringent time-step restriction. Equation (8) can be written in a linearized discrete-integral delta form to cover a broad class of implicit schemes[16]:

$$[I + \frac{\theta\Delta\tau}{1+\psi} M_{\bullet}^{n}]\Delta Q^{n} = - \frac{\Delta\tau}{1+\psi} R^{n} + \frac{\psi}{1+\psi} \Delta Q^{n-1} \tag{23}$$

Some of the time differencing schemes represented are (θ=1, ψ=1/2) three point backward, (θ=1, ψ=0) backward Euler, and (θ=1/2, ψ=0) trapezoidal.

Formally, all terms appearing in this equation should result from a single theory. Superior results have been obtained, though, by evaluating the residual term R^n with flux difference split theory, and the left-hand-side (LHS) operator with flux vector split theory, see [17]. The rationale behind this is unclear and needs further investigation. Hence the following expressions complete Eq.(23) for this hybrid scheme.

$$M_{\bullet}^{n} = \delta_i A_{\bullet}^{+} + \delta_i A_{\bullet}^{-} + \delta_j B_{\bullet}^{+} + \delta_j B_{\bullet}^{-} + \delta_k C_{\bullet}^{+} + \delta_k C_{\bullet}^{-} \tag{24}$$

with

$$A^{+} = \left(\frac{\partial F^{+}}{\partial Q}\right)^{n}$$

$$A^{-} = \left(\frac{\partial F^{-}}{\partial Q}\right)^{n}$$

$$B^{+} = \left(\frac{\partial G^{+}}{\partial Q}\right)^{n}$$

.

.

.

and

$$R^n = \delta_i F + \delta_j G + \delta_k H, \tag{25}$$

where, F, G, and H result from flux difference split theory.

The LHS of Eq. (23) tends to be cumbersome and difficult to invert, not to mention very costly. In light of this, the LHS was approximately factored into the product of two operators, each of which involve the passage of selected information. Here a forward and a backward operator are used as in [1] (block LU factorization), yielding the following two step (LU) scheme:

$$\left[I + \frac{\theta\Delta\tau}{1+\psi} M_{\bullet}^{+}\right]\left[I + \frac{\theta\Delta\tau}{1+\psi} M_{\bullet}^{-}\right]\Delta Q = -\frac{\Delta\tau}{1+\psi} R^n + \frac{\psi}{1+\psi}\Delta Q^{n-1} \tag{26}$$

or

$$\left[I + \frac{\theta\Delta\tau}{1+\psi}\left(\delta_i A_{\bullet}^{+} + \delta_j B_{\bullet}^{+} + \delta_k C_{\bullet}^{+}\right)\right]\Delta Q^{*} = -\frac{\Delta\tau}{1+\psi} R^n + \frac{\psi}{1+\psi}\Delta Q^{n-1} \tag{27a}$$

$$\left[I + \frac{\theta\Delta\tau}{1+\psi}\left(\delta_i A_{\bullet}^{-} + \delta_j B_{\bullet}^{-} + \delta_k C_{\bullet}^{-}\right)\right]\Delta Q^{n} = \Delta Q^{*} \tag{27b}$$

$$Q^{n+1} = Q^n + \Delta Q^n \tag{27c}$$

Although factoring has been shown to degrade the unconditional stability of Eq. (23), the (LU) scheme apparently retains this touted attribute. Equations (27) are in the final form of the mathematical model developed for the time-accurate analysis of turbomachinery.

Since the approximate Riemann solver is a characteristic based scheme, the characteristic variable boundary conditions developed in [3] relative to a three-dimensional time-dependent body-fitted reference frame for inflow, outflow, and impermeable boundaries are employed where applicable. As in [3] phantom cells are utilized to implement these

boundary conditions. The change in dependent variables, ΔQ^n, is set to zero in the phantom cells for inflow, outflow, and impermeable boundaries.

LOGICAL APPROACH WITH COMPUTATIONAL OPTIMIZATION

Selected Similarity Multi-Block

As mentioned in the introduction, the scope of the field of turbomachinery can encompass extremely complex flowfields. In order to adequately resolve these, a vast number of computational cells must be used. Several reasons can be cited for the segmentation of one virtually insurmountable flow environment into several smaller more manageable flow environments. This is commonly referred to as compositely gridding the field. There exist three basic methods of compositely gridding a field: overlaid, patched, and blocked. The approach taken here, a blocked grid method, is similar to that taken by Belk[4]. Belk has investigated many of the pitfalls encountered when constructing a code of a general multi-block nature. The emphasis in [4] was on developing a computer algorithm to handle a completely arbitrary block arrangement, unfortunately this added to the complexity of the code.

For the case of turbomachinery, the nature of the geometry suggests possible block arrangement and characteristic restrictions which can yield significantly simpler algorithm logic. The block structure proposed here for the specific case of dynamic cylindrical geometries (generally, bladed or finned bodies of revolution) is referred to as selected similarity multi-block.

Selected similarity multi-block is a method by which the benefits of a composite environment are attained while maintaining a reasonably simple, general algorithm. This is accomplished by requiring a fixed-map block arrangement with selectively-similar block characteristics. Hence by establishing prearranged, selectively-similar blocks with a

predefined solution path, algorithm logic can be simplified, thus enhancing code readability. In short, an attempt is being made to walk the fine line separating code generality from code simplicity. The fundamental restrictions applied in this approach deal with block arrangement and characteristics.

A natural arrangement presents itself when dealing with cylindrical frames. Consider partitioning the macro-block (with accompanying global curvilinear axes) into partial cylinder blocks, see Fig. 2. Further considerations into the particular eccentricities of rotating geometries indicate a desire for additional partitioning in the axial direction, as shown in Fig. 3. Each partition in the axial, ξ, direction will henceforth be referred to as a blade row, although it is not necessary for each axial partition to contain blades. Each blade row is granted limited rotational freedom about the ξ axis relative to the adjacent blade rows. With partitioning completed, a block referencing scheme must be installed. An obvious and simple choice is one which keys off the (global) curvilinear axes, also shown in Fig. 3.

With an arrangement selected, the second fundamental restriction, selectively-similar block characteristics, can be addressed. To begin, the local block axes are assumed to follow the global axes relative to orientation and direction, i.e. there exists global similarity (see Fig. (4)). The block ξ,η,ζ (i,j,k) index limits (NI,NJ,NK) are restricted such that for all blocks within a blade row NI, NJ, and NK remain constant. In addition, NJ must remain constant between blade rows. NI and NK are allowed to vary between blade rows, although the blade row circumferential cell count (BRCCC) must match.

For example consider Fig. (5), a three blade row configuration containing four partitions in the first blade row, i.e. four blade row blocks. Each of these inter-blade row blocks contain $NI_1 x NJ_1 x NK_1$ cells, where the subscript indicates the blade row. From the aforementioned restrictions on NJ, it is immediately established that $NJ_3=NJ_2=NJ_1$. The BRCCC for blade row 1 is the product of NK_1 and the number of blade row 1 blocks. Hence the BRCCC matching restriction implies that NK_2 and the number of blade row 2 blocks are limited to being factors of the BRCCC. In addition their product must equal the BRCCC. A similar condition exists for NK_3 and the number of blade row 3 blocks. Summarizing these restrictions in tabular form

Block Characteristic Restrictions

INDEX	SIMILARITY	LIMITATION
ξ or I	inter-blade row	none
η or J	global	none
ζ or K	inter-blade row	BRCCC matching

Memory Management

Those researchers with access to the genuine state-of-the art Cray 2 with its 256M words of internal memory might not concern themselves with in-core requirements. The vast majority of research though, is performed on equipment with lesser in-core capabilities. Also, it is understood that even Cray 2 users could find geometries that would push in-core requirements to the limits and beyond. Whence came an impetus

to manage the use of secondary memory, whether it is Cray's rapid access Solid State Storage Device or any sequential storage medium for that matter.

The use of computational blocks of differing cell count, and consequently differing memory requirement, necessitates the development of some special techniques to optimize memory utilization. A method referred to as ribbon vector dynamic memory management[4] is used here. As pointed out in [4], the principle of ribbon vector storage is frequently used to compress in-core storage to avoid wasting memory cells. The idea is to store the entire field on secondary memory with the exception of the block currently under execution. Ribbon vector dynamic memory management allows the adjustment of the in-core field length to accommodate each block, regardless of size (limited, of course, by the total in-core capability), without excess memory cells. A version similar to that proposed by Belk has been integrated into this computer algorithm.

Related to the topic of efficient memory use is the optimization of the processing of the memory. The time spent manipulating memory involves two general classes referred to as IO wait time and CPU execution time. An effort to minimize both involves maximizing the efficient use of the hardware capability of the available equipment, as the ultimate limiting factor is the equipment itself.

In the interest of minimizing IO wait time, the use of unblocked data transmission to and from secondary memory, namely SSD, made significant reductions. Essentially, all data contained within the ribbon vector is transferred between in-core and secondary memory via unblocked standard FORTRAN IO statements. Some padding is required to maintain

proper unblocked data length (multiples of 512 words). Regarding CPU execution time, efforts are made to utilize vector hardware to its fullest, including a procedure to vectorize a backward or forward substitution.

The (LU) scheme involves point simultaneous solutions and backward (or forward) substitutions. The substitution portion poses quite a problem when attempting to utilize vector hardware. The difficulty is referred to as a vector dependency[18] or recursion problem. A method has been developed in order to circumvent this problem in a multi-dimensional space.[2] The procedure in a three-dimensional space is to "simultaneously" process those cells lying on a special diagonal plane in computational space. Cells whose indices satisfy the equation of this diagonal plane,

$$i + j + k = \kappa \tag{28}$$

where κ is a constant designating the plane level, are computationally independent. Hence with the aid of indirect addressing, diagonal plane processing is used to facilitate the vectorization of a backward or forward substitution.

Interface Control

Advancing a flowfield simulation in time requires solving the system model, Eqs. (27), at each computational cell within the global domain for the new time level dependent variable vector Q. The solution path advocated here requires two sweeps through the global domain, one forward and one backward. The global path indicating the order in which the blocks are swept to insure analogous single block solution advance-

ment is shown in Fig. (6). All blocks are forward swept then all are backward swept. The forward sweep within each block consists of operating on (applying Eq. (27a) to) each cell on a diagonal plane, beginning with the lowest level interior plane and advancing (forward) to higher level planes. The forward sweep is complete when the highest level interior plane has been updated. The backward sweep is similar although the planes are traversed in decreasing (backward) order, and Eqs. (27b,c) is the applied operator.

Basic to the concept of multi-block is the requirement of continuous grid lines between blocks. This provides a relatively simple means of communication between blocks with stationary interfaces in computational space. Values from within the domain of one block can be extracted and then injected as phantom values in an adjacent block. As previously mentioned, each blade row can rotate relative to the adjacent blade rows. This rotation is presently governed by requiring continuous grid lines across the shearing block interface at all time levels. This is accomplished by maintaining equidistant circumferential cell spacing on the blade row boundaries. A shearing block interface in physical space yields a dynamic interface in computational space, i.e., the blocks continue to change communication partners. This complicates the simple communication procedure prescribed by a multi-block technique, but with appropriate index bookkeeping the extraction-injection procedure is still valid.

When dealing with multi-block techniques, an issue of the accuracy necessary for the communication between blocks arises. Ultimately, the best communication possible is that analogous to a singly gridded field. The approach taken here with regard to internal block boundary communi-

cation is to mimic the cell communication within the blocks, thereby incurring no boundary induced error. Hence internal block boundaries are maintained with up to third-order spatial accuracy. For the case of re-entrant block boundary communication, periodicity and previous ΔQ or zero ΔQ are satisfactory. Block boundaries involving the farfield or impermeable surfaces are maintained as mentioned in the third section dealing with boundary conditions.

RESULTS

Even Blade Count (GE UDF 8-8)

The computer algorithm resulting from the procedures outlined in the previous sections has application in the simulation of even or uneven blade count, single rotating, counter-rotating, or rotor-stator, axi-symmetric or non-symmetric, multi-stage, at angle of attack geometries. To support this claim several configurations are presently being analyzed, the results of which shall appear in subsequent publications. A configuration previously studied using a similar algorithm[19] will be examined here. Since in [19] the interface fluxes were given by flux vector split theory and the block interfaces were only maintained to first order, it will be of interest to compare the two algorithms.

The configuration is that of a counter-rotating unducted fan (UDF) emersed in an oncoming M=.72 axial flow, see Fig. 7. The configuration has two fan rows with eight blades per row. The highly swept, tapered, twisted, thin blades are designed to reduce the axial Mach number through the blading to alleviate compressibility losses. The forward row rotates clockwise and the aft row counter-clockwise. Both blades rotate with an advance ratio of 2.8.

The solution appearing herein was obtained using only two blocks (benefitting from solution symmetry). Although only two blocks were used, axial interblock communication was implemented with a full buffer disk (temporary storage area for injected or extracted data). The procedure involves extracting data, imaging the data to form a full 360 degree communication buffer disk, then allowing the appropriate data to

be injected based on the positional relationship between the blocks and the buffer disk. The mesh was H-type in all directions and each block contained 56x21x10 (i,j,k) cells.

The solution (scalar quantities) of the UDF are periodic with respect to blade passages, i.e. the solution repeats each time the blades are re-aligned. Although this is the case, the velocity vector components in a Cartesian reference frame are only periodic with respect to blade position. The periodic behavior of scalar quantities gives a good measure of the "convergence" of such geometries. The convergence history of the change in density, chosen to be defined as $Log_{10}((\Delta\rho)^{n}_{rms}/(\Delta\rho)^{1}_{rms})$, is shown in Fig. 8. The transients associated with the arbitrary initial condition of uniform flow have diminished considerably after 3 complete revolutions of each blade row (480 timesteps), yet the results presented herein still retain some of the more stubborn transients.

There exists evidence of the inherent unsteady behavior of the flow, though it is not prominent. Figures 9 and 10 support the comments in [19] regarding the lack of variation in the blade surface relative Mach number for both forward and aft blade rows. Though little variation is shown overall, there is more variation in the aft blade row, as expected. Note, the previous comments regarding equidistant circumferential cell spacing result in ten timesteps per blade passage. Hence each blade passage can be marked accordingly, with Step 1 representing an axial alignment of the blades (or as close to an alignment as one can get with highly swept, tapered, and twisted blades). Thus the suction

surface and pressure surface variations in Figs. 9 and 10 represent the respective maxima, and the steps indicated are those from which these maxima were computed.

Typical performance parameters are presented for comparison in Figs. 11. At first glance, one can see in Fig. 11a that the power coefficient (Power In) for the FDS scheme falls closer to experiment. The question arises whether this is the expected behavior. Consider the following plausible explanation. The FDS scheme is touted for less dissipation relative to the FVS scheme. Hence a reduction in numerical dissipation would tend to make for a 'more inviscid' flow, resulting in stronger, crisper shocks. This in-turn could yield pressure distributions which produce lower torque coefficient values, hence lowering the power coefficient from that of the FVS scheme. Figures 12 and 13, indicating the blade surface relative Mach numbers for the two numerical algorithms, do not necessarily support this claim. It is believed the evidence needed to support this claim lies between the root and the midspan plots, although since these did not appear in [19] a comparison cannot be made here.

The reduced efficiency relative to the FVS scheme plotted in Fig. 11b could also come from altered pressure distributions due to stronger shockwaves. In addition, the ratio of the blade torques given in Fig. 11c indicate an increase relative to the FVS. It is difficult to access the cause of this increase, due in part to the influence on pertinent parameters such as flow velocity and flow angle relative to aft blade produced by the upstream blade row. Without benefit of a detailed flowfield snapshot between blade rows for both the FVS and the FDS schemes, no plausible arguments can be proposed. The conclusions drawn here re-

sult from a limited source of comparable data, further investigation is warranted.

CONCLUSIONS

A computer program has been developed whose framework centers around the analysis of general rotating machinery. This general applicability has not been at the expense of algorithm simplicity. Painstaking thought has yielded an algorithm capable of simulating even or uneven blade count, single rotating, counter-rotating or rotor-stator, axi-symmetric or non-symmetric, multi-stage, at angle-of-attack geometries. The code presently stands at under 2000 lines, most of which are the numerics. A version of the numerical routine is resident on the NASA Marshall EADS Computer System.

Future emphasis will focus on the development of simple axial interface control allowing non-uniform circumferential cell spacing. This could relax (or remove) the inherent dependency of the time-step on the dynamic interfaces. Parametric studies of the effect of time-step on solution accuracy, evaluating the compatibility of characteristic variable boundary conditions with an approximate Riemann solver, and development of better grids for the simulated flowfields are in order.

BIBLIOGRAPHY

1. Janus, J.M., and Whitfield, D.L., "Advanced 3-D Viscous SSME Turbine Rotor Stator CFD Algorithms", Final Report, NASA Marshall Space Flight Center, September 1986.

2. Janus, Jonathan Mark, untitled, Ph.D. Dissertation, Mississippi State University, to be published August 1988.

3. Janus, Jonathan Mark, "The Development of a Three Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications", M.S. Thesis, Mississippi State University, August 1984.

4. Belk, D.M., "Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids", Ph.D. Dissertation, Mississippi State University, August 1986.

5. Lax, P.D., "Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shockwaves", SIAM Regional Conference Series in Applied Mathematics, No. 11, 1973.

6. Godunov, S.K., "Finite Difference Method for Numerical Computation of Discontinuous Solutions of the Equations of Fluid Dynamics", Mat. Sbornik, Vol. 47, No. 3, pp. 271-306, 1959. Translated as JPRS 7225 by U.S. Dept. of Commerce, November 1960.

7. Glimm, J., "Solutions in the Large for Nonlinear Hyperbolic Systems of Equations", Communications on Pure and Applied Mathematics, Vol. 18, pp. 697-715, 1965.

8. Chorin, A.J., "Random Choice Solution of Hyperbolic Systems", Journal of Computational Physics, Vol. 22, pp. 517-533, 1976.

9. Harten, A., and Lax, P.D., "A Random Choice Finite Difference Scheme for Hyperbolic Conservation Laws", SIAM Journal of Numerical Analysis, Vol. 18, pp. 289-315, 1981.

10. Sod, G.A., "A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws", Journal of Computational Physics, Vol. 27, pp. 1-31, 1978.

11. Roe, P.L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes", Journal of Computational Physics, Vol. 43, pp. 357-372, 1981.

12. Osher, S., and Chakravarthy, S., "Very High Order Accurate TVD Schemes", ICASE Report No. 84-44, Sept. 1984.

13. Chakravarthy, S.R., "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws", AIAA Paper No. 85-0363, Jan. 1985.

14. Chakravarthy, S.R., Szema, K.Y., Goldberg, U.C., Gorski, J.J., and Osher, S., "Application of a New Class of High Accuracy TVD Schemes to the Navier-Stokes Equations", AIAA Paper No. 85-0165, Jan. 1985.

15. Roe, P.L., "Some Contributions to the Modelling of Discontinuous Flows", Large Scale Computations in Fluid Mechanics, edited by B. Engquist, S. Osher, and R. Somerville, Lectures in Applied Mathematics, Vol. 22, part 2, pp. 163-193, 1985.

16. Beam, R.M., and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", AIAA Journal, Vol. 16, No. 4, pp. 393-402, April 1978.

17. Whitfield, D.L., Janus, J.M., and Simpson, L.B., "Implicit Finite Volume High Resolution Wave-Split Scheme for Solving the Unsteady Three-Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids", Mississippi State Engineering and Industrial Research Station Report No. MSSU-EIRS-ASE-88-2, February 1988.

18. Cray X-MP and Cray-1 Computer Systems (Fortran (CFT) Reference Manual SR-0009), Cray Research, Inc., Mendota Heights, Minnesota, 1984.

19. Whitfield, D.L., Swafford, T.W., Janus, J.M., Mulac, R.A., and Belk, D.M., "Three-Dimensional Unsteady Euler Solutions for Propfans and Counter-Rotating Propfans in Transonic Flow", AIAA Paper No. 87-1197, June 1987.

20. Smith, L.H., Jr., "Aerodynamic Performance Tests of Unducted Fan Models", Advanced Turboprop Workshop Presentation, NASA Lewis Research Center, Cleveland Ohio, Nov. 5-6, 1986.

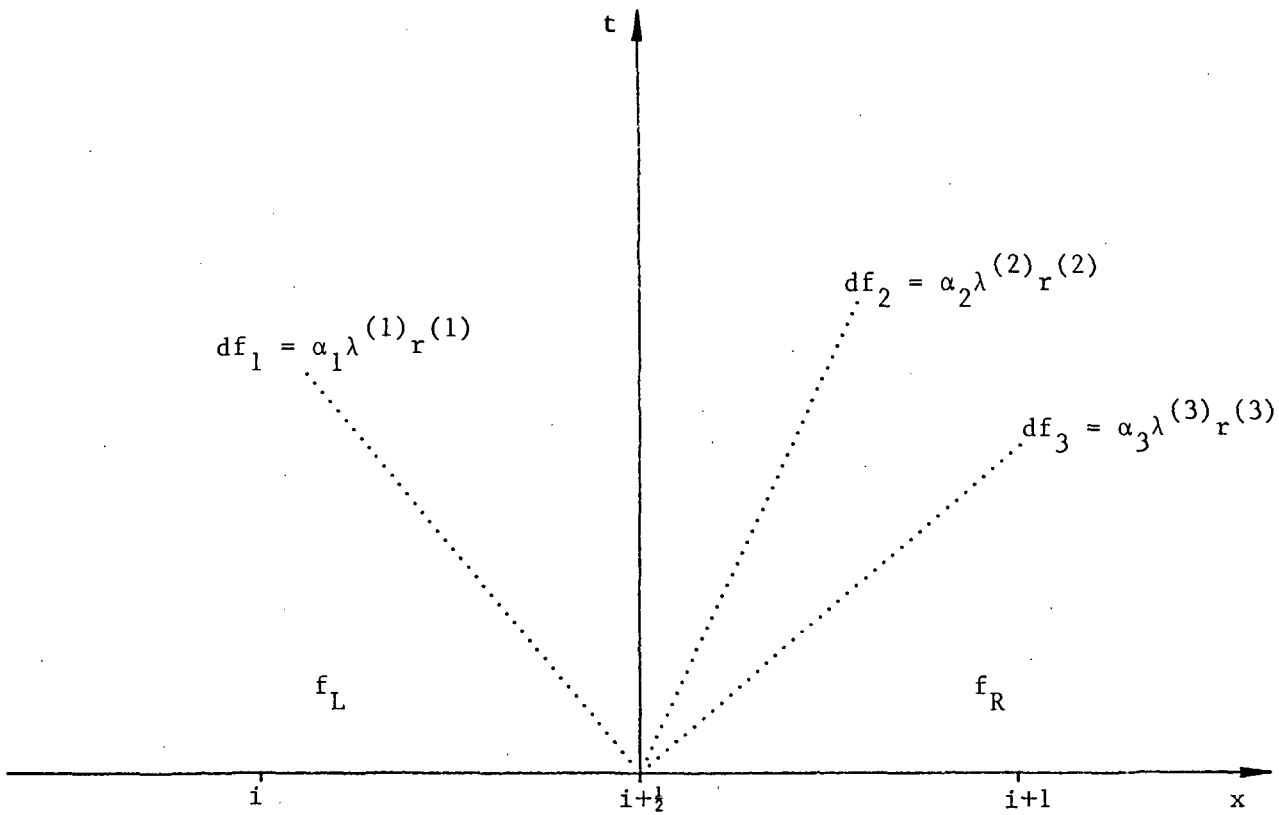


Figure 1. Interface Flux

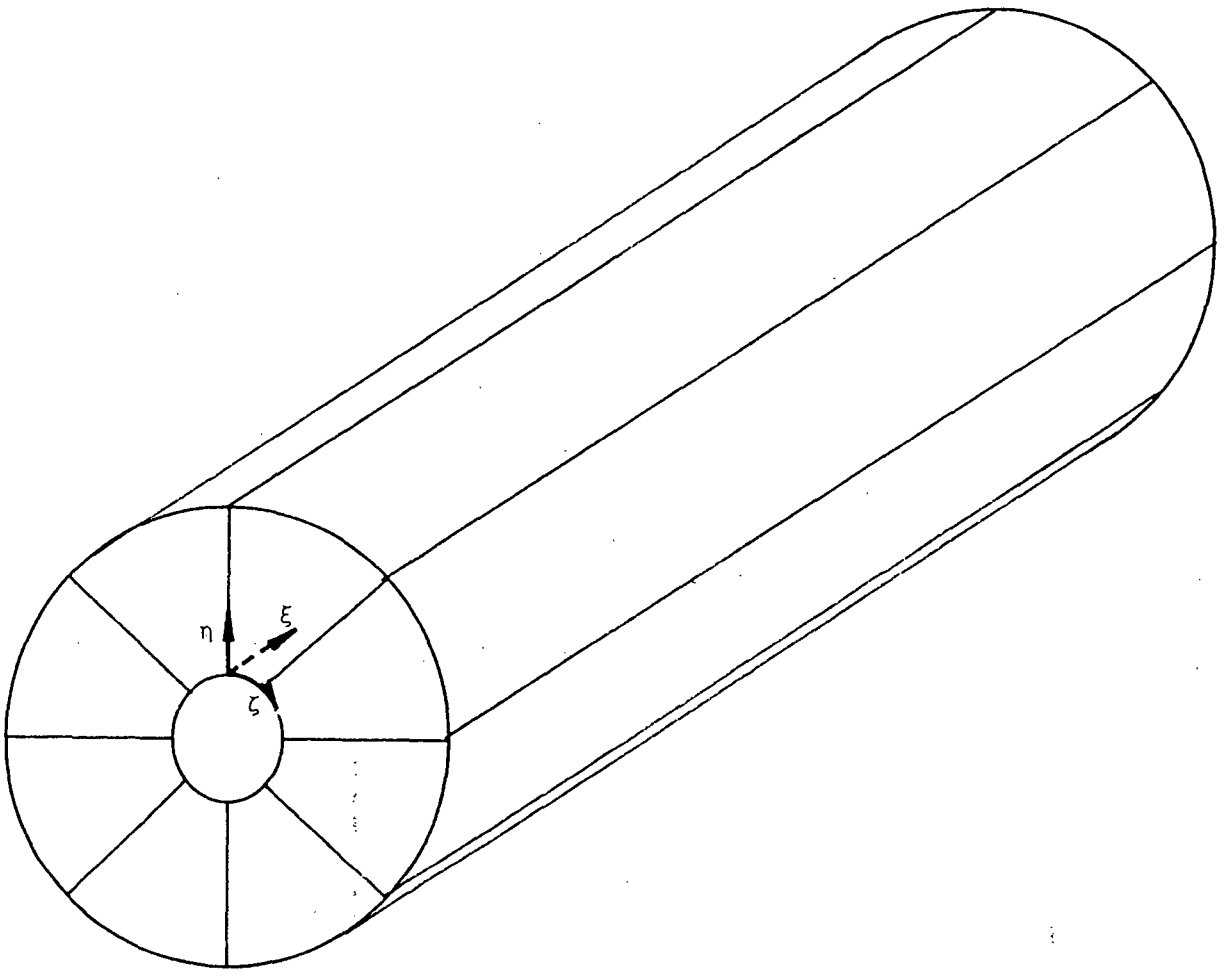


Figure 2. Macro-Block Circumferential Partitioning

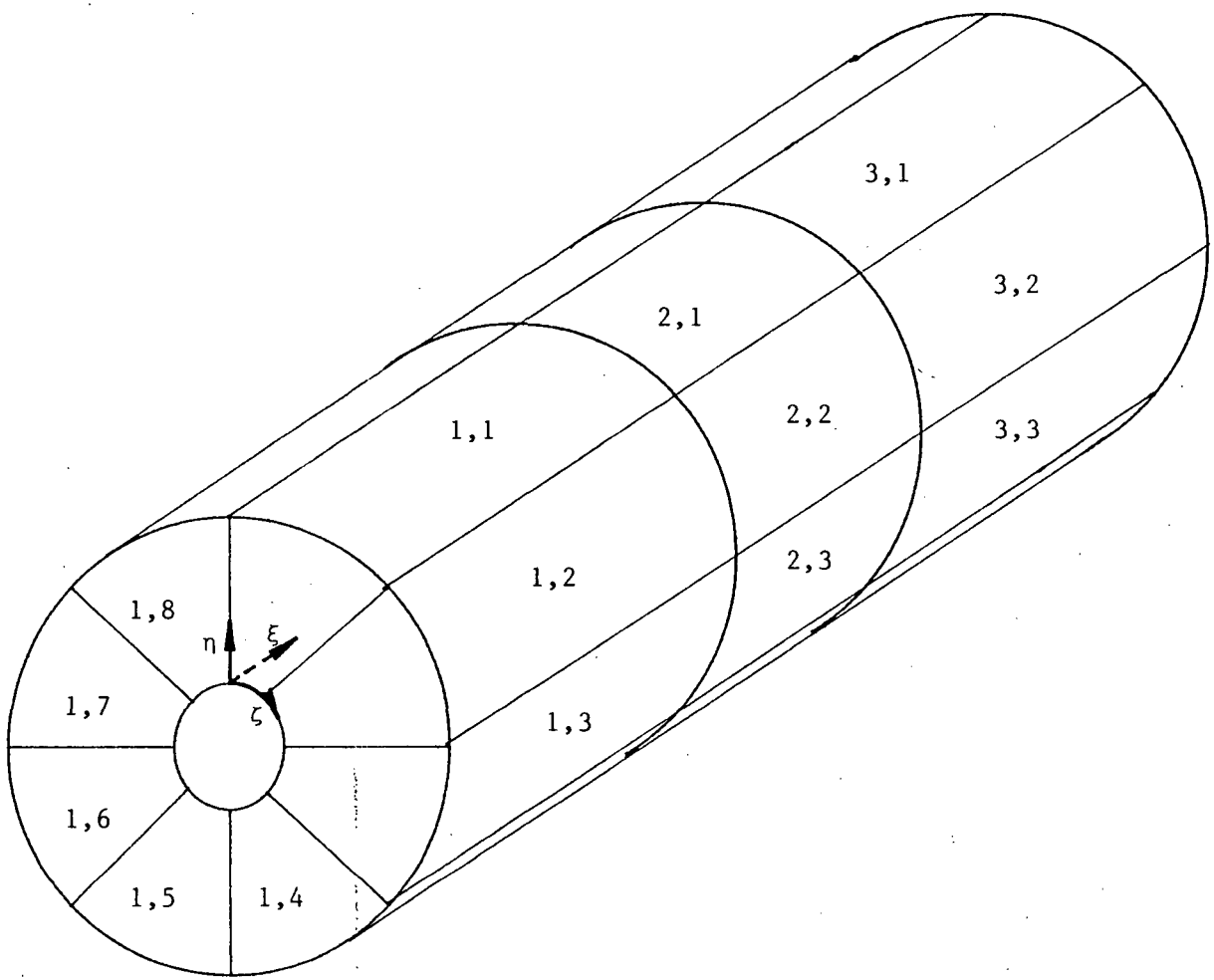


Figure 3. Macro-Block Axial Partitioning with Indexing

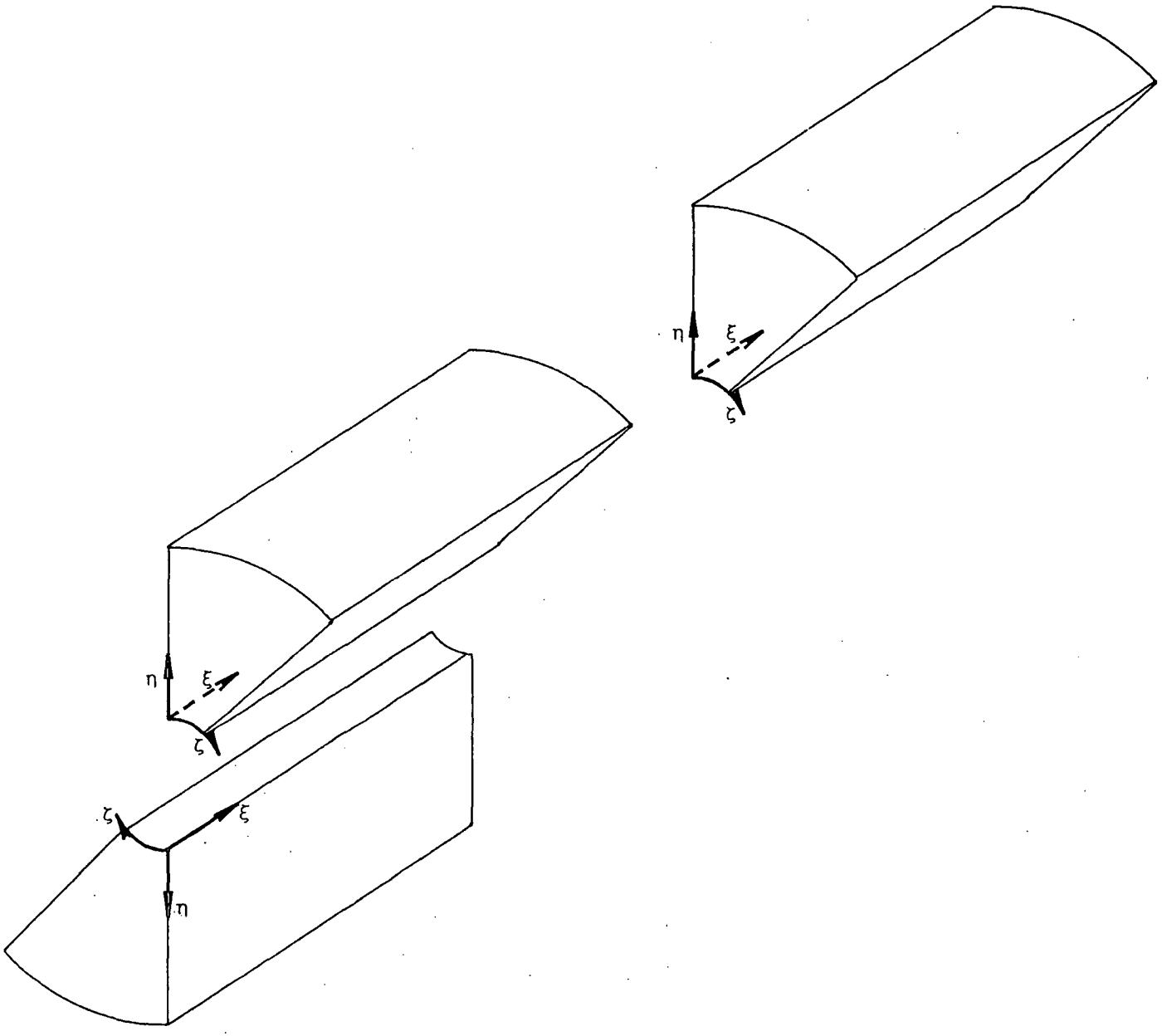
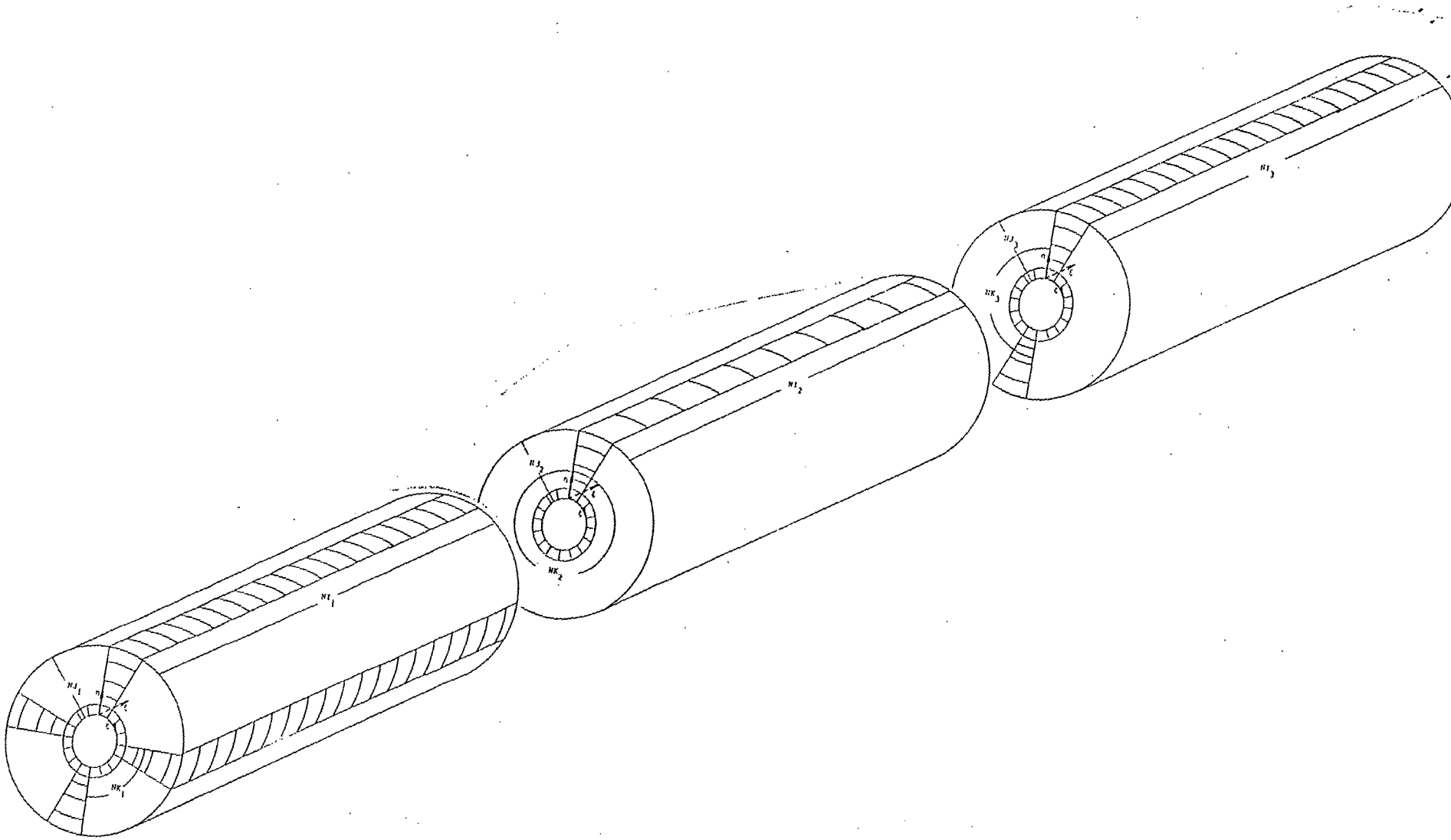
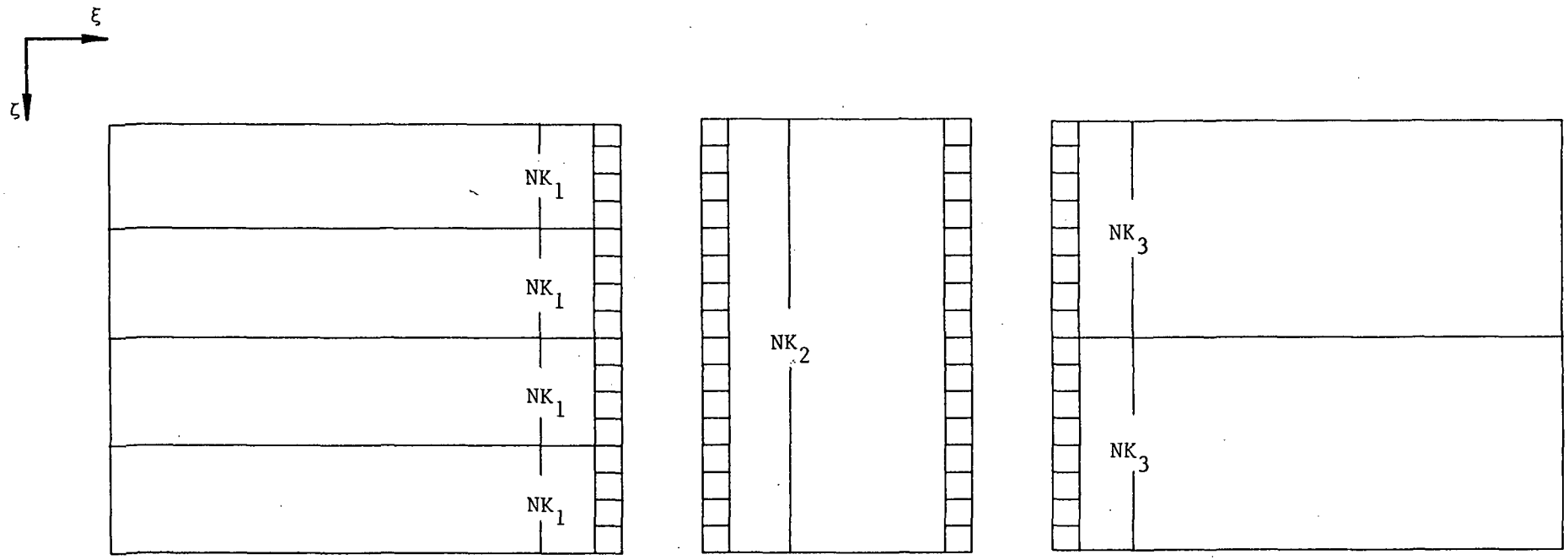


Figure 4. Block Coordinate System Orientation and Direction



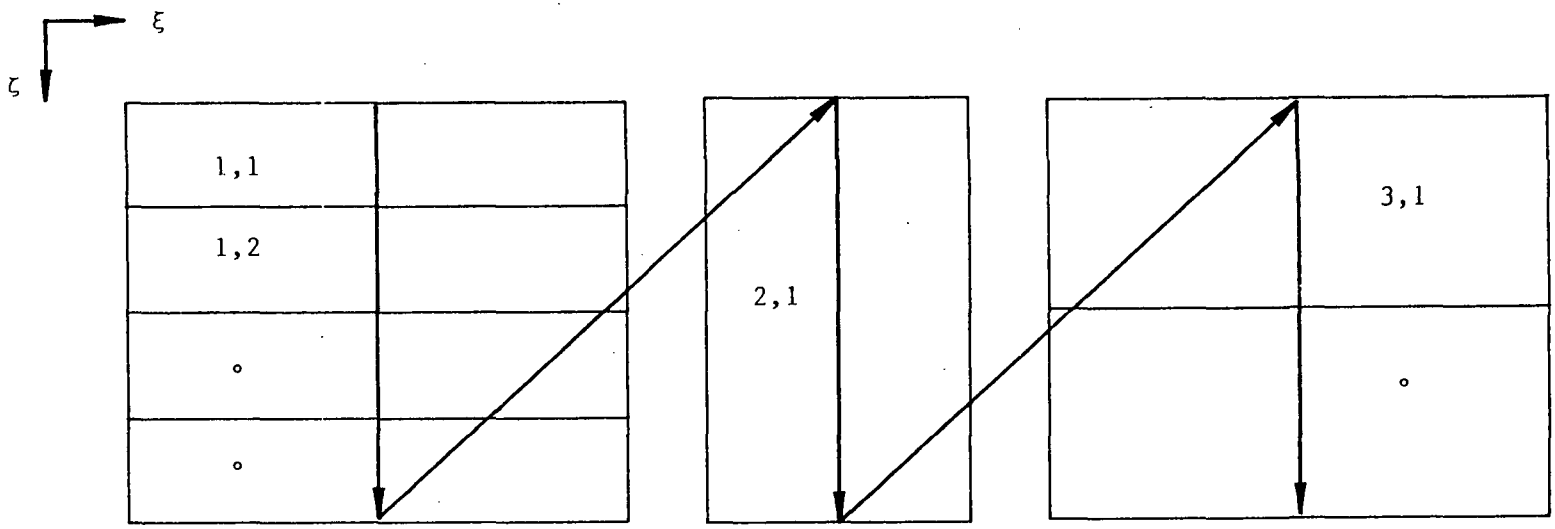
a. Physical Space

Figure 5. Typical Configuration

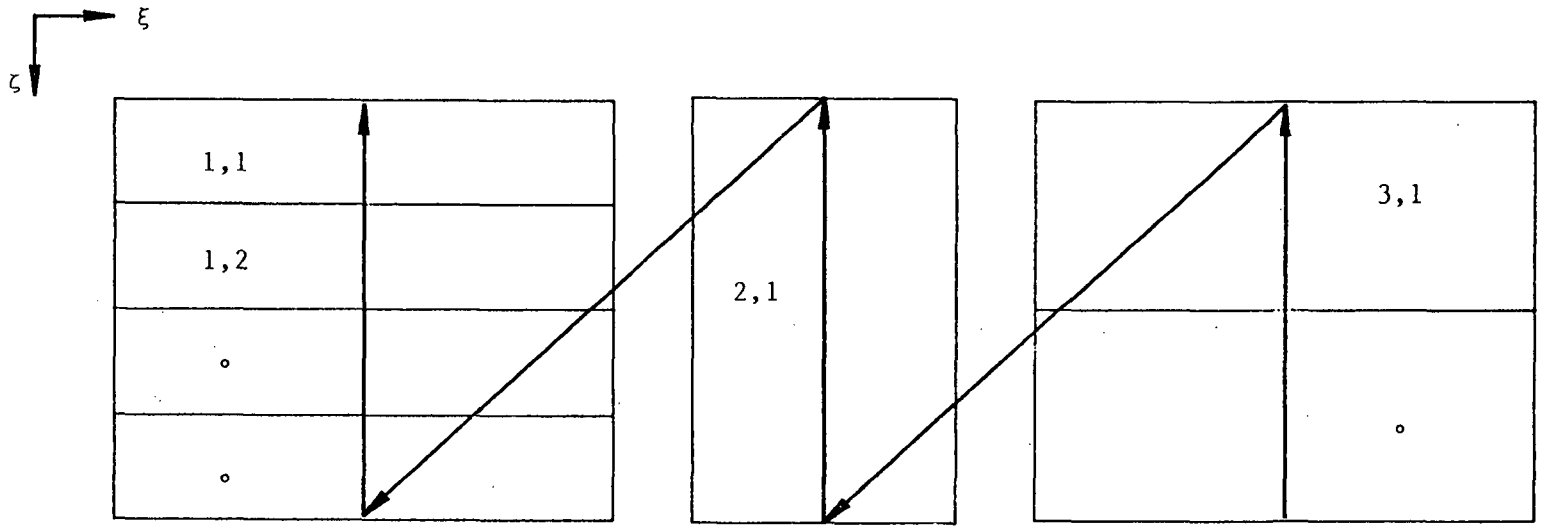


b. Computational Space

Figure 5. Typical Configuration



a. Forward Sweep



b. Backward Sweep

Figure 6. Global Block Solution Path

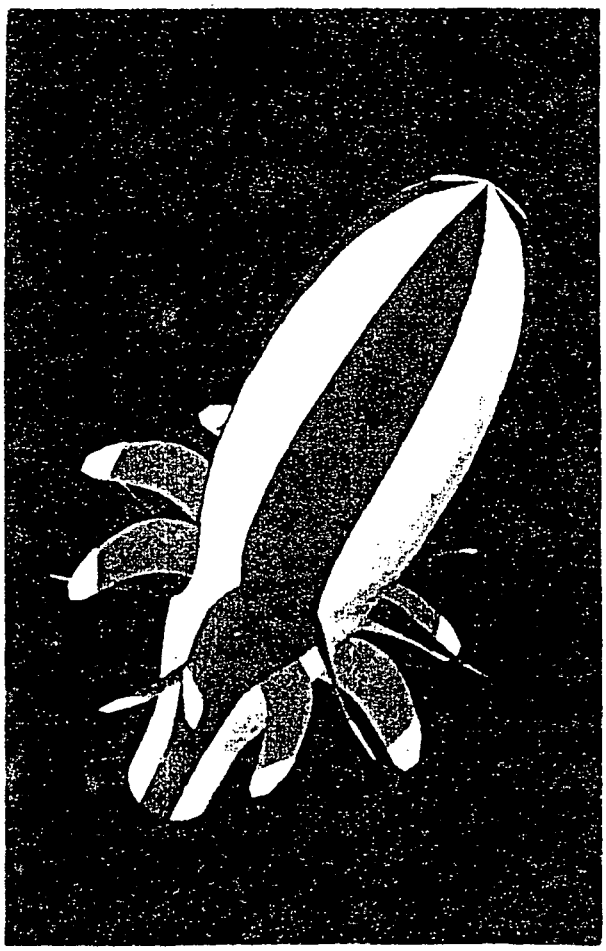


Figure 7. UDF Configuration

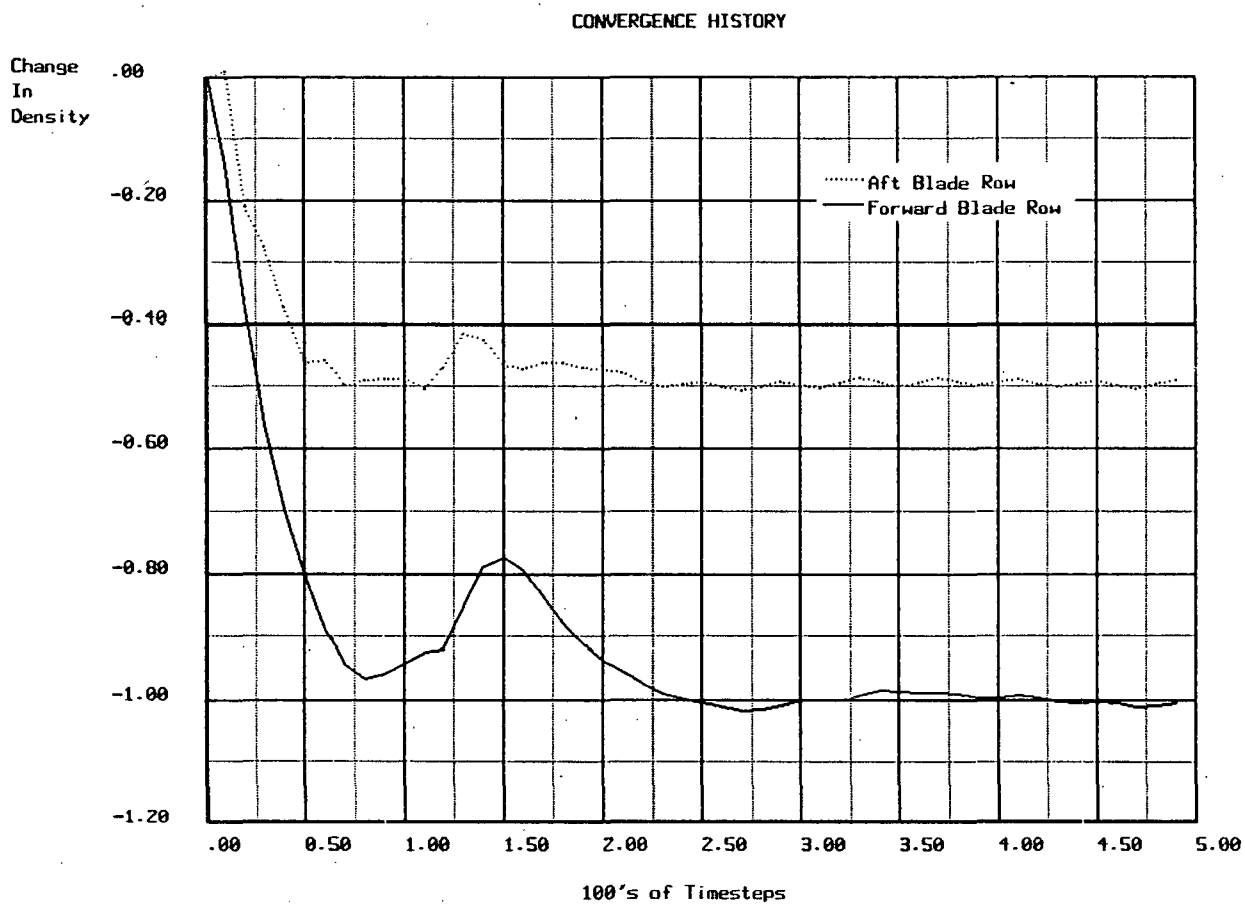
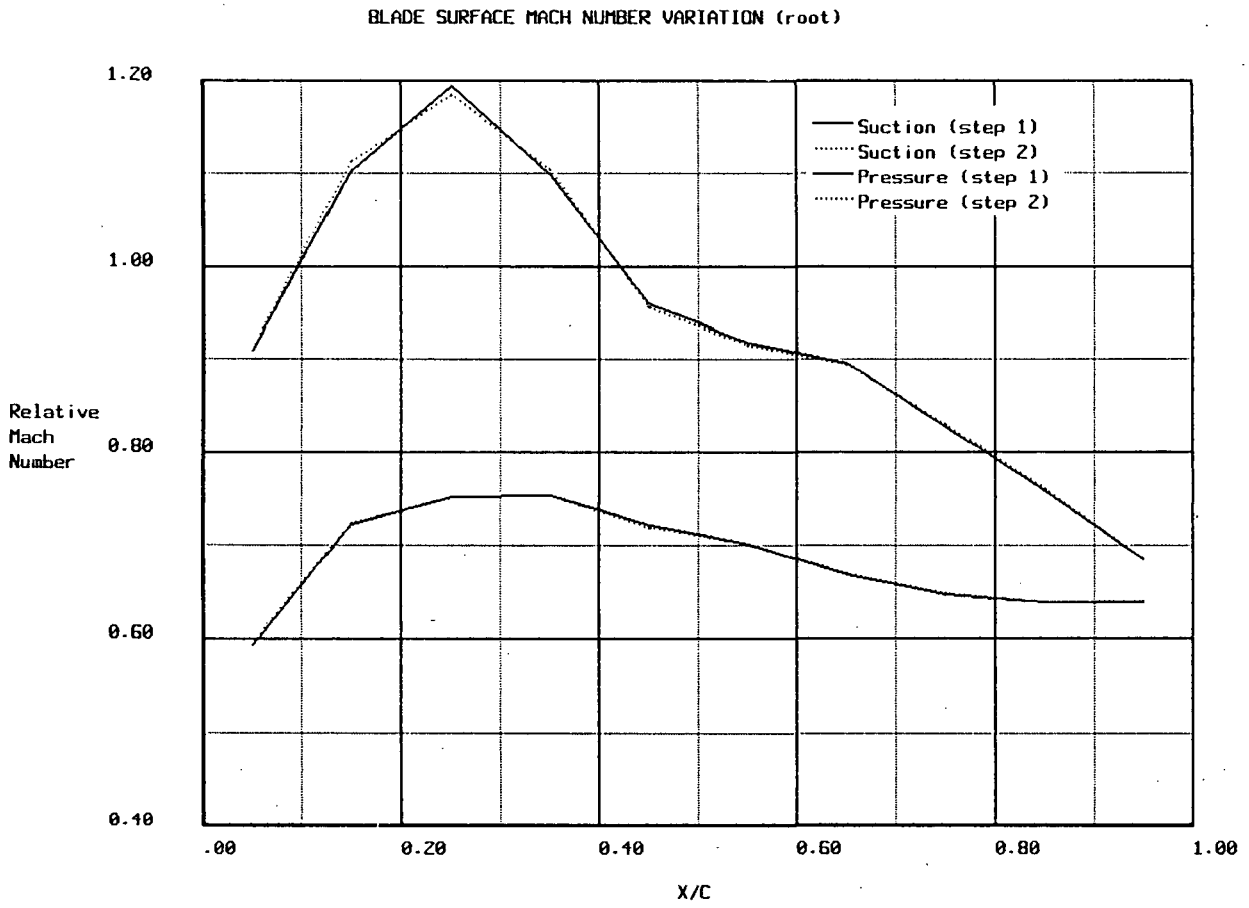
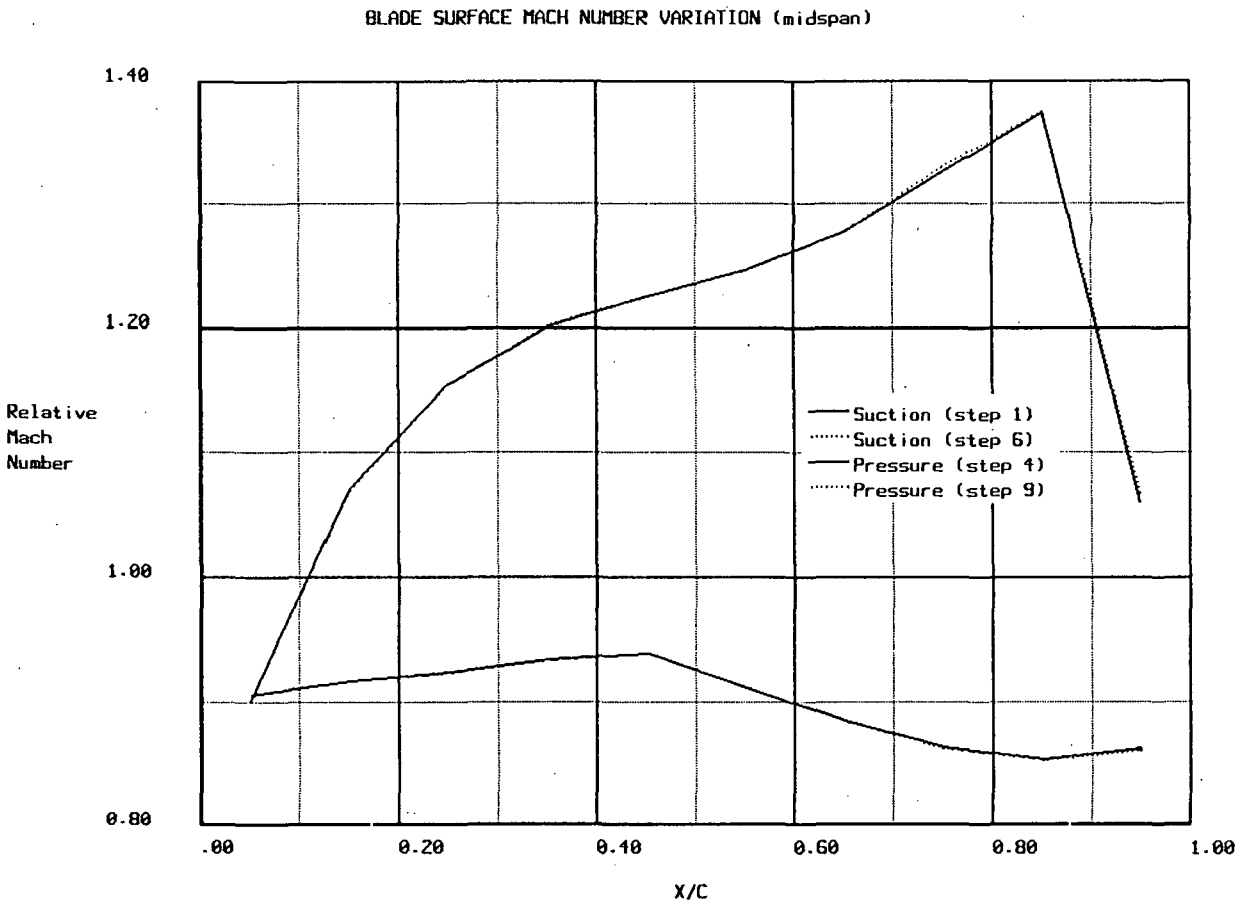


Figure 8. UDF Convergence History



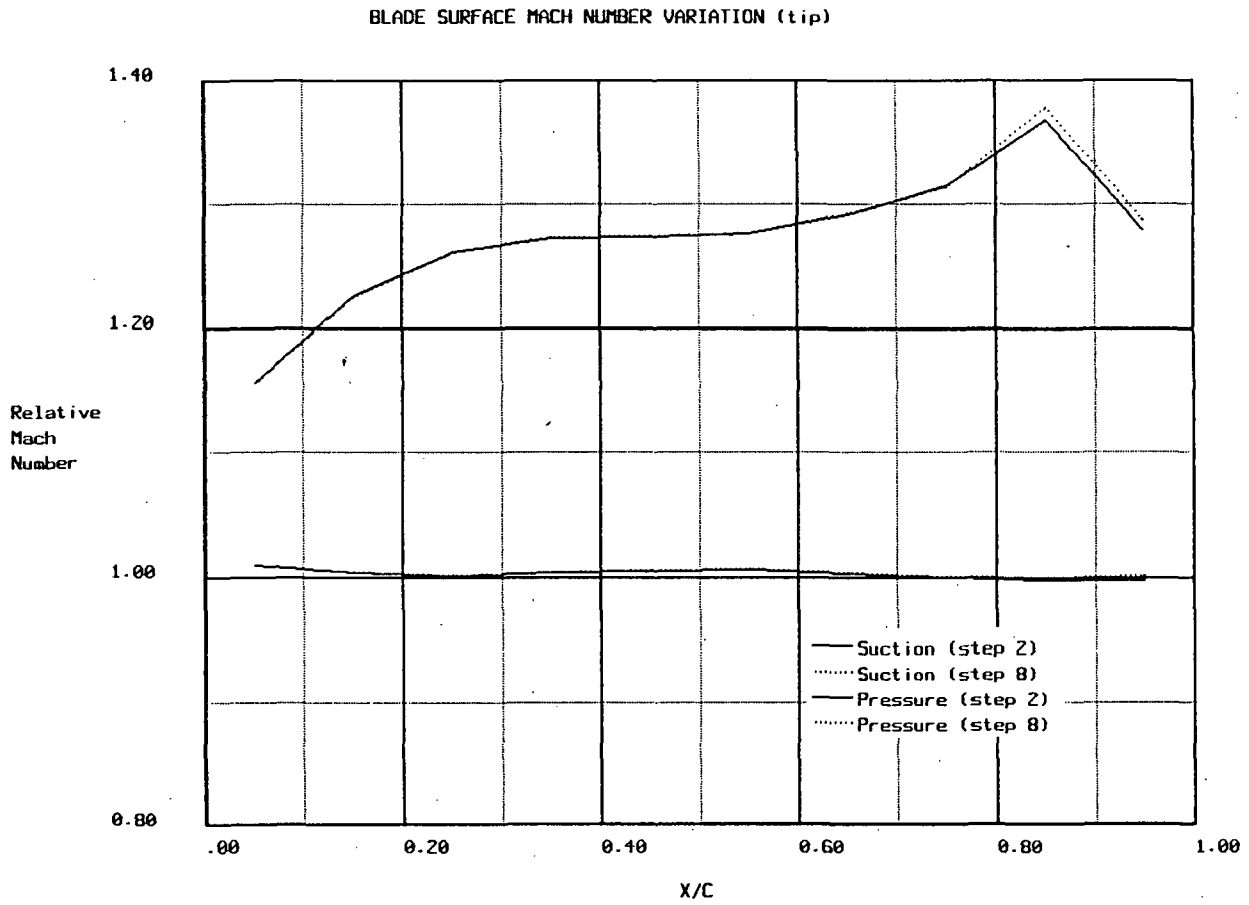
a. root

Figure 9. Forward Blade Row



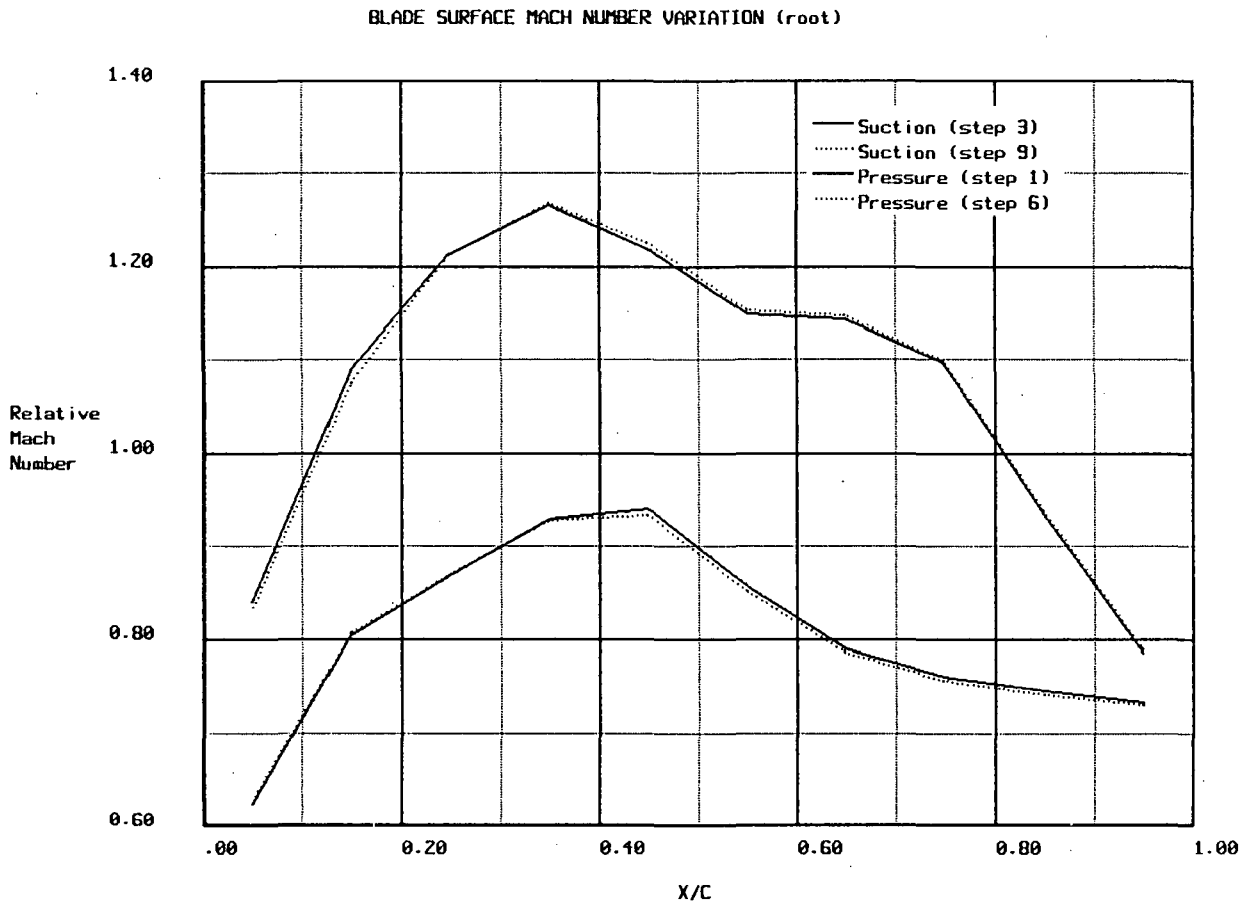
b. midspan

Figure 9. Forward Blade Row



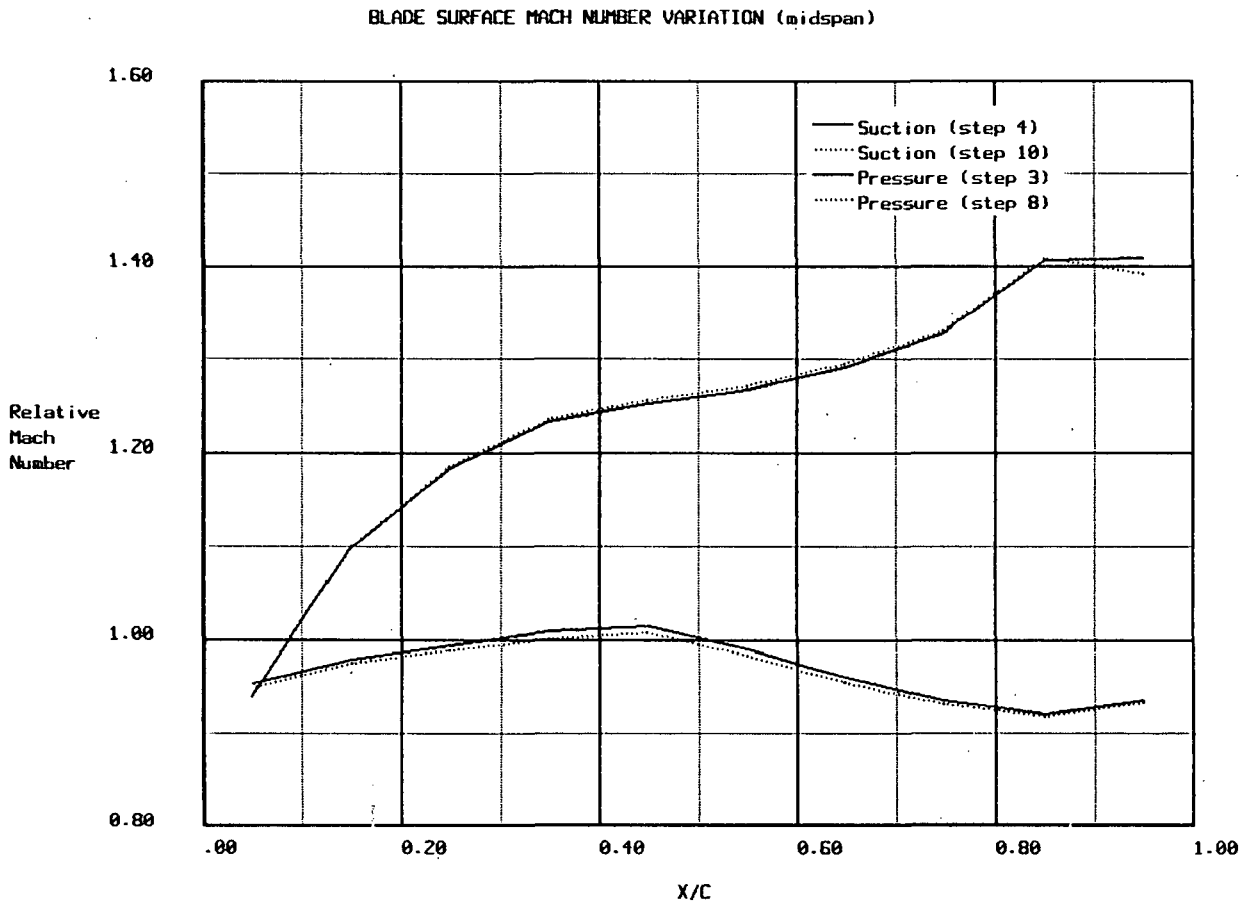
c. tip

Figure 9. Forward Blade Row



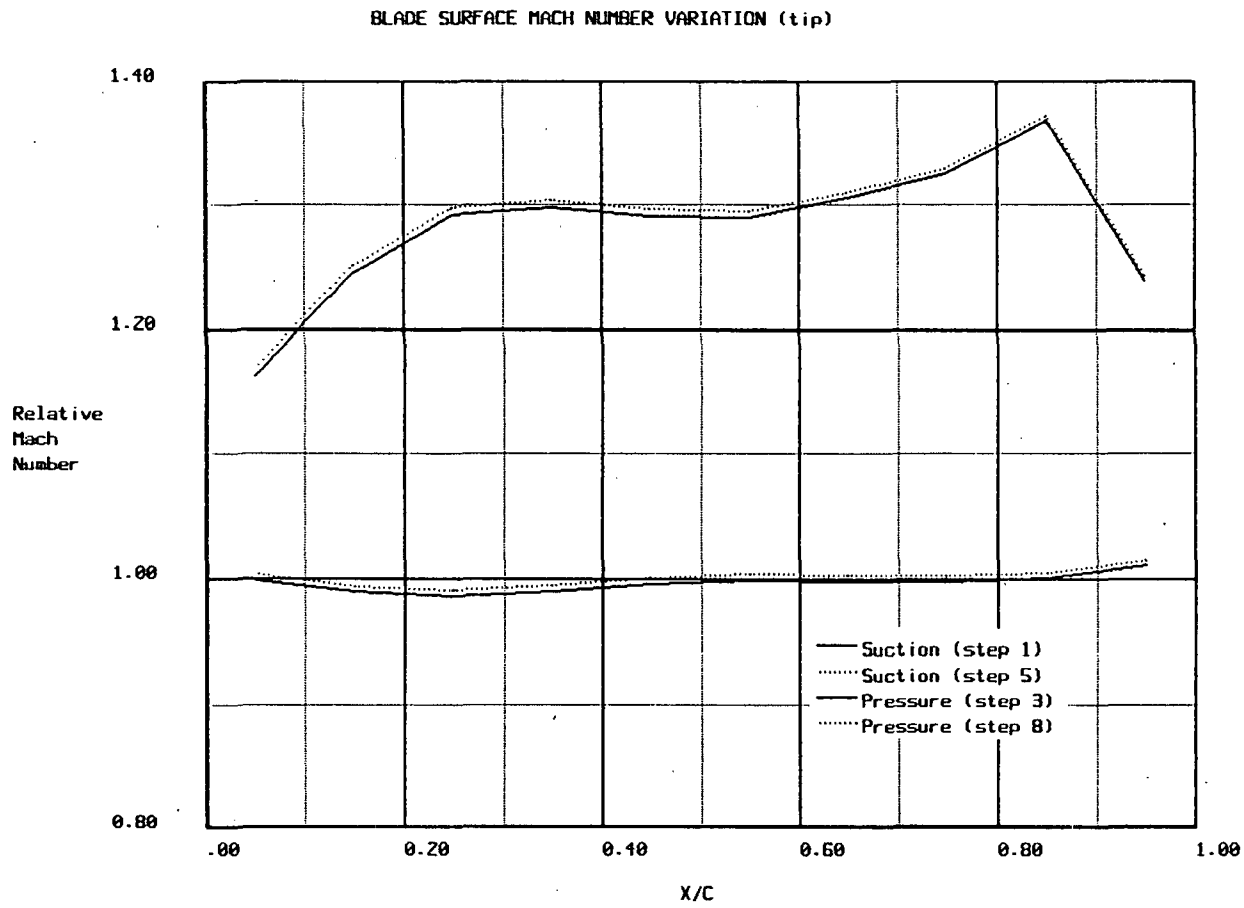
a. root

Figure 10. Aft Blade Row



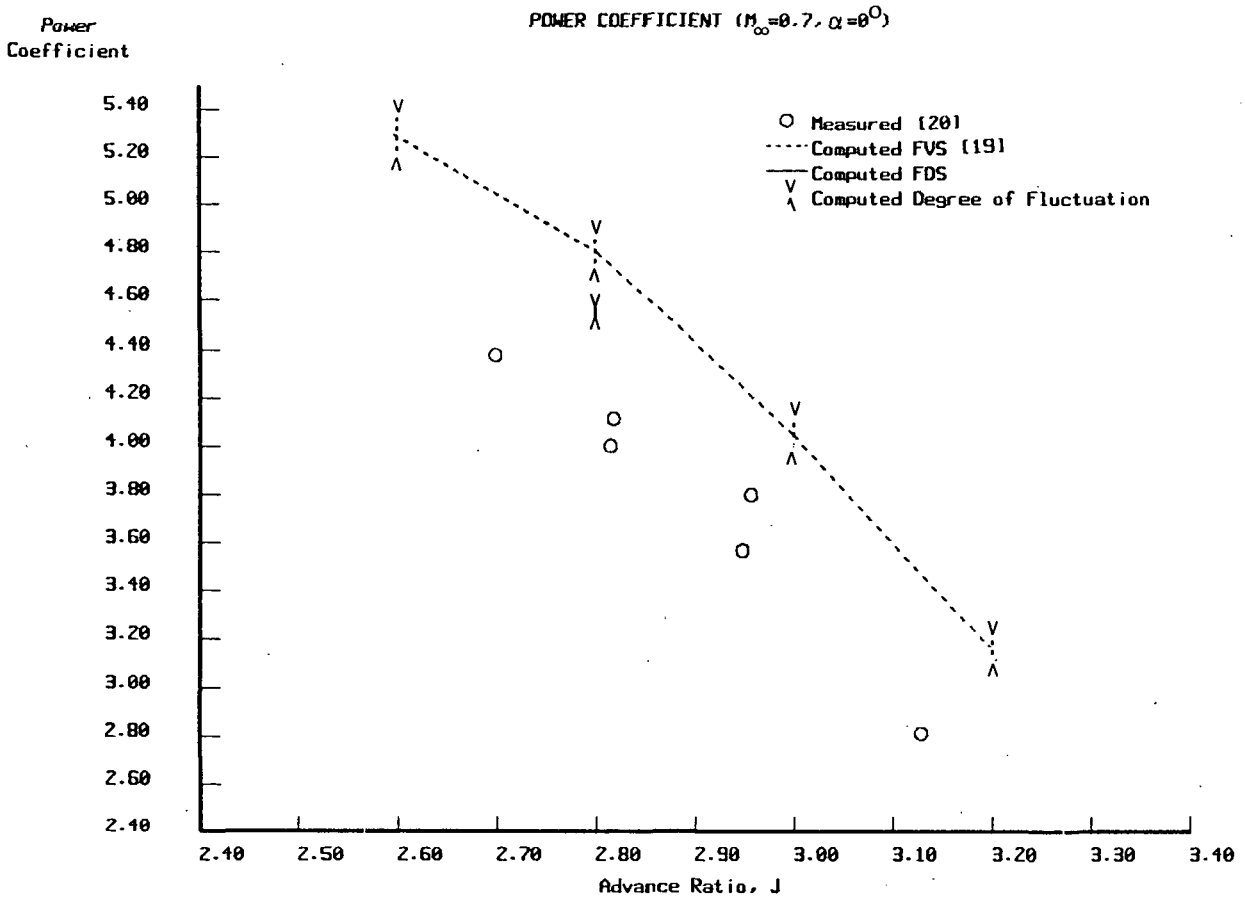
b. midspan

Figure 10. Aft Blade Row



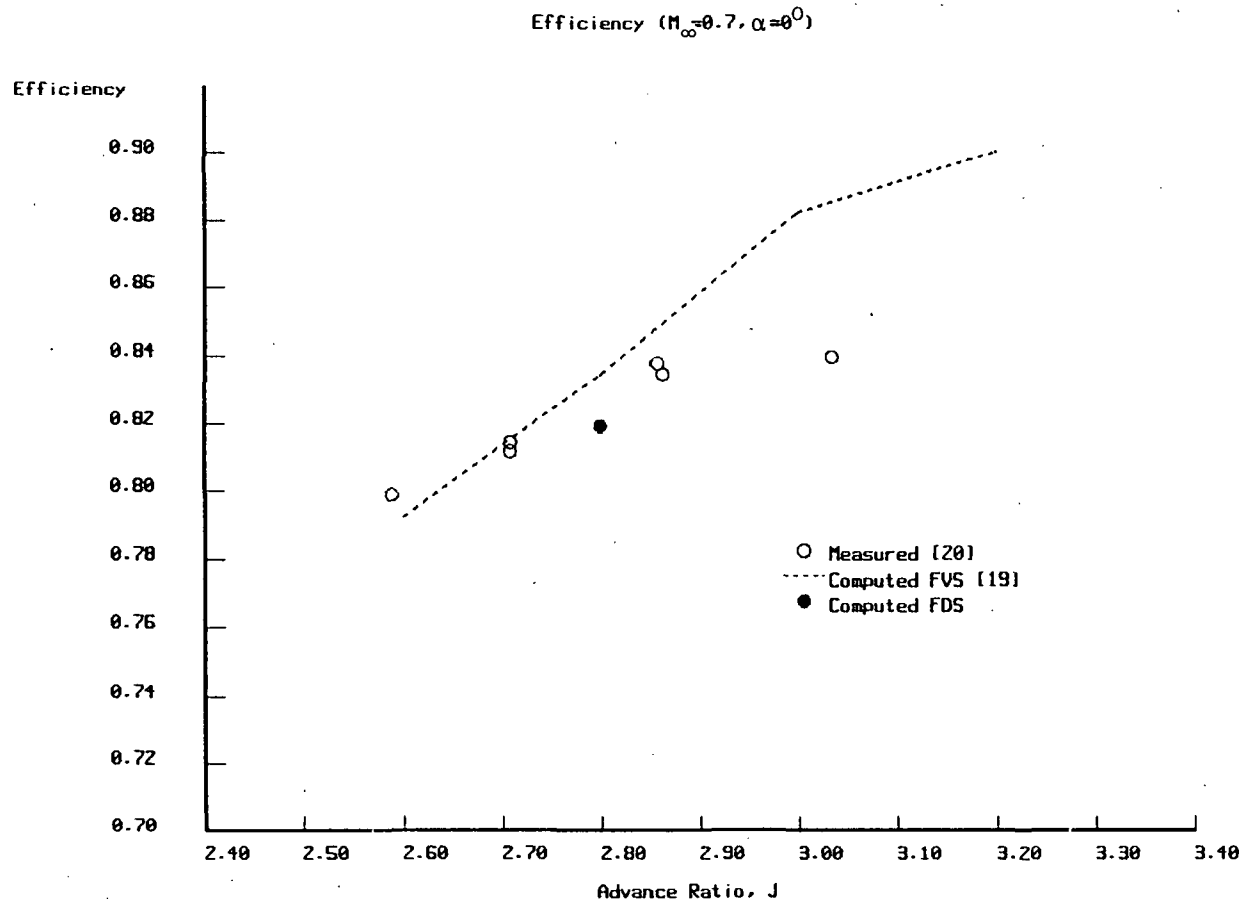
c. tip

Figure 10. Aft Blade Row



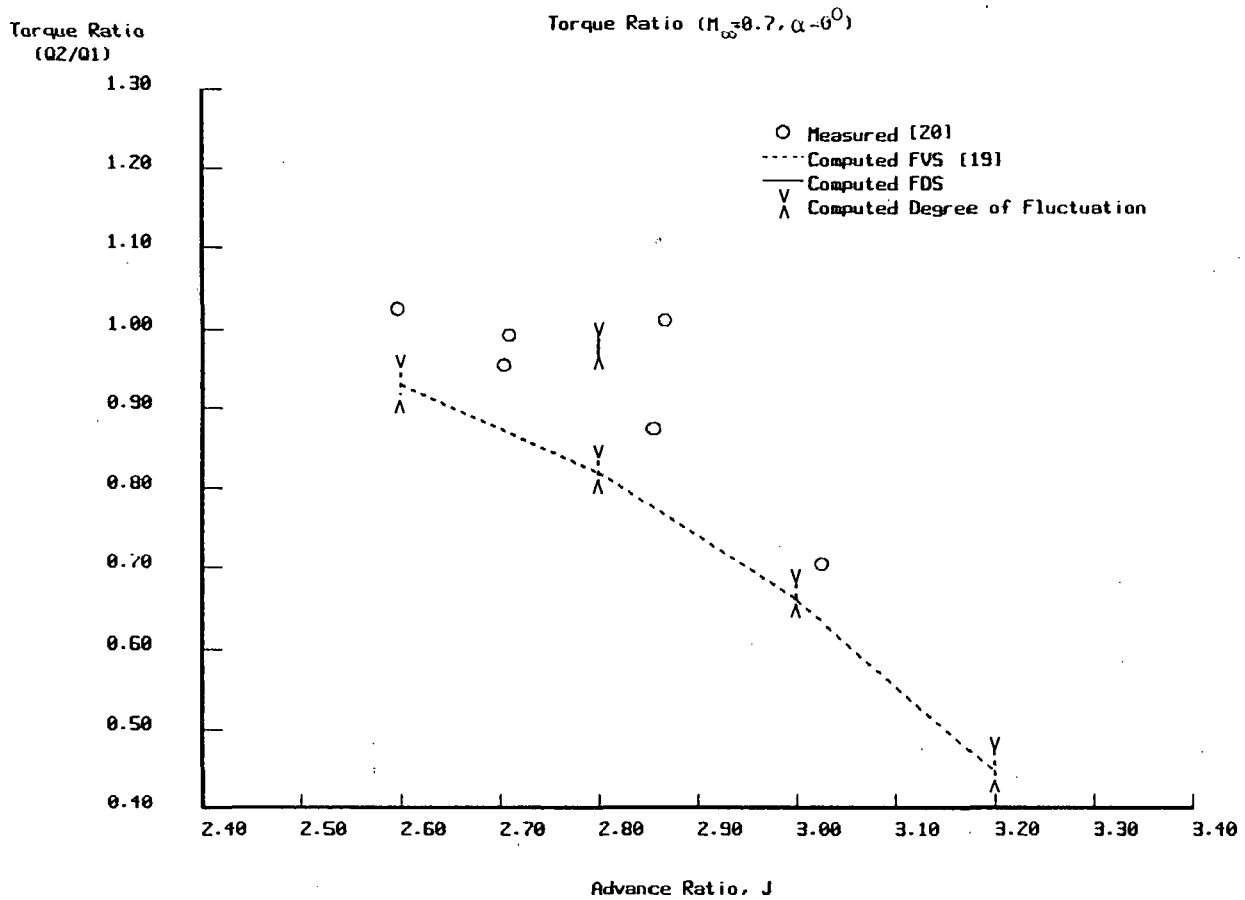
a. Power Coefficient

Figure 11. Performance Parameters



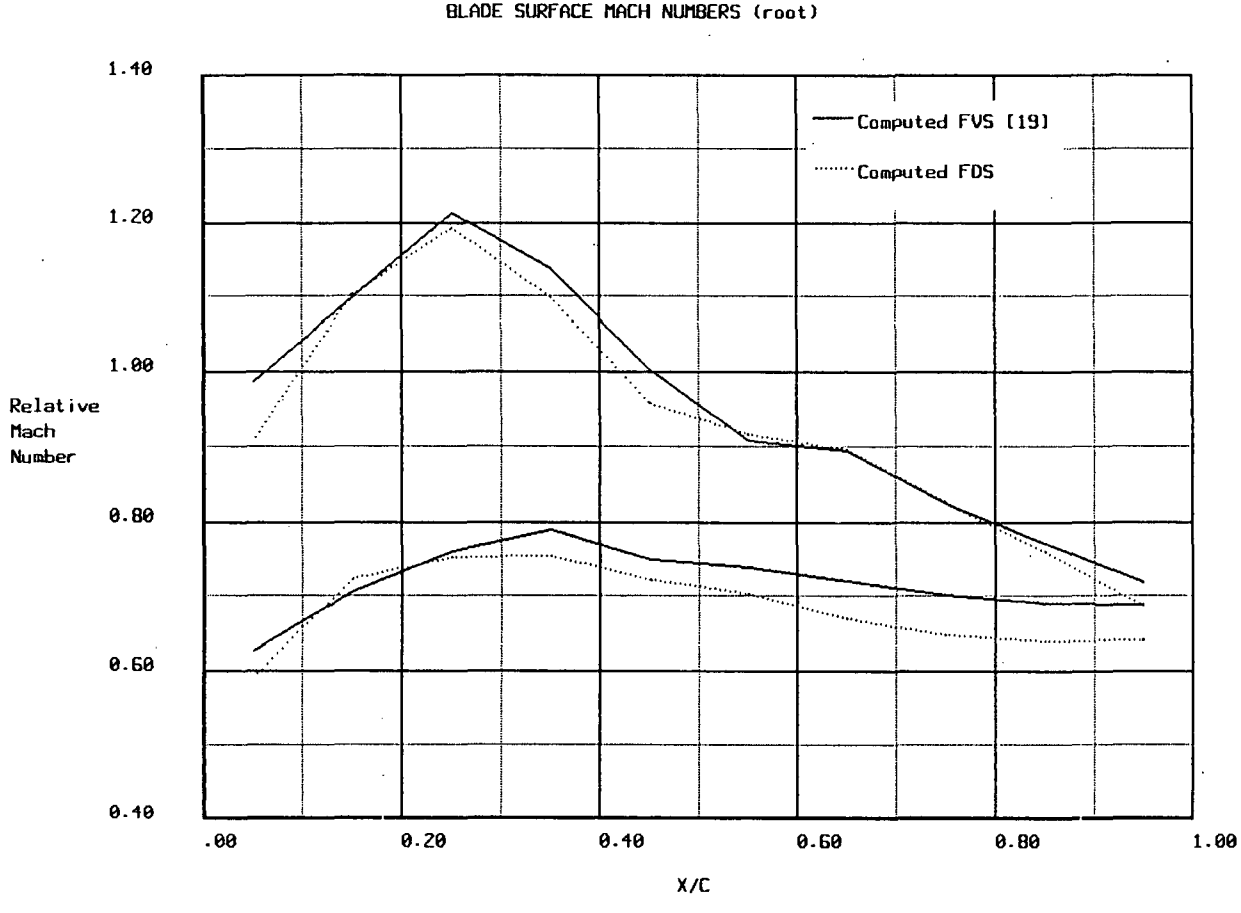
b. Efficiency

Figure 11. Performance Parameters



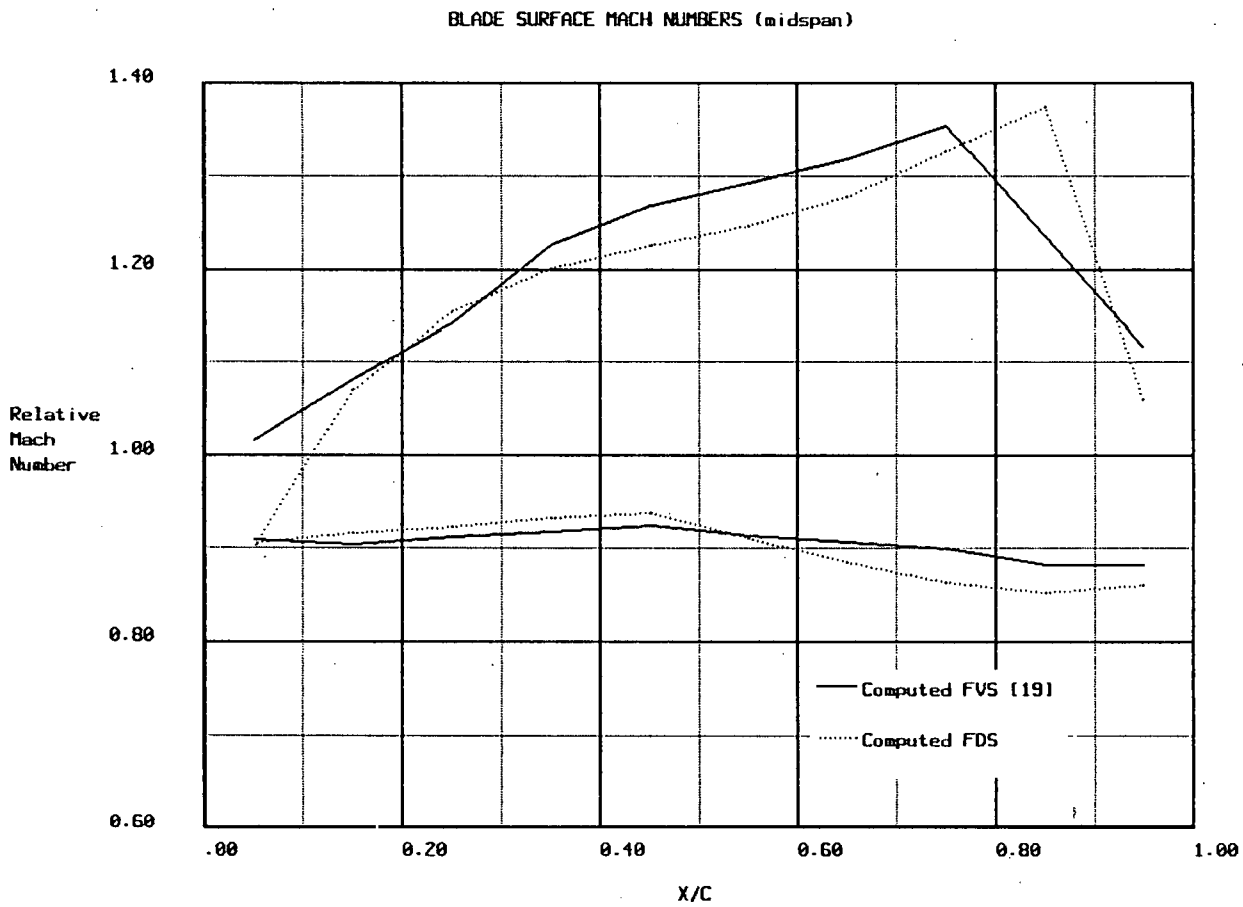
c. Torque Ratio

Figure 11. Performance Parameters



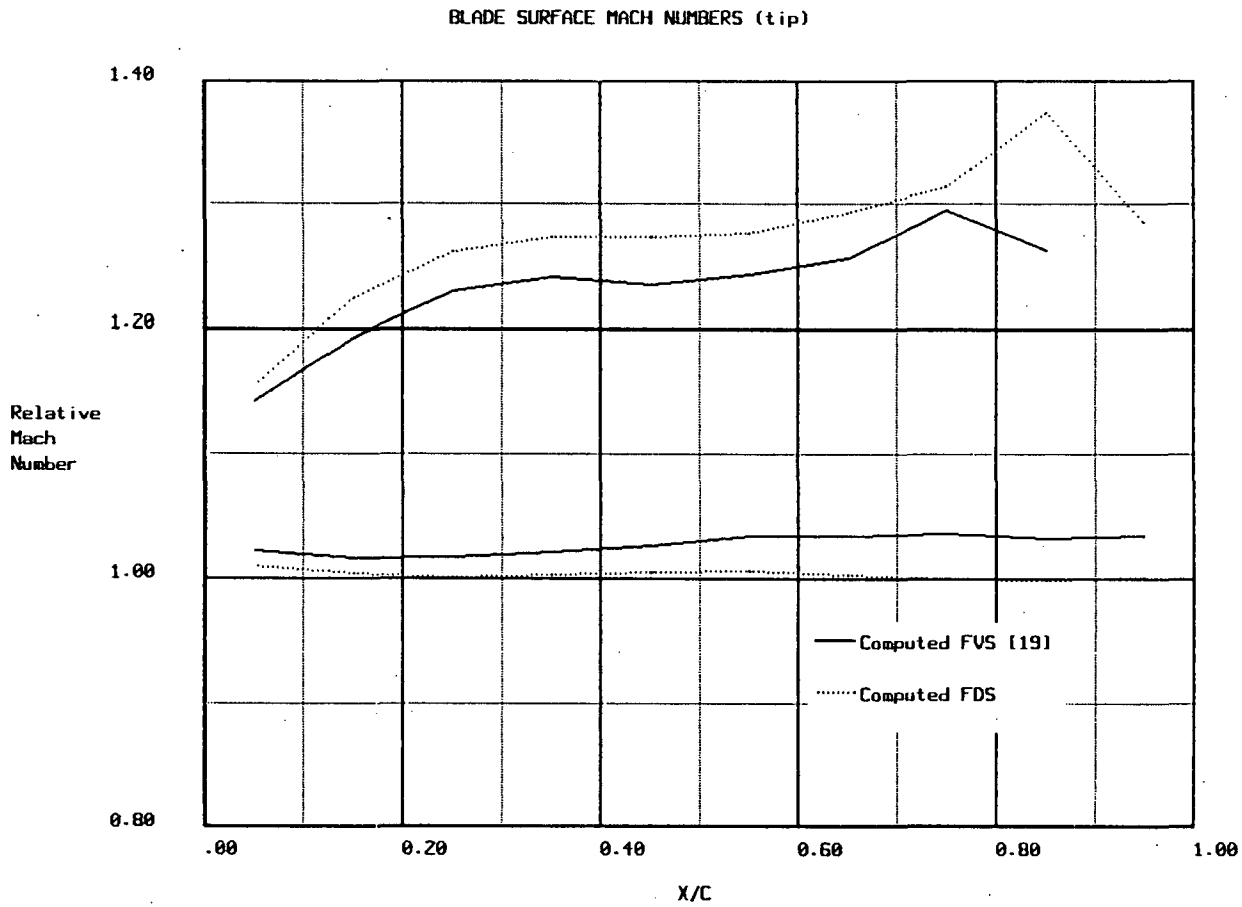
a. root

Figure 12. Forward Blade Row



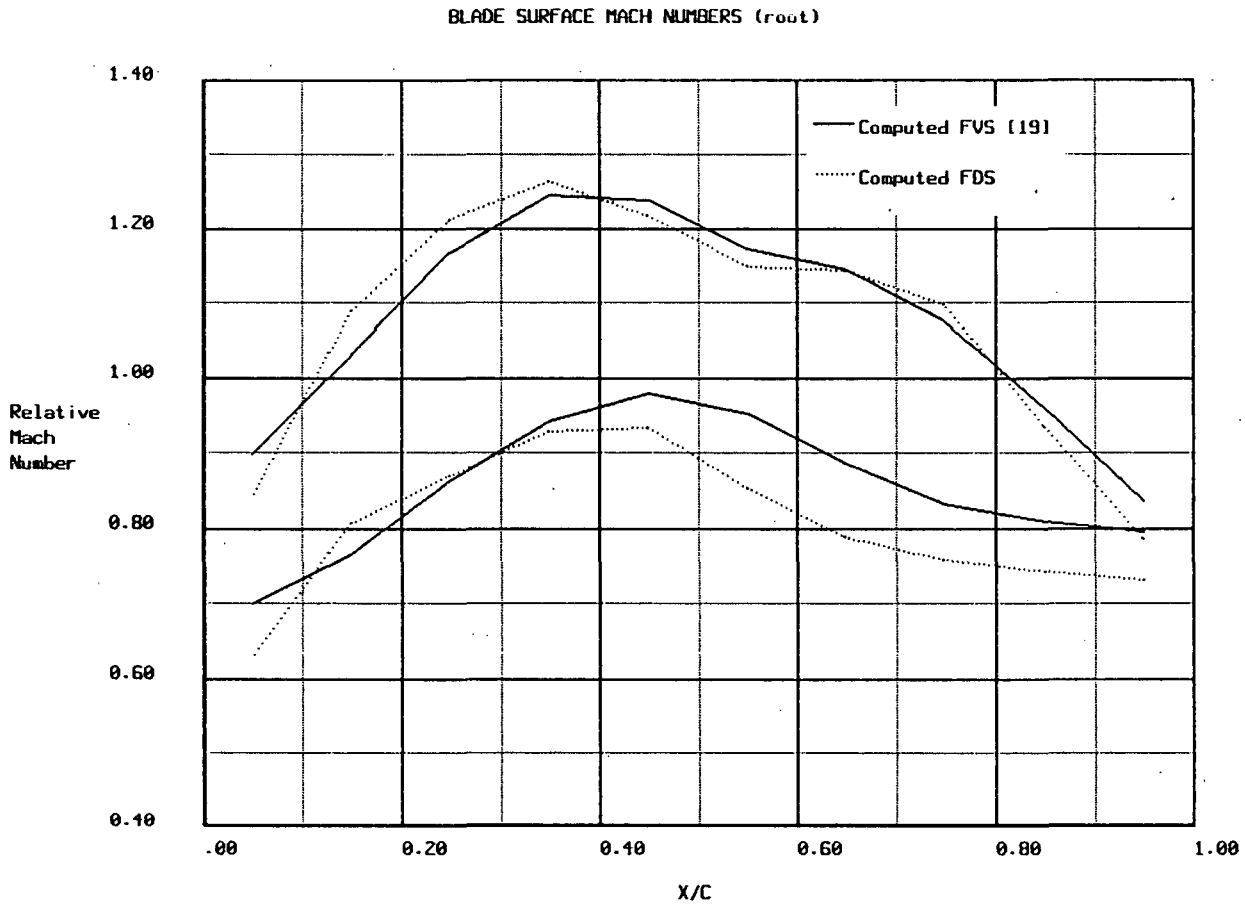
b. midspan

Figure 12. Forward Blade Row



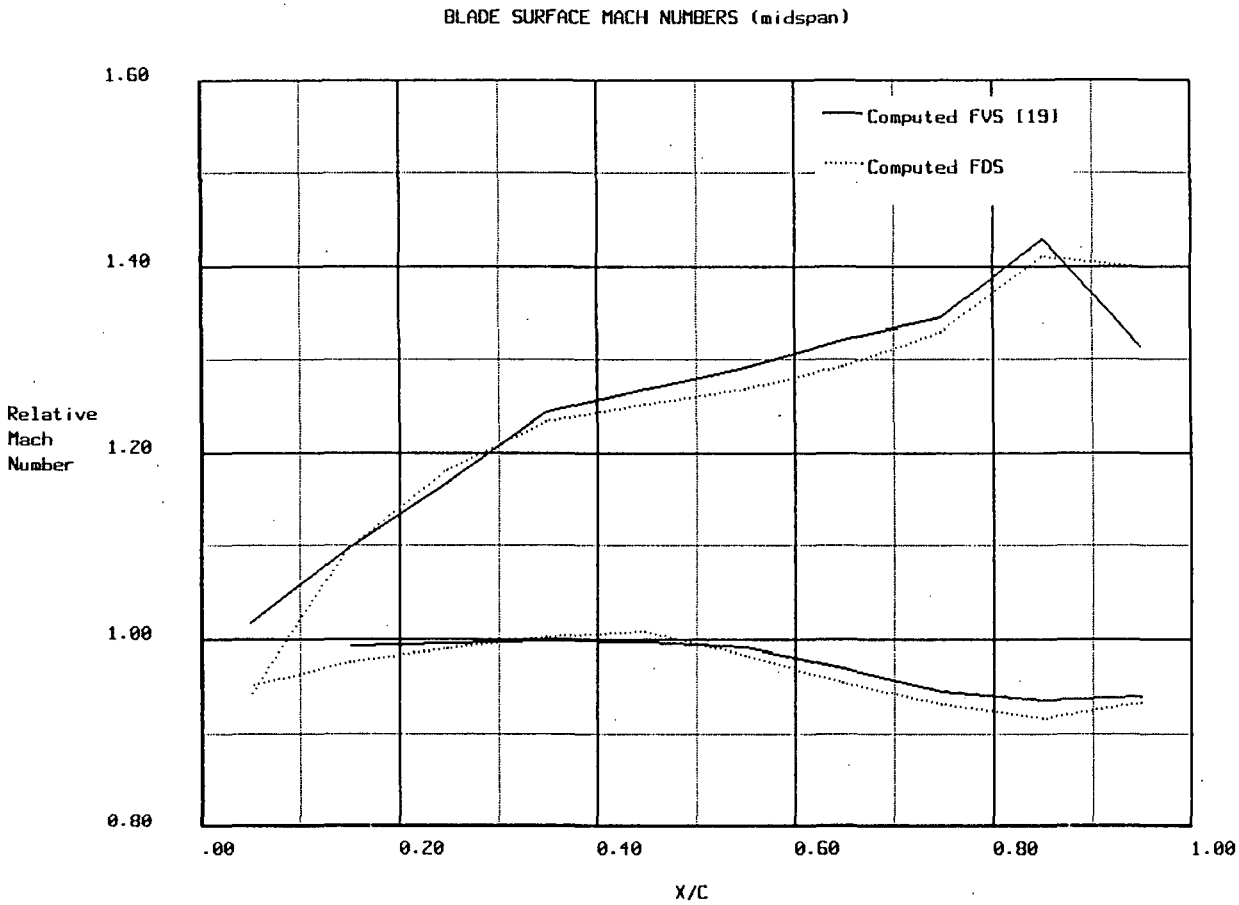
c. tip

Figure 12. Forward Blade Row



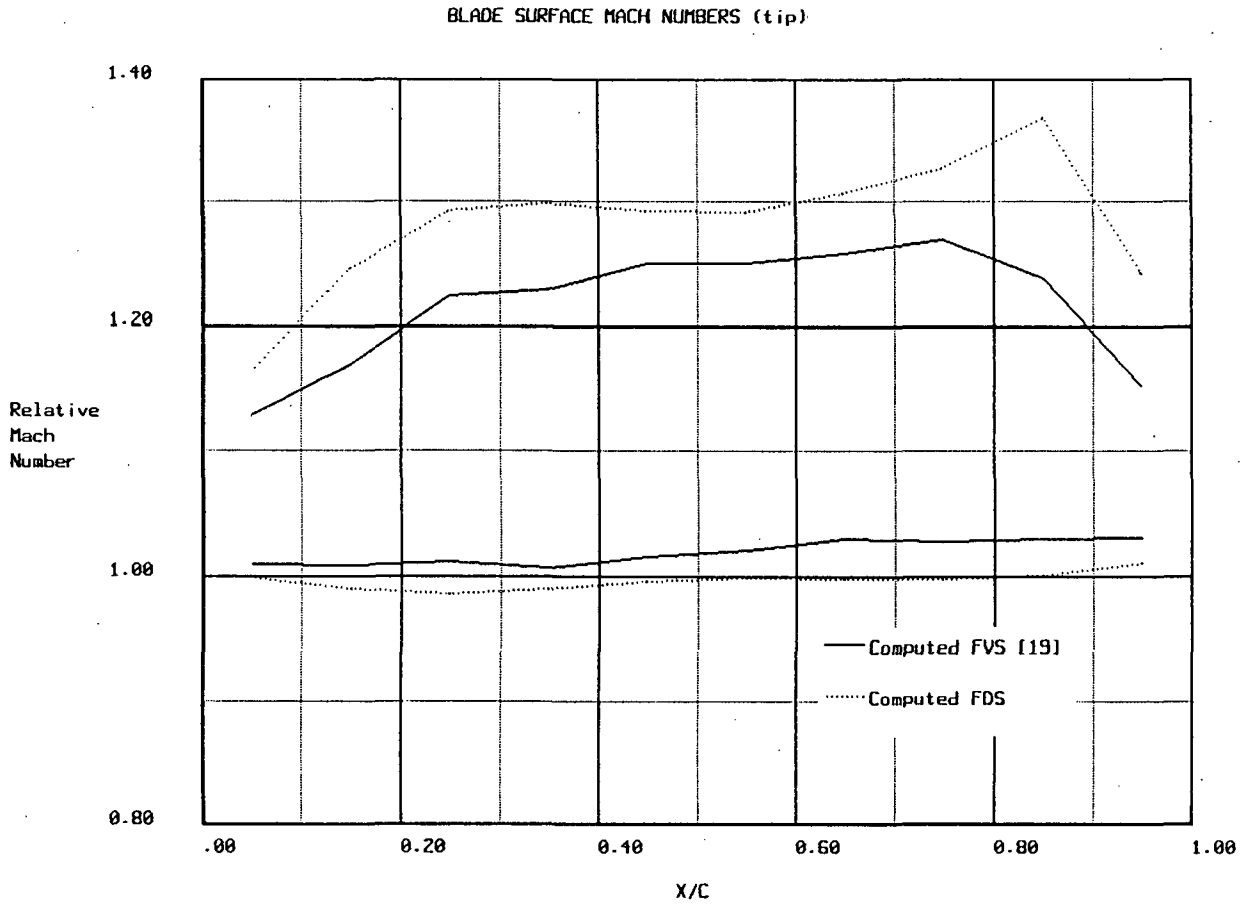
a. root

Figure 13. Aft Blade Row



b. midspan

Figure 13. Aft Blade Row



c. tip

Figure 13. Aft Blade Row