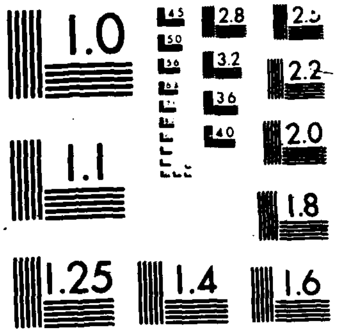
AD-A172 727   COMPUTATIONAL METHODS FOR COMPLEX FLOW FIELDS(U)   1/1
MASSACHUSETTS INST OF TECH CAMBRIDGE DEPT OF
AERONAUTICS AND ASTRONAUTICS E M MURMAN ET AL.
UNCLASSIFIED 28 JUN 86 AFOSR-TR-86-0868 AFOSR-82-0136   F/G 20/4   NL


1.0
4.5
5.0
5.6
6.3
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 86-0868

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE *(and Subtitle)*
1985 Annual Report
Computational Methods for Complex Flowfields

5. TYPE OF REPORT & PERIOD COVERED
Annual Report
June 1, 1984 - May 31, 1985

6. PERFORMING ORG. REPORT NUMBER

AUTHOR(s)
Earll M. Murman
Judson R. Baron

8. CONTRACT OR GRANT NUMBER(s)
AFOSR-82-0136

PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139

10. PROGRAM ELEMENT. PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F
2307/A1

CONTROLLING OFFICE NAME AND ADDRESS
AFOSR/NA

12. REPORT DATE
June 28, 1985

13. NUMBER OF PAGES

MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*
AFOSR
Bld 410
BAFB DC 20332

15. SECURITY CLASS. *(of this report)*
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
N.A.

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for Public Release. Distribution Unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*
Euler equations
Embedded grids
Adaptive grids
Airfoils
Computational Fluid Dynamics

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The development of solution algorithms for complex flowfields is the continuing objective of the research. The physical events are used to determine appropriate subdomains for both preselected and adaptive divisions of the field. Adapted embedding procedures have been completed for two-dimensional Euler flow with considerations of most suitable decision parameters. Development of an algorithm which combines cell and nodal-centered features continues for applications to less restrictive embedded topology. A mixed implicit/explicit approach for different coordinate directions is under study for Navier-Stokes flow. The Dec. 1984 Jerusalem Workshop is summarized briefly.

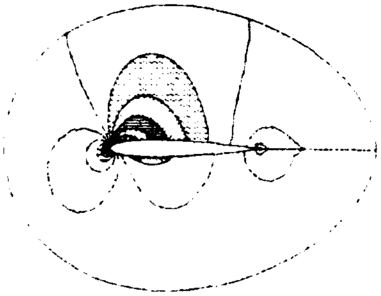
AFOSR-TR- 86-0868

1985 ANNUAL REPORT

COMPUTATIONAL METHODS FOR COMPLEX FLOWFIELDS

Earll M. Murman
Judson R. Baron

Principal Investigators



COMPUTATIONAL FLUID DYNAMICS LABORATORY
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

AFOSR-TR-85-

1985 ANNUAL REPORT

COMPUTATIONAL METHODS FOR COMPLEX FLOWFIELDS

Earll M. Murman
Judson R. Baron

Principal Investigators

AFOSR Grant 82-0136

OSP 95008

June 28, 1985

Computational Fluid Dynamics Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

TABLE OF CONTENTS

1. Introduction

The minimum requirement for the numerical evaluation of a flowfield is an adequate combination of a physical description, discrete grid distribution and procedural algorithm. All three must allow for events throughout the domain of interest. For realistic, complex flows this imposes varying demands on each since the disturbance field is generally nonuniform and frequently exhibits local concentrations where there are major changes in state properties. It is natural to attempt adjustment of either the physical description, the grid pattern and/or the algorithm itself so that in some sense there is a match of the available numerical power to those smaller subdomains and the specific kinds of behavior found there. Such considerations also appear on a global scale when choices are made of inviscid or viscous descriptors, or of grid generation techniques, and of methods that are specifically qualified for hyperbolic or other systems.

The CFD challenge is to devise ways in which appropriate subdivisions of a global domain may be selected and if need be altered, and how the resulting regions may be linked across their common interfaces during a computation. Efficient and robust algorithms must be expected to reduce both the computational speed and the memory requirements in proportion to the reduction in grid nodes and the simplicity of the description. Ideally, coarse grids and a minimum of physics are favored. Our focus is on procedures which will isolate and couple different portions of a complex domain such that in each subdomain the evaluation will disclose the details of the scale events to a sufficient and relevant accuracy.

2. Research Objectives and Tasks

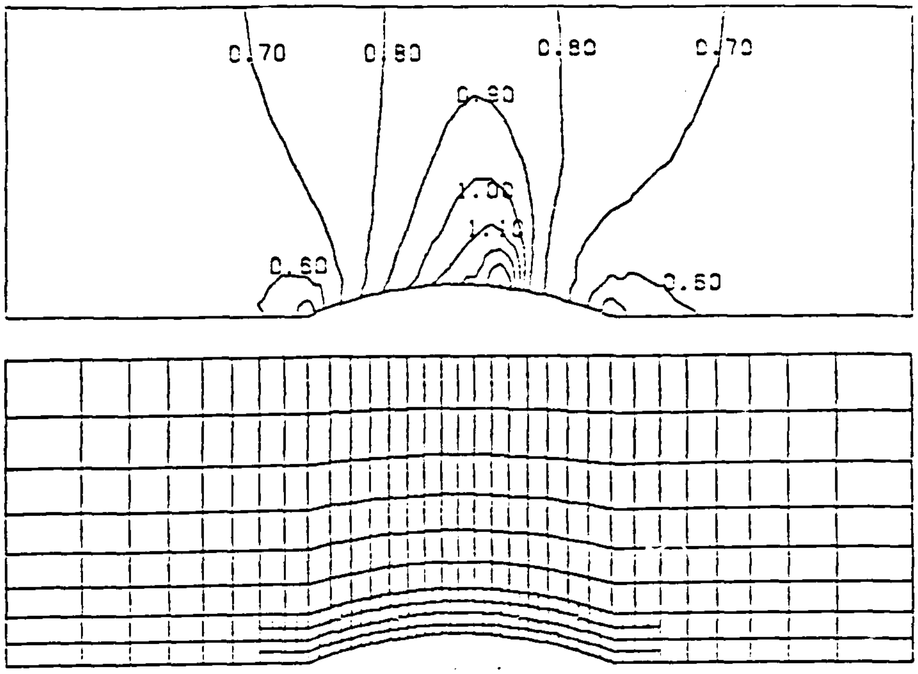
The specific purpose of the research is to develop procedures which will permit subdivision of the field and description, focus the computation on those portions which require a more precise evaluation, and control the coupling between the subdomains so as to maintain accuracy. Some problems have clear regions in which the dominant physics allows pre-selection of subdomains. We refer to this as a non-adaptive procedure in contrast to an adaptive approach in which the evolving events guide and change the choices for embedded regions.

The program's initial effort began with a determination of interface constraints and necessary adjustments of the background Ni (finite volume, multiple grid) method. Thereafter, threshold and decision parameters were developed for adaptation, and alternate algorithms have been considered in order to generalize the allowable mesh topology, the specific basic integrator, and the concept of embedding or zonal subdivision of an overall domain. Example evaluations have emphasized cascade and airfoil flowfields. A brief summary of the last year's efforts is described in Sections 3, 4 and 5, and some details for several tasks are included in attached Appendices.

3. Nonadaptive Embedded Subdomains

In last year's annual report research was reported on an algorithm which it is hoped will combine the best features of node-based methods such as Ni's and multistage methods such as Jameson's. Work has continued throughout this year, and good progress has been made in developing suitable damping and distribution operators. A detailed energy analysis was completed which identified requirements for these operators, and eventually led to suitable formulations for them. We refer the reader to Appendix A of last year's report for the basic outline of the algorithm. We have not included details of the further developments of this year, but instead show some results. Figure 3-1 shows a grid and Mach number plot for a transonic flow over a circular arc bump in a channel. The grid has an embedded region around the body. The computed results are in reasonable agreement with those of Dannenhoffer and Baron using Ni's method. Figure 3-2 shows supersonic flow past a 4% bump in a channel. The Mach number plots compare favorably with those of Ni for this problem. The basic algorithm seems to now be working. As yet we have not attempted a multigrid version which would significantly improve the rate of convergence.

However, we feel that it is now more important to attempt some calculations with this algorithm for which previous algorithms had difficulty. As discussed last year, this algorithm has been formulated without any assumption of regularity of the mesh or the number of dimensions. We have just started to try a calculation for a cylinder using the mesh shown in Figure 3-3. It should be noted that this is a very severe test of any algorithm for the Euler equations, and we might encounter significant problems. However, we think it is important to see how irregular a mesh

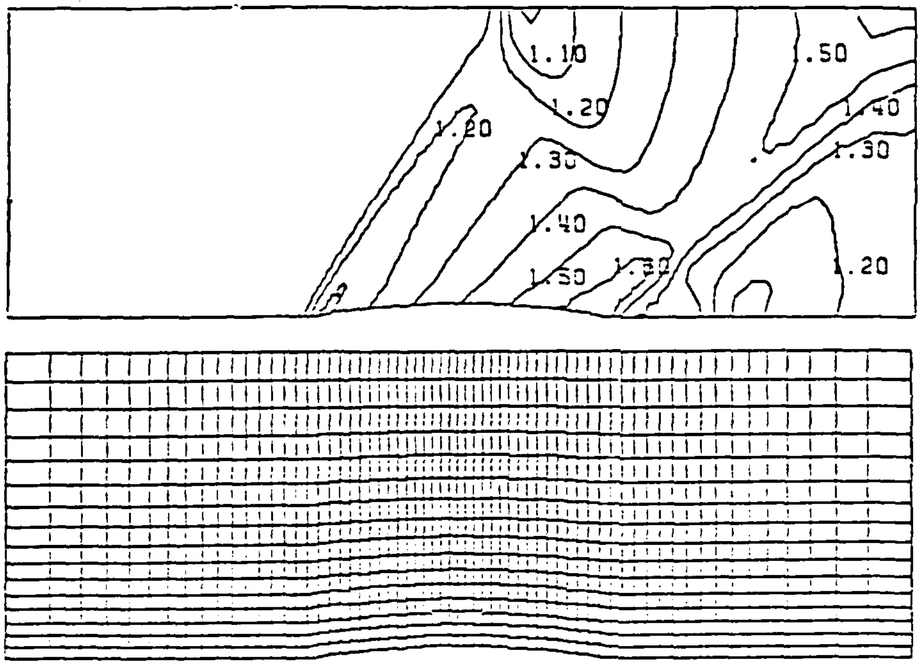


(a) Mach Number Distribution

Ni Bump
10% Blockage
Free Stream Mach
Number = 0.675

(b) Singly Embedded Grid

Figure 3-1 Transonic Channel Flow



(a) Mach Number Distribution

Ni Bump
10% Blockage
Free Stream Mach
Number = 1.40

(b) Grid

Figure 3-2 Supersonic Channel Flow

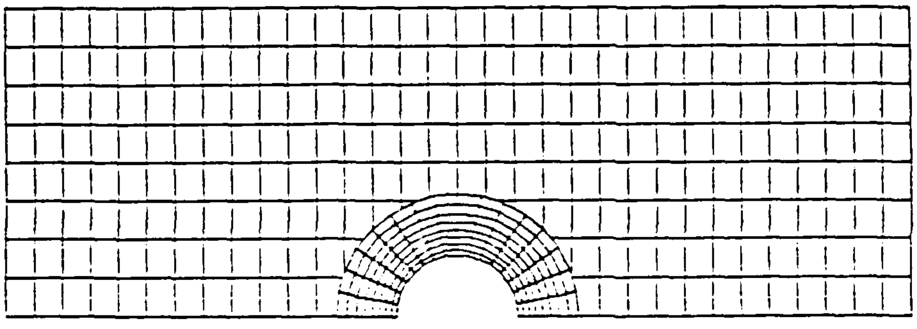


Figure 3-3 Patched Grid

can really be handled in order to assess the feasibility of a general grid approach. The current algorithm is written to be suitable for vector or parallel processors.

Approach which has less technical risk and may prove to be more practical than the above is the zonal approach. The flowfield is divided into different zones, and within each zone a standard algorithm of some type may be employed. A zone could have an Euler method on a regular grid and an adaptive method (within that zone), a Navier-Stokes method, or many other of a variety of suitable approaches. The important feature is that the zones communicate through the boundary data. Some thought has been given this year as to how such a zonal method should be organized to be suitable for multiprocessor architectures. A general strategy has been conceived, but not yet developed in detail nor tested. This will be done during the first six months of the next grant year.

As part of the zonal approach, an idea for a more efficient Navier-Stokes algorithm has been conceived. It is based upon the observation that Navier-Stokes approaches generally require very fine grid spacing normal to the body when compared to the other two directions. This leads to a stiff set of discrete equations, which then requires an implicit method for solution. The standard approach is to use a fully implicit method which requires approximate factorization (e.g. Beam-Warming). A more attractive approach would be to use an implicit method only for the differences in the body normal direction, and explicit approaches for the other two directions. This algorithm would require a block tridiagonal inversion, but no factorization. A stability analysis is given in Appendix A for a model problem, and the results look very promising. We will pursue this further in the next grant year.

4. Adaptive Embedded Subdomains

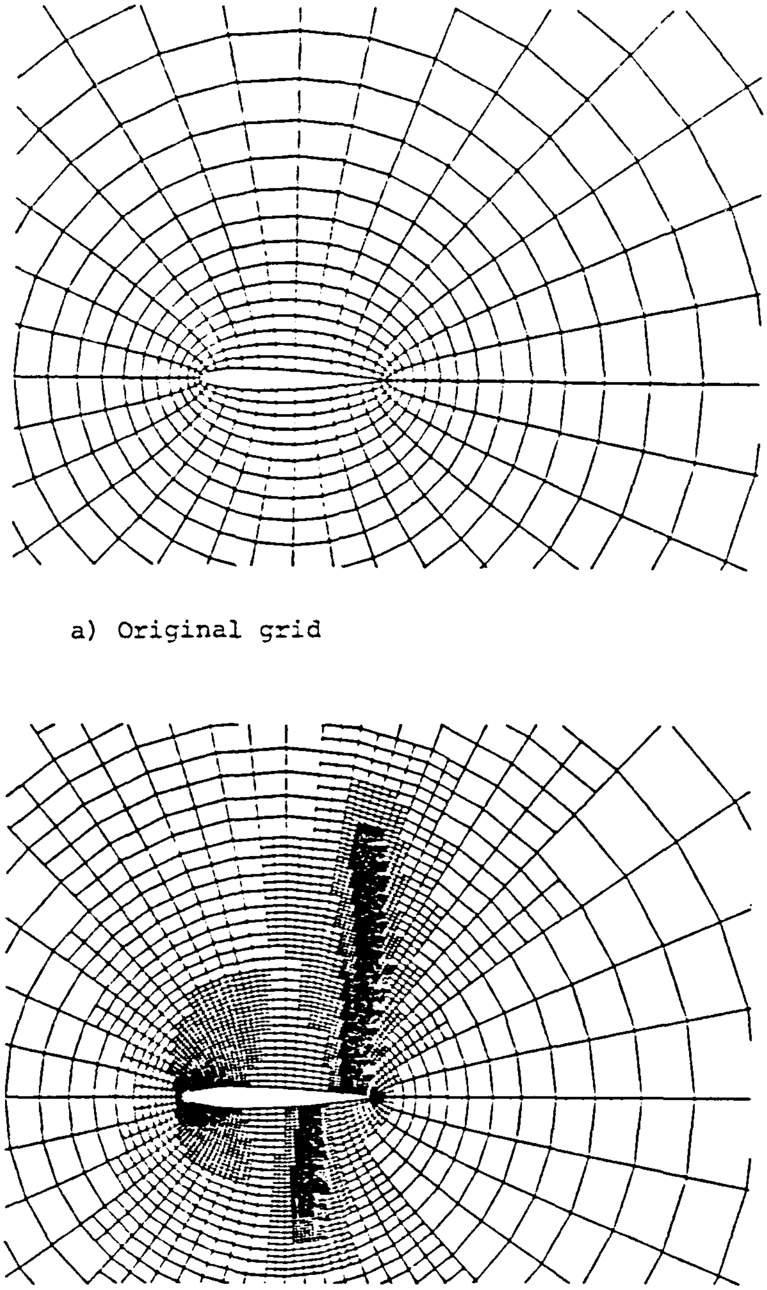
The goal of the adaptive concept segment of the effort has been to minimize computational work by appropriately locating, controlling and managing the embedded fine grid regions. The details involve procedures which recognize flow features as they develop, decide on adequate thresholds for adaptation to be carried out, and modify the discreteness formulations in order to proceed with irregular and multiple embedded grids.

During the first part of the last year the initial applications to two-dimensional airfoils were completed. Those calculations included a sensitivity study of the influence of different refinement parameters and their rates of change as the basis for adaptation decisions. Primitive parameters such as density, pressure, velocity and entropy were used as threshold selectors on the basis of their first and second difference distributions throughout the domain. Field evaluations for NACA 0012 and RAE 2822 lifting airfoils at transonic speeds indicated density first differences to be most efficient and therefore preferable in defining subdomains with dominant activity. This extension of the work to two-dimensional fields was reported on in AIAA Paper 85-0484 and is included here as Appendix C.

The effort has continued with a further development of the algorithm so as to introduce a capability for additional self evaluation. Previously consideration was given to the type of feature expected and an arbitrary but fixed number of adaptive levels. A new strategy scheme has been developed which allows for automatic selection of the number of embedded regions, in the multiple sense of sub-embedding as well as for distinct features. A rule based system monitors the progress and measures the CFD needs against such criteria as the changes in a global parameter of interest. A number of test cases have been completed for airfoil configurations that have been considered by the AGARD Working Group 07. Others are planned.

Some preliminary results are shown in Figures 4-1 through 4-4. The initial and final grids (convergence decided five adaptations) for the NACA 0012 airfoil at $M = 0.85$ and 1.0° incidence [AGARD WG 07 Test Case 2] appear in Figure 4-1. Those finest embedded regions which extend outward from the upper and lower surfaces indicate shock locations. The adaptation process includes an allowance for growth of all embedded regions during the final stages of an evaluation. Note that a non-adapted grid which is globally refined in order to attain the same feature resolution would contain a factor of 25 times more nodes.

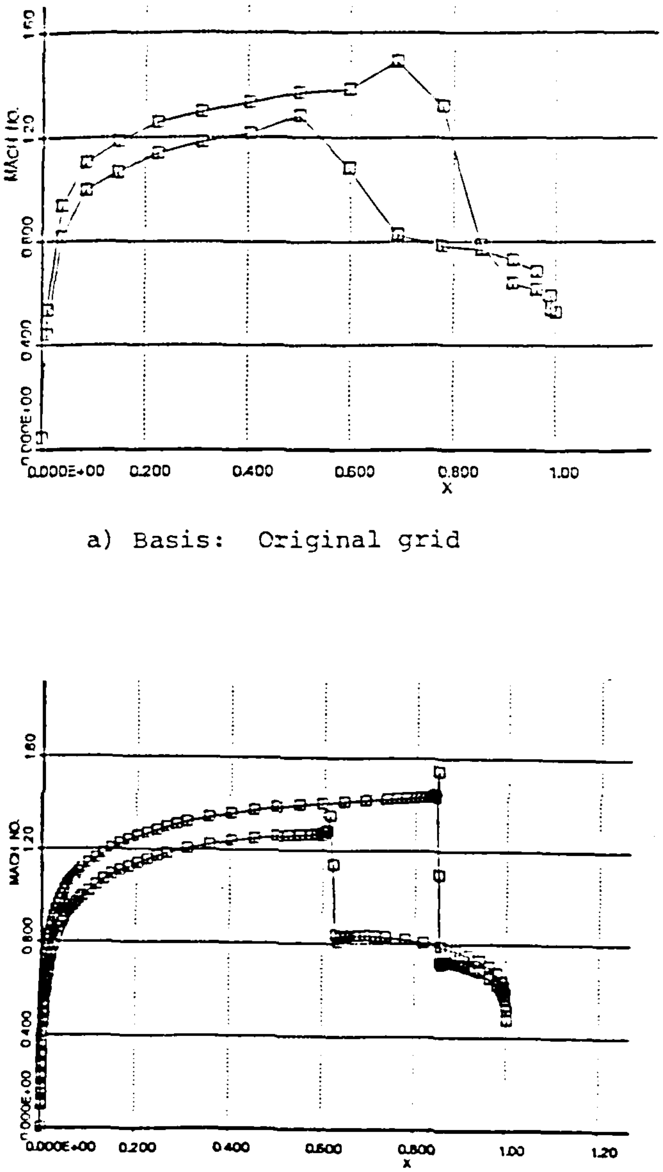
The surface Mach number distributions corresponding to the grids in Figure 4-1 are shown in Figure 4-2. The Mach number contours in the field and convergence histories for lift and drag coefficients are in Figure 4-3. Lastly, results for $M = 0.95$ are shown in Figure 4-4. Supersonic ($M = 1.2$)



a) Original grid

b) Fifth (final) grid adaptation

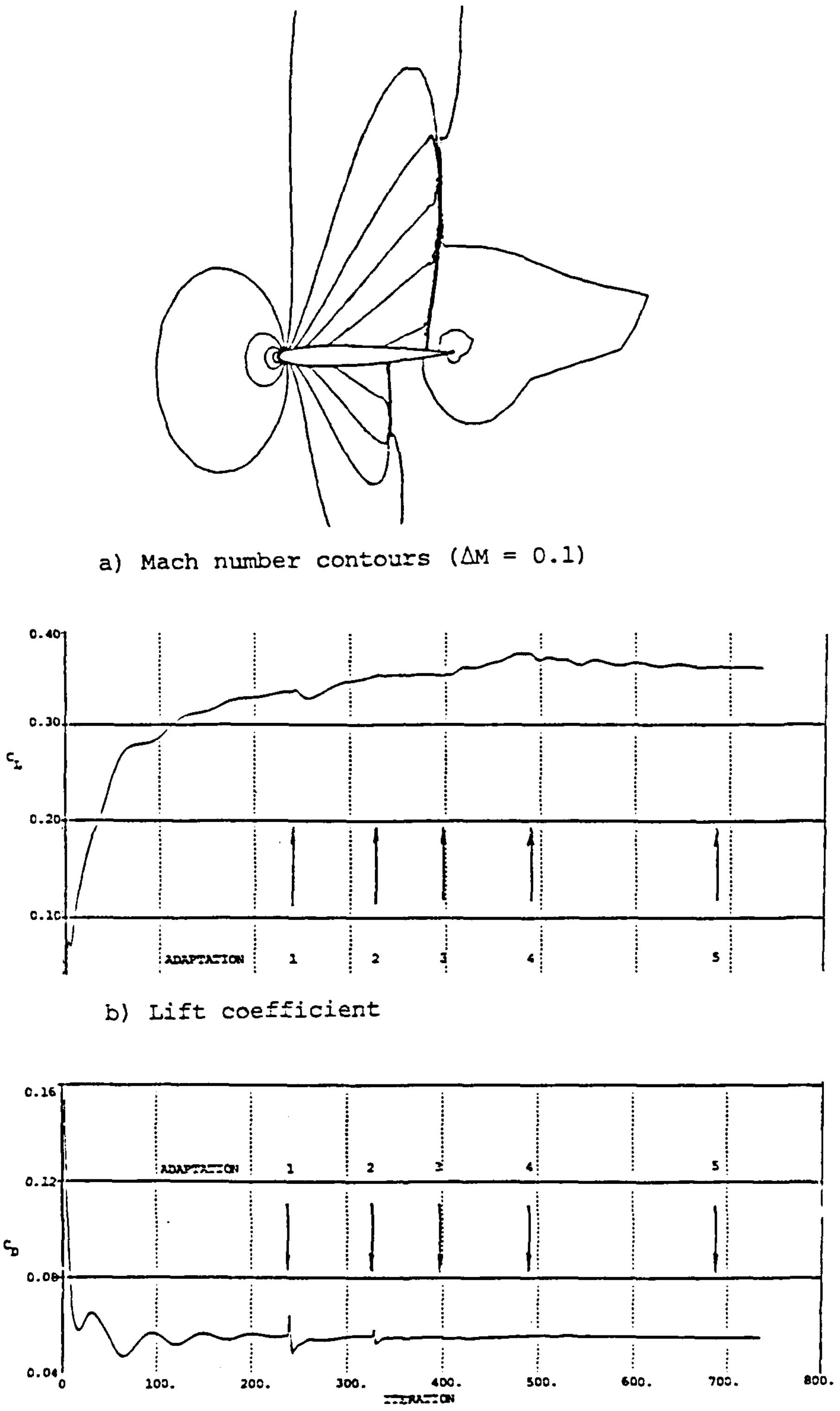
Fig. 4-1 Solution grids
NACA 0012, $M = 0.85$, $\alpha = 1.0°$



a) Basis: Original grid

b) Basis: Fifth (final) grid adaptation

Fig. 4-2 Surface Mach number distributions
NACA 0012, M = 0.85, α = 1.0°

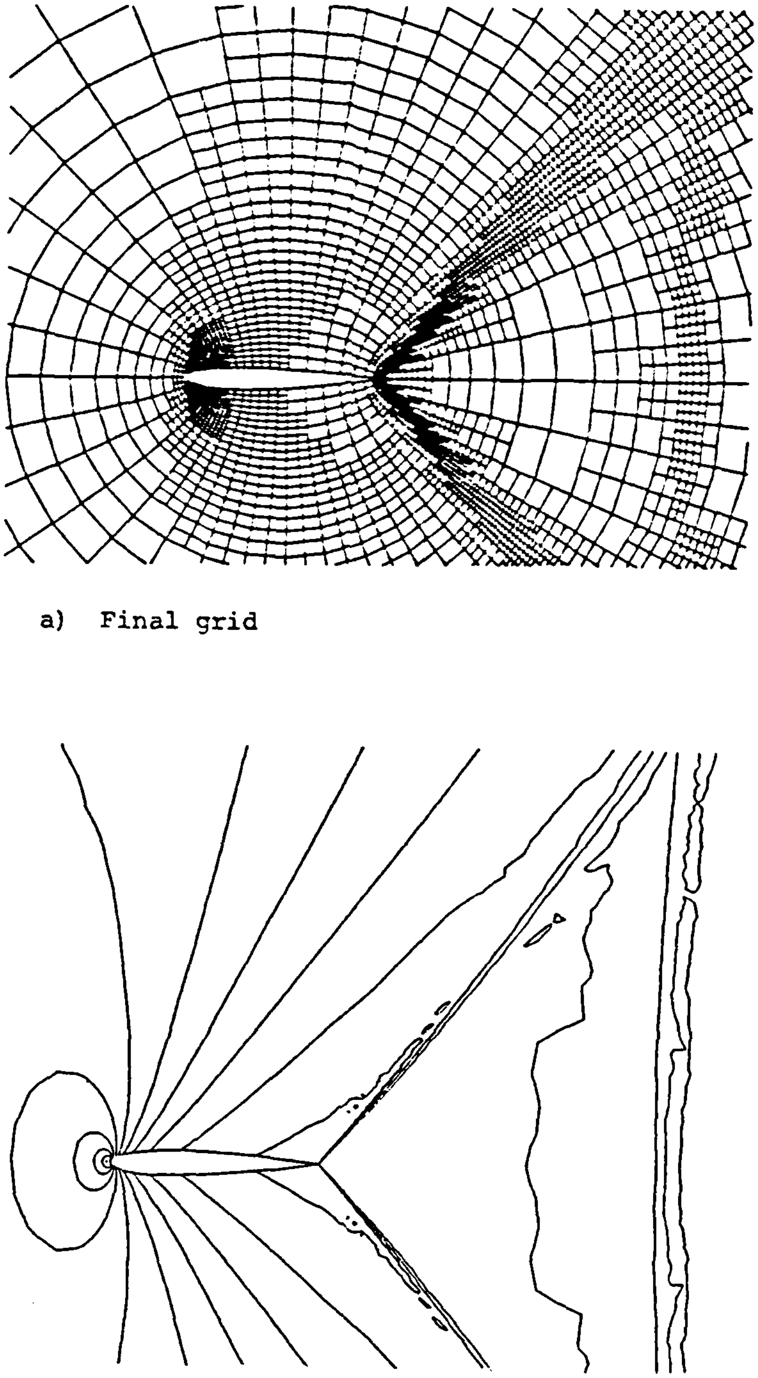


a) Mach number contours ($\Delta M = 0.1$)

b) Lift coefficient

c) Drag coefficient

Fig. 4-3 M contours and global parameter convergence
NACA 0012, $M = 0.85$, $\alpha = 1.0°$



a) Final grid

b) Mach number contours

Fig. 4-4 Adaptive solution
NACA 0012, M = 0.95, α = 0.°

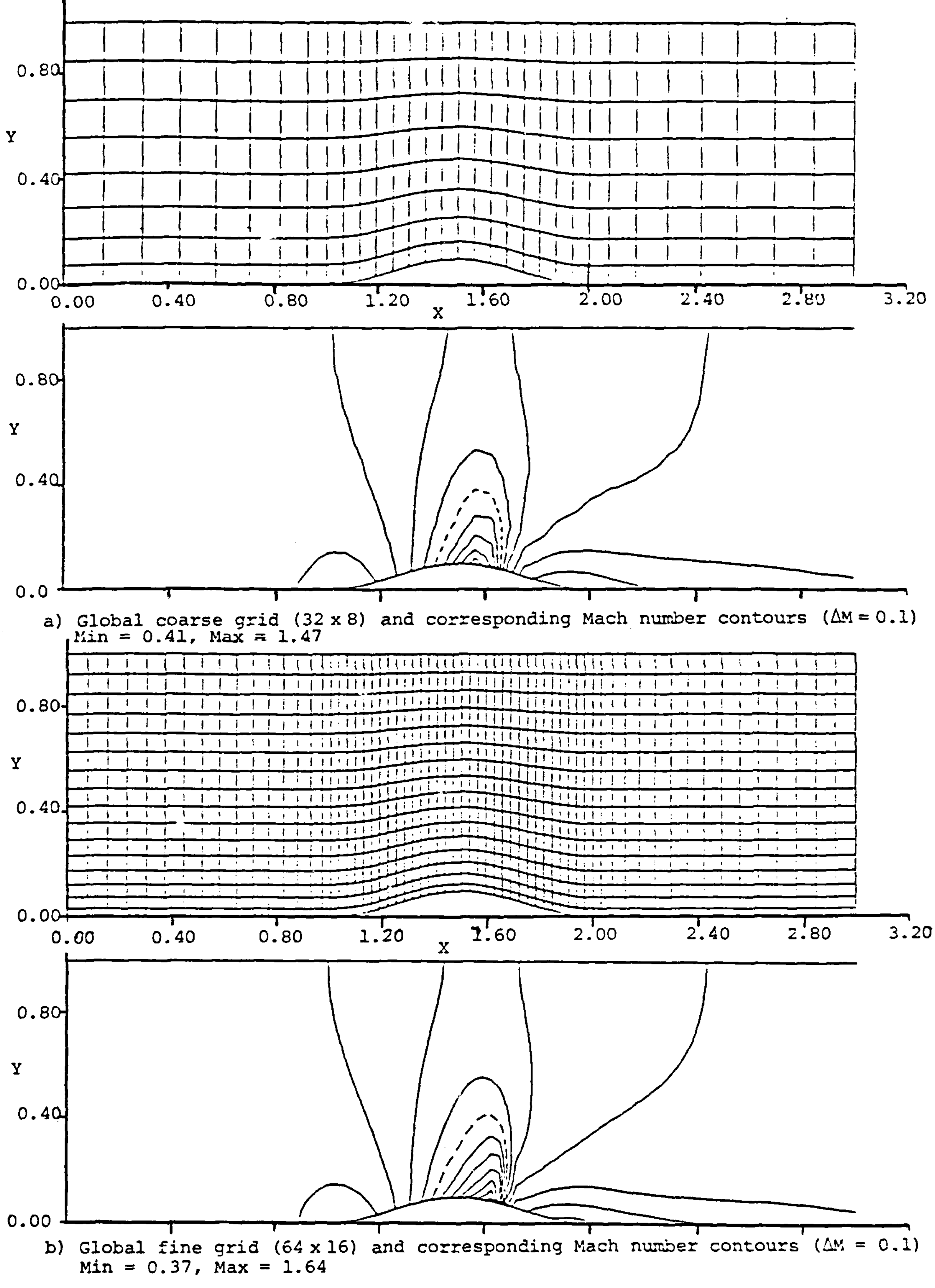
cases with detached bow and trailing edge fishtail shocks also have been completed. The algorithm is robust and accurate with virtually no discernible effect due to the sub-grid interfaces (see Figure 4-2). Generally, an order of magnitude reduction in computing time is achieved by introducing adaptation in place of global refinement.

A vectorized version of the adaptive code is currently being developed and tested. The Cyber 205 at the NASA Langley Research Center is being used for that purpose. The intent is to complete both globally refined and adaptive solutions so as to provide precise comparisons as well as a demonstration of the suitability for vectorization. In that regard the data structure associated with the adaptation procedures appears to reduce the benefits of vectorization only slightly. A paper is in preparation to describe some of the above findings.

All of the above has continued the use of the Ni method as the basic integrator for the algorithm. A separate study was undertaken to consider the embedding technique with an alternative finite volume scheme [Jameson, Schmidt and Turkel, AIAA 81-1259]. The motivation is a more efficient time marching procedure, but the focus of the study is on the interface constraints and the achievement of conservative, accurate and smooth solutions in the presence of abrupt mesh scale changes. Figure 4-5 shows grids and Mach number contours for coarse, embedded and globally fine cases of transonic flow ($M = 0.675$) past a typical channel bump (10%) configuration. To date analysis reveals that second order accurate, conservative flux formulations are not possible at interfaces. Smoothing also implies nonconservative contributions. Three classes of interface formulations are suggested: zeroth order accurate and fully conservative, first order accurate for conservative fluxes and nonconservative smoothing, and second order accurate but fully nonconservative. The Mach number contours in Figure 4-5 illustrate the advantages of global refinement. The interface effects are also apparent along the superimposed interface locus for the embedded case. This work with the Jameson scheme is continuing.

5. Jerusalem Meeting

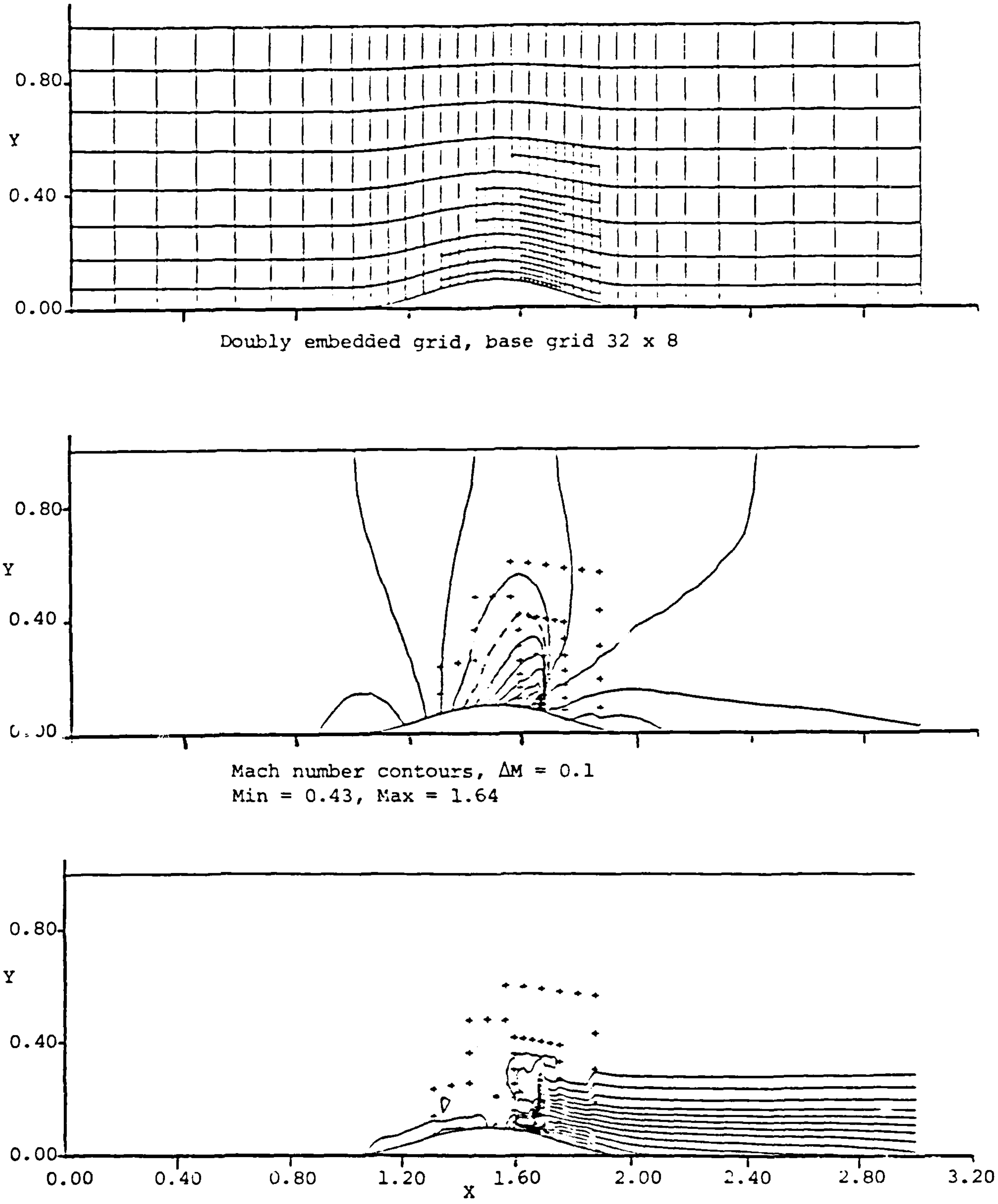
A workshop was held in Jerusalem, Israel in December 1984 entitled "The Impact of Supercomputers on the Next Decade of CFD." The participation of



a) Global coarse grid (32 x 8) and corresponding Mach number contours ($\Delta M = 0.1$) Min = 0.41, Max = 1.47

b) Global fine grid (64 x 16) and corresponding Mach number contours ($\Delta M = 0.1$) Min = 0.37, Max = 1.64

Fig. 4-5 Transonic channel flow, M = 0.675, 10% thick bump



Doubly embedded grid, base grid 32 x 8

Mach number contours, ΔM = 0.1
Min = 0.43, Max = 1.64

Total pressure loss contours, Δ = 0.01

Fig. 4-5 (concluded)

Professor Murman, one of the organizers of the workshop, was sponsored by this grant. Professors Murman and Abarbanel have prepared a summary report of the workshop which is attached as Appendix C. A number of important issues relative to this grant were brought out at the workshop and are recorded in Appendix C. Perhaps the most important are the observations that current algorithms cannot simply be scaled up to 256 megaword machines. They then will converge so slowly that the procedure will be impractical. Embedded and adaptive mesh approaches look very favorable as a way to proceed.

6. Cumulative List of Publications

1. Usab, W.J. and Murman, E.M., "Embedded Mesh Solutions of the Euler Equation Using A Multiple Grid Method." AIAA Paper 83-1946 CP, July 1983. Also in RECENT ADVANCES IN NUMERICAL METHODS IN FLUIDS, Vol. 3, Editor W.G. Habashi, Pineridge Press.

2. Usab, W.J., "Embedded Mesh Solutions of the Euler Equations Using A Multiple Grid Method." MIT PhD Thesis, January 1984. Also MIT CFDL-TR-84-2, May 1984.

3. Dannenhoffer, J.F. and Baron, J.R., "Adaptive Solution Procedure for Steady State Solution of Hyperbolic Equations." AIAA Paper 84-0005, January 1984.

4. Dannenhoffer, J.F. and Baron, J.R., "Grid Adaptation for the 2-D Euler Equations." AIAA Paper 85-0484, January 1985.

5. Dannenhoffer, J.F. and Baron, J.R., "Robust Grid Adaptation for Complex Transonic Flows." Submitted to the AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 1986.

6. Allmaras, S.R. and Baron, J.R., "Embedded Mesh Solutions of 2-D Euler Equations: Evaluation of Interface Formulations." Submitted to the AIAA 24th Aerospace Sciences Meeting, Reno, Nevada, January 1986.

7. Professional Personnel Associated with Research Effort

Principal Investigators
Earll M. Murman, Professor
Judson R. Baron, Professor

Others
Saul S. Abarbanel, Senior Lecturer, MIT; Professor, Tel-Aviv University
William J. Usab, Jr., PhD Candidate (degree awarded February 1984)
John F. Dannenhoffer III, PhD Candidate
Gregory L. Larson, PhD Candidate
Steven R. Allmaras, SM Candidate

8. Interactions

- AIAA Paper 85-0484 presented at AIAA 23rd Aerospace Sciences Meeting, Reno, Nevada, January 1985.
- Seminar on Adaptive Grid Algorithm presented by Baron at McDonnell Aircraft Company, St. Louis, MO, April 1985.
- Conferences with Dr. Joseph Shang, FDL, WPAFB.

9. New Discoveries

None

(Appendix A)

Stability Analysis of a Semi-Implicit Scheme for Hyperbolic Equations

Saul S. Abarbanel

Consider the 2-D linearized inviscid Burger's equation:

$$u_t = u_x + u_y \tag{1.1}$$

We shall consider two finite-difference schemes: an explicit Runga-Kutta three-stage scheme and a semi-implicit version of the same. The stability of both algorithms is analyzed.

1. Explicit Scheme

Using standard notation eq(1.1) has the following finite difference representation which is second order accurate in space and first order accurate temporally:

$$u^0_{j,k} = u^n_{j,k} \tag{1.2}$$

$$u^1_{j,k} = u^0_{j,k} + \Delta t(\delta_x + \delta_y)u^0_{j,k} \tag{1.3}$$

$$u^2_{j,k} = u^0_{j,k} + \Delta t(\delta_x + \delta_y)u^1_{j,k} \tag{1.4}$$

$$u^3_{j,k} = u^0_{j,k} + \Delta t(\delta_x + \delta_y)u^2_{j,k} \tag{1.5}$$

$$u^{n+1}_{j,k} = u^3_{j,k} \tag{1.6}$$

where

$$\delta_x u^n_{j,k} = \frac{u^n_{j+1,k} - u^n_{j-1,k}}{2\Delta x} \tag{1.7}$$

and

$$\delta_y u^n_{j,k} = \frac{u^n_{j,k+1} - u^n_{j,k-1}}{2\Delta y} \tag{1.8}$$

"Telescoping" the stages (1.3)-(1.5) we get the following expression for $u^{n+1}_{j,k}$:

$$u^{n+1}_{j,k} = u^n_{j,k} + \left\{\Delta t(\delta_x + \delta_y) + (\Delta t)^2(\delta_x + \delta_y)^2 + (\Delta t)^3(\delta_x + \delta_y)^3\right\}u^n_{j,k} \tag{1.8}$$

We Fourier decompose $u^n_{j,k}$ in the usual manner: $u^n_{j,k} = \hat{u}^n e^{ij\theta} e^{ik\phi}$.

Then the Fourier representation of (1.8) becomes:

$$\hat{u}^{n+1} = \left\{1 + i(\lambda_x \sin\theta + \lambda_y \sin\phi) - (\lambda_x \sin\theta + \lambda_y \sin\phi)^2 - i(\lambda_x \sin\theta + \lambda_y \sin\phi)^3\right\}\hat{u}^n$$

where $\lambda_x = \Delta t/\Delta x$ and $\lambda_y = \Delta t/\Delta y$.

Define $z = \lambda_x \sin\theta + \lambda_y \sin\phi$; the amplification factor, $G = \hat{u}^{n+1}/\hat{u}^n$ then becomes:

$$G = 1 + iz - z^2 - iz^3 = (1 - z^2)(i + iz) \tag{1.9}$$

and therefore

$$|G|^2 = (1 - z^2)^2(1 + z^2) = 1 - z^2 - z^4 + z^6 \tag{1.10}$$

The Von Neumann stability requirement $|G| \leq 1$, then leads to the following inequality:

$$z^4 - z^2 - 1 \leq 0 , \tag{1.11}$$

leading to:

$$z \leq \left(\frac{1 + \sqrt{5}}{2}\right)^{1/2} \simeq 1.27202 \tag{1.12}$$

Recall now that $z = \lambda_x \sin\theta + \lambda_y \sin\phi$, and requirement (1.12) takes the form

$$\frac{\Delta t}{\Delta x} + \frac{\Delta t}{\Delta y} = \lambda_x + \lambda_y \leq 1.2720 \tag{1.13}$$

or, upon rearranging

$$\Delta t \leq 1.2720 \frac{\Delta x \cdot \Delta y}{\Delta x + \Delta y} \tag{1.14}$$

The stability condition, (1.14), clearly shows that if in the calculation we choose the time step accordingly, i.e.

$$\Delta t \leq \min_{j,k} \left(1.2720 \frac{\Delta x_{j,k} \cdot \Delta y_{j,k}}{\Delta x_{j,k} + \Delta y_{j,k}} \right) \tag{1.15}$$

we may get very severe restriction if the mesh is highly stretched so that somewhere in the field $(\Delta y/\Delta x)_{j,k}$, say, is very small.

To overcome this difficulty without paying an inordinately high price of two matrix inversions per stage we propose a semi-implicit algorithm.

2. Semi-Implicit Scheme

Using the previous notation, the semi-implicit 3-stage Runga-Kutta algorithm takes the form:

$$u^0_{j,k} = u^n_{j,k} \tag{2.1}$$

$$(1 - \Delta t\delta y)u^1_{j,k} = u^0_{j,k} + \Delta t\delta_x u^0_{j,k} \tag{2.2}$$

$$(1 - \Delta t\delta_y)u^{2}_{j,k} = u^{0}_{j,k} + \Delta t\delta_x u^{1}_{j,k} \tag{2.3}$$

$$(1 - \Delta t\delta_y)u^{3}_{j,k} = u^{0}_{j,k} + \Delta t\delta_x u^{2}_{j,k} \tag{2.4}$$

$$u^{n+1}_{j,k} = u^{3}_{j,k} \tag{2.5}$$

Again "telescoping," and joined directly to Fourier space, the amplification factor is found to be:

$$G = \frac{1 - \lambda_x^2\xi^2 - \lambda_y^2\eta^2 + \lambda_x\lambda_y\xi\eta + i\lambda_x\xi - 2i\lambda_y\eta - i\lambda_x^3\xi^3}{(1 - i\lambda_y\eta)^3} \tag{2.6}$$

where $\xi = \sin\theta$, $\eta = \sin\phi$; i.e. $-1 \leq \xi \quad \eta \leq 1$. The Von Neumann stability requirement leads to the following inequality:

$$-(x+y)^2 - (x^2-y^2)^2 + y^2(x^2-y^2) + 2xy(x^2-y^2) + (x^2-y^2)(x^4+x^2y^2+y^4) \leq 0 \tag{2.7}$$

where $x = \lambda_x\xi$ and $y = \lambda_y\eta$; i.e. $-\infty < \quad < \infty$.

We carry out the investigation of (2.7) by first considering the case of $y^2 > x^2$. Divide (2.7) by $y^2 - x^2 > 0$ to get

$$-\frac{(y+x)^2}{y^2-x^2} - (y^2-x^2) - y^2 - 2xy - x^4 - x^2y^2 - y^4 \leq 0 \tag{2.8}$$

The worst case is that of $xy < 0$. Let $x \rightarrow -x$, $y \rightarrow y$, then the new x,y satisfy $xy > 0$ and the inequality:

$$-\frac{(y-x)^2}{y^2-x^2} - 2y^2 + x^2 + 2xy - x^4 - x^2y^2 - y^4 \leq 0 \tag{2.9}$$

Note that in (2.9) it does not matter (with $y^2 > x^2$ and $xy > 0$) whether x and y are both negative or positive. We shall, therefore, consider the $x > 0$, $y > 0$ case. We rewrite (2.9) slightly:

$$-\frac{y-x}{y+x} + x^2 - 2y^2 + 2xy - x^4 - x^2y^2 - y^4 \leq -\frac{y-x}{y+x} + x^2 - x^4$$
$$\leq -1 + x^2 - x^4 \quad 0 \tag{2.10}$$

The inequality (2.10) is <u>always satisfied</u>. Therefore, we have shown that (2.7) is satisfied when $y^2 > x^2$ for all $-\infty < x,y < \infty$. Notice also, directly from (2.7) that when $y = \pm x$ the inequality is satisfied (if $y = -x$ we have the equality). Therefore, (2.7) is satisfied for all $\underline{y^2 > x^2}$, $|x|,|y| < \infty$.

Next we consider the case $x^2 > y^2$. We see from (2.7) that now the worst case is that of $xy > 0$. Again x and y can both be either negative or positive without changing the left hand side of (2.7). We shall use $x > y$; $x > 0$, $y > 0$. Divide (2.7) $x^2 - y^2 > 0$:

$$- \frac{(x+y)^2}{x^2 - y^2} - (x^2 - y^2) + y^2 + 2xy + x^4 + x^2y^2 + y^4 \leq 0 \qquad (2.11)$$

Since $x > y$, take $y = \beta x$, $(0 < \beta < 1)$. Inequality (2.11) becomes:

$$- \frac{1+\beta}{1-\beta} + (2\beta^2 + 2\beta - 1)x^2 + (\beta^4 + \beta^2 + 1)x^4 \leq 0$$

or, with $w = x^2$,

$$f(w_j\beta) \equiv (\beta^4 + \beta^2 + 1)w^2 + (2\beta^2 + 2\beta - 1)w - \frac{1+\beta}{1-\beta} \leq 0 \qquad (2.12)$$

The smallest w, <u>for all $0 < \beta < 1$</u>, that solves $(w_j\beta) = 0$ is <u>$w = 4/3$</u> for <u>$\beta = 1/2$</u>. This is the stability criterion. Thus we have

$$\lambda_x|\xi| \leq (4/3)^{1/2}$$

and so the <u>stability condition</u> becomes $\Delta t/\Delta x = \lambda_x < 2/\sqrt{3}$, or

$$\Delta t \leq 1.1547 \left(\min_{j,k} \Delta x_{j,k}\right) \qquad (2.13)$$

Notice that now we have a conditional stability that does not depend on the mesh stretching. This condition is still superior to (1.15) even when $\Delta x_{j,k} = \Delta y_{j,i}$ everywhere, because then (1.15) yields $\Delta t \leq 0.636 \left(\min_{j,k} \Delta x_{j,k}\right)$.

AIAA'85

Appendix B

AIAA-85-0484

GRID ADAPTATION FOR THE 2-D EULER EQUATIONS

JOHN F. DANNENHOFFER, III
JUDSON R. BARON

Massachusetts Institute of Technology
Cambridge, MA

AIAA 23rd Aerospace Sciences Meeting
January 14-17, 1985/Reno, Nevada

AIAA-85-0484

GRIL ADAPTATION FOR THE 2-D EULER EQUATIONS

John F. Dannenhoffer, III *
Judson R. Baron **

Computational Fluid Dynamics Laboratory
Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

An adapted grid technique for accurate and efficient solutions of the steady, two-dimensional Euler equations is presented. Embedded meshes, which are coupled to a fixed global mesh via a multiple-grid technique, are employed. The effect of choosing various adaptation strategies is examined. Solutions are presented for isolated lifting airfoils in subcritical as well as transonic flow. The supersonic flow over a circular-arc cascade with a complex shock structure is also presented. In all cases, the adapted solution achieved the same accuracy as global refinement, but required a factor of between 5 and 7 less computer time and between 3 and 8 less storage.

INTRODUCTION

The numerical solution of the two-dimensional Euler equations is typically carried out by discretizing the flow field and then solving a coupled set of equations for each node. This technique suffers from the introduction of errors due to the discrete approximation of the continuous governing equations. These errors are related to the local mesh spacing and the local solution smoothness.

To obtain more accurate results, one can decrease the mesh spacing. But since the domain size is fixed, this results in more required grid points, with the associated increase in computational work and required storage. It has been noted by many prior researchers [1] that for the Euler equations applied to problems of interest, much of the flow field is smooth, resulting in small errors. There are however isolated areas in the flow field where the solution is not smooth, resulting in appreciably larger errors. Thus, it is an efficient strategy to refine the mesh only where necessary.

Unfortunately, one does not in general know _a priori_ where these unsmooth regions are, and thus it often takes a few trials and errors to put the dense mesh where it is really required. To circumvent this problem, adaptive grid techniques have been developed to search for regions of high errors, and then refine the mesh there. Thus, the solution and the grid evolve simultaneously.

There are currently two methods for adapting grids. The first (and most popular) is the grid point redistribution scheme which has been reviewed extensively by Thompson [1]. In this technique, there is a fixed number of grid points which move around, congregating in regions of interest. It is reasonably easy to program, since at every iteration (or at least every few iterations), some statistic of the flow field is sensed (for example entropy gradient), and the grid points move either by an attraction model or by using a Poisson equation grid generator. Redistribution however has a few drawbacks. First, since there ia a fixed number of grid points, moving them toward flow features deprives other flow regions of adequate resolution. Thus, one could say that the solution becomes both good uniformly and at the same time uniformly bad. Second, due to topological restrictions, grid lines which properly congregate near a shock at an airfoil surface must propagate out into the flow field, often to the far field boundary where the increased resolution is unnecessary. Thompson also points out that excessive grid skewing and stretching can occur, producing inherent inaccuracies.

The second grid adaptation technique involves locally embedding sub-grids in regions of interest. This results in an increase in the number of grid points, and hence an increase in required computational resources. Also, artificial internal boundaries are created in the flow field; care must be taken to ensure that the solution will smoothly transfer across such an interface, while maintaining conservation and stability. Since global grid points are never moved, the accuracy is not reduced away from flow features as in redistribution methods.

* Research Assistant, Member AIAA
** Professor, Associate Fellow AIAA

Berger and Jameson [2] have used local grid embedding for the two-dimensional Euler equations. In their approach, the flow features are detected and "best-fit" rectangles are superimposed over the global grid. This results in a mesh interface across which it is very difficult to enforce conservation. In addition, determination of the best-fit rectangle requires complex clustering algorithms.

An alternative embedded mesh procedure has been proposed by Dannenhoffer and Baron [3]. In this approach, irregularly shaped embedded regions which are topologically connected to the global grid are used. This results in both a simpler embedded mesh interface as well as a simpler embedding scheme which doesn't require clustering.

In reference [3], the authors discussed their technique for the one-dimensional Euler equation and two-dimensional Burgers equation. The extension of their technique to the two-dimensional Euler equations is presented here.

This paper first discusses the overall adaptation strategy. The governing equations are presented and the embedded mesh algorithm is described. Included in that section are discussions of the embedded mesh interface as well as the adapted smoothing algorithm employed. Appropriate choices for a refinement parameter are discussed in the next section. Computed results for the Euler flow over airfoils and a model problem with complicated shock topology complete the discussion.

ADAPTIVE SOLUTION ALGORITHM

The basic approach of the adapted solution procedure described here is to use a fixed global grid and then embed irregularly shaped grids where necessary. This is accomplished by solving initially only on the global grid. When quasi-convergence is reached, a refinement parameter is computed at each node and any cells which are connected to nodes where the refinement parameter is above some threshold are divided. The program then reintegrates on the new grid (global and embedded, coupled by a multiple-grid scheme). After a prespecified number of adaptations (typically 2), the program iterates to the final converged solution.

The integration of the governing equation is performed using Ni's multiple-grid algorithm [4]. This scheme is composed of two parts -- a finite-volume form of a standard single-step, Lax-Wendroff integration applied on a fine mesh, and a coarse grid accelerator which operates on residuals transported from the fine mesh solver. In both parts, a "change" is computed at the center of each cell and then transferred to the adjacent nodes by means of "distribution formulae".

The use of irregularly shaped embedded regions introduces a bookkeeping problem which must be addressed. Since Ni's scheme is cell-based, each cell can be integrated independently which is an important consideration in the choice of this method. Cells communicate with each other only through the variables at the shared nodes from the previous explicit pseudo-time step. A data structure which is based on that property was first introduced by Usab and Murman [5], a variation of which is used in the current program.

GOVERNING EQUATIONS

The unsteady, two-dimensional Euler equations in conservation law form are given by

$$\frac{\partial U}{\partial t} = -\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y} \qquad (1)$$

where

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ E \end{bmatrix} \qquad F = \begin{bmatrix} \rho u \\ \rho uu+p \\ \rho uv \\ \rho uh_0 \end{bmatrix} \qquad G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho vv+p \\ \rho vh_0 \end{bmatrix}$$

and

$$h_0 = \frac{E+p}{\rho} = \frac{\gamma}{\gamma-1}\frac{p}{\rho} + \frac{1}{2}\left(u^2+v^2\right)$$

In equation (1), ρ is the density, u and v are the velocity components in the x- and y-directions, E is the total internal energy per unit volume, p is the pressure, h_0 is the total enthalpy, and γ is the ratio of specific heats.

TIME MARCHING PROCEDURE

Basic Scheme

Consider the fine mesh cell shown in figure 1. To calculate the "change" in the dependent variables at the center of this cell, the divergence theorem is applied to the governing equations, giving

$$\Delta U = \frac{\Delta t}{\Delta V}\left(\bar{F}_w - \bar{F}_e + \bar{F}_s - \bar{F}_n\right) \qquad (2)$$

where $\bar{F}$ denotes the contravariant flux through each given face, Δt denotes the pseudo-time step, and ΔV the cell volume. The contravariant fluxes are computed by trapezoidal integration along each cell face, based on the dependent variables at the corner nodes. Equation (2) is called the cell flux balance.

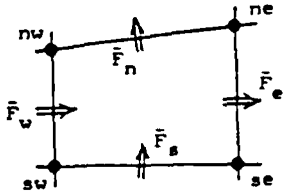


Figure 1. Typical computational cell.

The distribution formulae serve to transfer the "change" (ΔU) from the center of the cell to the four corner nodes. The formulae are derived from a second-order-accurate Taylor series expansion of U (with respect to time), and are given by

$$\delta U_{\substack{SW\\SE\\NE\\NW}} = \frac{1}{4}\left(\Delta U \substack{-\\+\\+\\-} \frac{\Delta t}{\Delta x}\Delta F \substack{-\\-\\+\\+} \frac{\Delta t}{\Delta y}\Delta G \right) \qquad (3)$$

where

$$\Delta F = \frac{\partial F}{\partial U}\Delta U \quad \text{and} \quad \Delta G = \frac{\partial G}{\partial U}\Delta U \qquad (4)$$

are the unsteady fluxes based upon the Jacobians of F and G evaluated at the center of the cell. The first term in equation (3) (ΔU) is the first-order-change-in-time for the Taylor series expansion while the last two terms represent a second-order-change-in-time which is necessary for stability. The latter terms bias the distribution of the "change" in the windward direction, which is similar to the stabilizing effects of upwind differencing [6].

To accelerate solution convergence, Ni introduced a multiple-grid algorithm [4] which propagates the fine grid changes at coarse grid speeds. The coarse grid is generated by eliminating every other fine grid line. The coarse grid acceleration is accomplished by transporting the changes previously calculated from the fine meshes to the center of a coarse grid cell and then distributing that change to the coarse grid nodes by use of the distribution formulae (3).

Boundary Conditions

For the problems considered in this paper, there are two types of boundary conditions. The first is a solid wall boundary condition where "no flow through the surface" is enforced. If one views the solid wall as a streamline, then the effect of a pseudo-cell just inside the body can be considered. By combining the effects of the distribution formulae (3) on each side of the wall, one finds that the boundary condition reduces to doubling the boundary changes predicted by the interior cell, and cancelling the normal velocity component.

The other boundary condition type is a far field condition. For an airfoil calculation, the far field is composed of a uniform flow plus the effect of a vortex, whose strength is set based upon an approximate lift coefficient. For the cascade problem, the free stream is assumed to be simply a uniform flow both upstream and downstream.

The boundary conditions are applied at far field nodes by using a characteristic analysis in the local streamwise direction; the characteristic variables which enter the domain remain unchanged, while for those exiting, it is assumed that the changes in the characteristic variables are predicted correctly by the interior scheme.

Embedded Mesh Interface

The embedded grid is composed of cells which were formed by dividing a global (or previously divided) cell. Locally, this appears exactly like the fine and coarse grid cells used in the multiple-grid algorithm.

Using a multiple-grid accelerator to couple the global and embedded meshes was first suggested by Brandt [7] and was implemented by Brown for the full potential equation [8]. With this technique, waves propagate through the embedded regions at coarse grid speeds. Usab has shown that this coupling results in convergence rates which are as fast as coarse-grid-alone solutions [9]. This is significant when one considers the consequences of simply coupling global and embedded regions at the interface [2]. In the latter technique, wave propagation is restricted to the embedded (fine) grid speed, resulting in slower convergence rates.

Points at the edge of the embedded domain must be carefully treated in order to maintain global conservation and computational stability. Consider the embedded mesh interface shown in figure 2. In the present scheme, nodes 2, 5, and 9 are considered part of the fine domain. Thus the flux balance and distribution formulae can be applied as usual to cells B, C, D, and E. Although the changes in the dependent variables are computed at nodes 2, 5, and 9 due to cells B and E, the changes must not be applied when operating on the embedded mesh level. Instead, they must be stored and applied only after the (explicit) flux balance has been performed on cell A.

Since cell A is a fine cell on the global grid level, the appropriate integration consists of a flux balance and distribution. Each of these steps must be modified due to the presence of node 5. The flux balance now consists of the sum of the fluxes through five faces (1-2, 2-5, 5-9, 9-8, and 8-1). This can be easily incorporated into the trapezoidal integration. Since the fluxes through the interface (for example 2-5) cancel in the adjoining cells (for example A and B), the scheme maintains global conservation.

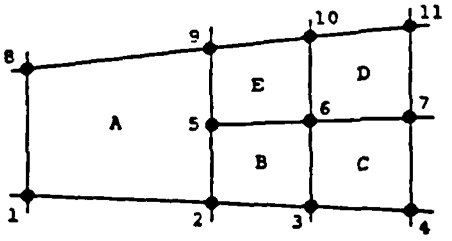


Figure 2. Detail of embedded mesh.

The distribution formulae given by equation (3) remain valid for cell A. However, the change at the center of the cell must also be distributed to node 5. This is accomplished by averaging the distribution to nodes 2 and 9, or in general

$$\delta U_5 = \frac{1}{4}\left(\Delta U \pm \frac{\Delta t}{\Delta x} \Delta F \right)$$

and (5)

$$\delta U_5 = \frac{1}{4}\left(\Delta U \mp \frac{\Delta t}{\Delta y} \Delta G \right)$$

Coarse cell BCDE is treated the same as any other coarse cell interior to the embedded region. This yields an apparent inconsistency at nodes 2, 5, and 9 due to the absence of a coarse cell underlying cell A. At convergence however, the residuals transferred to BCDE do vanish as do the inconsistencies.

Adapted Smoothing

As with most solutions of the Euler equations, artificial viscosity is needed to damp out spurious oscillatory solutions and to capture shocks. Different levels of artificial viscosity are needed for these two purposes. Jameson [10] uses fourth order smoothing over the whole flow field to damp out spurious oscillations, with second order smoothing blended in near shocks to capture them. This blending has the effect of putting the lower order smoothing only where it is needed.

An alternative way of using the proper smoothing only where needed is to use second order smoothing globally, but to vary the smoothing coefficient so that only the required amount is used locally. Typically, the coefficient is related to the second difference of density or pressure.

For illustrative purposes, the following will be developed in one dimension, with its extension to two dimensions being straightforward.

The smoothing at any point i is given by

$$\text{smooth}_i = \mu_i \, (u_{i-1} - 2u_i + u_{i+1}) \tag{6}$$

where u is the parameter being smoothed and μ_i is the spatially varying smoothing coefficient (which includes the appropriate geometric parameters).

To vary the smoothing coefficient from node to node, a second difference of density is used in the current program. This takes the form

$$\mu_i = \eta \left[1 + \delta \, \frac{|\rho_{i-1} - 2\rho_i + \rho_{i+1}|}{(\rho_{i-1} + 2\rho_i + \rho_{i+1})} \right] \tag{7}$$

Here η is the level of background smoothing needed to control spurious oscillations and δ is the factor by which the smoothing has to be increased (above the background level) near shocks. Typical values are η=0.025 and δ=50.

A desirable characteristic of any smoothing scheme is that it be conservative, i.e., the sum of the smoothing terms over the whole domain vanishes. If μ_i is a constant in (6), the contributions of the interior nodes in fact do cancel, leaving only boundary contributions. If however μ_i varies, this is not true. To circumvent this problem, one can use an averaging of the smoothing coefficients, giving

$$\begin{aligned} \text{smooth}_i = &+ (\mu_{i-1} + \mu_i) * (u_{i-1} - u_i) \\ &- (\mu_i + \mu_{i+1}) * (u_i - u_{i+1}) \end{aligned} \tag{8}$$

Thus, the sum of the smoothing over all i results in the cancellation of the contributions of all the interior intervals, leaving only the boundary terms, or

$$\begin{aligned} \sum_{i=2}^{N-1} (\text{smooth}_i) = &+ (\mu_1 + \mu_2) * (u_1 - u_2) \\ &- (\mu_{N-1} + \mu_N) * (u_{N-1} - u_N) \end{aligned} \tag{9}$$

This leaves two possibilities. First, one can accept the amount of non-conservation near the boundary and allow the creation of mass (or momentum); the resulting error is not serious. In practice, a more serious problem with this approach is that the smoothing operator looks convective (rather than diffusive) near the boundary. This is not an acceptable numerical model for a smoothing operator. The other approach (and the one adopted here) is simply not to smooth normal to the ends of the domain. As long as smoothing is not required there (and numerical experiments show that it is not), this is a valid approach.

A spatially varying smoothing coefficient can also cause the smoothing term to look convective within the flow field. This occurs if μ_i is allowed to vary too rapidly within the field. It can easily happen near shocks where the second difference of density can change rapidly from point to point. It is reasonable therefore to require a smooth distribution of μ_i, which requires smoothing the smoothing coefficient. This can be accomplished by calculating μ_i at every point, and then smoothing the values before use in the smoothing step. Unfortunately, this requires three sweeps through all the computational nodes, which is expensive. Therefore, in the present technique, a special smoothing coefficient equation

$$\mu_i^{n+1} = \mu_i^n + \alpha \left[\left(\pi_i^n - \mu_i^n\right) + \beta\left(\mu_{i-1}^n - 2\mu_i^n + \mu_{i+1}^n\right)\right] \quad (10)$$

is added to the set of equations being solved, where n is the iteration count and π_i is the desired smoothing coefficient computed from equation (7). In equation (10), α governs the rate of convergence for μ_i, which must be faster than the rate of convergence of the Euler equations. The parameter β in (10) governs the smoothness of μ_i. This equation assumes that μ_i does not vary much from the previous iteration, and thus using the prior values for the diffusive terms does not cause significant errors. Typical values are α=0.1 and β=1.

COMPUTED RESULTS

The computed results are presented in two sections. In the first, various refinement parameters are examined, and appropriate choices for the two-dimensional Euler equations are discussed. The second section compares adapted solutions with solutions computed with global refinement. Here, global refinement refers to the process of subdividing every cell in the domain. These comparisons are made with respect to both accuracy and efficiency.

Feature Detection

Adaptation requires that the program be able to sense where and when refinement is required. For the Euler equations there are many values which might be sensed for this purpose. For example, one may use first or second differences, or even the truncation error. Those operators can be applied to such primitive values as density or energy, or to derived quantities such as the entropy.

In choosing an appropriate refinement parameter, one must consider the kinds of features which are to sensed and things about them which are different than the background flow. For example, if one is searching for shocks, it is clear that a change in density, pressure, and entropy is generated across the shock, but that there is no change in mass. Similarly, if one was looking for the slip line behind an airfoil, it may be useful to use density or entropy, but not static pressure since it is continuous across the slip line.

In addition to choosing which variable to sense, one must also select a way of measuring how it varies. Typically first or second differences might be examined. Here, the undivided differences were chosen instead of the gradient or Laplacian of the chosen variable since it is important that the measure react to prior adaptation. For example, if one used the entropy gradient as a measure near a shock, adaptation would seem to have the wrong effect, since after adaptation, the gradient would increase, indicating a need for more adaptation, etc. On the other hand, if the first difference of density were used, then the refinement parameter would not increase as a result of prior adaptation.

Figure 3 shows the effect of choosing different refinement parameters during actual computations. The embedded grids were computed for an RAE 2822 airfoil at Mach number 0.75 and angle of attack 3.0 degrees. This case has a shock on the upper surface at about 75 percent chord. In each of the cases, there are two sequential adaptations, with the threshold selected automatically as described below.

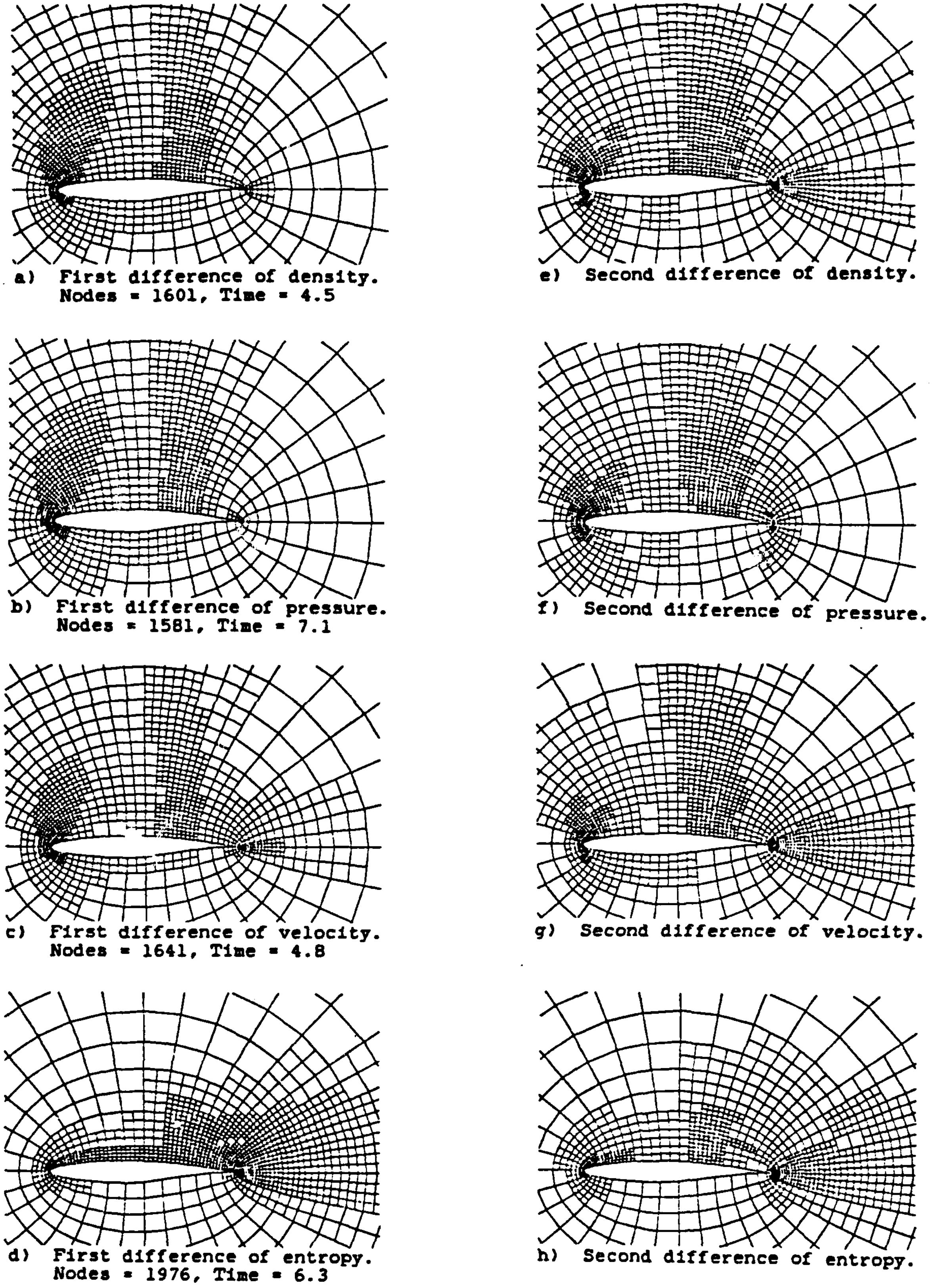
Figure 3a shows the grid resulting from using the first difference of density as the refinement parameter. In this case, double embedding was automatically generated around the stagnation point, the shock, and the trailing edge. In addition, double embedding followed the expansion fan generated at the leading edge. The remainder of the airfoil surface is singly embedded, except on the lower surface near the trailing edge.

Figure 3b shows the grid resulting from using the first difference of pressure as the refinement parameter. The grid is very similar to that in figure 3a, except that there is less adaptation near the trailing edge and in the near wake region. This is due to the fact that pressure is continuous across the slip line emanating from the trailing edge, whereas the density has a jump.

In figure 3c, the first difference of velocity served as the refinement parameter. Here again, the pattern of adaptation is very similar to the previous two examples. The main difference is that the extent of the adaptation behind the airfoil is slightly larger than either of the other two.

The first difference of entropy is used as the refinement parameter in figure 3d. It can be seen that the adaptation very faithfully follows the entropy gradient set up in the wake, even out to the far field boundary. Since one must control the increase in required computational resources, this results in fewer points being available for other parts of the field, with the lower surface of the airfoil being almost completely ignored. Also, the adaptation region around the shock is rather small, resulting in an appreciable strength shock passing through the edge of the embedded region.

The computed lift and drag coefficients for each of these cases were all within 1.4 percent of each other, indicating that the adaptations were equally good from the standpoint of accuracy. One can see from the figures that the normalized CPU times to convergence ranged from 4.5 to 7.1. However, these times were not directly



a) First difference of density. Nodes = 1601, Time = 4.5

b) First difference of pressure. Nodes = 1581, Time = 7.1

c) First difference of velocity. Nodes = 1641, Time = 4.8

d) First difference of entropy. Nodes = 1976, Time = 6.3

e) Second difference of density.

f) Second difference of pressure.

g) Second difference of velocity.

h) Second difference of entropy.

Figure 3.
Effect of choice of refinement parameter.
RAE 2822, Mach=0.75, α=3.0 deg.

related to the number of nodes for each case, due to the varying number of iterations required to reach convergence. From these cases as well as others which the authors have reviewed, it appears that the adaptations which completely surround flow features (shocks, leading and trailing edges) perform the best.

Figures 3e through 3h show the effect of using the second difference of density, pressure, velocity, and entropy respectively. In each case, the edge of the embedded region proves to be relatively ragged. There are voids in the embedded region as well as islands of adaptation, as easily evidenced in figure 3f. This results in convergence difficulties in the solver. Consequently, all of the cases using second differences resulted in unacceptably long computation times due to such topological problems.

Based on these findings, as well as on the accuracies and computation times for all these methods, it appears that the first difference of density is the best refinement parameter, especially for transonic airfoils with shocks, stagnation regions, and slip lines.

In addition to selecting an appropriate refinement parameter, it is necessary to determine a threshold; adaptation is performed around any nodes whose refinement parameter exceeds this threshold. This is a classical signal-to-noise discriminator.

Figure 4 shows the distribution of refinement parameter arbitrarily plotted versus node number for the airfoil calculation previously discussed. The refinement parameter for this case is the first difference of density, normalized by the average first difference over the whole domain. There are four levels of possible threshold shown.

Choosing A as the threshold results in very few cells being adapted. Alternatively, choosing D results in almost all the cells being adapted, yielding an almost global embedding, which is a very ineffective strategy. It appears from figure 4 that an appropriate threshold value would be either B or C.

This data is presented again in figure 5 which is the cumulative distribution function corresponding to figure 4. On the abscissa are the possible values of the threshold and on the ordinate is the fraction of points whose refinement parameter exceeds the selected threshold.

There is a knee in the figure near point C. For a slight decrease in the threshold value, the fraction of points which would be newly included in the adaptation would rise rapidly (a large slope). This corresponds to the decrease in the threshold below the value of noise in figure 4. If on the other hand there is a slight increase in the threshold, there are very few nodes which would no longer become adapted. Therefore, the "knee" in figure 5 is used to automatically set the threshold.

One must however include guards into this algorithm. For example it seems foolish to choose a threshold value which is less than the average refinement parameter. It is also important to ensure that too much adaptation is not done since that increases the required computational resources. This results in constraining the possible values of threshold by the cross-hatched lines in figure 5.

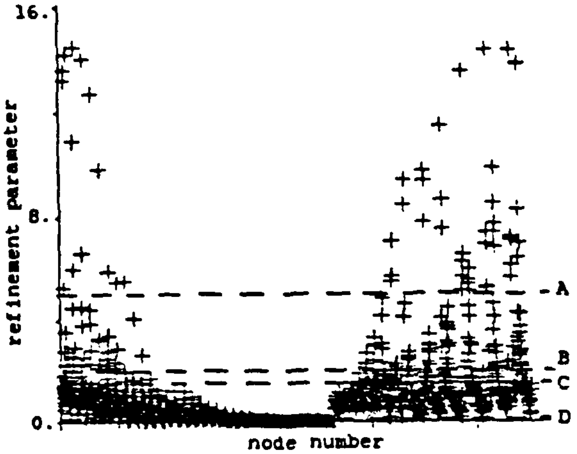


Figure 4.
Distribution of refinement parameter.

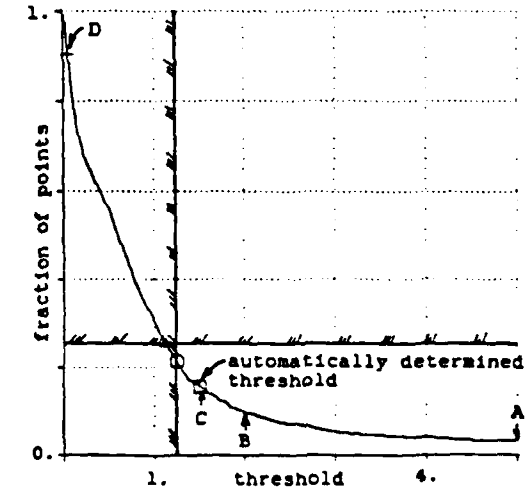


Figure 5.
Cumulative distribution of refinement parameter.

A sensitivity study was conducted to ascertain the importance of the parameters used for the automatic threshold determination. Figure 6 (a, b, and c) shows the effect of choosing the threshold at levels A, B, and C respectively. It can be seen that the extent of the embedded region increases as the threshold value decreases. However, the computed lift and drag coefficients show a slight degradation from C to B (about 1 percent) and a larger degradation from C to A (about 5 percent). This is consistent with the previous comment that the adaptation is best when it completely surrounds the flow features of interest.

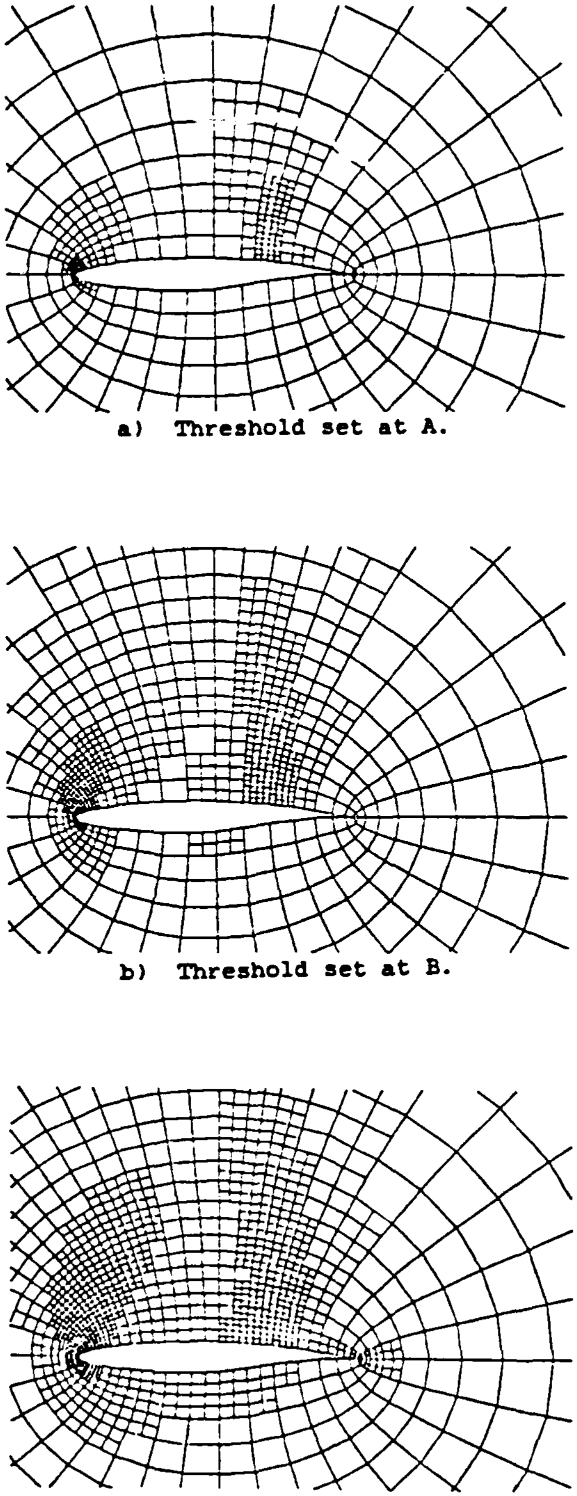
Adaptation Effectiveness

To determine the effectiveness of an adaptive grid scheme, one must consider its accuracy and efficiency as compared with standard techniques. Adaptation effectiveness can be measured in two ways; the first is with respect to required CPU time and the second is with respect to required computer storage. Adaptation accuracy can be measured by comparing solutions computed with adapted refinement with those based upon globally refined grids.

Figure 7 shows the computed accuracy versus efficiency for three different airfoils (to be discussed below). The accuracy is measured by the difference between the computed lift coefficient and a reference lift coefficient (figures 7a and 7c) or the difference between the computed drag coefficient and a reference value (figures 7b and 7d). The efficiency is measured either by the required CPU time normalized by the time required for the base solution (figures 7a and 7b), or by the number of nodes similarly normalized (figures 7c and 7d). In each figure, for each airfoil (symbol), there are two lines. The solid line refers to adapted refinement and the dotted line refers to global refinement.

The first airfoil is the NACA 0012 at Mach number 0.40 and zero degrees angle of attack. For this case, which is denoted by squares, the reference values of lift and drag coefficients are both zero. It should be noted that all the solutions yielded zero lift, and this is shown as an error of 10^{-4} (the assumed accuracy of the calculation). The drag coefficients for the solution with one global and one adapted refinement were identical, as were the drags calculated for two global and two adapted refinements.

The second case (denoted by circles) is again for the NACA 0012, but this time at Mach number 0.63 and 2.0 degrees angle of attack. For this case, the reference lift coefficient is 0.3297 and the reference drag coefficient is zero. It can again be seen that the lift and drag coefficients are the same if global or adapted refinement is used.

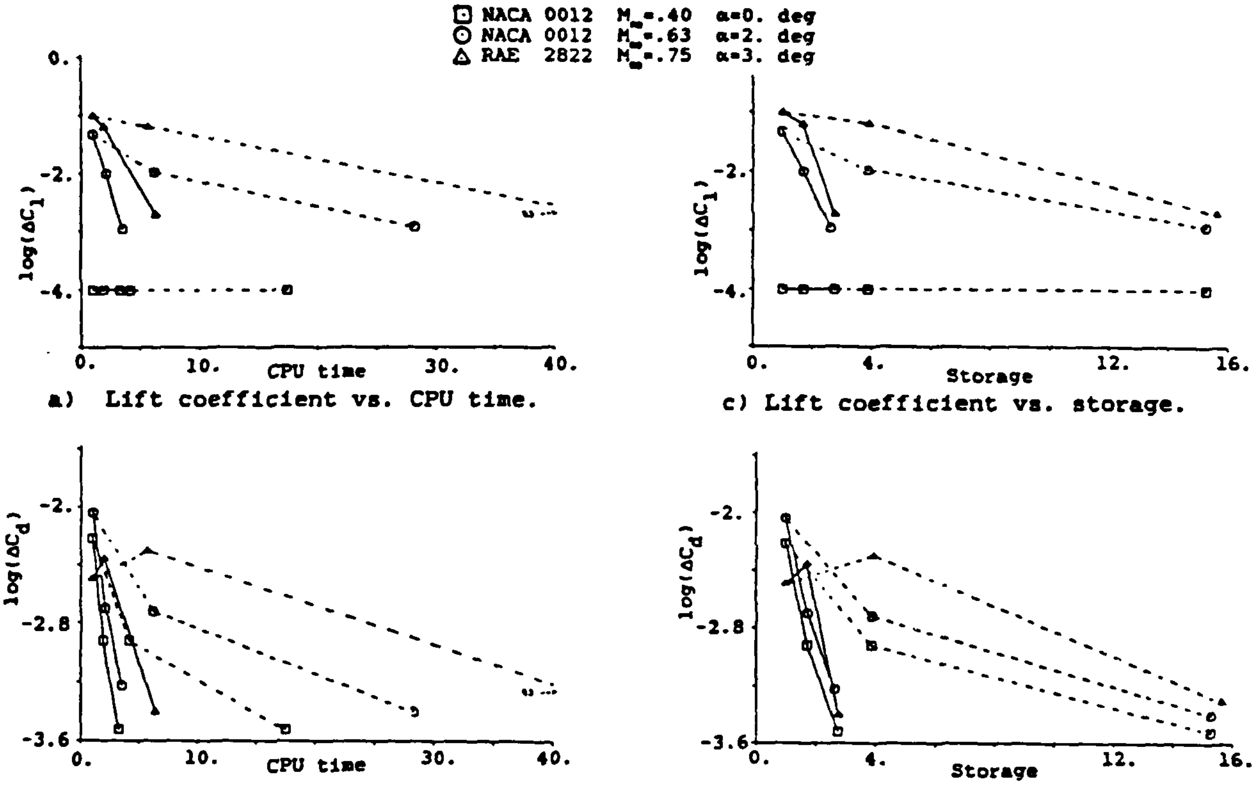


a) Threshold set at A.

b) Threshold set at B.

c) Threshold set at C.

Figure 6. Effect of choice of threshold.
First difference of density.
RAE 2822, Mach=0.75, α=3.0 deg.



a) Lift coefficient vs. CPU time.

c) Lift coefficient vs. storage.

b) Drag coefficient vs. CPU time.

d) Drag coefficient vs. storage.

Figure 7. Comparison of accuracy vs. efficiency for global and adapted embedding.

The third case (denoted by triangles) is the RAE 2822 airfoil at Mach number 0.75 and 3 degrees angle of attack. The pressure coefficient distribution for this airfoil (figure 8) shows a shock at about 75 percent chord on the upper surface, as do the Mach number contours for the same case (figure 9). The three lines in figure 8 are the computed solutions without refinement, with global refinement, and with adapted refinement; the global and adapted refinements are so close that they virtually appear as one line on the plot. For this case, the reference lift coefficient is 1.076 and the reference drag coefficient is 0.0424. The anomaly in drag coefficient for the base mesh is due to a trade-off between shock location (and strength) change and total pressure loss due to smoothing. Again, the error in the lift and drag coefficients are the same for global and adapted refinement.

For these cases, figure 7 demonstrates that adaptive refinement yielded the same accuracy as global refinement, but required only 12 to 33 percent as much resources. Alternatively, one notes that for a given resource allocation, the adapted solutions yielded considerably more accurate solutions.

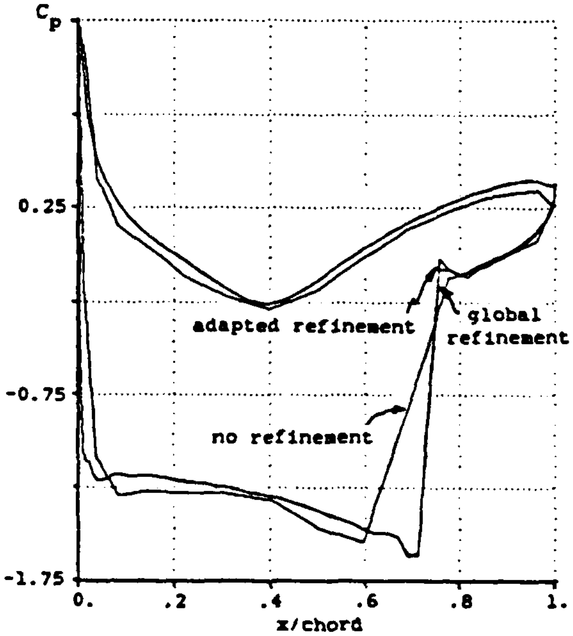


Figure 8. Pressure coefficient for non-embedded, global-embedded, and adapted-embedded runs. RAE 2822 Mach=0.75, α=3.0 deg.

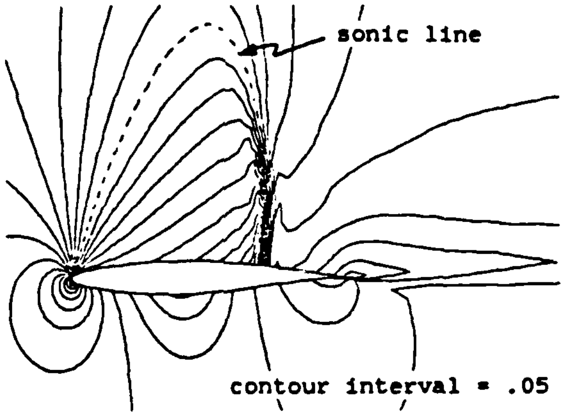


Figure 9. Mach number contours.
RAE 2822 Mach=0.75, α=3.0 deg.

The final test case is for the supersonic flow over an 8-percent-thick, circular-arc, unstaggered cascade at a free stream Mach number of 1.4. In this case, there is a complex shock pattern which forms from the intersection and eventual coalescence of the leading and trailing edge shocks. It can be seen that the adaptation method used here resolves the detailed shock interaction (figure 10a). If grid refinement by redistribution had been used, the case would have yielded significant topological difficulties. As before, the refinement parameter is the first difference of density, with the threshold chosen automatically.

Figure 10b shows that for the base grid, the shock intersection, reflection, and eventual coalescence is missed whereas for either the global (figure 10c) or adapted refinement (figure 10d), these phenomena are obtained in detail. For this case, the adapted solution ran 4.8 times faster and required one third the storage of the comparable globally refined solution.

CONCLUSIONS

- An adapted grid strategy which uses the embedded mesh procedure described herein yields solutions with the same accuracy as a globally refined mesh case but only requires between 12 and 33 percent of the resources for all cases tried to date.

- Various refinement parameters have been examined to deduce that the first difference of density is the most effective for two dimensional Euler solutions with shocks.

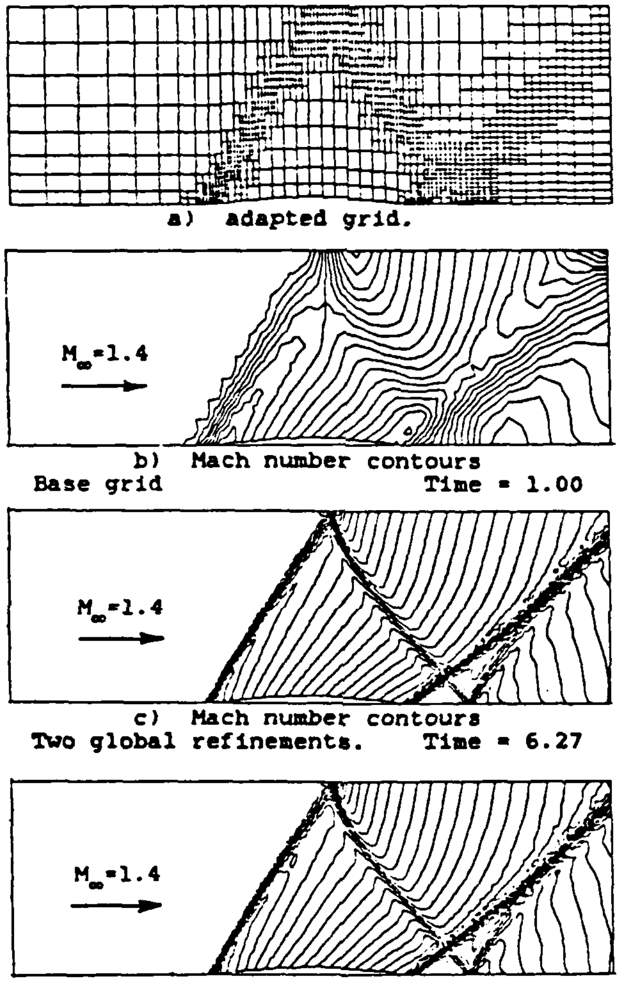
- A thresholding algorithm has been developed which automatically sets thresholds in order to detect and isolate features from the background field.

- An adaptive smoothing algorithm has been developed, which is conservative, small in smooth regions, and primarily diffusive rather than convective.

- The adaptation strategy presented here is extendable to three-dimensions, with even larger anticipated savings.

ACKNOWLEDGEMENTS

This work was supported by the Air Force Office of Scientific Research under grant AFOSR-82-0136, Dr. J.D. Wilson technical monitor.



a) adapted grid.

b) Mach number contours
Base grid Time = 1.00

c) Mach number contours
Two global refinements. Time = 6.27

d) Mach number contours
Two adapted refinements. Time = 1.31

Figure 10.
Solution for supersonic circular arc cascade.

REFERENCES

[1] Thompson JR, "Review On the State of the Art of Adaptive Grids", AIAA-84-1606, June 1984.

[2] Berger MJ and Jameson A, "Automatic Adaptive Grid Refinement for the Euler Equations", DOE/ER/03077-202, October 1983.

[3] Dannenhoffer JF and Baron JR, "Adaptive Procedure for Steady State Solution of Hyperbolic Equations", AIAA-84-0005, January 1984.

[4] Ni RH, "A Multiple-Grid Scheme for Solving the Euler Equations", AIAAJ, Vol 20, No 11, November 1982, pp 1565-1571.

[5] Usab WJ and Murman EM, "Embedded Mesh Solution of the Euler Equation Using a Multiple-Grid Method", AIAA-83-1946-CP, July 1983.

[6] Jameson A, "Iterative Solution of Transonic Flows over Airfoils and Wings, Including Flows at Mach 1", Comm on Pure and Applied Math, Vol 23, May 1974, pp 283-309.

[7] Brandt A, "Multi-Level Adaptive Solutions to Boundary-Value Problems", Math. of Comp., Vol 31, No 138, April 1977, pp 333-390.

[8] Brown JJ, "A Multigrid Mesh-Embedding Technique for Three-Dimensional Transonic Potential Flow Analysis", AIAA-82-0107, January 1982.

[9] Usab WJ, "Embedded Mesh Solutions of the Euler Equation Using a Multiple-Grid Method", Ph.D. Thesis, Massachusetts Institute of Technology, January 1984.

[10] Jameson A, "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method", MAE Report 1613, Princeton University, June 1983.

Appendix C

IMPACT OF SUPERCOMPUTERS ON THE NEXT DECADE OF COMPUTATIONAL FLUID DYNAMICS

Earll M. Murman
Saul S. Abarbanel

CFDL-TR-85-1 March 1985

Department of Aeronautics and Astronautics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

This report was prepared under the sponsorship of contract AFOSR-82-0136C, Dr. James Wilson, Technical Monitor. To appear in PROGRESS AND SUPERCOMPUTING IN COMPUTATIONAL FLUID DYNAMICS, Birkhauser Boston, Inc., publishers, August 1985.

IMPACT OF SUPERCOMPUTERS ON THE NEXT DECADE OF COMPUTATIONAL FLUID DYNAMICS

Earll M. Murman
Saul S. Abarbanel

Introduction

A small group of CFD researchers from the United States and Israel gathered in Jerusalem during December 1984 at a workshop entitled "The Impact of Supercomputers on the Next Decade of Computational Fluid Dynamics." The background of the workshop attendees ranged from CFD code developers to applied mathematicians to computer experts. During the workshop the participants presented and discussed results of their current research. They then engaged in discussion of the workshop theme. This article attempts to summarize their observations and speculations on what the impact of supercomputers will be on CFD during the next decade. First, however, we briefly summarize the papers in these proceedings and the current status of CFD.

The Present

Supercomputers and CFD have affected every aspect of fluid dynamics to some degree during the past decade. Perhaps the area which has experienced the most dramatic impact is the field of attached flow aerodynamics, typical of design point conditions for transport aircraft. In this situation the fluid flow is well behaved by design. Separated and unsteady flow are avoided. The turbulent flow models applicable to attached boundary layers are acceptable (though not perfect). There are no chemical reactions or phase changes taking place. The biggest challenges lie in solving the nonlinear inviscid flow equations (primarily transonic) and dealing with the complex geometry. During the past decade, the capability of the computers and algorithms has developed enormously in this one particular subdiscipline. In many instances they are as much utilized as wind tunnel testing, although by no means supplementing them.

In most other fields of science and engineering, many of the more difficult fluid dynamic phenomena which are absent in attached flow aerodynamics are of paramount importance. For example, turbomachines are dominated by three-dimensional viscous and unsteady phenomena which affect heat transfer and performance. In many devices for propulsion and chemical processes, multicomponent chemical reactions and turbulent mixing must be modeled. High performance aircraft and helicopters are strongly influenced by vortical and unsteady effects. Low drag bodies are dominated by the prediction of transition. Separated, unsteady wakes of automobiles influence their fuel consumption and handling capabilities. In large scale geophysical fluid dynamics, coriolis forces and stratification effects are dominant. These lead to multiple time scale wave phenomena. Unstable stratification produces turbulent, buoyant mixing. Turbulent flow is present in virtually every situation, yet can only be adequately modeled for the simplest of flows like attached boundary layers and jets. This does not exhaust the list, but the point made here is quite clear; only the tip of the iceberg has been seen by the progress made in attached flow aerodynamics. The biggest challenges are yet to come. The papers in the proceedings give one assessment of where the field stands in this respect.

The Papers

Fernbach's paper gives a comprehensive overview of the current performance of supercomputers and what is on the horizon. This field is now very active following a dormant period in the 70's. The basic message is that computer speed and main memory will both increase by about two orders of magnitude in the next decade. Also, all future supercomputers will be a combination of vector (pipeline) and parallel (multiprocessor) architectures. Algorithms will have to adapt to and exploit these architectural features to achieve the stated machine performance. Thompkins' paper demonstrates that supercomputing does not necessarily have to be done on supercomputers. It makes an interesting case for the need of personal-sized supercomputers with speeds about one order of magnitude slower than the mainframes, but with comparable memories. The idea of using higher level languages for multiprocessor applications is introduced by Thompkins. Navier-Stokes results are also presented for turbomachine cascades, illustrating the state of the art for CFD applied to these flows. The paper by Jameson, Leicher, and Dawson demonstrates one way that the current generation of algorithms for inviscid external flows can be modified for multiprocessor architectures.

The paper by Steger and Buning provides an overview of current issues regarding the computation of inviscid and viscous aerodynamic flows. In addition to a number of interesting results which are presented, the experience of these authors using the current generation of supercomputers is recorded. Vectorization of implicit algorithms is explained. The need for good graphical postprocessing of large data bases turns out to be mandatory. A similar message is given in the paper by Murman, Rizzi and Powell, which compares two independently obtained solutions for leading edge vortex flows for delta wings. This paper also illustrates that this class of compressible flows is relatively unexplored compared to shock wave dominated flows.

Several papers present new algorithms for the Euler and the incompressible or compressible Navier-Stokes equations. Since the solution of these equations will become more frequent with the higher power computers, this is an important topic for the future. The paper by Walters and Dwoyer introduces an upwind differencing line relaxation algorithm for the Euler equations. McCormack presents a new algorithm for the compressible Navier-Stokes equations which has some similarities to the algorithm of Walters and Dwoyer. Turkel presents methods for accelerating the convergence to a steady state solution of the Euler or Navier-Stokes equations by using preconditioning to alter the time consistency of the equation set. The paper by Glowinski addresses finite element methods for the incompressible Navier-Stokes equations and presents a number of results concerning entry to ducts and the subsequent internal flow. Remarks are also included on finite element methods for compressible Navier-Stokes equations and applications on supercomputers. Israeli gives an algorithm for the parabolized Navier-Stokes (PNS) equations. Brandt and Ta'asan present the latest multigrid algorithms for quasi-elliptic systems which arise from discrete approximations to the Navier-Stokes and related equations. The importance of algorithms, such as multigrid methods, which have convergence rates (spectral radius) independent of the number of mesh points will be mentioned later.

Perhaps no problem is more central to fluid mechanics than the prediction of transition and turbulent flows. Three papers deal with this topic. Brachet, Metcalfe, Orszag and Riley present new results for instability of free shear flows based upon direct computations of the Navier-Stokes equations. Numerical experiments such as these can lead to new theoretical understanding of instability of rotational flows. Ferziger's paper gives a comprehensive overview of current and future approaches to turbulent flow computations using direct simulations of the Navier-Stokes equations, large eddy simulation, and turbulence models. Wolfshtein considers the latter topic in much greater detail. These two papers point out the capabilities and shortcomings of current turbulence models. The importance of having accurate algorithms is stressed by both authors.

Papers by Sulem and by Michelson illustrate how numerical results can be used to understand the nature of the solutions to partial differential equations. The use of spectral methods for problems which require high accuracy is receiving increased interest. The presentations by Abarbanel and Gottlieb, and Gottlieb and Tadmor consider some basic issues regarding the resolution of extreme gradients by spectral methods. The paper by Browning and Kreiss illustrates that many fluid problems with multiple time and length scales are exceedingly difficult to compute, even with "unlimited" computer power. It is important to understand that the powerful new supercomputers will only yield useful results if the mathematical and numerical analysis formulation is carefully done. The paper by Sever is another illustration of this.

The Next Decade

During the next decade supercomputer power will increase dramatically. The directly addressable high speed memory capacity will increase by about two orders of magnitude, from 2 million words to 256 million words. Processor speed will increase an order of magnitude from about 100 MFLOPS to 1000 MFLOPS, or perhaps more. It is likely that the corresponding parameters of smaller computers will increase by similar factors. History has shown that whenever an important parameter is varied by an order of magnitude, new discoveries are made. The difficulty is to have some feeling as to what these discoveries might be. The participants realized, of course, that forecasting the future is never an exact science, and one should proceed cautiously. It is interesting to note, however, that during the panel discussion almost everyone subscribed to the idea that the new supercomputers will not only allow tackling bigger problems, but will also lead to a better understanding of the physics of some complex problems such as turbulence, vortical flows, and chemically reacting flows.

In the remainder of this article we summarize the feelings expressed by the attendees concerning four questions which were posed by the panel.

Impact of Supercomputers on CFD

In the field of aerodynamics, the preliminary design of transport aircraft will primarily be done on supercomputers. The modeling and computing capability will be basically in place. Unlike earlier predictions that computers will make wind tunnels obsolete, few people subscribe to that viewpoint now. What is more likely to happen is that the use of wind tunnels by researchers and design engineers will change. Less and less of the exploratory design will be done by tests as the predictive methods become more reliable. This has already happened in several instances with the current generation of computers. The next generation will provide enough resolution and speed that a realistic model of an actual cruising transport aircraft can be computed.

The capability to model "off-design" or "unclean" aerodynamic flows will increase. These are flows which are separated, unsteady, vortex dominated, and the like. Such flows are of great importance for maneuvers of high performance aircraft or for emergency situations for transport aircraft. The loads developed in these regimes often determine the required strength of the aircraft components. The same phenomena often dominate rotary wing and rotating machinery aerodynamics. Capacity of computers and algorithms up to now has not been adequate to support a frontal assault on this class of problems. The payoff for analysis of unsteady, separated, vortical flows will be much greater than for the clean flows representing cruise aerodynamics. This is because little theory has ever been developed for them.

The complexity of problems which the researcher and the engineer will be dealing with will grow in some proportion to the new computer power. This will have a number of impacts on the daily life of the fluids mechanician. Problems under investigation will have many length and time scales. Analysis of results will be more challenging. It will be harder to understand the solution due to the number of interacting physical phenomena present. This is illustrated by the paper of Brachet, Metcalfe, Orszag, and Riley. New methods of analyzing and presenting results will have to emerge in order to deal with this. Graphical output is crucial, and maybe artificial intelligence types of technology will help out.

One difficulty which can be foreseen is the problem of verifying the accuracy or fidelity of a computed result. Up to now, it has generally been possible to compare computed results with theory for limiting conditions. For example, a transonic wing calculation can be compared with linear wing theory for low Mach numbers, or a Navier-Stokes solution can be compared with a laminar boundary layer. But as the computations move into more nonlinear flows, the past theoretical framework will become less and less applicable. Comparison with experiment is an essential, and independent computations of the same problem by different researchers will be necessary. Perhaps a renewed interest in theory will result from this need.

Impact of Supercomputers on Basic Sciences

As one participant stated, the great masters of fluid mechanics in the past solved all the linear problems and left us with only the nonlinear ones. Since most fluid mechanic problems are nonlinear, we can speculate that the ability to model highly nonlinear problems with powerful computers will lead to many new discoveries. Another participant thought that the impact of supercomputers will influence the basic way we think about physical problems. New information will be discovered from numerical experiments and provide insight for modeling. In this sense, computational experiments are akin to laboratory experiments which have provided insight and ideas throughout the history of fluid mechanics and other scientific disciplines.

In the past, computational methods have made a major impact on our ability to compute and understand potential flows and inviscid flows dominated by shockwaves. One can conclude that the classes of flows are well understood both from the physical and algorithmic points of view. Although the ability to analyze shock dominated flows has been a major step forward in fluid mechanics, much is left to be done. For example, only limited CFD studies have been done for vortical flows, and little is understood about the algorithm requirements for inviscid rotational flows. Many studies have been done for two-dimensional separated flows, but only limited studies for three-dimensional separated flows. Although efficient algorithms for steady flows are under development, indications are that the flows these algorithms are to be applied to may be unsteady in nature. See for example Thompkins or Murman, Rizzi and Powell.

Perhaps no area is more tempting to speculate on than the field of turbulence. This is an area in which progress has been relatively stagnant since Reynolds introduced all the unknowns without introducing any new equations. In the past decade, computational models and laboratory experiments have opened a new look at turbulence. The idea of organized or coherent structure has emerged. On the other hand, mathematicians have shown that solutions to fairly simple dynamical systems have chaotic behavior. An interesting question which was posed is "What will be the resolution of the speculation that there is both determinism and chaos in nonlinear equations?" Computational experiments could provide a framework for helping to answer this question.

Another area which will probably be strongly influenced by more powerful computational approaches is the coupling of chemical reactions and heat release to fluid flow problems. Even fairly "simple" reactions involve many species with many time scales of reactions. In the past, computers simply were not large enough to tackle many of these problems. Rate constants are always an uncertain factor in such calculations. Perhaps being able to model the experimental conditions under which the rate constants are measured will lead to more accurate measurements of their values.

One issue on which there was quite a difference of opinion is the degree to which modeling will be required prior to computing. On the one hand, many participants felt that the time was upon us to tackle the full three-dimensional Navier-Stokes equations, possibly adding models only for subgrid scale turbulence. Others felt that the past practice of selecting simplified sets of equations such as inviscid or parabolized viscous will still be prudent. It is likely that some level of modeling will always be required as computer capability will never be big enough to solve a complex problem from first principles. In fact, for most problems this is unnecessary. The question is, will the type of modeling appropriate for the future be different from that used in the past when computer memory, speed, and accessibility were more limited?

Impact of Supercomputers on Algorithms and Languages

An important issue regarding algorithms arises from the multiprocessor and vector architectures of supercomputers. Algorithms which cannot be efficiently used on these architectures will be of limited utility. Many fluid mechanic problems are solved using time dependent integration procedures for initial boundary value problems. Both explicit and implicit methods are utilized. In general, explicit algorithms are easier to vectorize than implicit ones. The latter usually involve recursive steps in the matrix inversion. However, the paper by Steger and Buning demonstrates an effective vectorization strategy for simultaneous inversion of a large number of tridiagonal matrices applicable to approximate factorization methods. Explicit algorithms are also easier to adapt to multiprocessor architectures as the solution domain can be subdivided without introducing difficulties in the algorithm. The paper by Jameson, Leicher and Dawson reports a strategy and results which illustrate this. Effective strategies for adapting implicit algorithms for multiprocessor architectures need to be found also. Participants generally agreed that the real payoff is for algorithms which can work effectively on tens or hundreds of parallel processors, not just two, four, or eight.

It is clear that algorithm developers must be cognizant of the advantages and constraints which non-Von Neumann architectures will place on supercomputing. The paper by Thompkins is an indicator of what will come. In addition to the conceptual changes in algorithms due to multiprocessors and vector processors, efficiency limitations arise in the speed at which main memory can be accessed from various processors or the speed at which data can be exchanged between processors. The personal-size supercomputer reported by Thompkins involves two processors (a scalar host and an attached vector). Movement of the data must be carefully managed to avoid bottlenecks. In the future algorithm development must take into consideration not only traditional numerical analysis, but also computer science aspects.

Another algorithm issue arises from the shear size of the main memory of supercomputers. Problems with very fine meshes will be possible. With the exception of multigrid algorithms for elliptic equations, the asymptotic convergence rate (spectral radius) of iterative methods is dependent on the number of mesh points. Recent estimations done by researchers at NASA Langley indicate that the time required to reach convergence for three-dimensional Navier-Stokes equations on a 256 megaword machine is excessive to the point of being unrealistic. This indicates a real need for finding algorithms which have asymptotic convergence rates which are either mesh independent or vary (at worst) slowly with the number of grid points. Such algorithms must work for problems with widely varying length and time scales, not just model problems or model equations (see Jameson et al).

There was considerable discussion on the need for higher level languages that will make it easier to construct a solution approach for a new problem, as well as make it easier to utilize the new architecture. The general strategy for solving a problem by CFD is more or less common. A grid is generated, discretization of spatial derivatives and boundary conditions is done, an iteration or time integration method is selected, and various outputs are required. It was suggested that assembling these tools and manipulating them for embedded subdomains and the like would lend itself to a higher, and therefore simpler, language. FORTRAN is the language of the CFD community to date. The paper by Thompkins introduces some higher level constructs for managing the solution process in a multiprocessor environment.

Another area in which algorithm innovation may be required is in preprocessing (grid generation) and post-processing (data base analysis). The papers by Steger and Buning and Murman, Rizzi and Powell indicate that graphical analysis is imperative, but other ways of manipulating the data bases would be desirable. Maybe knowledge base programs ("expert systems") will be helpful in finding the important features in a given solution. Or perhaps pattern recognition approaches will be needed.

Impact on Subsystems

The workshop attendees for the most part represented users of supercomputers, and not hardware specialists. However, with the large data bases which will be generated, the participants felt that several of the supporting subsystems might well be inadequate to match the power of the high speed processors. Participants who have had experience with supercomputers felt that strong graphics capability is a number one requirement for analyzing results. As discussed above, some new graphics algorithms may well need to be devised to analyze complicated flow fields. But powerful processing and graphical display capabilities are also required. As the paper by Thompkins points out, a researcher will typically spend much of his or her time performing graphical analysis which requires a processor about an order of magnitude smaller than the supercomputer. A large number of operations are required to construct the output quantities which typically are different combinations of the dependent variables than those which are stored in the data base. Many researchers currently think that the plot files will be created on the supercomputer itself for these reasons. However, super graphical processors, which could be much cheaper and therefore more available to the users, should be developed. Typically, algorithms for creating the graphical data base are easily vectorized and could be done on array processors attached to most processors. These could lead to very powerful and inexpensive graphics workstations. Fortunately the technology is developing rapidly for medium to high resolution color display devices with interactive capability.

Most of the attendees at the workshop fall in the category of "remote" users. They are not located at the same site as the supercomputer. There was significant concern that remote communications will be a real bottleneck. Regular dial-up capability is barely adequate at present for editing files or transmitting small output files to remote users. It certainly would be impossible to do interactive graphics processing from a remote site or to transmit the entire data base for onsite analysis using even dedicated data lines currently available. The best way to communicate with remote facilities at present is to transmit magnetic tapes via express mail services. This inevitably leads to delays and slow turnaround. The impact of supercomputers on researchers who are not co-located with the machines will be minor if high bandwidth communications are not available.

Conclusions

The impact of supercomputers on the next decade of computational fluid dynamics will be substantial. With processor speeds and high speed memory increasing by two orders of magnitude, many changes will take place. The difficulty lies in accurately forecasting what those changes will be. The above discussion presents the thinking of one group of active CFD researchers. Perhaps their viewpoints will serve to help others to become aware of, and think about, these changes as they take place. The next decade has already started, twenty years ago!

List of Papers to Appear in PROGRESS AND SUPERCOMPUTING IN COMPUTATIONAL FLUID DYNAMICS, to be published by Birkhauser Boston, Inc., August 1985.

Earll M. Murman Saul S. Abarbanel	"The Impact of Supercomputers on the Next Decade of Computational Fluid Dynamics"
Sidney Fernbach	"Current Status of Supercomputers and What Is Yet to Come"
W. T. Thompkins	"Experience with a Personal Size Supercomputer - Implications for Algorithm Development"
Antony Jameson Stefen Leicher Jef Dawson	"Remarks on the Development of a Multiblock Three-Dimensional Euler Code for Out of Core and Multi-processor Calculations"
Joseph L. Steger Pieter G. Buning	"Developments in the Simulation of Compressible Inviscid and Viscous Flow on Supercomputers"
Earll M. Murman Arthur Rizzi Kenneth G. Powell	"High Resolution Solutions of the Euler Equations for Vortex Flows"
Robert W. Walters Douglas L. Dwoyer	"An Efficient Iteration Strategy for the Solution of the Euler Equations
Robert W. McCormack	"Numerical Methods for the Navier-Stokes Equations"
Eli Turkel	"Algorithms for the Euler and Navier-Stokes Equations for Supercomputers"
R. Glowinski	"Viscous Flow Simulation by Finite Element Methods and Related Numerical Techniques"
Moshe Israeli	"Marching Iterative Methods for the Parabolized and Thin Layer Navier-Stokes Equations"
Achi Brandt Shlomo Ta'asan	"Multigrid Solutions to Quasi-Elliptic Schemes"
Marc E. Brachet Ralph W. Metcalfe Steven A. Orszag James J. Riley	"Secondary Instability of Free Shear Flows"
Joel H. Ferziger	"Turbulent Flow Simulation - Future Needs"
Micha Wolfshtein	"Numerical Calculation of the Reynolds Stress and Turbulent Heat Fluxes"
P. L. Sulem	"Numerical Investigation of Analyticity Properties of Hydrodynamic Equations Using Spectral Methods"

Daniel Michelson	"Order and Disorder in the Kuramoto-Sivashinsky Equation"
Saul Abarbanel David Gottlieb	"Information Content in Spectral Calculations:
David Gottlieb Eitan Tadmor	"Recovering Pointwise Values of Discontinuous Data Within Spectral Accuracy"
G. Browning Heinz-Otto Kreiss	"Numerical Problems Connected With Weather Prediction"
Michael Sever	"Order of Dissipation Near Rarefaction Centers"