INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 7, 1211–1228 (1987)

ADAPTIVE FINITE ELEMENT METHODS FOR HIGH-SPEED COMPRESSIBLE FLOWS

J. T. ODEN, T. STROUBOULIS AND PH. DEVLOO

Texas Institute for Computational Mechanics, Department of Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, Austin, TX 78712-1085, U.S.A.

SUMMARY

We describe an adaptive finite element algorithm for solving the unsteady Euler equations. The finite element algorithm is based on a Taylor/Galerkin formulation and uses a very fast and efficient data structure to refine and unrefine the grid in order to optimize the approximation. We give a general version of the method which can be applied to moving grids with sliding interfaces and we present the results for a transient supersonic calculation of rotor–stator interaction.

KEY WORDS Adaptive Methods Finite Elements Compressible Flow Turbomachinery

INTRODUCTION

In 1946, von Neumann, realizing the limited capabilities of analytical methods in fluid dynamics, envisioned the use of numerical simulation as a powerful tool for solving complicated problems in fluid dynamics. Since then tremendous progress has been made in the production of very fast computing devices for such calculations. But it is a debatable issue as to whether or not similar progress has been made in the production of general and fully reliable numerical algorithms for fluid mechanics simulations.

Most of the popular methods in use in fluid dynamics calculations employ a local finite difference approximation of the flow equations on a structured grid. The selection of an appropriate finite difference scheme for the problems of interest is typically based on an analysis of the performance of the scheme for a one-dimensional problem. The selected finite-difference scheme is then applied to the multidimensional case by using dimensional splitting. Such procedures are based on the construction of smooth, orthogonal, well-structured meshes along the directions of a system of generalized co-ordinates. The construction of such meshes for complicated flow domains often presents a formidable task.

In this paper, we present a finite element formulation which generalizes Richtmeyer's two-step Lax–Wendroff method to multidimensional problems and arbitrary cell geometries. This method was used under different forms by several authors,[1–3] and is based on the Taylor–Galerkin ideas of Donea[4,5] and Oden.[13] In contrast with most finite difference methods, which require an elaborate structured grid generation, our method requires a coarse initial mesh which models only the basic geometrical features of the flow domain. Our algorithm estimates the local error and has

Based on an invited Lecture

0271-2091/87/111211-18$09.00

the ability to adapt the grid in order to produce optimal approximations.

The grid is dynamically adapted during the calculation by locally refining elements with large error and by unrefining 'groups' of elements with small 'group error'. When changing the structure of the grid a major problem of bookkeeping arises. We solve the data management problem by using a 'fast' data structure which permits local grid changes using a local modification of the data structure. The fact that we do not have to search through the whole list of elements in order to modify one element makes the data structure very attractive for transient adaptive calculations.

Following this Introduction, we give a brief description of the solution algorithm, we present the adaptive finite element pocedures for steady-state and transient calculation and we close the paper with the presentation of some numerical examples including the adaptive calculation of rotor–stator interaction.

FINITE ELEMENT ALGORITHMS

We consider the motion of a perfect gas through a time-dependent two-dimensional domain $\Omega(t) \subset \mathbb{R}^2$, $t \in [0,T]$. If $\mathbf{U} = \mathbf{U}(\mathbf{x}, t)$ is the vector of conservation variables with ρ the mass density, $\mathbf{m}$ the linear momentum and e the total energy, it satisfies the following space-time weak formulation.

Find $\mathbf{U} = \mathbf{U}(\mathbf{x}, t) \in V$ such that

$$-\int_{\tau_1}^{\tau_2}\int_{\Omega(t)} \mathbf{U}^{\mathrm{T}}\boldsymbol{\phi}_t \,\mathrm{d}\Omega\,\mathrm{d}t + \int_{\Omega_{\tau_2}} \mathbf{U}^{\mathrm{T}}(\mathbf{x},\tau_2)\boldsymbol{\phi}(\mathbf{x},\tau_2)\,\mathrm{d}\Omega - \int_{\Omega_{\tau_1}} \mathbf{U}^{\mathrm{T}}(\mathbf{x},\tau_1)\boldsymbol{\phi}(\mathbf{x},\tau_1)\,\mathrm{d}\Omega$$
$$= \int_{\tau_1}^{\tau_2}\int_{\Omega(t)} \mathbf{Q}(\mathbf{U}):\nabla\boldsymbol{\phi}\,\mathrm{d}\Omega\,\mathrm{d}t - \int_{\tau_1}^{\tau_2}\int_{\partial\Omega(t)} \boldsymbol{\phi}^{\mathrm{T}}[\mathbf{Q}(\mathbf{U}) - \mathbf{U}(\mathbf{u}^{\mathrm{G}}\cdot\mathbf{n})]\,\mathrm{d}s\,\mathrm{d}t, \quad \forall\phi \in \mathbf{W}. \tag{1}$$

Here the sets V and W are appropriately defined,[3] $\boldsymbol{\phi}_t = \partial\boldsymbol{\phi}/\partial t$, $\mathbf{Q}(\mathbf{U})$ is the Euler flux tensor,

$$\mathbf{Q}(\mathbf{U}) = \begin{bmatrix} m_1 & m_2 \\ \rho^{-1}m_1 + p(\mathbf{U}) & \rho^{-1}m_1m_2 \\ \rho^{-1}m_1m_2 & \rho^{-1}m_2^2 + p(\mathbf{U}) \\ \rho^{-1}m_1[\mathrm{e} + p(\mathbf{U})] & \rho^{-1}m_2[\mathrm{e} + p(\mathbf{U})] \end{bmatrix}, \tag{2}$$

and $p(\mathbf{U})$ is the thermodynamic pressure

$$p(\mathbf{U}) = (\gamma - 1)(e - \rho^{-1}\mathbf{m}\cdot\mathbf{m}/2). \tag{3}$$

Moreover, the following notation is used:

$$\mathbf{U}^{\mathrm{T}}\boldsymbol{\phi}_t = \sum_{\alpha=1}^{4} U_\alpha \partial\phi_\alpha/\partial t,$$
$$\mathbf{Q}:\nabla\boldsymbol{\phi} = \sum_{i=1}^{2}\sum_{\alpha=1}^{4} Q_{\alpha i}\,\partial\phi_\alpha/\partial x_i. \tag{4}$$

We obtain a finite element approximation of (1) by partitioning the space–time domain $D = \bigcup_{0\le t\le T}\Omega(t)$ into subdomains $D_n = \bigcup_{t_n\le t\le t_{n+1}}\Omega(t)$ with $0 = t_0 < t_1 < \ldots < t_n < t_{n+1} < \ldots < t_N = T$, by discretizing each subdomain and by using the discrete spaces of test and trial functions defined by the discretization. In order to discretize (1) in the case of variable domains we note that[6]

$$\partial\boldsymbol{\phi}_\alpha/\partial t = -\mathbf{u}^{\mathrm{G}}\cdot\nabla\boldsymbol{\phi}_\alpha$$

where $\mathbf{u}^{\mathrm{G}}$ denotes the grid velocity.

Using numerical quadrature to approximate the space–time integrals in (1),[3] we obtain the following two-step scheme.

First step

For each element Ω_e calculate a constant element vector $U_{\alpha,e}^{n+1/2}$ from

$$A_e^{n+1} U_{\alpha,e}^{n+1/2} = \sum_{i=1}^{4} \left(\int_{\Omega_e^n} \phi_i \, d\Omega \right) U^{i,n} - \frac{\Delta t}{2} \frac{A_e^{n+1}}{A_e^{n+1/2}} \left(\int_{\Omega_e^{n+1/2}} \frac{\partial \phi_i}{\partial x_\beta} d\Omega \right) Q_{\alpha,\beta}^{i,n}$$
$$+ \sum_{i=1}^{4} \sum_{j=1}^{4} \Delta t \int_{\Omega_e^{n+1/2}} \frac{\partial}{\partial x_\beta} (\phi_i \phi_j) u_\beta^{i,G} U_\alpha^{j,n} \, d\Omega . \tag{5}$$

Second step

For each node calculate U_α^{j+1} by solving the following system of equations:

$$\sum_{j=1}^{N} \left(\int_{\Omega^{n+1}} \phi_i \phi_j \, d\Omega \right) U_\alpha^{j,n+1} = \sum_{j=1}^{N} \left(\int_{\Omega^n} \phi_i \phi_j \, d\Omega \right) U_\alpha^{j,n} + \Delta t \int_{\Omega^{n+1}} Q^{n+1/2} \frac{\partial \phi_i}{\partial x_\beta} d\Omega$$
$$- \Delta t \int_{\partial\Omega^{n+1}} n_\beta (Q_{\alpha\beta}^{n+1/2} - Q_{\alpha\beta}^n) \phi_i \, ds - \Delta t \int_{\partial\Omega^{n+1}} n_\beta Q_{\alpha\beta}^n \phi_i \, ds . \tag{6}$$

Here $\bar{Q}_{\alpha\beta}^n$ denotes an elementwise averaged value of the flux and $Q_{\alpha\beta}$ is defined by

$$\bar{Q}_{\alpha\beta} = Q_{\alpha\beta} - U_\alpha u_\beta^G . \tag{7}$$

Equations (6) and (7) define a two-step TG/FELW (Taylor–Galerkin/finite-element Lax–Wendroff) method which was introduced in the finite difference literature by Burstein[7] used for the first time in a finite element context by Oden,[13] and is one of the Taylor–Galerkin methods introduced by Donea,[4] studied by Baker and Kim[5] and employed by Löhner *et al.*[1] and others[2,3] to the solution of compressible flow equations. The second step of the scheme involves a global calculation of the form

$$\mathbf{M}\{\mathbf{U}\}^{n+1} = \{\mathbf{R}\} . \tag{8}$$

Here $\mathbf{M}$ denotes the consistent mass matrix, $\{\mathbf{R}\}$ the load vector whose definition can be easily deduced from (6) and $\{\mathbf{U}\} = [\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3, \ldots, \mathbf{U}_M]^T$ is the global vector of nodal unknowns. The inversion of the mass matrix can be performed by Jacobi iteration[1,8] or a preconditioned Jacobi conjugate gradient.[3]

The TG/FELW method provides us with a fast, multidimensional time-stepping algorithm with high 'resolution' (high order of accuracy) in smooth regions of the flow and which applies to unstructured adaptive grids. Artificial diffusion is added in order to stabilize the scheme in the presence of discontinuities[7,9] and to suppress hour-glassing modes.[3]

Artificial diffusion terms prevent the occurrence of non-linear instabilities, but do not eliminate non-physical oscillations from the solution. In recent years, Boris, Book and others[8,10,11] have developed the theory of flux-corrected transport (FCT) in an attempt to correct finite-difference transport schemes in order to avoid non-physical oscillations in the solution. Fully multi-dimensional FCT schemes were presented by Zalesak,[11] and recently Löhner *et al.*[8] presented a flux-corrected procedure for the TG/FELW. We now give a short exposition of the FCT-TG/FELW algorithm which we employed in some of our examples.

The FCT procedure consists in solving equation (8) by using first a 'strong' diffusion step [10,11] to obtain a 'transported and diffused' solution which is free of non-physical oscillations; then an antidiffusion step with flux limiting[11] in order to steepen the solution at discontinuities. In particular, we have:

Step I: 'diffusion' step

Compute $\{\mathbf{U}_{\text{td}}^{n+1}\}$ from

$$\mathbf{M}_{\text{L}}\{\mathbf{U}_{\text{td}}^{n+1}\} = \{\mathbf{R}\} + \{\mathbf{V}\}. \tag{9}$$

Here $\mathbf{M}_{\text{L}}$ denotes the lumped mass matrix and $\{\mathbf{V}\}$ denotes the vector of added diffusion, with nodal contributions of the form

$$\mathbf{V}_i = \int_\Omega \left(D_x \frac{\partial \phi_i}{\partial x}\frac{\partial \mathbf{U}^n}{\partial x} + D_y \frac{\partial \phi_i}{\partial y}\frac{\partial \mathbf{U}^n}{\partial y} \right) \mathrm{d}\Omega. \tag{10}$$

For a mesh of quadrilaterals we let

$$D_x = D_y = cA_e,$$

where c is a constant and A_e denotes the area of the element Ω_e.

Step II: 'antidiffusion' step

Compute $\{\mathbf{U}^{n+1}\}$ as the limit of the sequence of iterates $\{\mathbf{U}_{[i]}^{n+1}\}, i = 1, 2, 3, \ldots$ defined by

$$\mathbf{M}_{\text{L}}\{\mathbf{U}_{[i+1]}^{n+1} - \mathbf{U}_{\text{td}}^{n+1}\} = \mathbf{l}(\mathbf{F}_{[i]}), \tag{11}$$

$$\mathbf{F}_{[i]} = (\mathbf{M}_{\text{L}} - \mathbf{M})\mathbf{U}_{[i]}^{n+1} - \mathbf{V}.$$

Here $\mathbf{l}$ denotes the flux limiting function of Zalesak.[11]

ADAPTIVE PROCEDURES

We now present adaptive procedures for the equations of compressible flow. The organization of the adaptive procedure depends on the desired result. Adaptive procedures for steady state problems are different from adaptive procedures for the accurate integration of the transient response. To initiate the adaptive procedure for a given flow domain, a coarse finite element mesh is defined which contains only a number of elements sufficient to model the basic geometric features of the flow domain (see Figure 1(b)). Each element of the initial mesh is assigned a 'level' equal to zero. Then the initial mesh is bisected uniformly several times in order to construct an initial grid which has the 'group' structure. Note that when an element is refined a group of four elements is defined and each one of the four new elements has a level one unit higher than the parent element (for more details about the data structures see Reference 3).

An adaptive procedure for steady-state solutions of the Euler equations involves the following steps:

1. For a given finite element grid determine the steady-state solution.
2. Compute error indicators θ_e overall elements in the grid. Let

$$\theta_{\text{MAX}} = \max_{1 \leq e \leq M} \theta_e$$

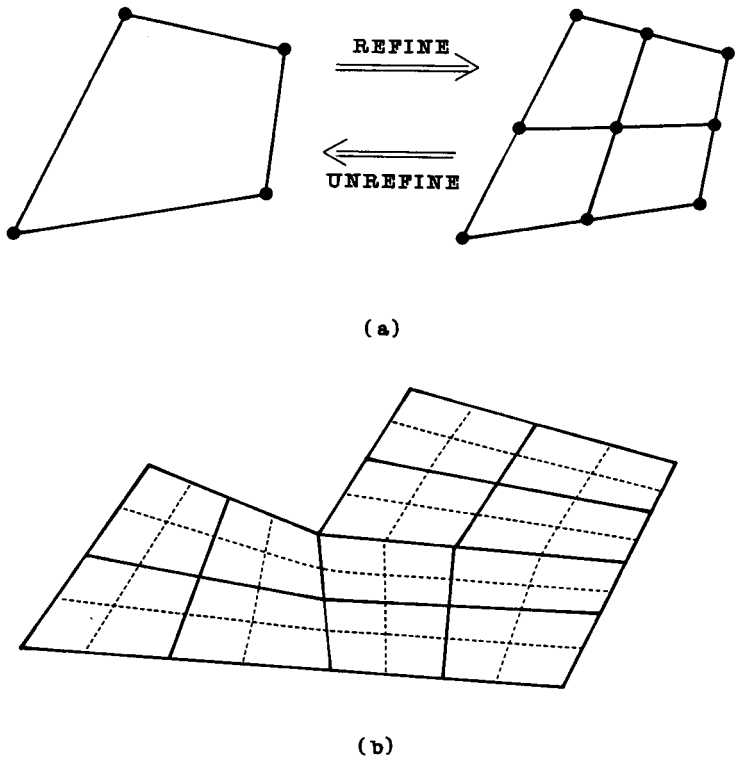

Figure 1. (a) Refinement and unrefinement of a four-element group and (b) a coarse initial mesh consisting of four-element groups

3. Scan groups of four elements and compute

$$\theta^m_{\text{GROUP}} = \sum_{k=1}^{4} \theta_{m_k}$$

where m_k is the k^{th} element in group m.

4. For given tolerances $\alpha, \beta \in \mathbb{R}, 0 < \alpha, \beta < 1$, if $\theta_e \geqslant \beta\theta_{\text{MAX}}$ refine element Ω_e by bisecting it into four new elements. If $\theta^m_{\text{GROUP}} \leqslant \alpha\theta_{\text{MAX}}$ unrefine group m by replacing the group by a single new element with the nodes coincident with the corner nodes of the group.
5. Go to step 1.

We also present an example of an h-refinement/unrefinement strategy for transient calculation. The goal of this algorithm is accurately to follow the features of the transient response and to preserve temporal accuracy. The basic steps of the algorithm are

(a) Advance the solution N time steps.
(b) Do the following until no more elements can be refined:
  (1) compute the element error indicators θ_e
  (2) refine all elements with $\theta_e \geqslant \beta\theta_{\text{MAX}}$
  (3) integrate the last N time steps with the updated (refined) mesh

(4) go to (1)

(c) Compute the element error indicators θ_e and unrefine all groups with $\theta^m_{\text{GROUP}} \leqslant \alpha\theta_{\text{MAX}}$.

(d) Go to (a).

We note that the 'do loop' in step (b) converges when no more elements can be refined (the maximum level of refinements is fixed). Although the iteration in step (b) guarantees a *'fully upated' mesh* it may lead to an expensive scheme if more than a few passes are required for convergence of the 'do loop'. A cheaper alternative is presented by the following 'two-pass' scheme:

(a) Advance the solution N time steps.

(b) Compute the element error indicators θ_e.

(c) Refine all elements with $\theta_e \geqslant \beta\theta_{\text{MAX}}$.

(d) Integrate the last N time steps with the refined mesh obtained in step (c).

(e) Compute the element error indicators θ_e and

(1) unrefine all groups with $\theta^m_{\text{GROUP}} \leqslant \alpha\theta_{\text{MAX}}$

(2) refine all elements with $\theta_e \geqslant \beta\theta_{\text{MAX}}$.

(f) Go to step (a).

The adaptive procedures presented above require the computation of reliable error indicators θ_e. In our calculation we employed the following definition:

$$\theta_e = \frac{\sqrt{A_e}}{\rho_e} \sup_{i=1,2} \left| \frac{\partial \rho}{\partial x_i} \right|.$$

Here ρ_e denotes an average value of density for element Ω_e and A_e denotes the element area. The numerical examples show that our error indicator captures well the location of variable shocks. For alternative definitions of error indicators see References 3 and 8.

NUMERICAL EXAMPLES

Supersonic flow over a sharp corner

We consider the problem of a uniform Mach 3 ($\gamma = 1{\cdot}40$) flow which is deflected by a 20° ramp. The flow enters with uniform flow conditions through the left boundary and, travelling horizontally with constant supersonic speed, arrives at the corner and turns discontinuously 20° into the direction of the other leg of the angle travelling again at constant velocity. The flow domain consists of two constant states which are separated by an oblique shock front attached to the corner tip.

The calculation of the steady flow was performed with the adaptive scheme described above with constants $\alpha = 0{\cdot}05, \beta = 0{\cdot}15$. Figures 2–4 show the computed steady-state pressure contours with zero, one and two levels of refinement, respectively. The results were obtained using the FCT version of the algorithm with 'viscous' constant, $c = 0{\cdot}125$ and C.F.L. number equal to 0·25.

Supersonic rotor–stator interaction

We applied the 'two-pass' version of the adaptive procedure for transient calculations to a problem of rotor–stator interaction.[6,12] We consider two rows of unsymmetric aerofoils which are shown, together with the initial discretization of the domain, in the upper part of Figure 5(a). We assumed that the stator and the rotor have the same number of aerofoils and we perform the computation on domains corresponding to one rotor and one stator aerofoil, simulating the

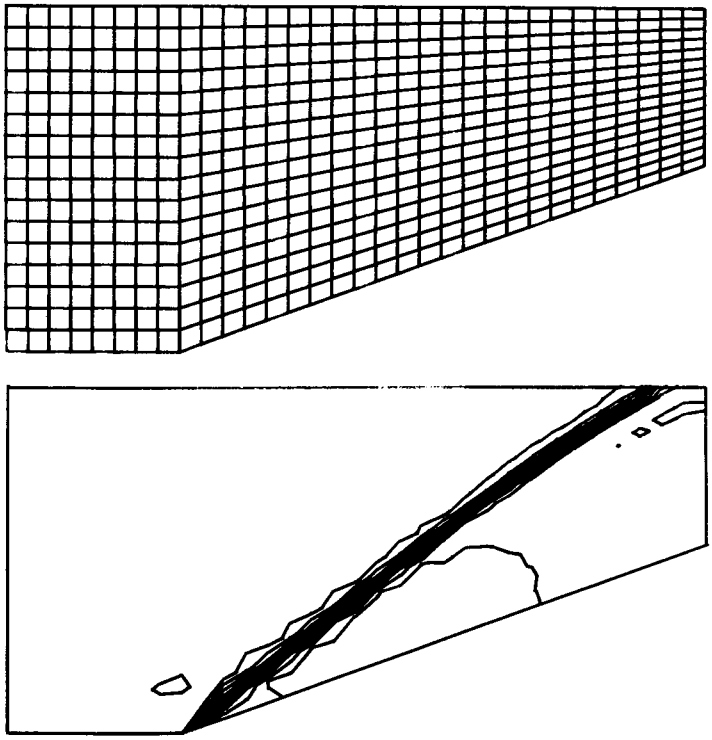

Figure 2. Upper: initial coarse mesh, and lower: corresponding pressure contours for supersonic flow over a ramp

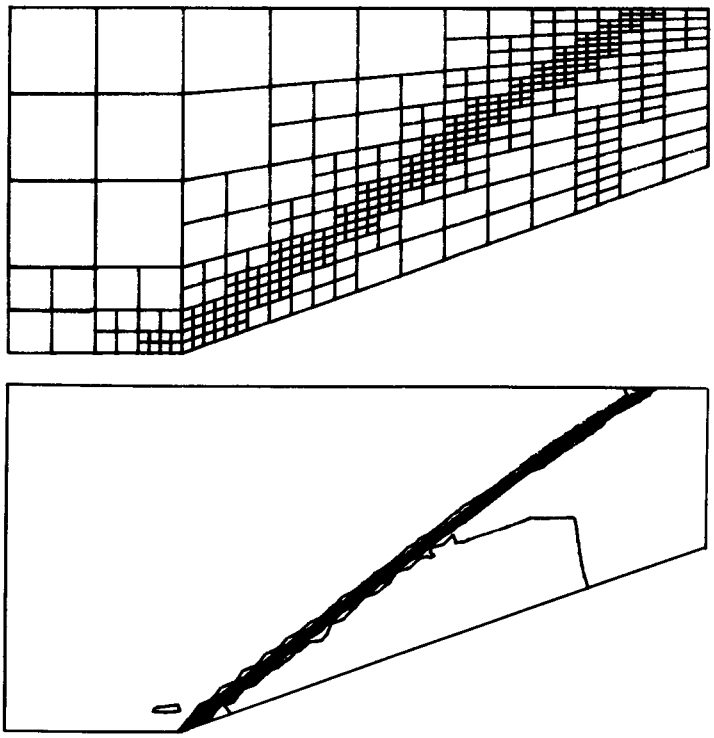

Figure 3. Upper: first level refinement, and lower: corresponding pressure contours for supersonic flow over a ramp

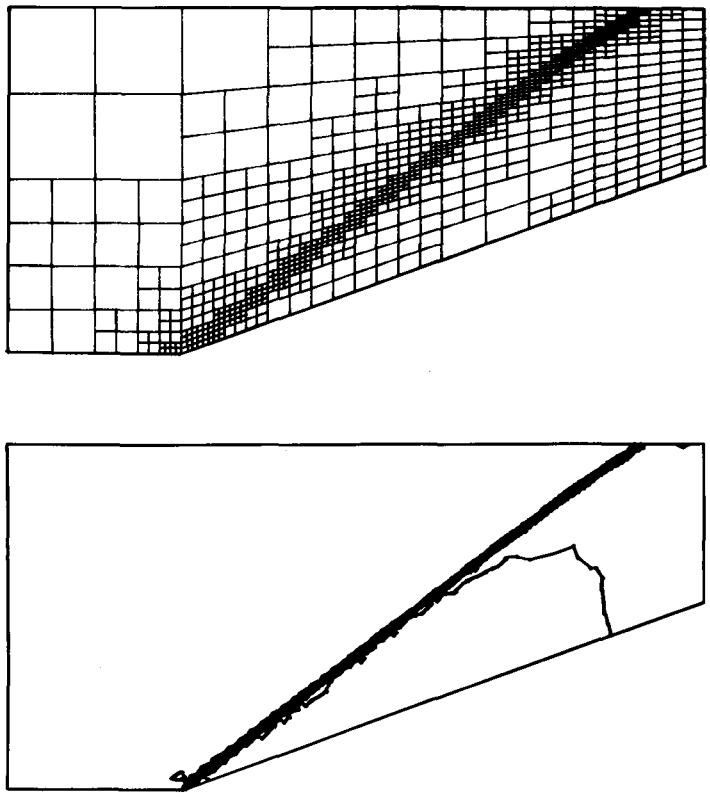

Figure 4. Upper: second level refinement, and lower: corresponding pressure contours for supersonic flow over a ramp

presence of the remaining aerofoils by periodic boundary conditions. We imposed supersonic inflow boundary conditions at the stator corresponding to a free-stream Mach number equal to three ($\gamma = 1{\cdot}40$). Boundary conditions of supersonic outflow were assumed on the right boundary of the rotor. In the beginning, all the aerofoils are kept fixed in order to obtain a steady flow pattern (lower part of Figure 5(a)) which was used as an initial condition in our calculations. After the steady state is reached we start moving the left row of aerofoils (stator) upward with a velocity equal to one-third of the inflow velocity.

Figures 5(a)–5(i) show the computed adaptive finite element meshes and the corresponding density contours for various stages of the stator motion. Figure 5(a) shows the initial mesh and the density contours for the initial conditions. The remaining plots show the evolution of the mesh and the density contours for 1/4, 2/4, 3/4, 4/4, 5/4, 6/4, 7/4, 8/4 cycles of motion. The results show that the error indicator captures well discontinuities of variable strength and that the mesh is dynamically adapted to follow the prevailing features of the solution.

ACKNOWLEDGEMENTS

Support of a portion of our research on adaptive methods has been provided by ONR under Contract N00014-84-K-0409 for error estimation and transient problems in solid mechanics. The data management scheme and CFD codes described here were completed under support of the Aerothermal Loads Branch of NASA Langely Research Center, Contract NAS1-17894.

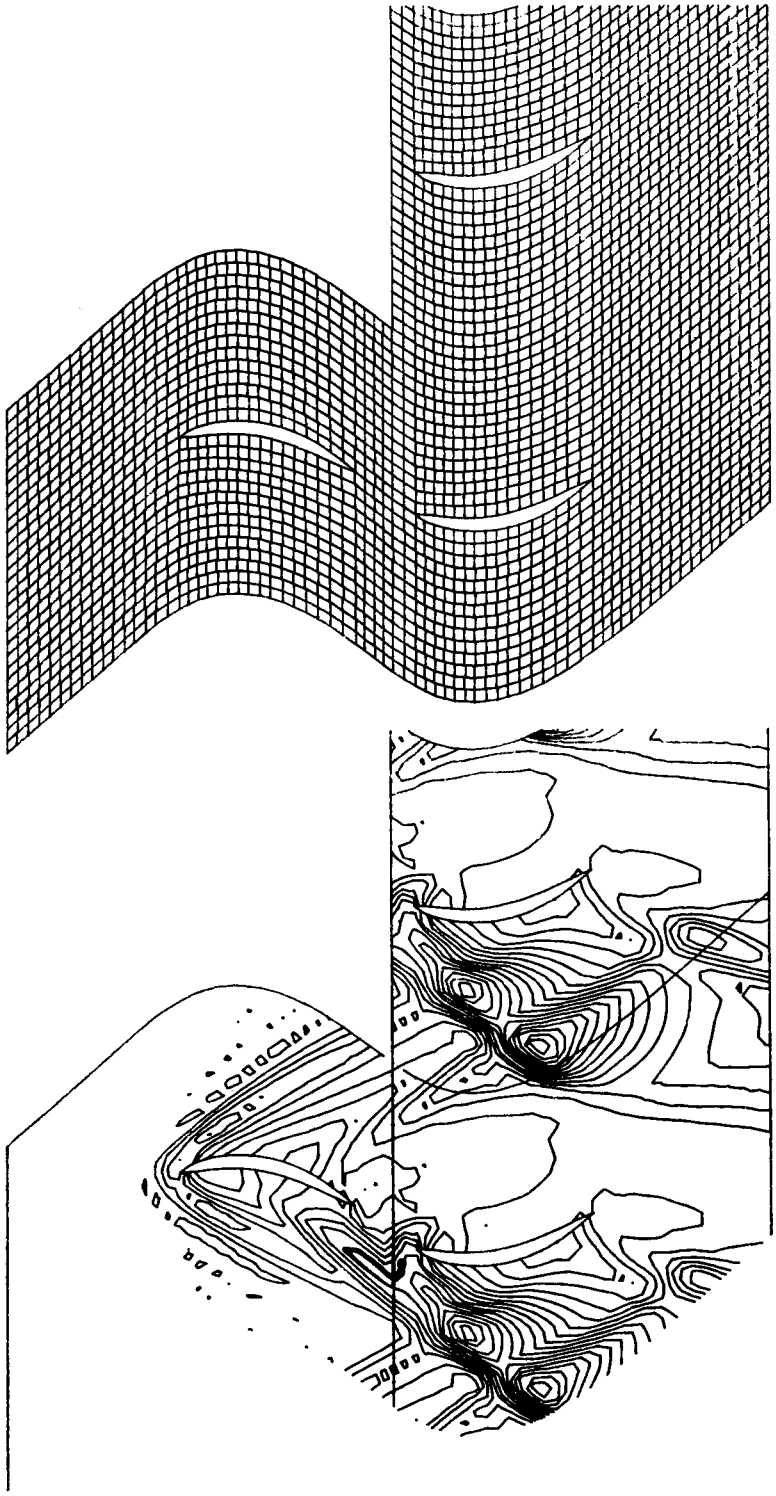

Figure 5(a). Upper: initial mesh for dynamic rotor–stator flow interaction problem, and lower: computed initial pressure distribution. The next sequence of Figures shows the dynamically refined and unrefined meshes and corresponding density contours for various time instants measured in fractions of a rotor–stator period P

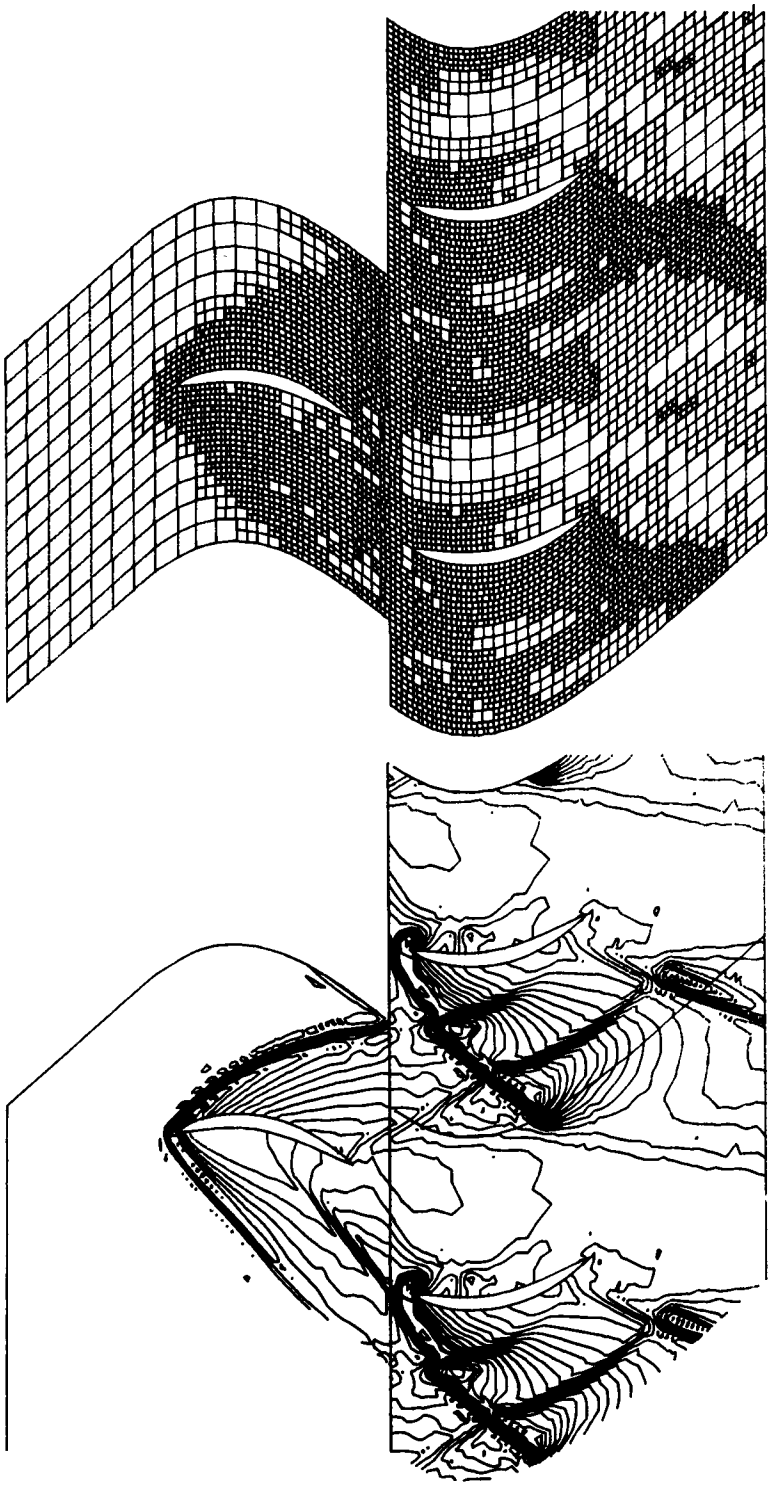

Figure 5(b). 1/4 P

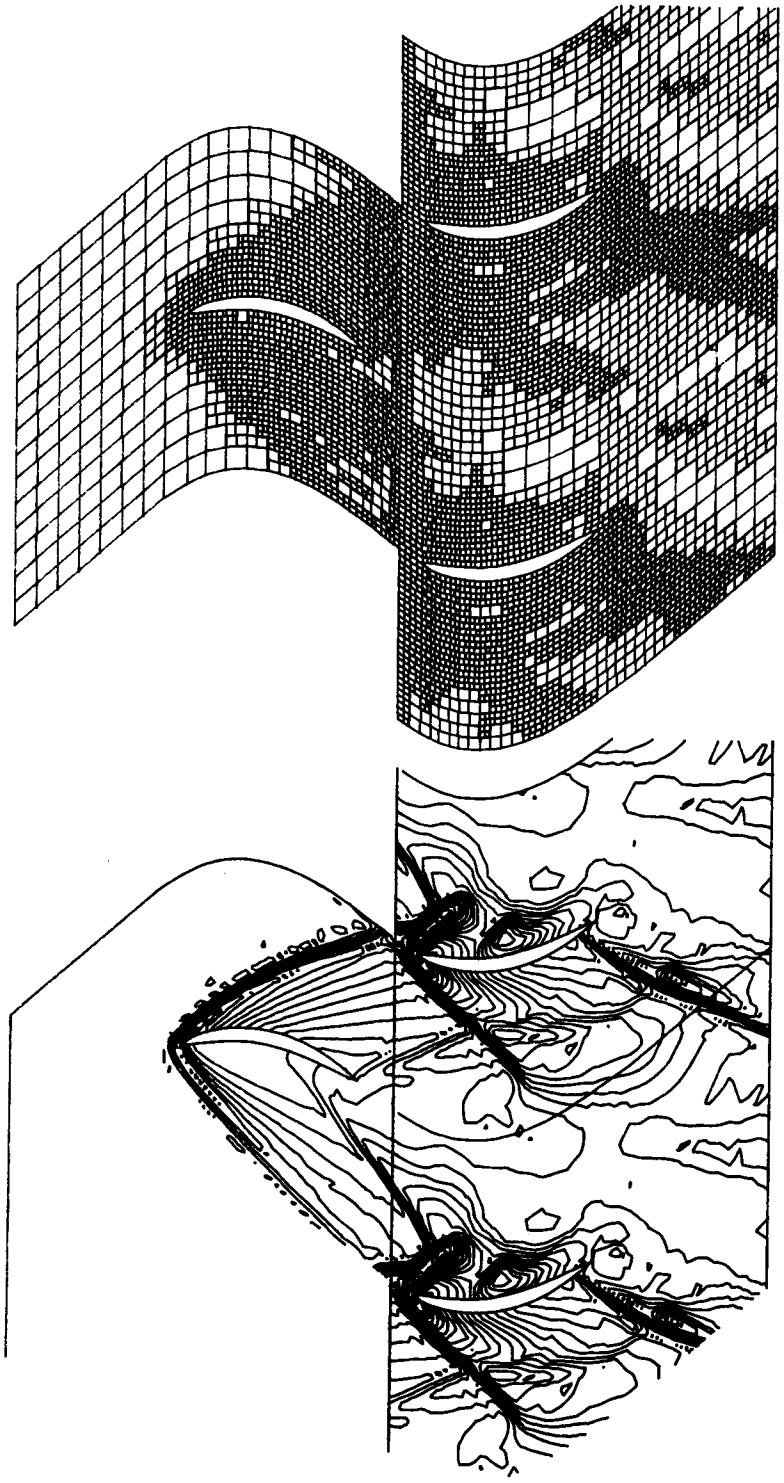

Figure 5(c). 2/4 *P*

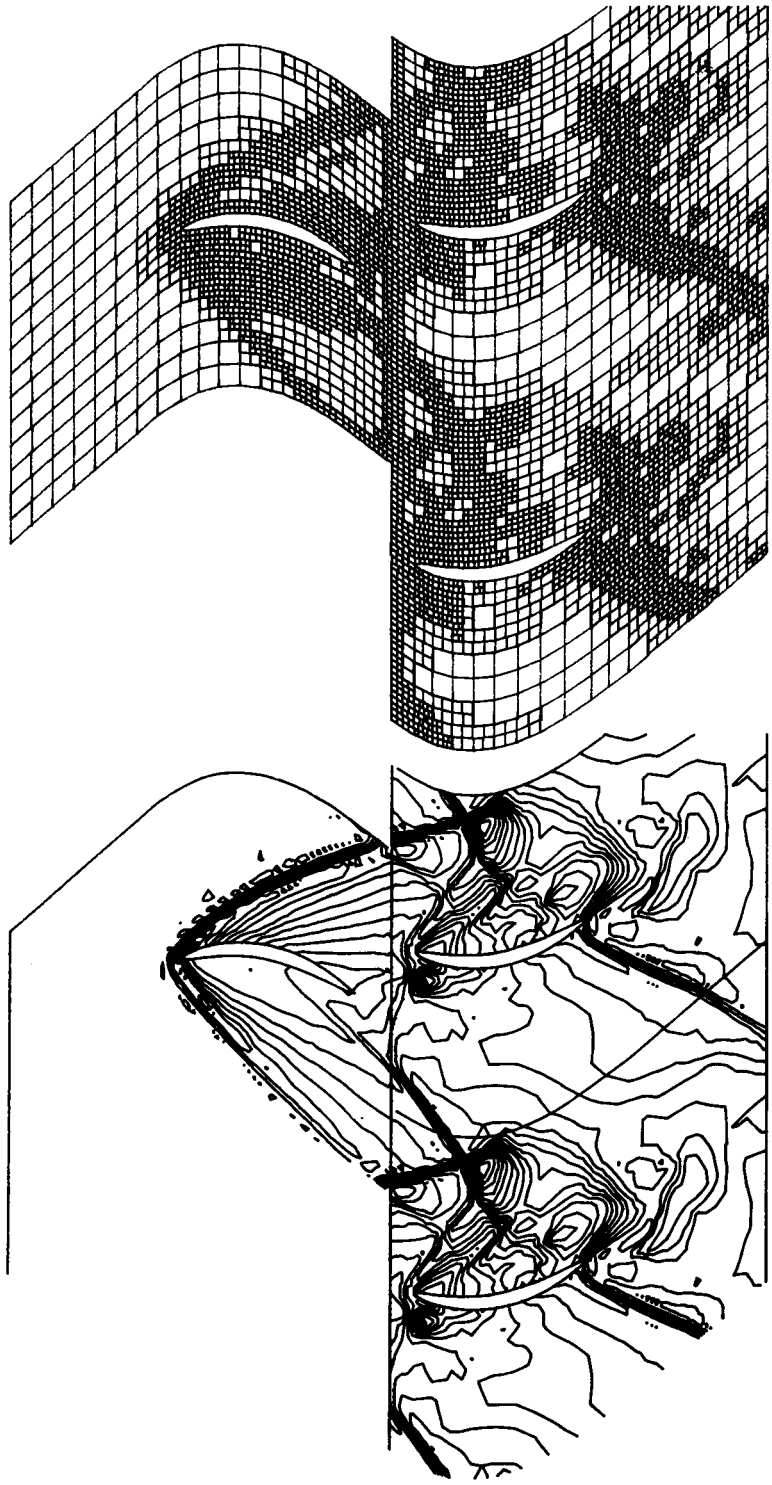

Figure 5(d). 3/4 *P*

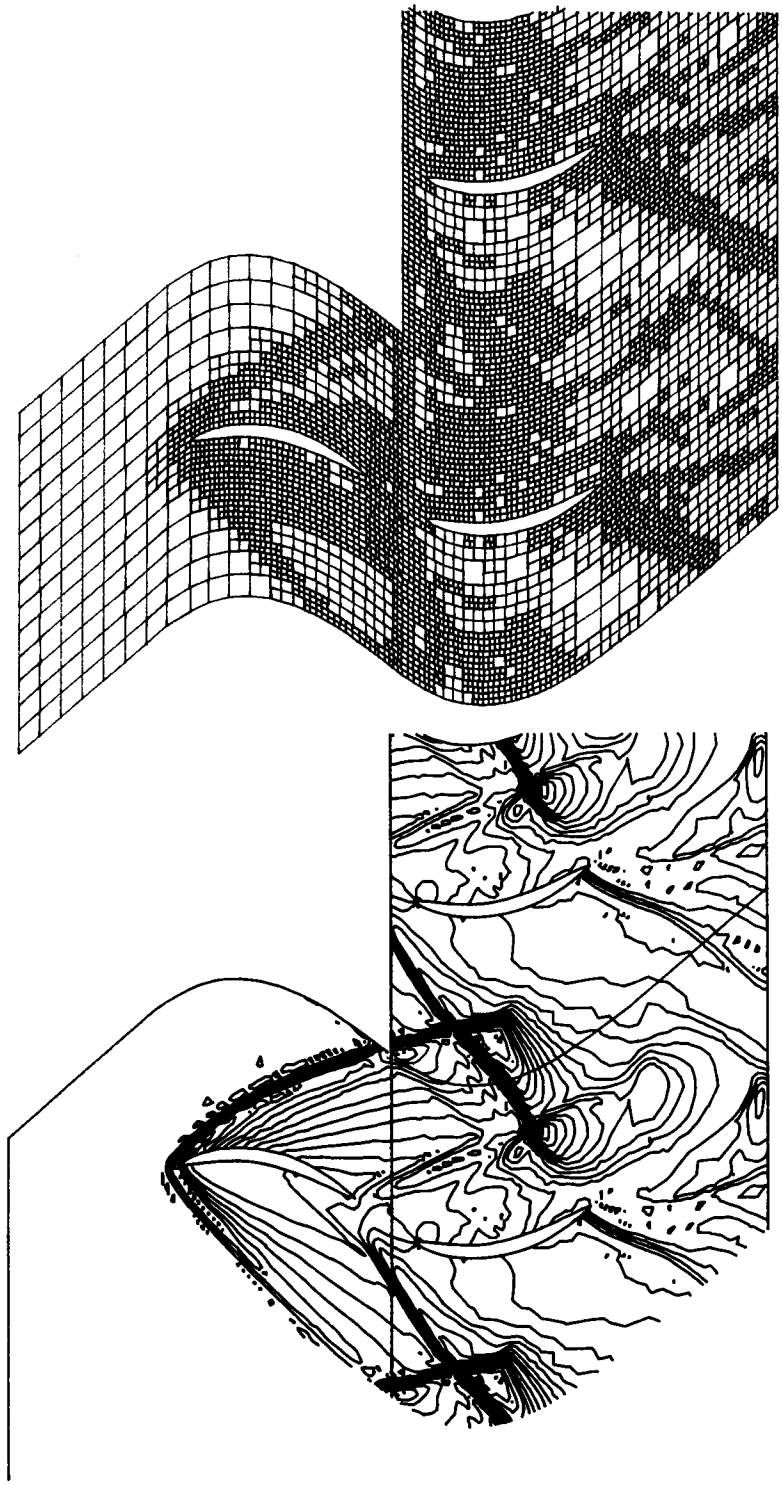

Figure 5(e). 4/4 *P*

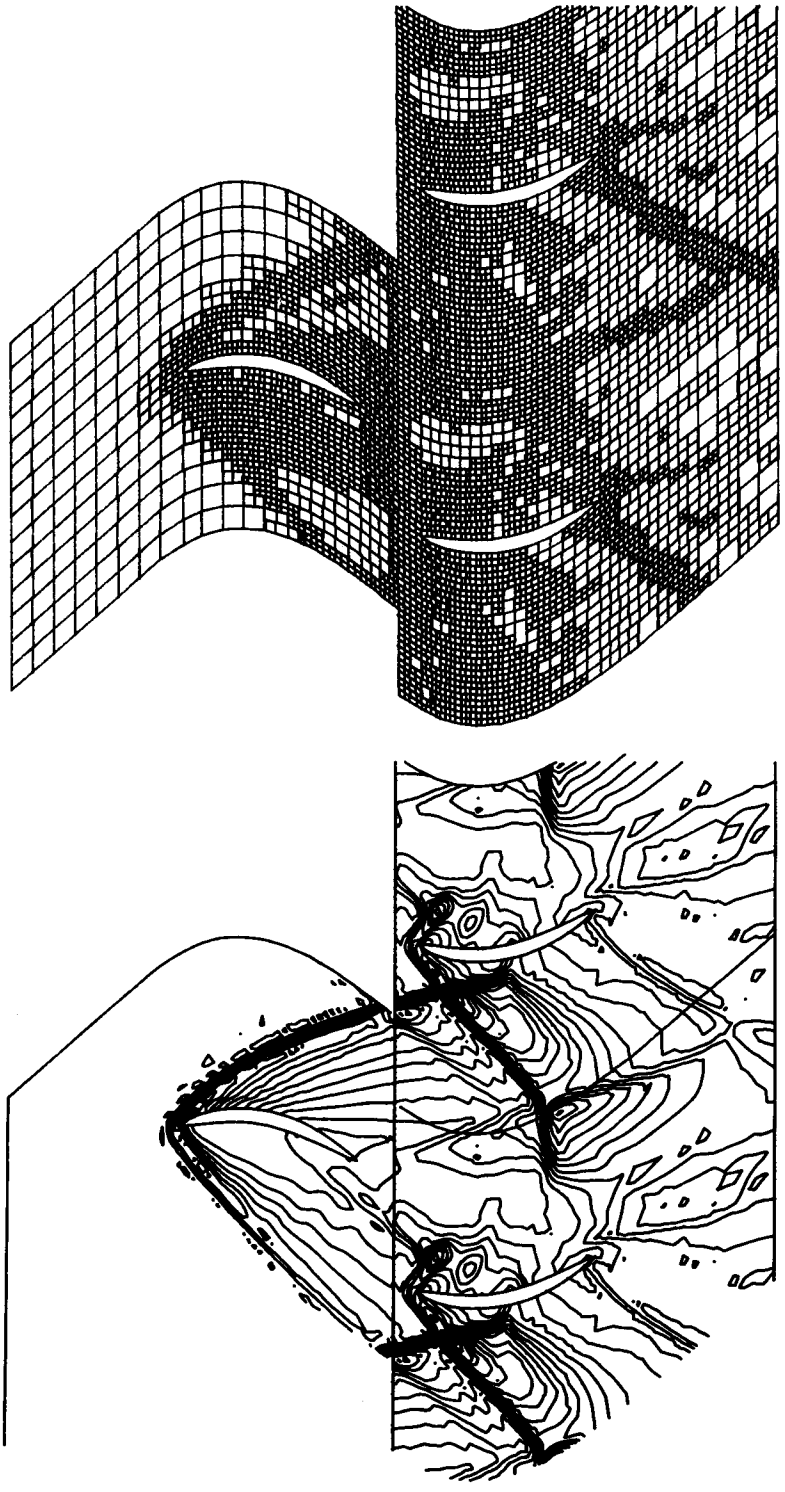

Figure 5(f). 5/4 *P*

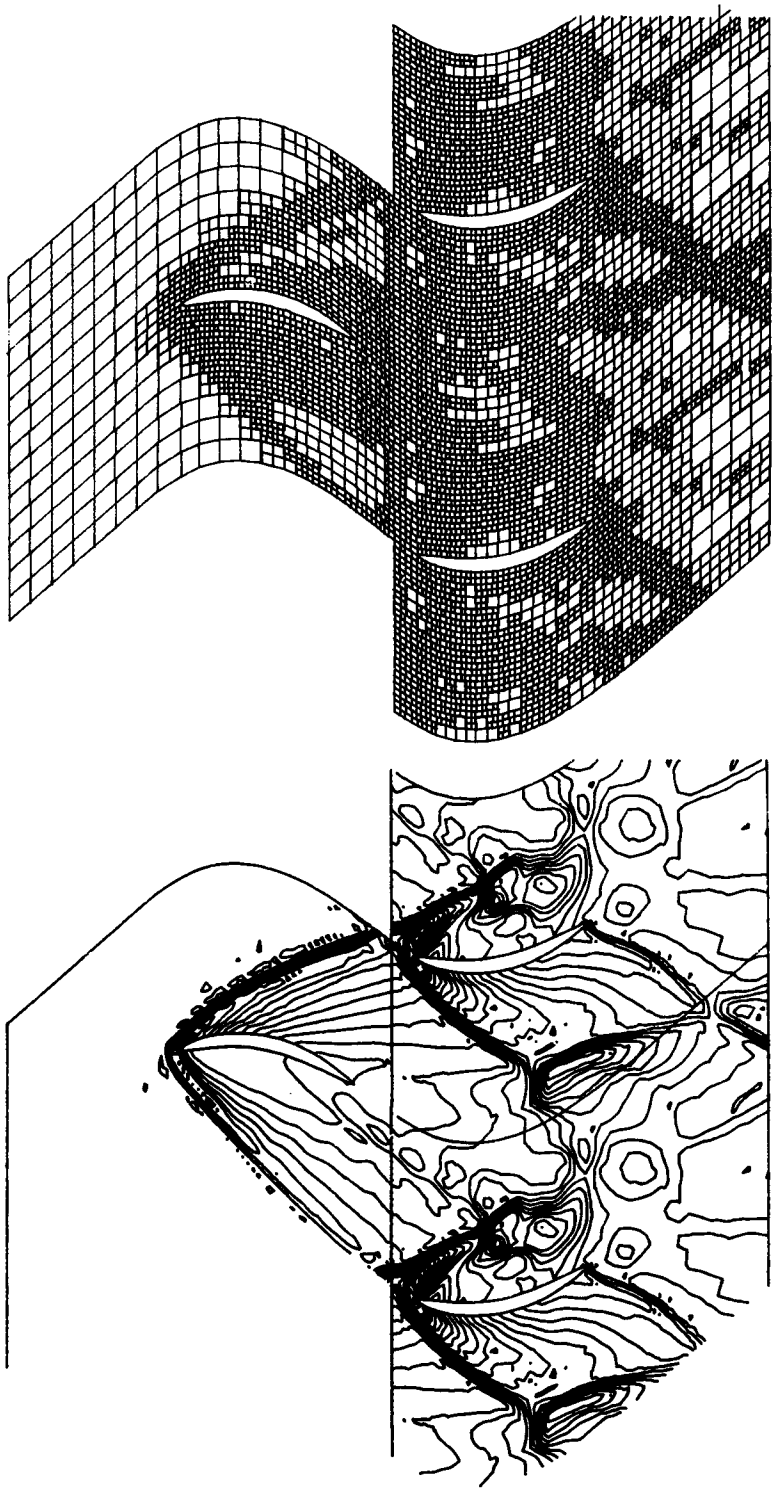

Figure 5(g). 6/4 *P*

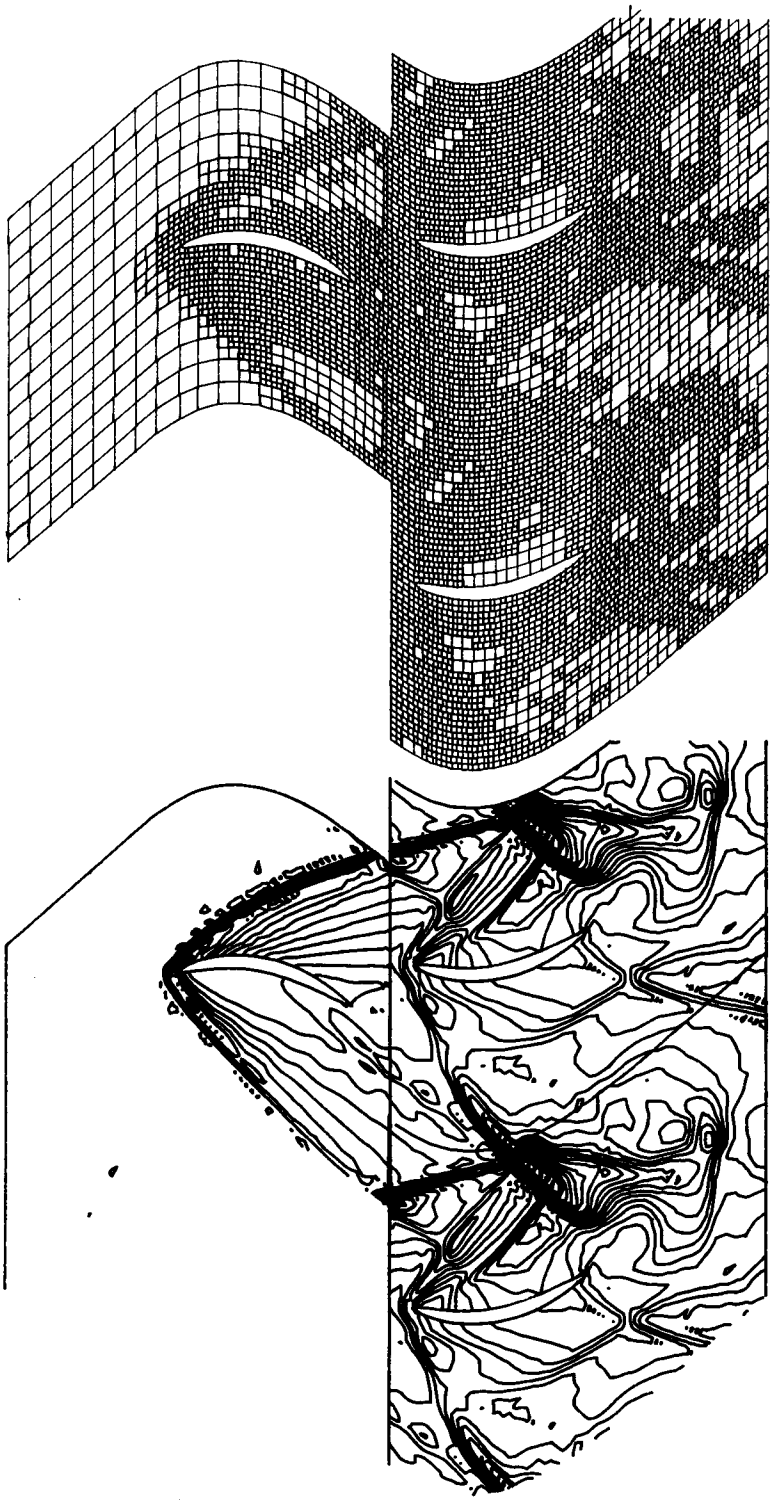

Figure 5(h). 7/4 P

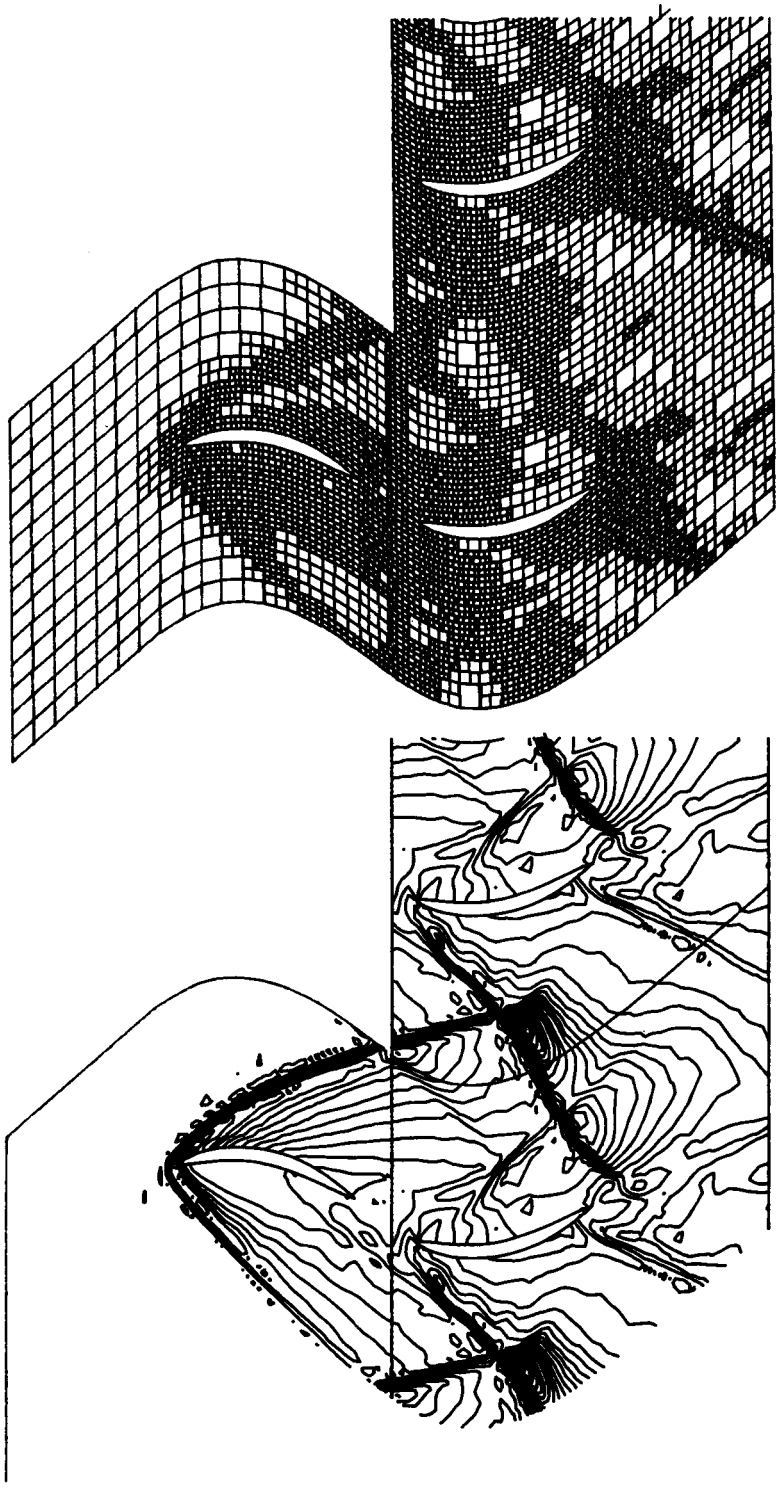

Figure 5(i). 8/4 *P*

REFERENCES

1. R. Löhner, K. Morgan and O. C. Zienkiewicz, 'An adaptive finite element procedure for compressible high speed flows', *Comput. Meth. Appl. Mech. Eng.*, **51**, 441–466 (1986).
2. K. S. Bey, E. A. Thornton, P. Dechaumpai and R. Ramakrishnan, 'A new finite element approach for prediction of aerothermal loads—progress in inviscid flow computations', 7th AIAA Comp. H Dyn. Conf., Cincinnati, *AIAA Paper No. 85-1533-CP*. 1985.
3. J. T. Oden, T. Strouboulis and Ph. Devloo, 'Adaptive finite element methods for the analysis of inviscid compressible flow: I. fast refinement/unrefinement and moving mesh methods for unstructured meshes', *Comput, Meth. Appl. Mech. Eng.*, **59**(3), 327–362 (1986).
4. J. Donea, 'A Taylor–Galerkin method for convective transport problems', *Int. j. numer. meth. eng.*, **20**, 101–120 (1984).
5. A. J. Baker and J. W. Kim, 'Analysis of a Taylor weak-statement algorithm for hyperbolic conservation laws', *Tech. Rept. CFDL/86-1*, Computer fluid Dynamics Laboratory, Univ. of Tennessee, Knoxville, TN, 1986.
6. T. Strouboulis, Ph. Devloo and J. T. Oden, 'A moving-grid finite element algorithm for supersonic flow interaction between moving bodies', *Comput. Meth. Appl. Mech. Eng.*, **59**(2), 235–255 (1986).
7. S. Z. Burstein, 'Finite difference calculations for hydrodynamic flows containing discontinuities', *J. Computational Physics*, **20**, 198–222 (1967).
8. R. Löhner, K. Morgan, M. Vahdati, J. P. Boris and D. L. Book, 'FEM-FCT: combining unstructured grids with high resolution', (to appear).
9. A. Lapidus, 'A detached shock calculation by second order finite differences', *J. Computational Physics*, **2**, 154–177 (1967).
10. J. P. Boris and D. L. Book, 'Flux corrected transport I: SHASTA, A fluid transport algorithm that works', *J. Comp. Physics*, **11**, 38–69 (1973).
11. S. T. Zalesak, 'Fully multidimensional flux-corrected transport algorithm for fluids', *J. Computational Physics*, **31**, 335–362 (1979).
12. J. T. Oden, T. Strouboulis, Ph. Devloo, S. J. Robertson and L. W. Spradley, 'Adaptive and moving mesh finite element methods for flow interaction problems', *Proceedings Sixth Inter. Conference on Finite Element Methods in Fluids*, Antibes, France, June 1986.
13. J. T. Oden, 'Formulation and Application of Certain Primal and Mixed Finite Element Models of Finite Elastic Deformations', *Computer Methods on Applied Science and Engineering*, Part 1, Edited by R. Glowinski and J. L. Lions, Int'l. Symposium at Versaille, 334–365 (1973).
14. L. W. Spradley, I. F. Stalnaker, M. A. Robinson and K. E. Xiques, 'Finite element algorithms for compressible flow computation on a supercomputer', *Finite Elements in Fluids*, **6**, 135–156 (1985).