STEADY AND UNSTEADY THREE-DIMENSIONAL TRANSONIC FLOW COMPUTATIONS BY INTEGRAL EQUATION METHOD

by

Hong Hu, Principal Investigator

Hampton University
Hampton, Virginia 23668

Final Technical Report
of
Grant NAG-1-1170

for the period of
September 1990 - August 1994

Prepared for
National Aeronautics and Space Administration

STEADY AND UNSTEADY THREE-DIMENSIONAL TRANSONIC FLOW COMPUTATIONS BY INTEGRAL EQUATION METHOD

by

Hong Hu, Principal Investigator

Hampton University
Hampton, Virginia 23668

Final Technical Report
of
Grant NAG-1-1170

for the period of
September 1990 - August 1994

Prepared for
National Aeronautics and Space Administration

FOREWORD

The research reported in this document was performed under the grant, NAG-1- 1170, from the National Aeronautics and Space Administration (NASA), Langley Research Center through HBCU (Historically Black College and University) Program. Dr. Carson Yates, Jr. was the technical monitor for the period from September 1990 to August 1992. Mr. Walter Silva was the technical monitor for the period from September 1992 to August 1994. The computational time on CM-2 and CM-5 was provided by Numerical Aerodynamic Simulation (NAS) Program at the NASA Ames Research Center. Dr. Carson Yates, Jr. and Mr. Walter Silva have provided helpful suggestions and guidance during this investigation. Two graduate students, Mr. Terry Logan and Mr. Min Soe, and three undergraduate students, Mr. Issac Jackson, Ms. Jada Paysour and Mr. Jason Bryan, at Hampton University, have participated and contributed to this work.

CONTENTS

	Page
Part I. Unsteady Flows	2
Part II. Steady Flows with Shock Fitting	9
Part III. MPP Implementation and Performance Analysis	16
References	22
Figures	24
Tables	57
Program Lists	60
List of Publications	66
Attachments - Publications	

STEADY AND UNSTEADY THREE-DIMENSIONAL TRANSONIC FLOW COMPUTATIONS BY INTEGRAL EQUATION METHOD

Hong Hu
Hampton University
Department of Mathematics
Hampton, Virginia 23668

This is the final technical report of the research performed under the grant: NAG-1-1170, from the National Aeronautics and Space Administration. The report consists of three parts. The first part presents the work on unsteady flows around a zero-thickness wing. The second part presents the work on steady flows around non-zero thickness wings. The third part presents the massively parallel processing implementation and performance analysis of integral equation computations. In the end of the report, publications resulted from this grant are listed and attached.

PART I: UNSTEADY FLOWS

SUMMARY

This part presents the development of an unsteady integral equation (or called field-panel, field-boundary element) scheme for solving the full-potential equation for transonic unsteady zero-thickness wing flows. The unsteady full-potential equation is written in a moving frame of reference, in the form of the Poisson's equation. Compressibility and unsteadiness are treated as non-homogeneity. The integral equation solution in terms of velocity field is obtained by the Green's theorem. The solution consists of a wing surface integral term of vorticity distribution, a wake surface integral term of free-vortex sheet and a volume integral term of compressibility and unsteadiness over a small limited domain around the wing. Numerical solutions are obtained by a time-marching, iterative procedure. Time-derivative term is calculated by a second-order backward finite-difference scheme. To be consistent with the mixed-nature of flows, the Murman-Cole type-difference scheme is used to compute the derivatives of the density. The present scheme is applied to flows around a zero-thickness rectangular wing at transonic speed undergoing acceleration motion and transient pitching motion, respectively. The time history of wing surface pressure distributions is presented.

I-1. INTRODUCTION

Starting in 1970, a great deal of progress has been made in solving transonic flows by using the finite-difference method (FDM) and finite-volume method (FVM). Although the Navier-Stokes equation formulation for the transonic flow computations has been understood as the best model and the FDM and FVM are successful in dealing with transonic flows, the computation of the unsteady Navier-Stokes equations over complex three-dimensional configurations is expensive, particularly for time-accurate unsteady flow computations. There are also major technical difficulties in FDM and FVM for generating suitable grids for complex three-dimensional aerodynamic configurations.

The experience has shown that accurate solutions can been obtained for many transonic flows using the inviscid modeling of the full-potential equation. For transonic flows without strong shocks and massive separations, the full-potential equation is an adequate approximation to the Navier-Stokes equations. The integral equation method (IEM) for the potential equation is an alternative to the FDM and FVM. Moreover, the IEM has several advantages over the FDM and FVM. The IEM involves evaluation of integrals, which is more accurate and simpler than the FDM and FVM, and hence a coarse grid (field-panels) can been used in IEM. The IEM automatically satisfies the far-field boundary conditions and therefore only a small limited computational domain is needed. The IEM does not suffer from the artificial viscosity effects as compared to FDM and FVM for shock capturing in transonic flow computations. The generation of the three-dimensional grid for complex configuration is not difficult in the IEM, since the mapping from physical plane to computational plane is not required.

Because of these advantages associated with the IEM, it is highly desirable to fully develop the IEM to treat transonic flows. Integral equation methods for transonic flows have been developed by several investigators[1-14] during the past few years for steady airfoil, wing and aircraft configurations and unsteady airfoils. The capability, accuracy and efficiency of the integral equation method for steady transonic flows have been investigated. The possibility of treating unsteady transonic flows has been investigated for an airfoil undergoing pitching oscillation[13]. The solutions show that the unsteady effects and the motion of the shock have been predicted accurately and efficiently by the method[13]. This part presents the unsteady three-dimensional integral equation scheme, which is the extension of the unsteady two-dimensional scheme[13], coupling with the steady three-dimensional scheme for zero- thickness wing[9]. Two numerical examples are presented to demonstrate the capability of the unsteady wing flow computations using the integral equation method.

I-2. FORMULATION

For a general unsteady motion of a body, the governing equations are simple to solve if a moving (body-fixed) frame of reference formulation is used. This formulation does not require the grid-motion calculation since the grid is rigidly fixed in the frame. In addition to the space-fixed frame of reference $OXYZ$, a moving frame of reference $oxyz$ is introduced as shown in Figure 1. The moving frame of reference $oxyz$ is translating at a velocity of $\vec{V}_o(t)$ and rotating around a pivot point, $\vec{r}_p = (x_p, y_p, z_p)$, at an angular velocity of $\vec{\Omega}(t)$. The relation for the absolute velocity ($\vec{V}$), relative velocity ($\vec{V}_r$) and transformation velocity ($\vec{V}_o + \vec{V}_e = \vec{V}_o + \vec{\Omega} \times (\vec{r} - \vec{r}_p)$) is given by

$$\vec{V} = \vec{V}_r + \vec{V}_o + \vec{\Omega} \times (\vec{r} - \vec{r}_p) \tag{1}$$

where $\vec{r}$ is the position vector measured in the moving frame of reference.

I-2.1 Governing Equations

The non-dimensional unsteady full-potential equation in the moving frame of reference was derived as follows[14]:

$$\nabla^2\Phi = G \tag{2}$$

with

$$G = G_1 + G_2 \tag{3}$$

$$G_1 = -\frac{\nabla\rho}{\rho}\cdot\vec{V}_r \tag{4}$$

$$G_2 = -\frac{1}{\rho}\frac{\partial'\rho}{\partial t} \tag{5}$$

and

$$\rho = \left\{1 + \frac{\kappa-1}{2}\left[-V_r^2 + (\vec{V}_o + \vec{V}_e)^2 - 2(\frac{\partial'\Phi}{\partial t})\right]\right\}^{\frac{1}{\kappa-1}} \tag{6}$$

where the characteristic parameters of the wing surface panel length, the density and the speed of sound at infinity have been used; Φ is the absolute velocity potential ($\vec{V} = \nabla\Phi = \nabla'\Phi$), G_1 the compressibility, G_2 the unsteadiness, ρ the density, κ the gas specific heat ratio, and t the time; and the prime ($'$) refers to the derivative with respect to the moving frame of reference. The associated boundary conditions are described in the next sub-section.

I-2.2 Boundary Conditions

The boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, wake kinematic and dynamic conditions. They are described as follows:

$$\vec{V}_r\cdot\vec{n}_g = 0 \qquad \text{on} \quad g(\vec{r}) = 0 \tag{7}$$

$$\Delta C_p|_{sp} = 0 \tag{8}$$

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r}) = 0 \quad \text{and} \quad w(\vec{r},t) = 0 \tag{9}$$

$$\frac{1}{|\nabla w|}\frac{\partial' w}{\partial t} + \vec{V}_r\cdot\vec{n}_w = 0 \qquad \text{on} \quad w(\vec{r},t) = 0 \tag{10}$$

$$\Delta C_p = 0 \qquad \text{on} \quad w(\vec{r},t) = 0 \tag{11}$$

where $\vec{n}_g$ is the unit normal vector of the wing surface, $g(\vec{r}) = 0$; C_p is the surface pressure coefficient; the subscript sp refers to the edges of separation, and in the present scheme the only separation from the wing trailing edge is considered; and $w(\vec{r},t) = 0$ is the wake surface.

I-2.3 IE Solution

By using the Green's theorem, the integral equation solution of Eq. (2) in terms of the relative velocity field is given by

$$\begin{aligned}\vec{V}_r(x,y,z,t) = &-\vec{V}_o(t) - \vec{\Omega}(t)\times(\vec{r}-\vec{r}_p)\\ &+\frac{1}{4\pi}\int\int_g \frac{\vec{\gamma}_g(\xi,\eta,\zeta,t)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta)\\ &+\frac{1}{4\pi}\int\int_w \frac{\vec{\gamma}_w(\xi,\eta,\zeta,t)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta,t)\\ &-\frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta,t)}{d^2}\vec{e}_d d\xi d\eta d\zeta\end{aligned} \tag{12}$$

where $\vec{\gamma}$ is the surface vorticity distribution; the subscripts g and w refer to the wing and wake surfaces, respectively; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

In Eq. (12), the first integral term is the contribution of the wing surface vorticity; the second integral term is the contribution of the wake vorticity; and the third integral term is the contribution of the full compressibility and the unsteadiness. It should be noticed that the infinity condition, Eq. (9), is automatically satisfied by the integral equation solution. It should be also noticed that the integrand of the volume integral term, the third integral term in Eq. (12), decreases rapidly with increasing distance, d, not only because of the factor of $1/d^2$ but also $G(\xi,\eta,\zeta,t)$ diminishes rapidly with increasing distance. Consequently, for computational purposes, the volume integral term needs to be addressed only within the immediate vicinity of the body. This is the beauty of the IE methods.

I-3. COMPUTATIONAL SCHEME

I-3.1 Discretisation

In terms of discretisation, the wing and its wake are represented by triangular vortex panels. A uniform rectangular parallelopiped type of volume elements are used throughout the flow field. The discretized integral equation solution becomes

$$\begin{aligned}&\vec{V}_r(x,y,z,t)\\ &= -\vec{V}_o(t) - \vec{\Omega}(t)\times(\vec{r}-\vec{r}_p)\\ &+\frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{\vec{\gamma}_{g_{i,k}}(\xi,\eta,\zeta,t)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta)\\ &+\frac{1}{4\pi}\sum_{i=1}^{LW}\sum_{k=1}^{NW}\int\int_{w_{i,k}} \frac{\vec{\gamma}_{w_{i,k}}(\xi,\eta,\zeta,t)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta,t)\\ &-\frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV}G_{i,j,k}\int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{e}_d d\xi d\eta d\zeta\end{aligned} \tag{13}$$

where the indices, i, j and k refer to the surface panels and field elements; $LG \times NG$ is the total number of wing surface panels; $LW \times NW$ is the total number of wake surface panels; and $LV \times MV \times NV$ is the total number of field elements. A constant G-distribution is used over small field element, while a linear $\vec{\gamma}$-distribution is used over small surface panel.

I-3.2 Time-Marching, Iterative Scheme

Due to the nature of the nonlinearity of the flow, the solutions are obtained through a time-marching, iterative procedure, where the compressibility, G_1, unsteadiness, G_2, and the wake shape and its strength are updated within each iteration. The solution procedure follows the successful form of the unsteady two-dimensional scheme[13] and the steady three-dimension scheme[9], hence only a brief description is given:

Step 1 - at time step $(k = 0)$:
This step corresponds to the steady flow computation. At this step, $G_2^{(o)}$ and $(\partial'\Phi/\partial t)^{(o)}$ are set to be zero. Equations (3), (4), (6) and (13) with the boundary conditions are solved iteratively until the solution converges. Here, two loops are used. The inner loop is used to calculate and check the convergence of the non-linear term, $G^{(o)}$. The outer loop is used to update and check the convergence of the wake shape and wing surface pressure distribution.

Step 2 - at time step $(k = n)$:
This step is unsteady time marching step. At this step, the wing translation and angular velocities are calculated by the given functions, $\vec{V}_o = \vec{V}_o(t)$ and $\vec{\Omega} = \vec{\Omega}(t)$, respectively. The orientation of the wing changes due to the angular velocity. Two numerical examples are considered. In the first numerical example, the wing is given an acceleration motion. Translation Mach number, $M_o(t) = V_o(t)/a_\infty$, is given by

$$M_o(t) = M_i + \dot{M}_o t \tag{14}$$

where M_i is the initial value of $M_o(t)$ and $\dot{M}_o$ is the rate of change of $M_o(t)$. In the second numerical example, the wing is given an forced transient pitching motion. Angle of attack, $\alpha(t)$, is given by

$$\alpha(t) = \alpha_i + \dot{\alpha} t \tag{15}$$

where α_i is the initial value of $\alpha(t)$ and $\dot{\alpha}$ is the z-component of $\vec{\Omega}$ ($\vec{\Omega} = 0\vec{i} + 0\vec{j} + \dot{\alpha}\vec{k}$). The unsteadiness, $G_2^{(n)}$, is calculated numerically by a second-order accurate backward finite-difference scheme, which is given by

$$\begin{aligned} G_2^{(n)} &= -\left(\frac{1}{\rho}\frac{\partial'\rho}{\partial t}\right)^{(n)} \\ &= -\frac{1}{\rho^{(n)}}\frac{c_1\rho^{(n-2)} + c_2\rho^{(n-1)} + c_3\rho^{(n)}}{c_4} \end{aligned} \tag{16}$$

where $c_1 = 1$; $c_2 = -[(\Delta t^{(n-1)} + \Delta t^{(n)})/\Delta t^{(n)}]^2$; $c_3 = c_2^2 - 1$; and $c_4 = -(\Delta t^{(n-1)} + \Delta t^{(n)}) - c_2\Delta t^{(n)}$. The time derivative term of the potential, $(\partial'\Phi/\partial t)^{(n)}$, can been numerically calculated by $\Phi^{(n)}$ and $\Phi^{(n-1)}$, and hence $\Phi^{(n)}$ and $\Phi^{(n-1)}$ must be calculated by

integration of the velocity field numerically. In order to avoid numerical error when doing this numerical integration, Eq. (6) is used to compute $(\partial'\Phi/\partial t)^{(n)}$ distribution. Thus, Eq. (6) takes the form

$$\left(\frac{\partial'\Phi}{\partial t}\right)^{(n)} = \frac{1-\rho^{(n-1)^{\kappa-1}}}{\kappa-1} - \frac{1}{2}\vec{V}_r^{(n)^2} + \frac{1}{2}(\vec{V}_o^{(n)} + \vec{V}_e^{(n)})^2 \tag{17}$$

With $G_2^{(n)}$ obtained from Eq. (16) and $(\partial'\Phi/\partial t)^{(n)}$ obtained from Eq. (17), Step 1 is repeated until the solution converges.

Step 3 - at time step $(k = n+1)$:
Step 2 is repeated for time step $(n+1)$.

I-4. NUMERICAL EXAMPLES

The present scheme has been applied to a zero-thickness, rectangular wing with aspect ratio of 2. The half-span of the wing and the wake is divided into 10×6 and 10×10 quadrilateral panels, respectively. Each quadrilateral panel consists of 2 triangular panels. The one-half of the computational domain is divided into $23 \times 9 \times 9$ field volume elements in x, y and z directions, respectively. Two numerical examples are presented as mentioned before. The first one is the acceleration motion and the second one is the pitching motion.

Acceleration Motion

In this numerical example, the wing is given an acceleration motion at an angle of attack of 5 degrees. The translation Mach number is given by

$$M_o(t) = 0.7 + 0.1t \tag{18}$$

where t is given by $n\Delta t$; n is the time step index and Δt is the time step size, which is numerically chosen as 0.1 for this case. The time history of the surface pressure distributions is presented in Figures 2a through 2c at $z = 1.11$, $z = 4.44$ and $z = 7.78$, respectively. The surface pressure distributions at $M_o(t) = 0.80$ over different wing across-sections are shown in Figure 2d. It can be seen that the shock is located in the region between $x = 0.67$ and $x = 1.67$. The chord length is 10 units with the leading edge at x=0. Due to the lack of experimental data and other computational results, an integral equation solution[9] for steady flow is provided in Figure 2a to serve as an reference solution.

Transonic Pitching Motion

In this numerical example, the wing is given an forced transient pitching motion at a transonic translation Mach number of 0.7. In this case, the contribution of the rotation of the moving frame of reference has been included, which is represented by $\vec{\Omega} \times (\vec{r} - \vec{r}_p)$. To simplify the problem, only a pitching motion in xy-plane is considered. The angle of attack is given by

$$\alpha(t) = 5^o + 0.5^o t \tag{19}$$

where Δt is numerically chosen as 1 in this case. The time history of the surface pressure distributions is presented in Figures 3a through 3c at $z = 1.11$, $z = 4.44$ and $z = 7.78$,

respectively. The surface pressure distributions at $\alpha(t) = 10^o$ over different wing across-sections are shown in Figure 3d. The results presented here are self-explanatory, which show that the present scheme can capture unsteady effects, although the accuracy of the solution is to be determined.

I-5. CONCLUDING REMARKS

An unsteady integral equation scheme based on the full-potential equation formulation for transonic flows is developed. The scheme is capable of handling general unsteady motion in three dimensions, although only an acceleration motion and a pitching motion in xy-plane are implemented and tested. The computational results show that the scheme is capable of capturing shock and unsteady effects. The scheme is very stable, and the solution converges within 1-5 iterations per time step. The number of iterations for convergence decreases when t increases. The large time steps ($\Delta t = 1.0$, or 0.1) used in the present computations make this scheme very efficient for unsteady flow solutions. It is necessary to emphasize that the main aim here has been a demonstration of unsteady three-dimensional transonic flow computations using integral equation method, and hence the accuracy of the solution is less of a priority at this time.

PART II: STEADY FLOWS WITH SHOCK-FITTING

SUMMARY

This part presents the development of an integral equation shock-fitting field-panel method for three-diemnsion (3D) transonic flows. In this method, the full-potential equation, written in the form of the Poisson's equation, is solved by integral equation field-panel method. The solution consists of a wing surface source panel integral term, a field-volume integral term of compressibility over a small limited domain, and a shock panel integral term. Due to the non-linearity of flows, solutions are obtained through an iterative procedure. Instead of using a field-panel refinement procedure, a shock-fitting technique is used to fit the shock. Finally, numerical examples are provided to demonstrate the accuracy of the method. The major differenc of this part from the previous part is that this part deals with non-zero thickness wing flows and the accuracy of the solution is discussed.

II-1. INTRODUCTION

As mentioned in the previous part, integral equation methods for transonic flows have been developed by several investigators[1-14] during the past few years for steady airfoils, steady wing and aircraft configurations and unsteady airfoils. These methods solve either full-potential or transonic small disturbance equations using both surface and field panels. The shock capturing technique was applied in these methods. The method of Ref. 11 solves the full-potential equation for two-dimensional transonic flows, where both shock-capturing and shock-fitting techniques are applied. The capability of capturing shocks with shock-capturing technique and improvement of the shock with shock-fitting technique was presented in the Ref. 11. The method is efficient and engineering accurate. In this part a method for computing steady 3-D flows is presented along with numerical examples to demonstrate the capability, accuracy and the potential of the present IE scheme for subsonic and transonic flow computations. The method is the extension of the steady 2D method of Ref. 11 to three-dimensional flows. In order to use a coarse grid, which is particularly important in 3D calculations, the shock-fitting technique is applied to the present transonic flow calculations.

II-2. FIELD-PANEL FORMULATION

II-2.1 Governing Full-Potential Equation

The non-dimensional steady full-potential equation is given by:

$$\nabla^2\Phi = G_1 \tag{20}$$

with

$$G_1 = -\frac{\nabla\rho}{\rho}\cdot\vec{V} \tag{21}$$

and

$$\rho = [1 + \frac{\kappa-1}{2}(1-u^2-v^2-w^2)]^{\frac{1}{\kappa-1}} \tag{22}$$

where the characteristic parameters, ρ_∞, a_∞ and c have been used; a is the speed of the sound, ρ the density, and c the wing root-chord length; and Φ is the velocity potential ($\nabla\Phi = \vec{V} = (u,v,w)$), G_1 the compressibility, and κ the gas specific heat ratio.

Equation (20) is not in the conservative form but in the form of the Poisson's equation. By writting the full-potential equation in the Poisson's form, the nonlinearity of the transonic flows can be treated as non-homogeneity and in terms of the IE solution, this non-linearity is represented by field volume integral term. And hence the classical surface- panel method can be extended into field-panel method for non-linaer flows. The experience[1-14] has shown that such non-conservative formulation has produce accurate solutions as long as the shock is not very strong.

II-2.2 Boundary Conditions

The general boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, and wake kinematic and dynamic conditions as mentioned in Part

I. For the present non-lifting flows, the only surface no-penetration condition and infinity condition are needed and they are given by:

$$\vec{V} \cdot \vec{n}_g = 0 \qquad \text{on} \quad g(\vec{r}) = 0 \tag{23}$$

and

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r}) = 0 \quad \text{and} \quad w(\vec{r}) = 0 \tag{24}$$

where $\vec{n}_g$ is the unit normal vector of the wing surface, $g(\vec{r}) = 0$.

II-2.3 IE Solution

By using the Green's theorem, the integral equation solution of Eq. (20) in terms of the velocity field is given by

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty & \\ - \frac{1}{4\pi}\int\int_g \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d ds(\xi,\eta,\zeta) & \\ + \frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d d\xi d\eta d\zeta & \\ + \frac{1}{4\pi}\int\int_S \frac{q_S(\xi,\eta,\zeta)}{d^2}\vec{e}_d ds(\xi,\eta,\zeta) &\end{aligned} \tag{25}$$

where $\vec{V}_\infty$ is the free-stream velocity; q is the surface source distribution; the subscript, S, refers to the shock surface; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$. It can be seen that the infinity condition, Eq. (24), is automatically satisfied by the integral equation solution, since the integrals become zero when d is large enough.

II-2.4 Field-Panel Discretisation

The formulation presented here can be easily extended to general lifting flows by including surface and wake vortex-panel integral terms, although the present computations are only made to symmetric non-lifting flows. In this non-lifting computational model, the wing surface is represented by a number of uniform rectangular source panels. A uniform rectangular parallelopiped type of field-volume panels are also used throughout the flow field. Constant surface and volume source (q and G) distributions are assumed over wing / shock surface panels and field volume panels. The discretized integral equation solution

in terms of surface and field-volume panels then becomes

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty \\ &- \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG} q_{g_{i,k}} \int\int_{g_{i,k}} \frac{1}{d^2}\vec{e}_d ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV} G_{i,j,k} \int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{e}_d d\xi d\eta d\zeta \\ &+ \frac{1}{4\pi}\sum_{j=1}^{MS}\sum_{k=1}^{NS} q_{S_{j,k}} \int\int_{S_{j,k}} \frac{1}{d^2}\vec{e}_d ds(\xi,\eta,\zeta)\end{aligned} \tag{26}$$

where the indices, i, j and k refer to the surface and field panels; $LG \times NG$ is the total number of wing surface panels; $LV \times MV \times NV$ is the total number of field panels; and $MS \times NS$ is the total number of shock surface panels. A sketch of the computational model is given in Figure 4, while the detailed wing surface panelling is given in Figure 5 where the exact number of wing surface panels is shown for a typical case.

II-3. COMPUTATIONAL SCHEME

II-3.1 Iterative Scheme

Due to the nature of the non-linearity of transonic flows, solutions are obtained through an iterative procedure, where the wing surface source strength and the compressibility over selected volume elements are updated through each iteration. The solution procedure follows the successful form of Ref. 11 of two-dimensional computations. Here only the treatment of shocks for transonic flow is described.

II-3.2 Shock-Fitting Technique

It should be mentioned that mathematically the second (volume) integral term of Eq. (26) includes all compressibility effects including shock discontinuity. Since a relative coarse grid is used in the present IE computational domain where only 10 field panels are used over the wing chord, the contribution of the shock discontinuity is extracted from this volume integral term and it is represented explicitly by the third integral term of Eq. (26). It is very important to use coarse grid in 3D calculations, since the integral calculations over 3D field panels are very expensive. The strength of shock panels, q_S,is equal to the difference of normal velocity across the shock. This can be shown by integrating Eq. (20) over an infinitesimal volume around an infinitesimal area of the shock surface and applying the divergence theorem, one gets

$$\Delta(\nabla\Phi) = V_{2n} - V_{1n} = G_1\epsilon \tag{27}$$

where ϵ is the infinitesimal thickness normal to the shock surface. By letting $G_1\epsilon = q_S$ and using Rankine-Hugoniot relation, one finally obtains

$$q_S = \left[\frac{(\kappa-1)M_{1n}^2+2}{(\kappa+1)M_{1n}^2} - 1\right]V_{1n} \tag{28}$$

where the subscripts 1 and 2 refer to the conditions ahead and behind of the shock, respectively; and the subscript n refers to the normal component to the shock. The purpose to use Rankine-Hugoniot relation is to introduce the effect of entropy change across the shock sinec the full- potential formulation uses isentropic flow assumption which is not true in the shock region.

The constantly distributed, piece-wise continuous (in flow direction) oblique shock panels are used. The slope of shock panels is determined by the so called $\theta - \beta - M$ relation as given by

$$tan\theta = 2cot\beta[\frac{M_1^2 sin^2\beta - 1}{M_1^2(\gamma + cos2\beta) + 2}] \tag{29}$$

where θ is the flow-deflection angle, and β is the shock angle.

In the present calculation, the shock panel term, the last term of Eq. (26), becomes active only after the sonic line (and hence the shock location) is fixed. In other words, the shock-capturing technique is first used to locate the shock, where the Murman-Cole type-difference scheme is used in consistent with the mixed-nature of transonic flows. The use of the Murman-Cole scheme is equivelent to the introducing of the artificial dissipation. The use of this artificial dissapation scheme within a shock-fitting scheme seems contradictory since some of their effects will cancel each other. But if we consider the shock-fitting as the way to give a correct inviscid shock and the Murman-Cole scheme as the way to give the artificial viscous effect, then the use of the Murman-Cole scheme with shock-fitting scheme will give a correct viscous shock, this is what it should be.

II-4. NUMERICAL EXAMPLES

The presesnt scheme is applied to rectangular wings of symmetric sections with different aspect ratios (AR) at different free-stream Mach numbers. The half-span of the wing surface (including upper and lower surfaces) is divided into (20to24) $\times$ (6to12) quadrilateral panels depending on wing geometry and free-stream conditions. The one-half of the computational domain is divided into 20 $\times$ 16 $\times$ (9to18) field volume elements in chord, normal and span directions, respectively. The size of the computational domain is from $2c \times 1.5c \times 2.25c$ to $2c \times 1.5c \times 3c$ for different AR values in the chord, normal and span directions, respectively. It should be noted that the both surface- and field-panel sizes in chord (flow) direction are as large as 10% of chord length. Only symmetric flows with zero angle of attack (non-lifting flows) are considered, and attached flow assumption is also made.

The first numerical example is made to the flow around a wing with a 5% thick circular arc section of $AR = 3$ at free-stream Mach number of 0.7, a shock-free subsonic flow, where the non-linearity effect is small. Figure 6a is the calculated surface local Mach contoures which shows that the flow is purly subsonic. The calculated surface pressure coefficients are presented in Figure 6b in terms of contours and Figure 6c in terms of line plot, along with the computational results obtained by the non-linear LTRAN3 TSD FD code[15] and by the linear SOUSSA IE code[16] at three span stations located at 0%, 50% and 90% of semi-span. As the figure shows, the presently calculated pressure distributions are in close agreement with the non-linear LTRAN3 results over the entire wing surface and agreement with the linear IE SOUSSA results except the discrepancy over leading and trailing edges. The convergence of the solution is obtained by checking the relative error of surface pressure distribution over each iteration, and for this shock-free flows, the number

of iterations for a convergent solution is 6.

The second numerical example is made to a transonic flow around a wing with a 6% thick circular arc section of $AR = 4$ at a free-stream Mach number of 0.908. In order to show the capability of shock- fitting, the solutions obtained with and without shock-fitting are presented in Figure 7a through Figure 7d. Figures 7a and 7b are the surface Mach contours and surface pressure coeffcient contours without shock- fitting, respectively, where the shock is diffused but the supersonic flow region is clearly seen in the Figure 7a. Figures 7c and 7d are the solutions with shock-fitting, where the shock is clearly predicted. The effect of shock-fitting is self-explanatory from these figures. In order to verify the accuracy of the shock-fitting, the calculated results are ploted in Figure 7e along with the other reference solution. The calculated pressure distributions compare very well with a TSD FD result[17] and another IE result[18] except the discrepancy at the station near the wing tip. The location and the strength of the shock are correctly predicted by the present method. For the present transonic flow case, 16 iterations are used to get a convergent solution, where the first 10 iterations are used to locate shock and additional iterations are used to fit shock.

The discrepancy near wing tip may be caused by the different tip shapes used in different computational models, and hence to have different tip- release effects. To investigate this effect, the present computation is made for this case with different wing tip thickness. Figures 8a through 8c are the results obtained by tapering off wing tip to 75%, 50% and 0% of the value at root section, respectively. Figures 8a - 8c show the variation of the surface pressure coeffieients at the station near tip due to the tip- release effect.

While above two examples are for the flows around sharp leading-edge wings, the next two examples present the results for flows around round leading-edge wings. Figures 9a through 9c are the calculated results for the transonic flow around a wing with 6% thick NACA64A006 section of $AR = 4$ at a free-stream Mach number of 0.877. The calculated pressure at root section is compared with the 2-D experimantal data[19] and the comparison shows a good agreement.

The last numerical example is for the transonic flow around a wing with a symmetric 10.6% thick NACA64A010A section of $AR = 4$ at a free-stream Mach number of 0.8. The symmetric NACA64A010A wing section is obtained by averaging the upper and lower surface coordinates of NACA64A010A airfoil. The calculated pressure distribution is compared with the TSD finite-difference solution[20] as shown in Figure 10c, while FIgures 10a and 10b are local Mach and surface pressure contours, respectively. The comparison shows that the two solutions have good agreement in terms of both location and strength of the shock.

II-5. CONCLUDING REMARKS

An integral equation field-panel method based on the full-potential equation formulation for transonic flows is presented. The method can be extended for handling flows around general three dimensional configurations, although only non-lifting cases are implemented and tested. The calculated wing surface pressure distribution is reasonably correct including the location and the strength of the shock. As an alternative to the grid refinement, the shock-fitting technique applied here does give a correct shock both in location and in strength. The present IEM is effective in terms of the number of iterations

compared with those of FDM and FVM, althought the computational cost per IE iteration is more expensive than those of FDM and FVM. The large grid size (for example, $\Delta x = 0.1c, \Delta y = 0.1c, \Delta z = 0.25c$) used here makes the scheme even more efficient. The CPU time for a typical transonic flow case is around 165 seconds on Cray-YMP computer.

PART III: MPP IMPLEMENTATION AND PERFORMANCE ANALYSIS

SUMMARY

This part presents the massively parallel processing (MPP) implementation of integral equation calculations and the performance analysis. For both two- dimensional (2D) and three-dimensional (3D) flows, integral equation panel method computer codes are converted into parallel CM-FORTRAN codes. Comparative study of computational performance of CM-2 / CM-5 and Cray-YMP computers is made. The performance results are obtained on CM-2 with 8k, 16k and 32k processors, and on CM-5 with 32, 64 and 128 nodes along with those on Cray-YMP with a single processor. The comparison of the performance indicates that the parallel CM-FORTRAN code out-performs the equivalent serial FORTRAN code for most cases tested.

III-1. INTRODUCTION

In recent years, the processors of conventional vector supercomputers seem to be approaching the limit in computational speed inherent in their technology. However, the need for even faster computations continues to grow. As a consequence, massively parallel computers are being developed as a possible solution. Massively parallel computers, such as CM-2 and CM-5, are families of parallel computer architectures which may provide orders of magnitude improvement in computation performance in the near future over today's fastest conventional supercomputer. In fact the CM-5 computer with a maximum 16k nodes installed is a 2 TFLOPS computer in theory.

Computational fluid dynamics (CFD) is one of the areas which need super-fast computational power. The massively parallel computers has potential to become the main computational tool for CFD; it may replace the conventional supercomputers in the near future. The integral equation panel method has the nature for processing data in aparallel computing environment, and hence it is important to investigate this nature.

III-2. CM-2/CM-5 AND CM-FORTRAN

The Connection Machine CM-5 system is a scalable distributed-memory multiprocessor system. The major hardware elements of the system include front-end computers to provide developing and execution environments and a parallel processing unit, which consists of multiple nodes, to execute parallel operations. It supports both the SIMD (Single Instruction Multiple Data) data parallel and MIMD (Multiple Instruction Multiple Data) message passing programming models. The maximum possible configuration for a CM-5 system is $16k$ nodes, where $k = 1024$. The CM-5 used under the present study at NASA Ames Research Center has 128 nodes installed. Each node has 32 MB of memory, one SPARC processor and four vector processors for a theoretical peak performance of 128 MFLOPS. Therefore the CM-5 with 128 nodes has a theoretical peak performance of 16 GFLOPS.

Similar to the CM-5, the CM-2 is another MPP machine which was built before the CM-5. The CM-2 supports SIMD data parallel computing mode. The parallel processing unit contains up to 64k single-bit physical processors. The CM-2 used under this study at NASA Ames Research Center has 32k processors. The aggregate peak performance for this 32k CM-2 is about 2 GFLOPS.

The CM-FORTRAN language is an implementation of FORTRAN 77 supplemented with array-processing extensions from the standard FORTRAN 90. These array-processing features map naturally onto the data parallel (for SIMD model of parallel programming) architecture of the CM-5 system, since the CM- FORTRAN allows array elements to be evaluated simultaneously. The most important difference of CM-FORTRAN from FORTRAN 77 is the treatment of entire arrays as objects, thus explicit indexing in CM-FORTRAN is not always necessary. For example, it is not necessary to write Do-Loops or other such control constructs to have the operation repeated for each element of arrays. On the other hand for message passing models of parallel programming (MIMD), the program may be written in FORTRAN 77 along with message passing routines.

III-3. INTEGRAL EQUATION PANEL METHODS

For incompressible flows, the governing equation is given by the Laplace equation

which can be obtained from Eq. (20) after setting the compressibility to zero,

$$\nabla^2 \Phi = 0 \tag{30}$$

where Φ is the velocity potential, $\vec{V} = \nabla\Phi$.

The integral equation solution of Eq. (30) for source panel method, in terms of velocity field ($\vec{V} = \nabla\Phi$), is given by

$$\begin{aligned} \vec{V}(x,y,z) &= \vec{V}_\infty \\ &- \frac{1}{4\pi} \sum_{i=1}^{LG} \sum_{k=1}^{NG} \int\int_{g_{i,k}} \frac{q_{i,j}(\xi,\eta,\zeta)}{d^2} \vec{e}_d \, ds(\xi,\eta,\zeta) \end{aligned} \tag{31}$$

where the subscript ∞ refers to the free-stream condition, $LG \times NG$ is the total number of panels; $q_{i,j}$ is the wing-surface source distribution, which is unknown to be determined by applying boundary condition; ds is the infinitesimal surface area; the $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$, and where $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$. It should be mentioned that for two-dimensional flows, the surface integrals of Eq. (31) become line ntegrals and the surface panels become line segments.

The wing-surface zero-normal-velocity boundary condition is applied at each control point of all panels,

$$\vec{V}(x,y,z) \cdot \vec{n} = 0 \qquad \text{on} \quad CP_{I,K} : I = 1, NL; K = 1, NG \tag{32}$$

Applying Eq. (31) to Eq. (32), a system of equation is obtained,

$$[A]\{q\} = \{B\} \tag{33}$$

where $[A]$ is $N \times N$ aerodynamic influence coefficient matrix, and $N = LG \times NG$; $\{q\}$ is a $N \times 1$ unknown vector matrix containing q_j for $j = 1$ to N; and $\{B\}$ is a $N \times 1$ known vector matrix which is contributed from $\vec{V}_\infty$.

The solution procedure of the problem using source panel method involves three major steps: (1) evaluation of integrals for N^2 times to construct matrices $[A]$ and also $[B]$; (2) solving the resulting dense linear system of Eq. (4); and (3) post-processing of aerodynamic calculations. It should be noted that the Step (1) involves evaluating a large number of integrals. The total number of integrals can be very large for aerodynamic problems, for example, it can be in the order of 10^8 if $LG \times NG = 100 \times 100$. An important feature of the Step (1) is that the calculation for each (x,y,z) and each (ξ,η,ζ) can be performed simultaneously for all (x,y,z) and all (ξ,η,ζ). This feature of panel method calculation leads itself in a natural way for processing data in a SIMD parallel computing environment.

III-4. PARALLEL FORTRAN IMPLEMENTATION

The serial FORTRAN codes are converted into parallel CM-FORTRAN codes where the CMAX translator is used to partially convert the 3D code. However the most of the conversion is done manually.

List 1a is the program list of the subroutine for evaluating influence coefficient matrix, [A], and the matrix $\{B\}$ in serial FORTRAN. It is noted that this subroutine is nothing but a two-level Do-loops with a If-conditional statements, which provides for evaluating each element of [A] and $\{B\}$. When the code is in execution on Cray-YMP, the vectorization of the inner Do-loops is automatically done through the vectorization capability of the FORTRAN 77 compiler.

List 1b is the parallel CM-FORTRAM euqivalent of List 1a. A few things should be mentioned. First, no Do-loop is seen here since in CM- FORTRAN entire arrays are treated as objects and array elements are evaluated simultaneously. Second, the conditional If-statement is represented in the form of WHERE-ELSEWHERE-ENDWHERE format which allows the conditional processing to be done in parallel; Third, the CM-FORTRAN intrinsic function SPREAD is used here to create two-dimensional arrays from one-dimensional arrays by duplicating the elements in either row- or -column-directions as desired for easy implementing parallel processing of statements like, $X1(K,J) = X2(K) + X3(J)$; Fourth, temporary scale variables, such as DYJ, DXJ and so on in serial FORTRAN becomes two-dimensional arrays in CM-FORTRAN in order to implement parallel processing. However, such arrays increase the total memory requirement of CM- code significantly as compared with the serial FORTRAN code. For example, within the present investigation it has been found that the CM-FORTRAN code with 4096 panels exceeded the 2GB memory limit of the CM-2 computer with 16k processors.

The dense linear system of Eq. (33) is solved by the direct Gauss elimination method for 2D and 3D cases and by indirect Jacobi's iterative method for 2D case also. List 2a is the program list of serial FORTRAN version of Jacobi's iterative method, while List 2b is the parallel CM-FORTRAN version. From the list, it can be seen that the Jacobi's iteration method on CM-FORTRAN is fully parallelized.

List 3a and List 3b are program list of the subroutine for evaluating aerodynamic influence coefficients for three-dimensional flows in serial FORTRAN and parallel CM-FORTRAN version, respectively. Similar to List 1a and List 1b, the Do-loops in serial version are replaced by data parallel statement, and the If-conditional statement is replaced by parallel conditional statement. The major difference of 3D program from 2D program is that the 3D program is more complicated.

III-5. PERFORMANCE STUDY

Figure 11 shows the CPU time for solving linear system using Gauss elimination. On the CM-FORTRAN version, the system is solved by calling the Gauss elimination solver from the CMSSL library on the CM-2. The comparison shows that the CPU time required on CM-2 with 32k processors approaches to that required on Cray-YMP with the increase of the problem size. For example, the CPU time required on CM-2 with 32k processors is 0.125 seconds for $N = 32$ which is much larger than that for Cray-YMP of 0.00152 seconds; while this comparison becomes 277 seconds to 192 seconds when $N = 2048$. Therefore it can be expected that when the problem size becomes large enough the CM-2 with 32k

processors will near- or out-perform the Cray -YMP. But on the other hand, it is found that the direct method for large system of equations are very expensive for both serial and parallel versions.

Table 1 gives the detailed performance results for 2D calculation on Cray-YMP and CM-5 computers where the dense linear system is solved by a much more efficient indirect Jacobi's iterative method. The performance of CM-5 in terms of MFLOPS is the equivalent Cray-YMP performance. In Table 1, "Mat Coef" refers to evaluating aerodynamic enfluence coefficients; "Lin Syst" refers to solving linear system using Jacobi method; and "Total" refers to solving entire code. The sets of results from Table 1 have been extracted, and are presented in Figures 12-15.

Figure 12 shows execution time for evaluating aerodynamic influence coefficient matrix, $[A]$, and the matrix $[B]$ on Cray-YMP and CM-5 computers for different numbers of panels. It can be seen that the CPU execution time decreases with the increase of the number of CM-5 nodes after the size of the problem is large enough to fully use all nodes. For example when $N = 1024$, the CPU time of 0.216 seconds with 32 nodes is reduced to 0.114 seconds with 64 nodes, and then is further reduced to 0.061 seconds with 128 nodes. That is to say that whenever the number of nodes used is doubled, the CPU time is almost reduced by a factor of 2 - a near-perfect parallelization. It is also seen that when the problem size is large enough the CPU time required on CM-5, even with 32 nodes, is significantly (note that Log_{10}-axis is used for execution time !) less than that required on Cray-YMP.

Figure 13 is the CPU time for solving linear system using Jacobi iterations. The results tell us that, when the N is large enough, CM-5 out-performs Cray- YMP and the Jacobi method is very efficient.

Figure 14 shows the total CPU time for solving entire problem. The results are self-explanatory. Figure 15 is a partial reproduction of Figure 14 for performance results on Cray-YMP and CM-5 with 128 nodes, and it is represented in terms of MFLOPS. From this figure it is clearly seen that the CM-5 performs at about 2 GFLOPS when $N = 1024$. The speed achieved here is very encouraging, which is much faster than that achieved on Cray-YMP.

For three-dimensional flows, the performance results are listed in Table 2 through Table 5. Table 2 gives the detailed CPU time results on the Cray-YMP and the CM-5 to construct the matrices. It is founded that the CM-5 out-performs the Cray-YMP by small margin for $N = 24 \times 12$. When the size is increased to $N = 48 \times 24$ the CM-5 out-performs the Cray-YMP by a much larger margin, even with only 32 nodes.

Table 3 displays CPU time for the Gaussian elimination routine, and it tells us that the Cray-YMP out-performs the CM-5 for $N = 24 \times 12$. However its performance relative to the CM-5 decreases drastically when N increases; and in fact we can see that the CM-5 out-performs the Cray-YMP for $N = 48 \times 24$.

Table 4 lists the CPU time for post processing calculations where the parallel CM-FORTRAN version is not fully parallelized. Table 5 is the total CPU times and from the table it is seen that for the larger probel tested under this study, the CM-5 out-performs

the Cray-YMP.

III-6. CONCLUDING REMARKS

Source panel method computational codes for both 2D and 3D incompressible flows are successfully implemented on the MPP computers using CM-FORTRAN language. The linear system is solved by both direct Gauss elimination and efficient iterative Jacobi's methods. The detailed performance results are obtained and analysed. The parallel CM-FORTRAN code achieves a very high performance and for most of the cases tested here it out-performs Cray-YMP supercomputer, which is very encouraging. Through this study, it seems that the integral equation method is appropriate for parallel computation and a high performance can be achieved.

REFERENCES

1. Piers, W. J. and Sloof, J. W., "Calculation of transonic flow by means of a shock-capturing field panel method", AIAA Paper 79-1459, 1979.

2. Hounjet, M. H. L., "Transonic panel method to determine loads on oscillating airfoils with shocks", *AIAA Journal*, 1981, Vol. 19, No. 5, pp.559- 566.

3. Tseng, K. and Morino, L., "Nonlinear Green's function methods for unsteady transonic flows", *Transonic Aerodynamics*, (Ed. Nixon, D), 1982, pp.565-603.

4. Oskam, B., "Transonic panel method for the full potential equation applied to multi-component airfoils", *AIAA Journal*, 1985, Vol. 23, No. 9, pp.1327- 1334.

5. Erickson, L. L. and Strande, S. M., " A theoretical basis for extending surface-paneling methods to transonic flow", *AIAA Journal*, 1985, Vol.23, No. 12, pp.1860-1867.

6. Sinclair, P. M., "An exact integral (field panel) method for the calculation of two-dimensional transonic potential flow around complex configurations", *Aeronautical Journal*, June/July 1986, Vol. 90, No. 896, pp.227-236.

7. Kandil, O. A. and Yates, E. C., Jr., "Computation of transonic vortex flows past delta wings-integral equation approach", *AIAA Journal*, 1986, Vol. 24, No. 11, pp.1729-1736.

8. Madson, M. D., "Transonic analysis of the F-16A with under-wing fuel tanks: an application of the TranAir pull-potential code", AIAA Paper 87-1198, 1987.

9. Chu, L-C., "Integral equation solution of the full potential equation for three dimensional, steady, transonic wing flows", Ph.D. dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March 1988.

10. Sinclair, P. M., "A three-dimensional field-integral method for the calculation of transonic flow on complex configurations - theory and preliminary results", *Aeronautical Journal*, June/July 1988, Vol. 92, No. 916, pp. 235-241.

11. Kandil, O. A. and Hu, H., " Full-potential integral solution for transonic flows with and without embedded Euler domains", *AIAA Journal*, 1988, Vol. 26, No. 9, pp. 1074-1086.

12. Ogana, W., "Transonic integro-differential and integral equations with artificial viscosity", *Engineering Analysis with Boundary Elements*, 1989, Vol. 6, No. 3, pp. 129-135.

13. Kandil, O. A. and Hu, H., "Integral solution of unsteady full- potential equation for a transonic pitching airfoil", *Journal of Aircraft*, 1990, Vol. 27, No. 2, pp. 123-130.

14. Hu. H., "Full-potential integral solutions for steady and unsteady transonic airfoils with and without embedded Euler domains", Ph.D. Dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March,1988.

15. Guruswamy, P. and Goorjian, P. M.,"Comparisons between computations and experimental data in unsteady three-dimensional transonic aerodynamics, including aeroelastic applications", AIAA Paper 82-0690-CP, 1982.

16. Yates, E . C. Jr., Cunningham, H. J., Desmarais, R. N., Silva, W. A., and Drobenko, B., "Subsonic aerodynamic and flutter characteristics of several wings calculated by the SOUSSA P1.1 panel method", AIAA Paper 82-0727, 1982.

17. Bailey, F. R. and Steger, J. L., "Relaxation techniques for three- dimensional transonic flow about wings", AIAA Paper 72-189, 1972.

18. Tseng, K., "Application of the Green's function method for 2- and 3-dimensional steady transonic flows", AIAA Paper 84-0425, 1984.

19. Lambourne, N. C., et al., "Compendium of unsteady aerodynamic measurements", AGARD R-702, 1982.

20. Bland, S. R. and Seidel, D. A., "Calculation of unsteady aerodynamics for four AGARD standard aeroelastic configurations", NASA TM- 85817, 1984.

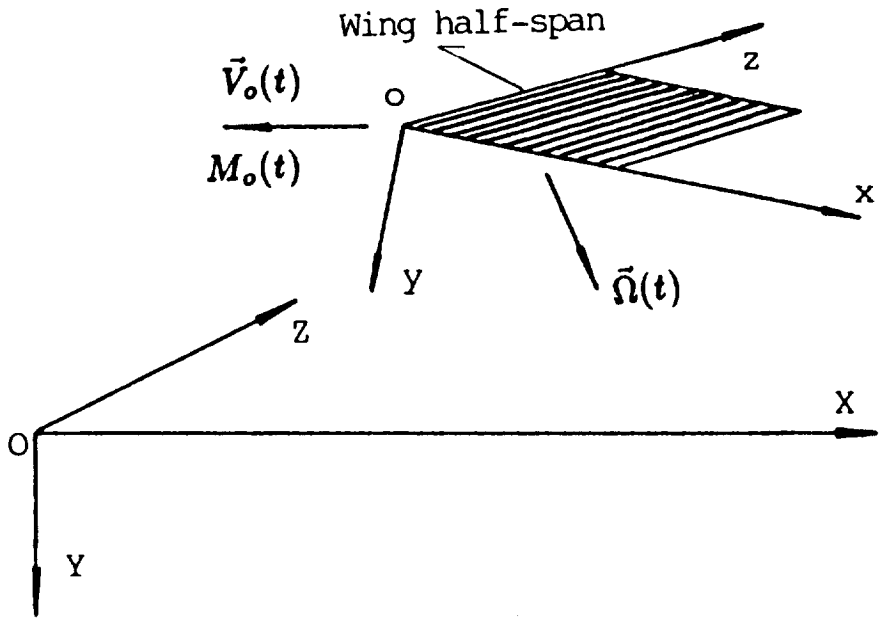


Figure 1. Frames of reference.

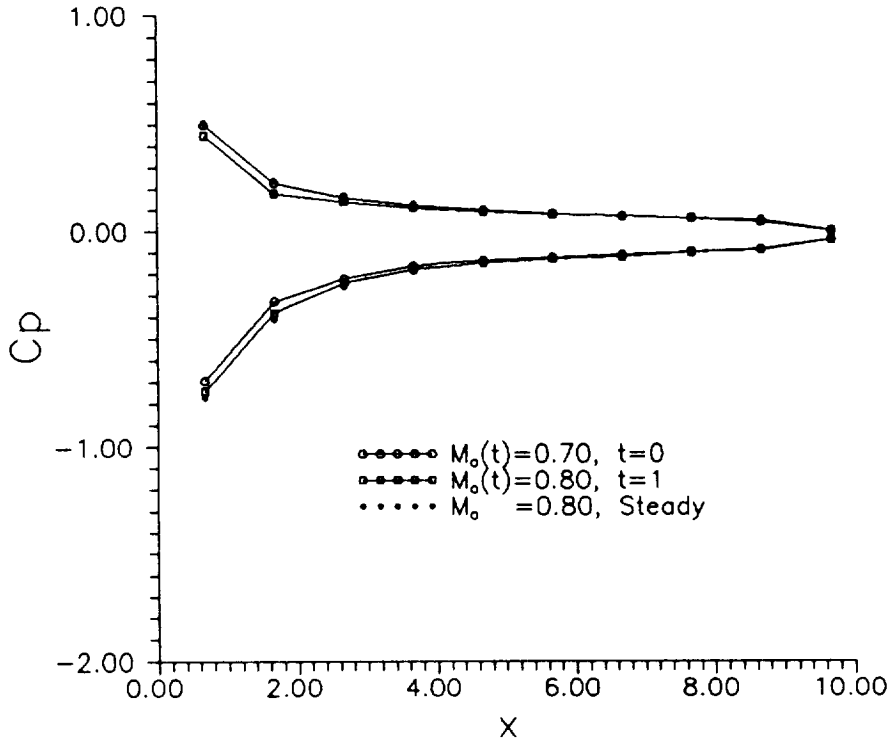


Figure 2a. Unsteady C_p history at $z = 1.11$, acceleration motion.

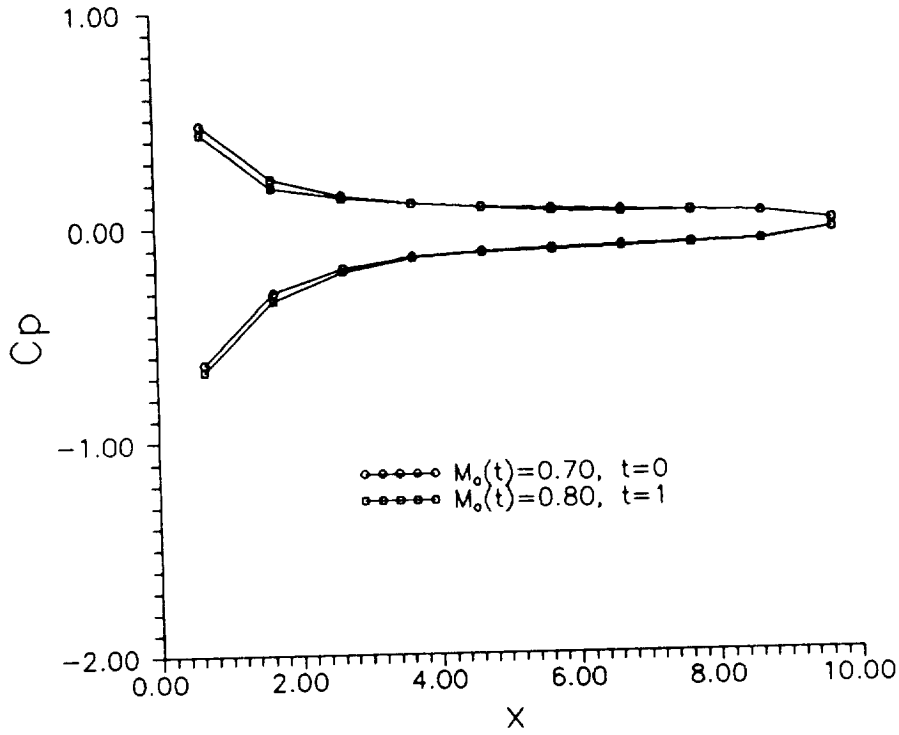


Figure 2b. Unsteady C_p history at $z = 4.44$, acceleration motion.

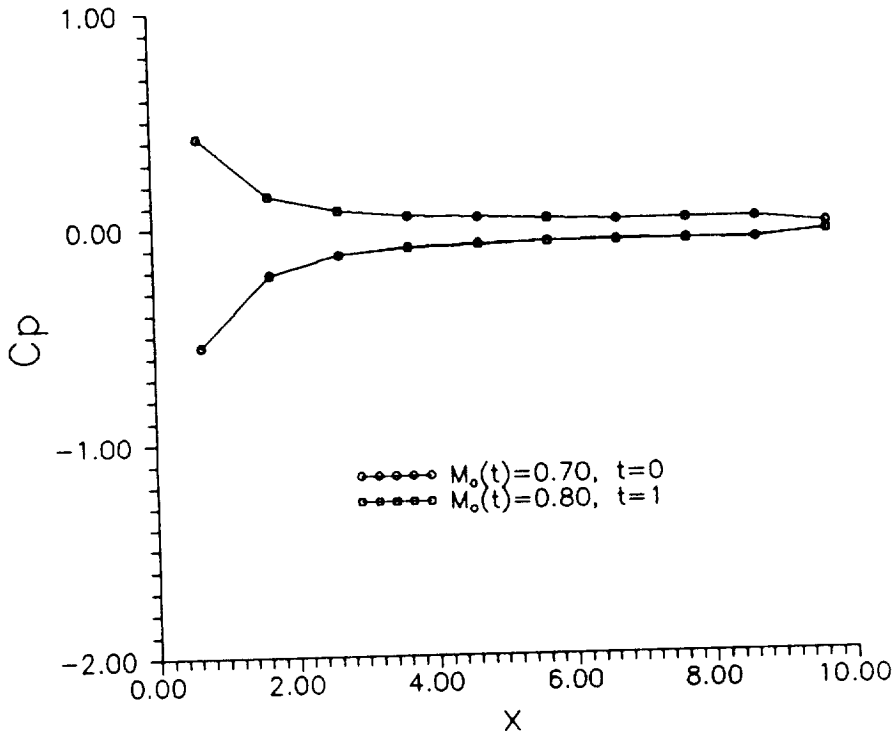


Figure 2c. Unsteady C_p history at $z = 7.78$, acceleration motion.

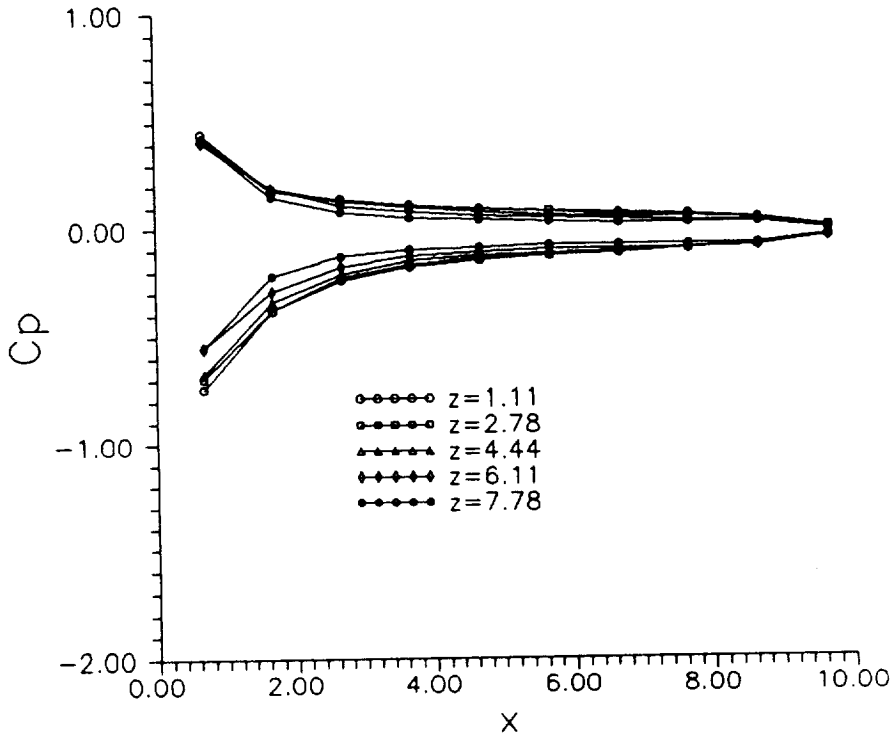


Figure 2d. C_p distribution at $M_o(t) = 0.8$ over wing surface, acceleration motion.

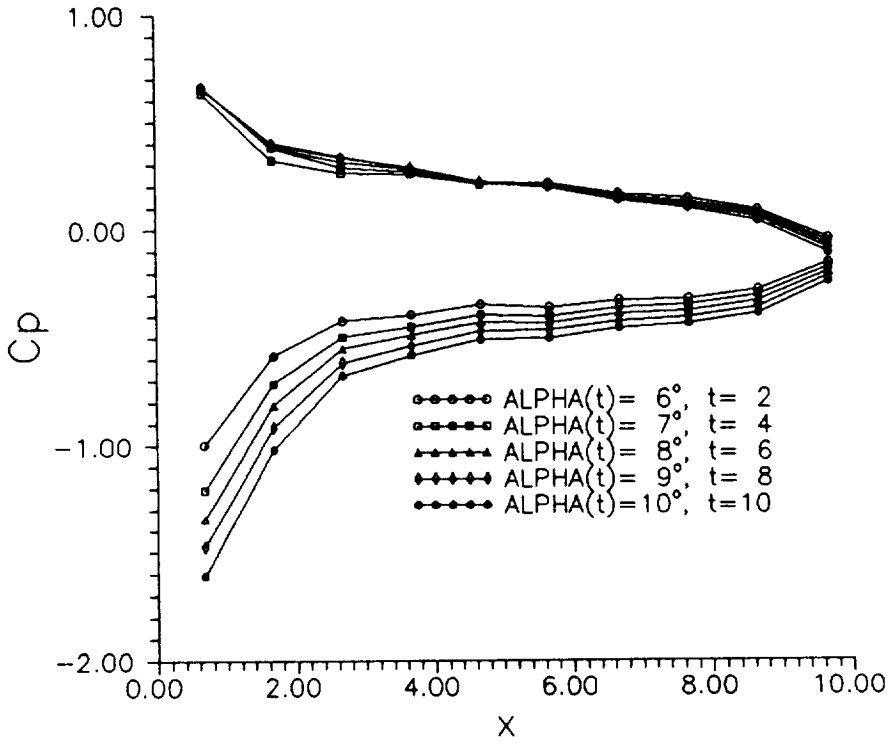


Figure 3a. Unsteady C_p history at $z = 1.11$, pitching motion.

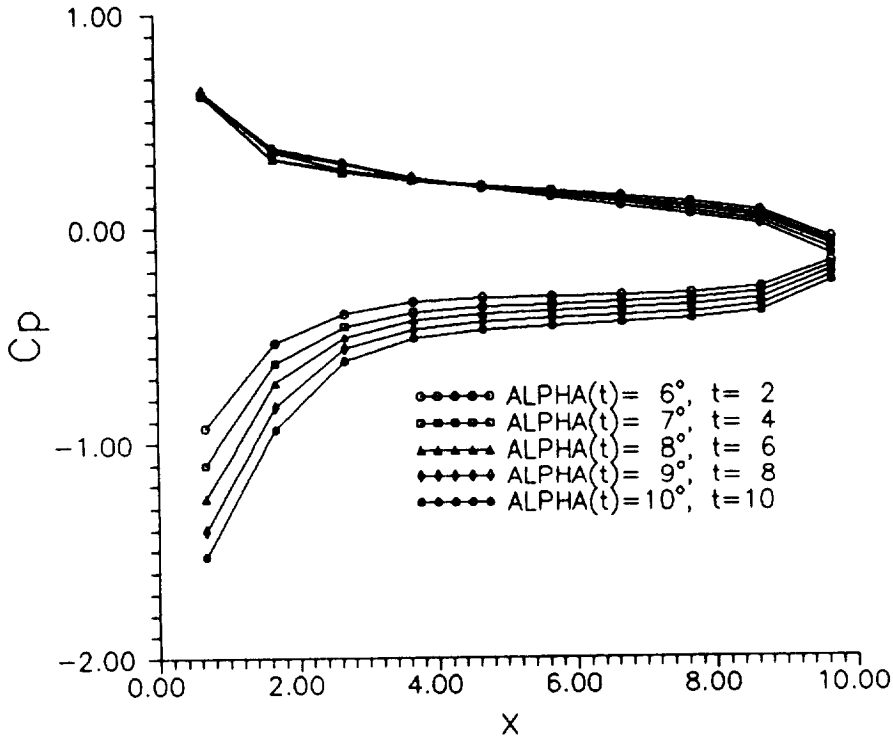


Figure 3b. Unsteady C_p history at $z = 4.44$, pitching motion.

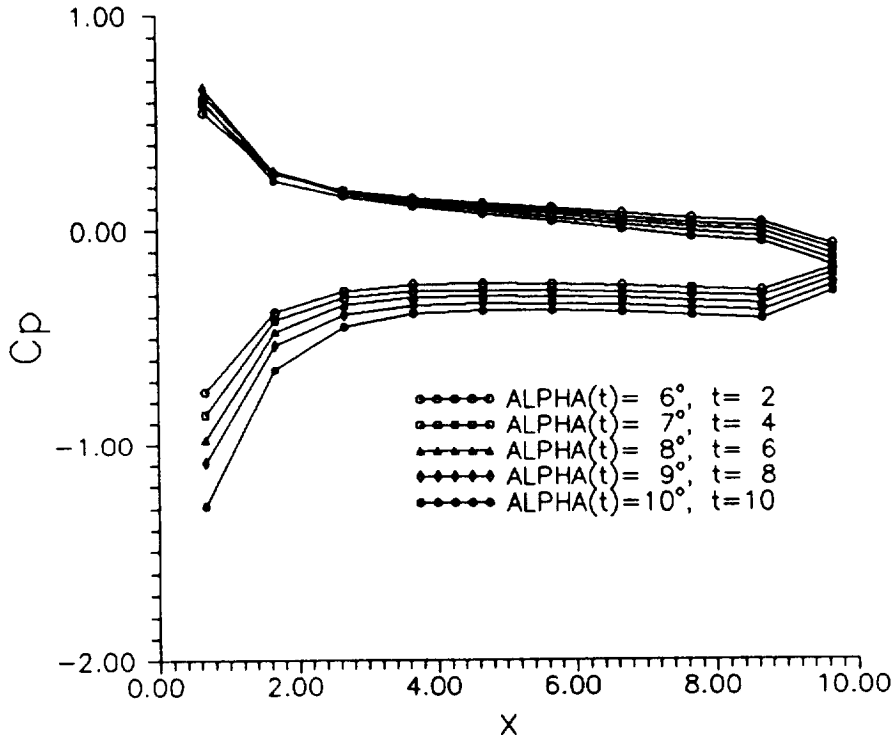


Figure 3c. Unsteady C_p history at $z = 7.78$, pitching motion.

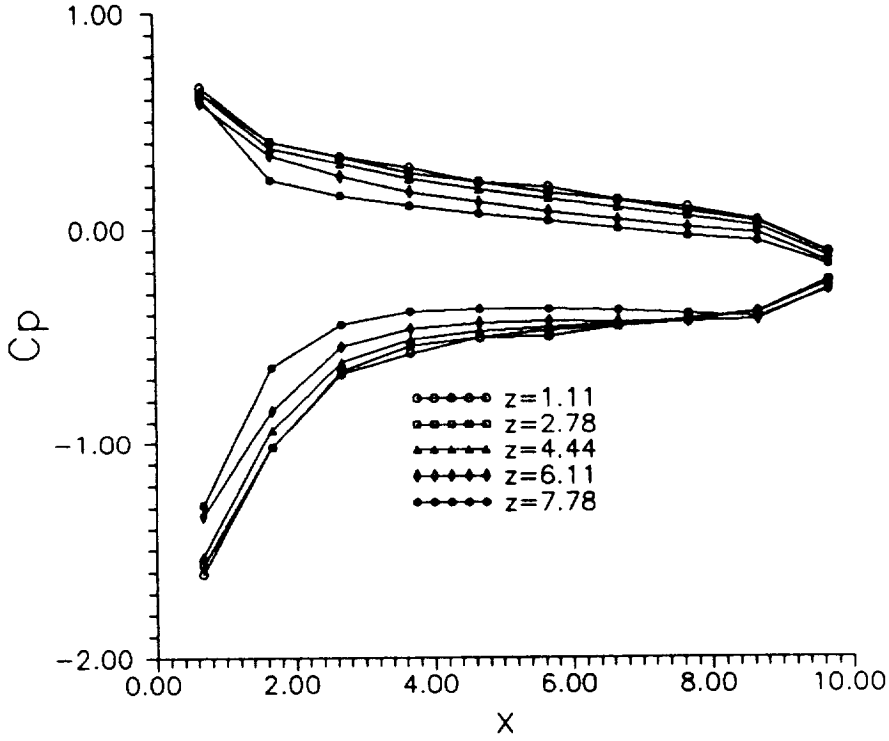


Figure 3d. C_p distribution at $\alpha(t) = 10^o$ over wing surface, pitching motion.

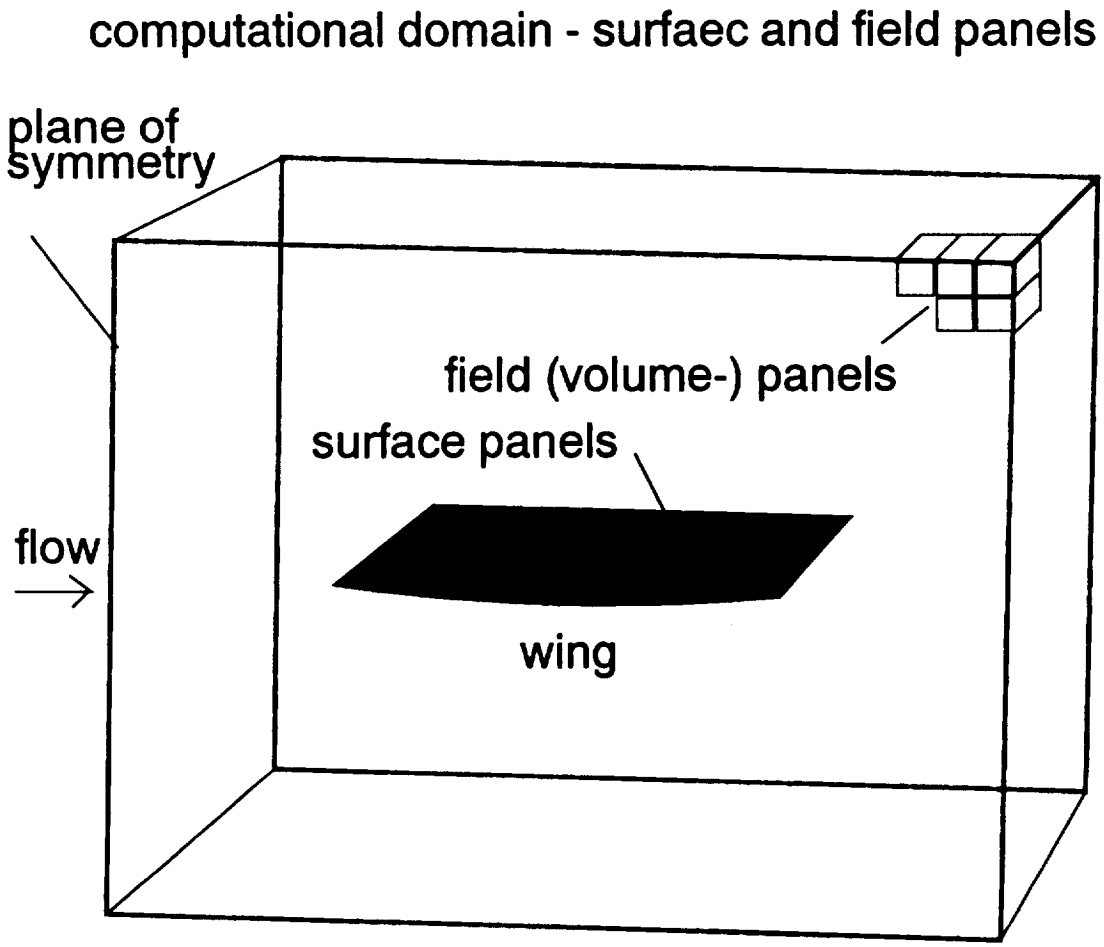


Figure 4. Computational model.

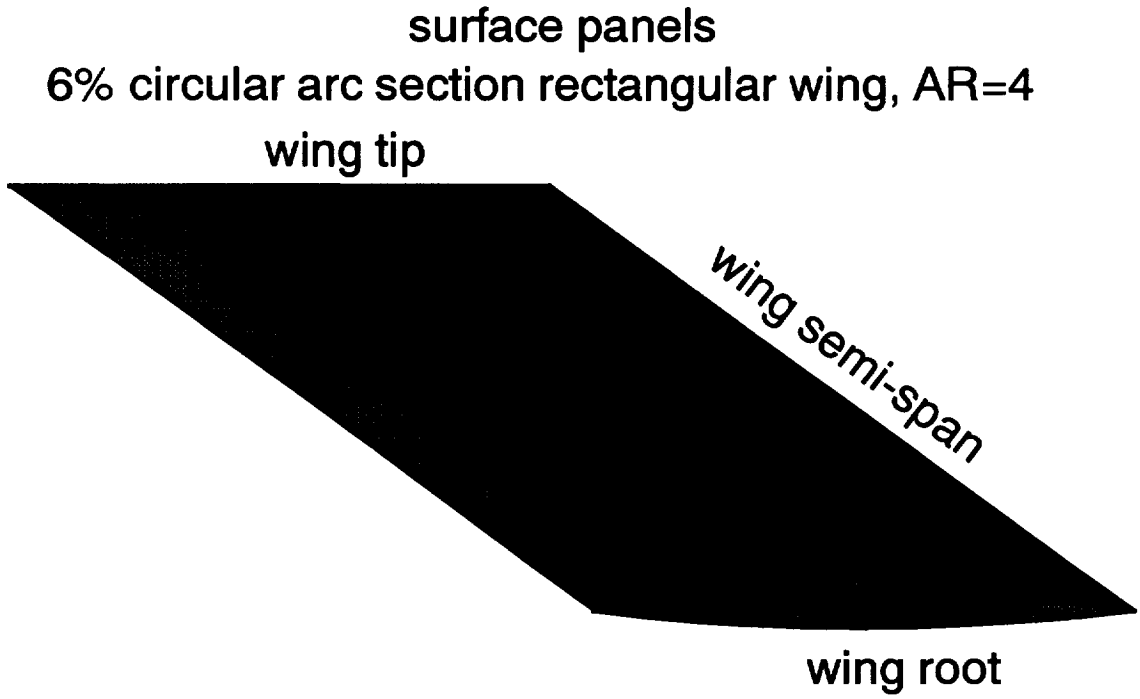


Figure 5. Surface panels.

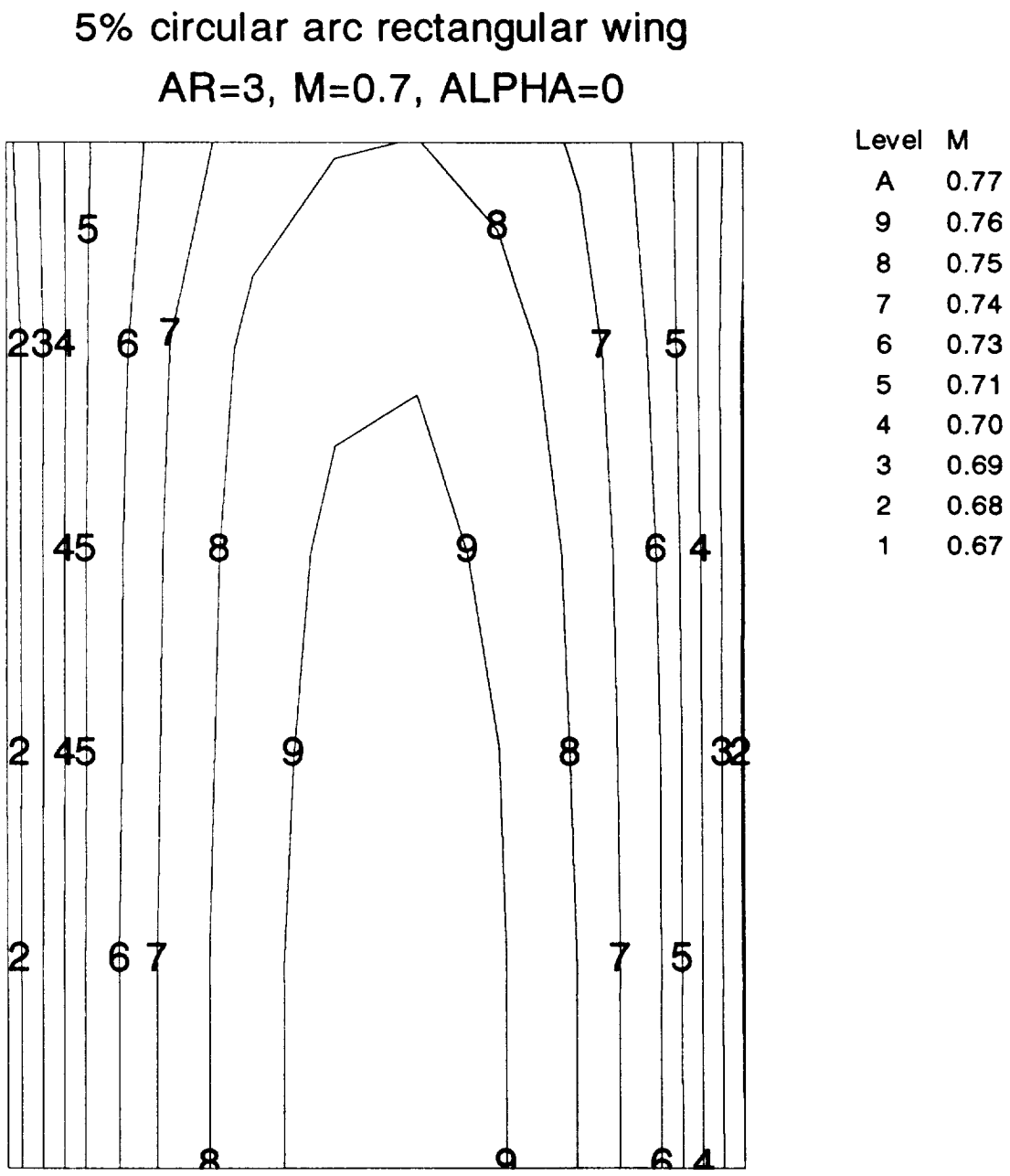


Figure 6a. Shock-free subsonic flow, surface local Mach contours.

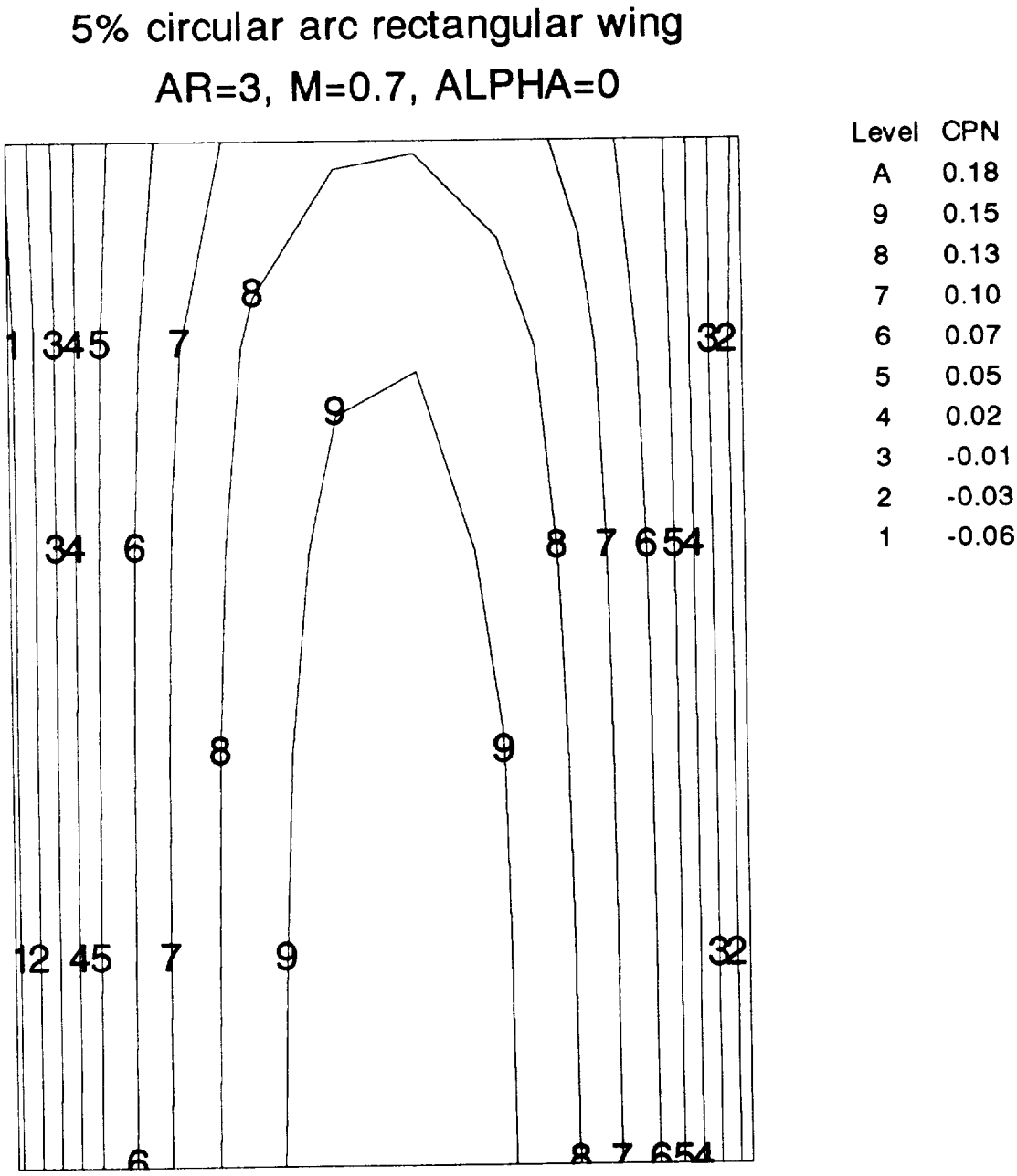


Figure 6b. Shock-free subsonic flow, surface pressure coefficient contours.

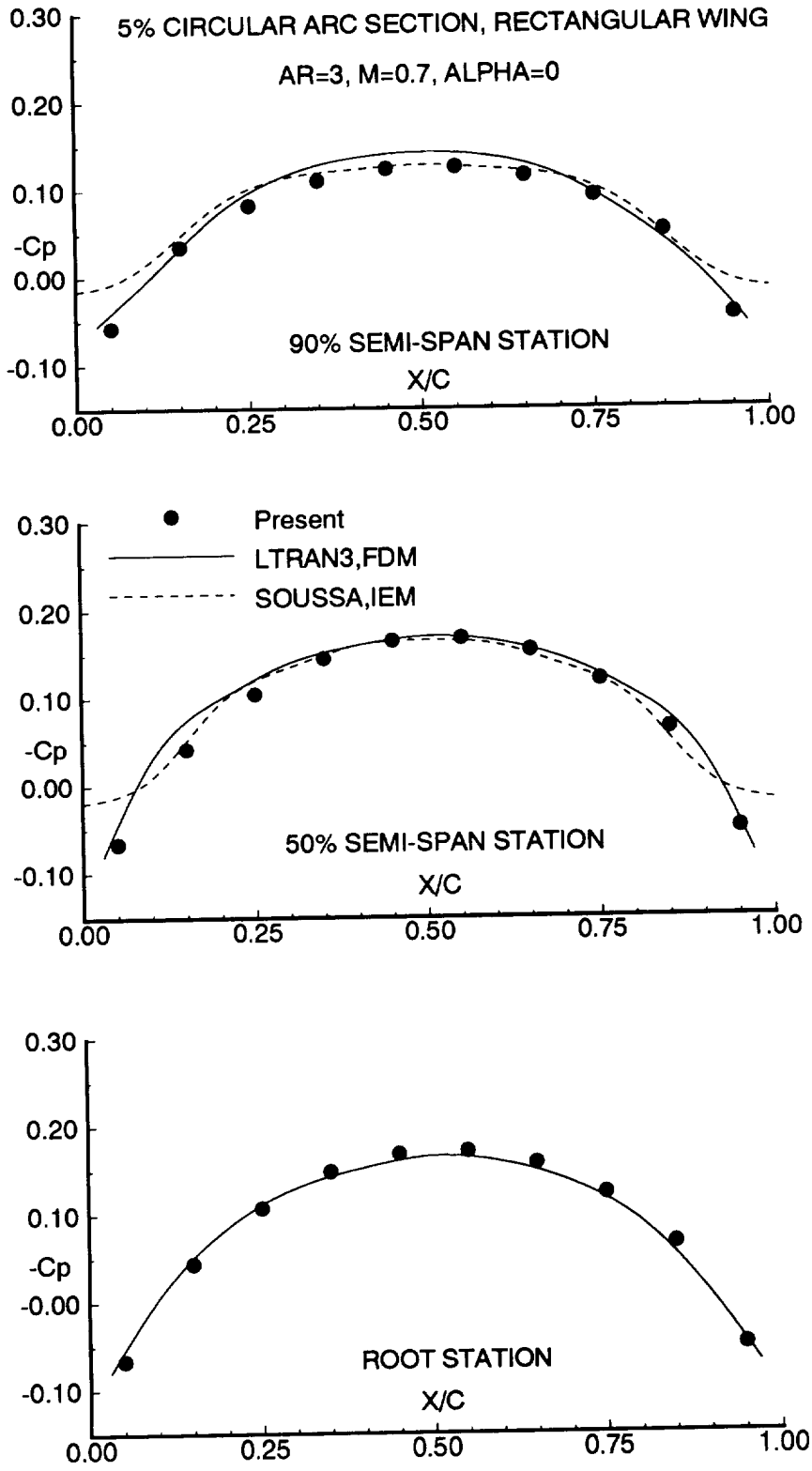


Figure 6c. Shock-free subsonic flow, surface pressure coefficients.

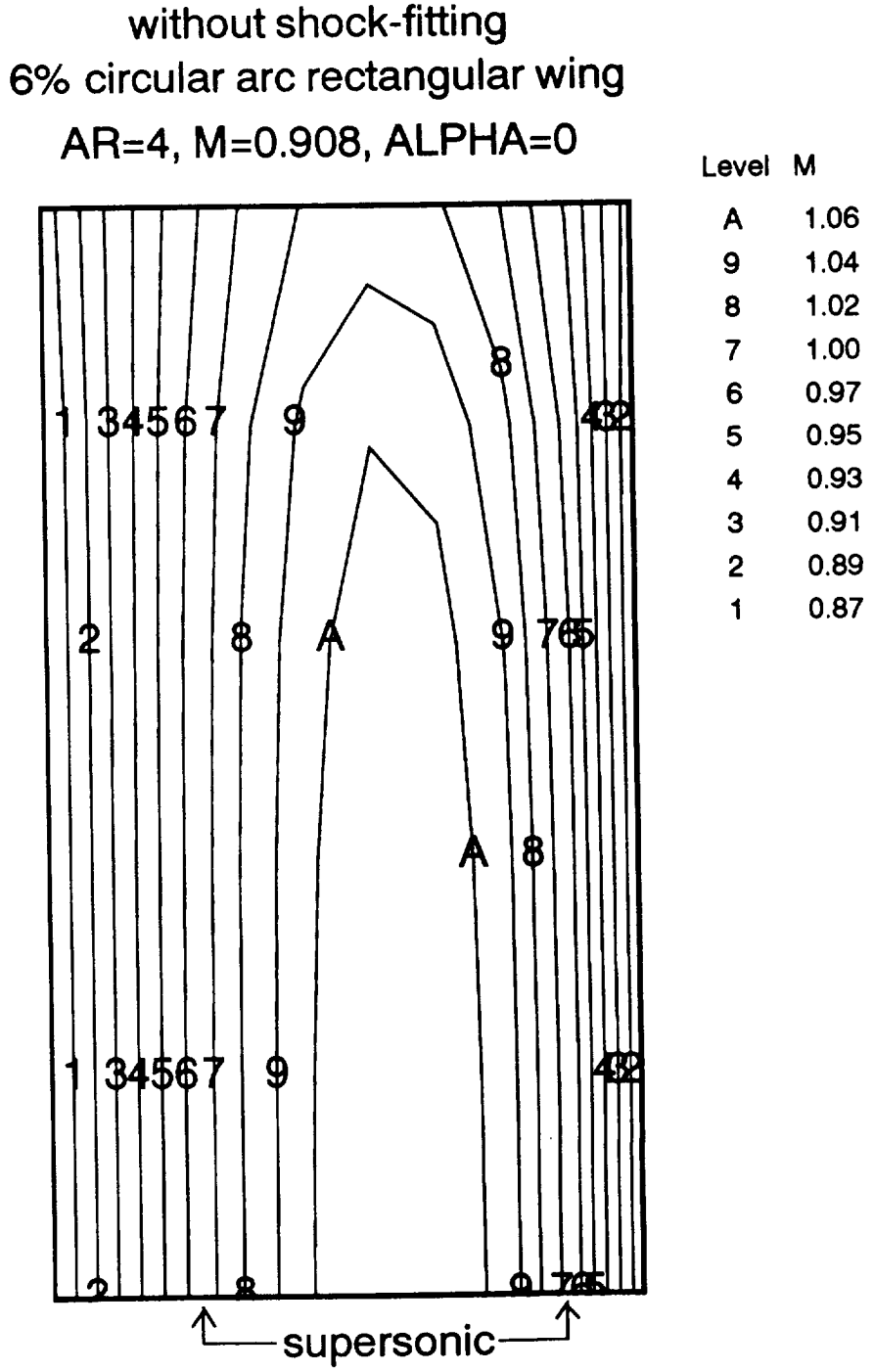


Figure 7a. Effect of shock-fitting, Mach contours, withou shock- fitting.

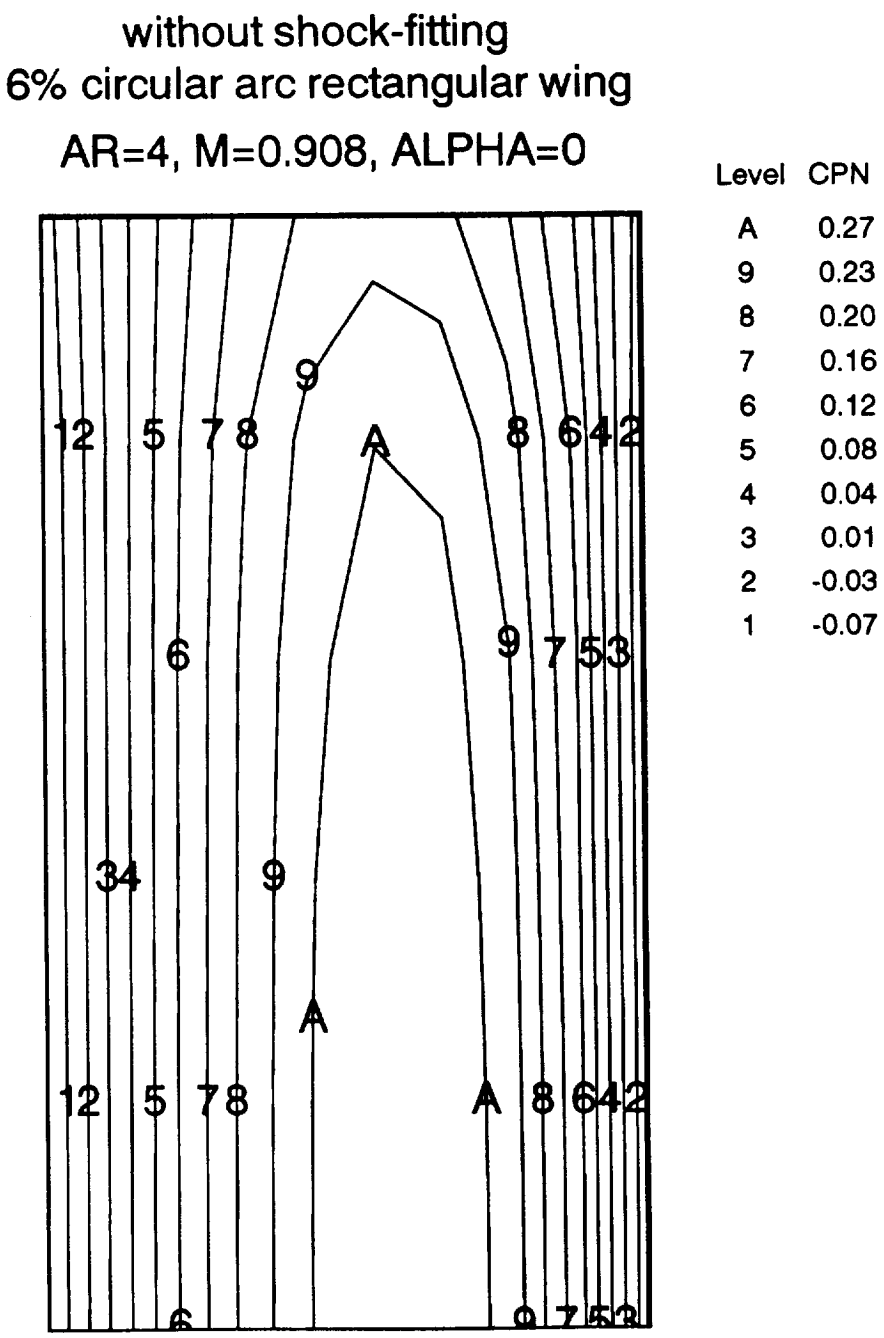


Figure 7b. Effect of shock-fitting, presure contours, without shock-fitting.

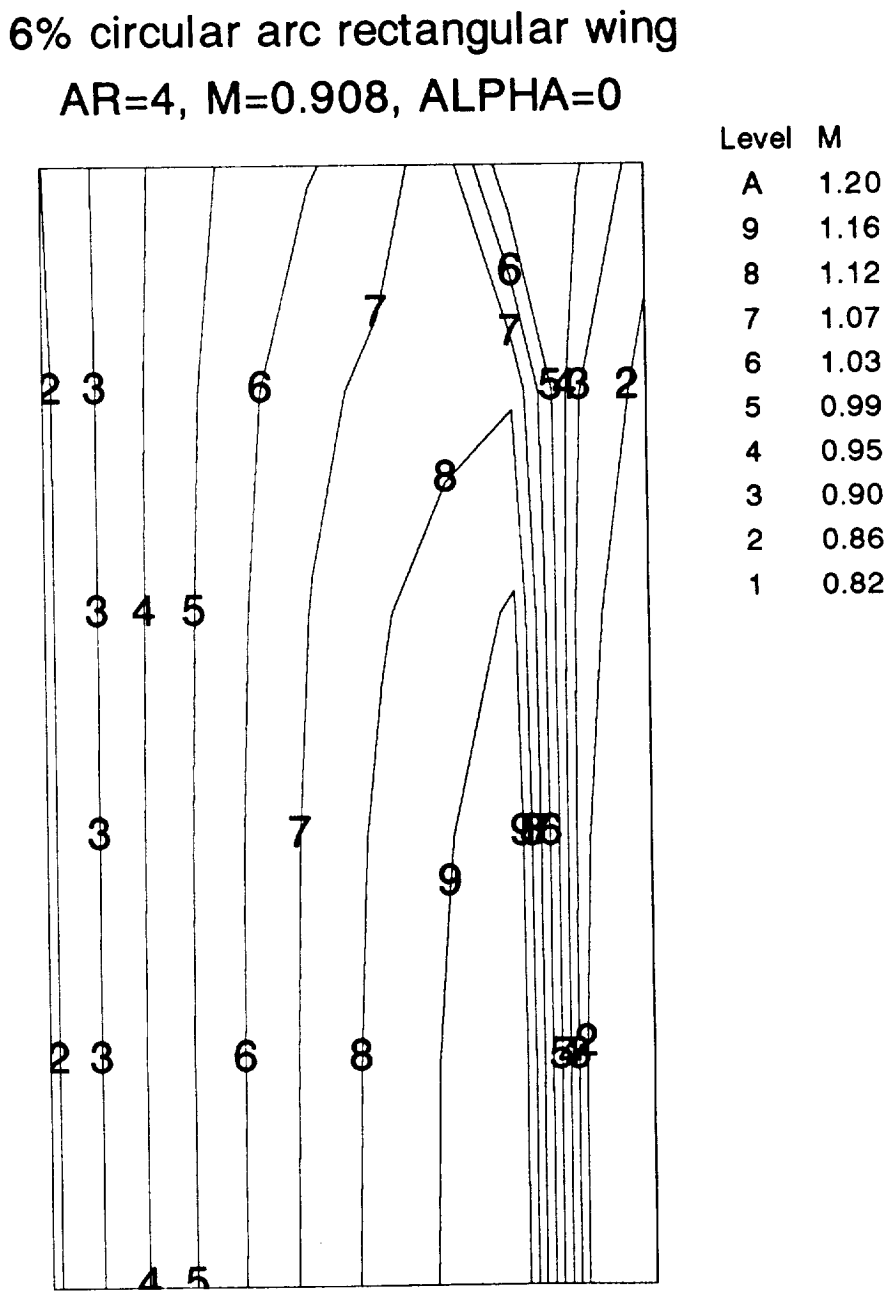


Figure 7c. Effect of shock-fitting, Mach contours, with shock fitting.

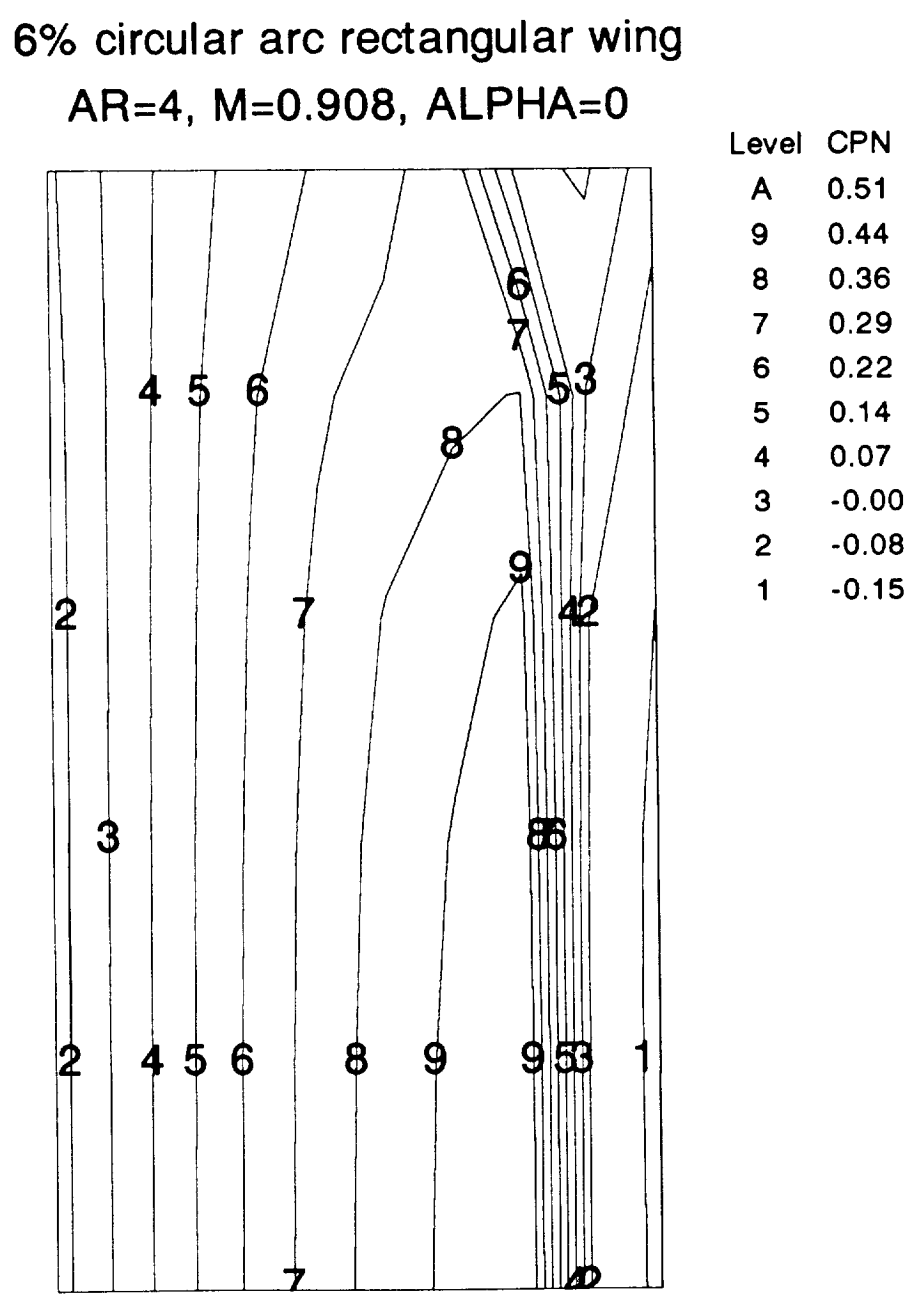


Figure 7d. Effect of shock-fitting, pressure contours, with shock-fitting.

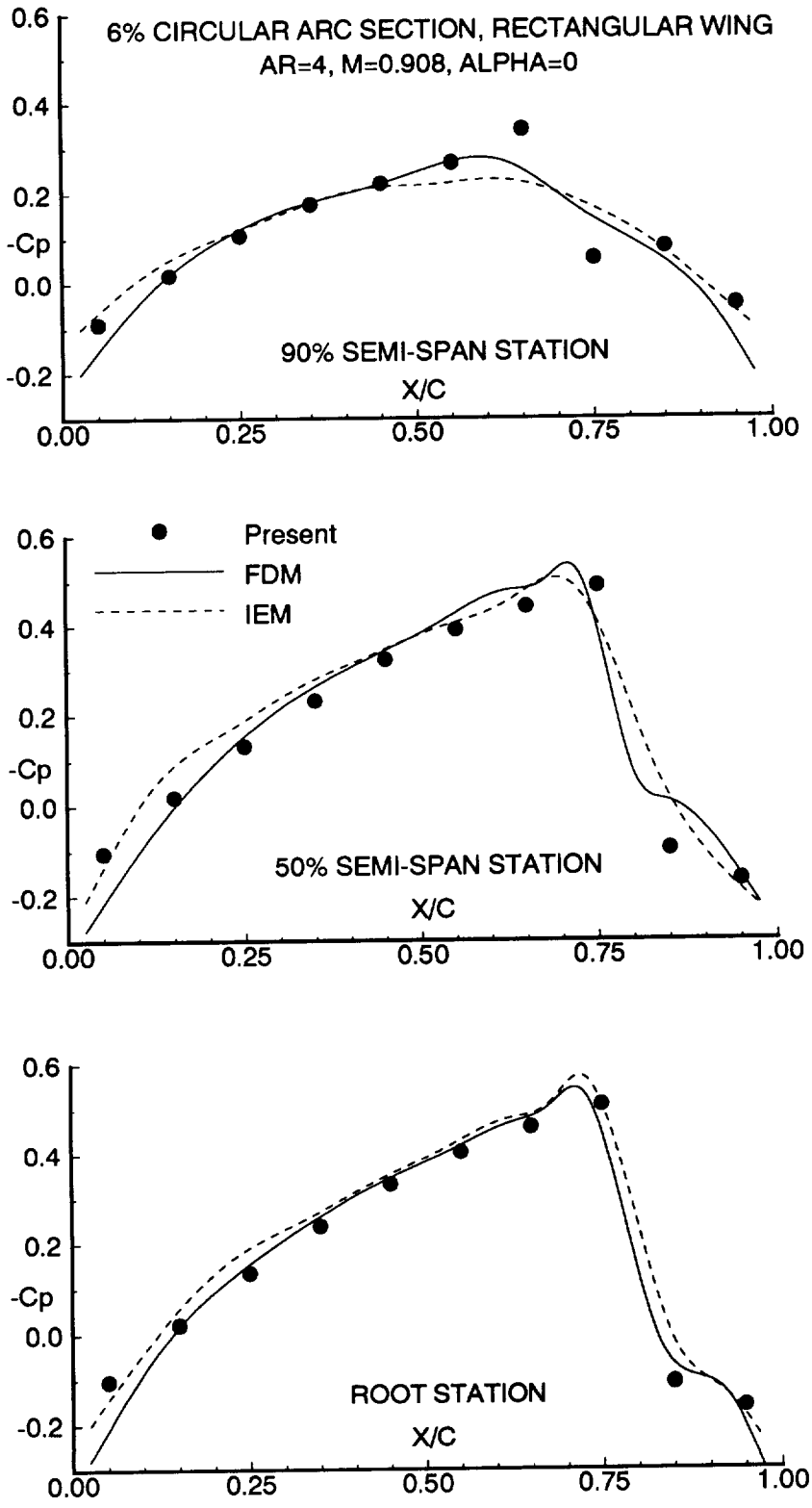


Figure 7e. Surface pressure coefficients, with shock-fitting.

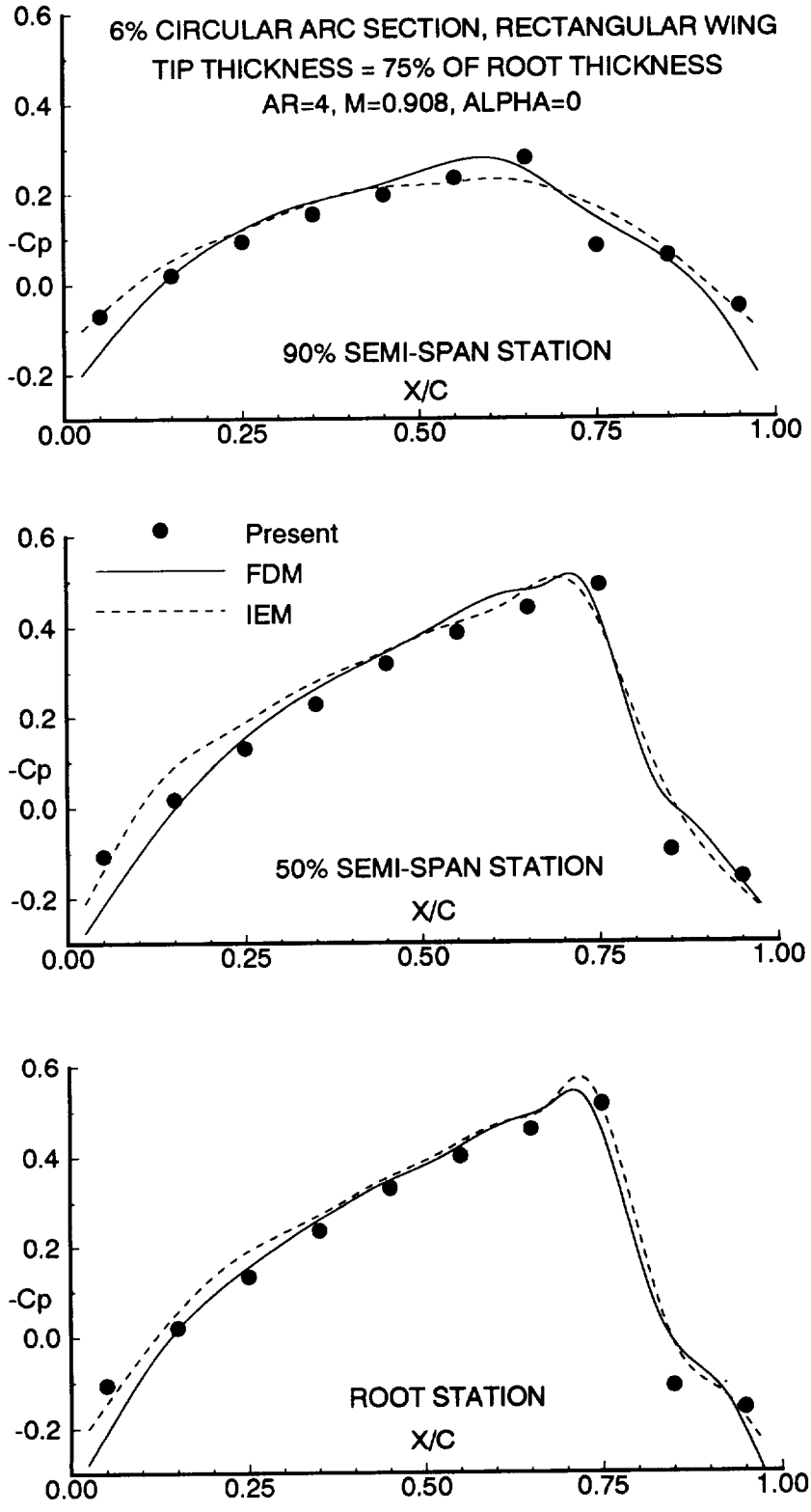


Figure 8a. Tip-release effect, tip thickness = 75% root thickness.

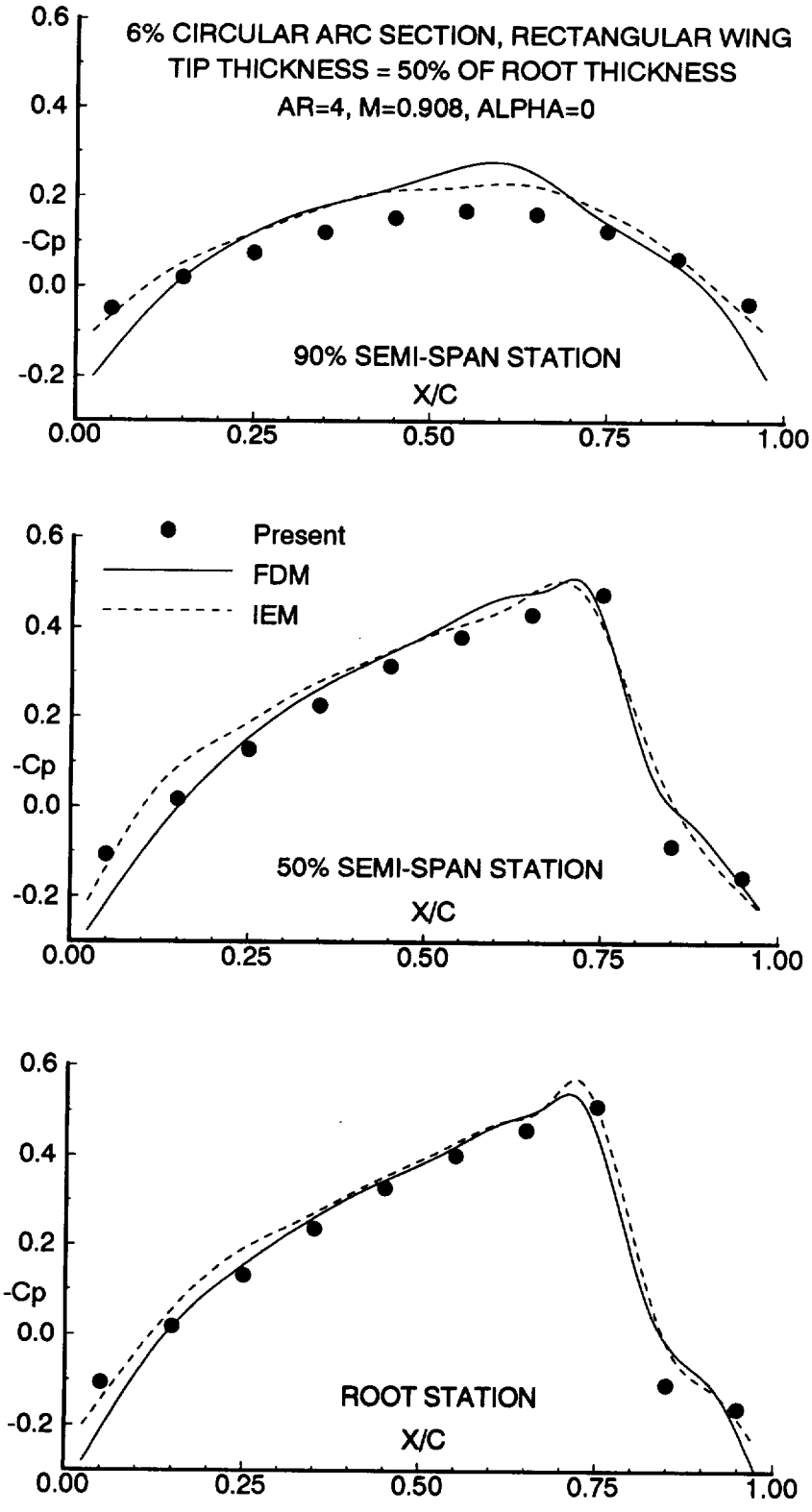


Figure 8b. Tip-release effect, tip thickness = 50% root thickness.

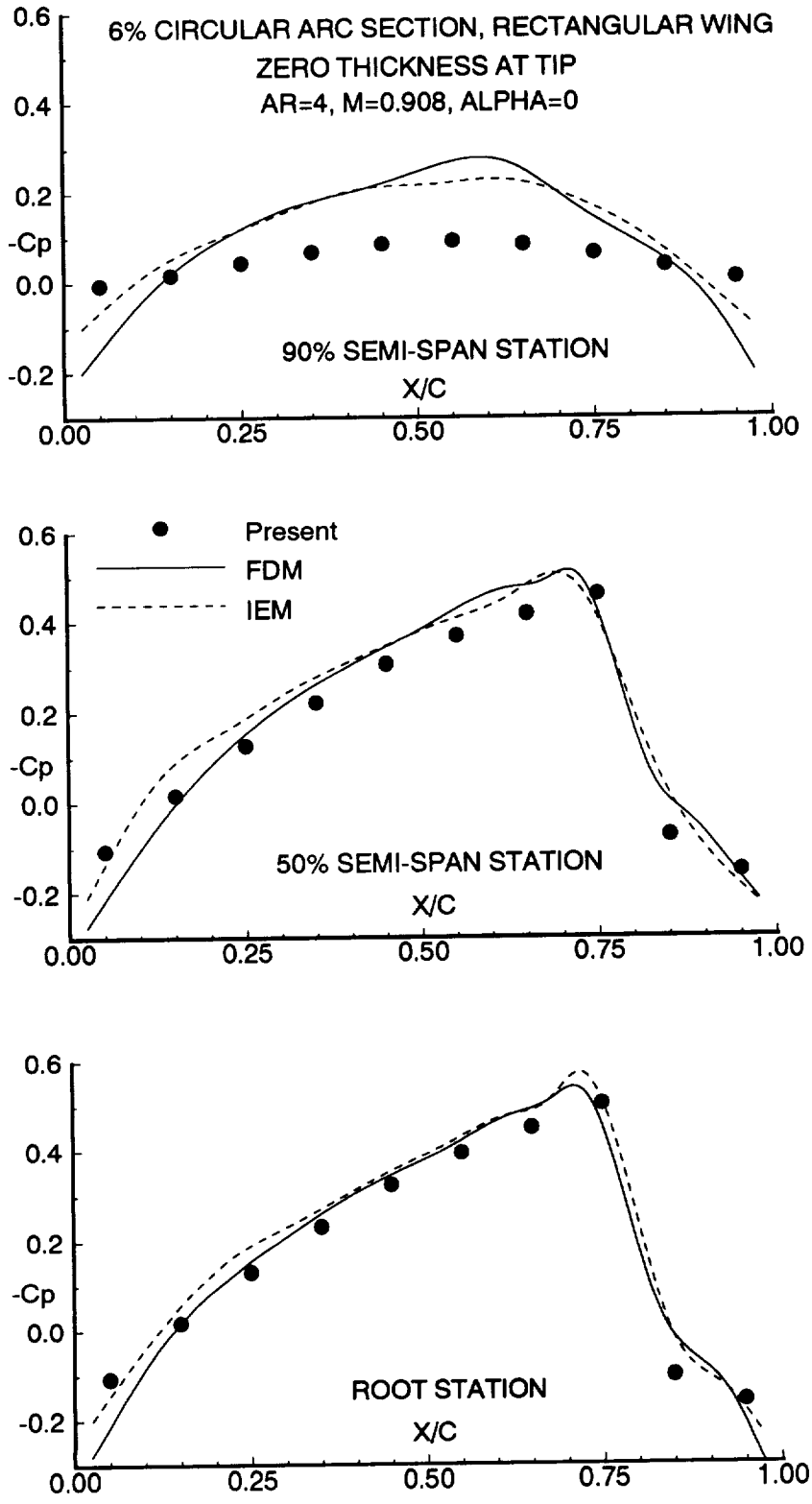


Figure 8c. Tip-release effect, tip thickness = 0.

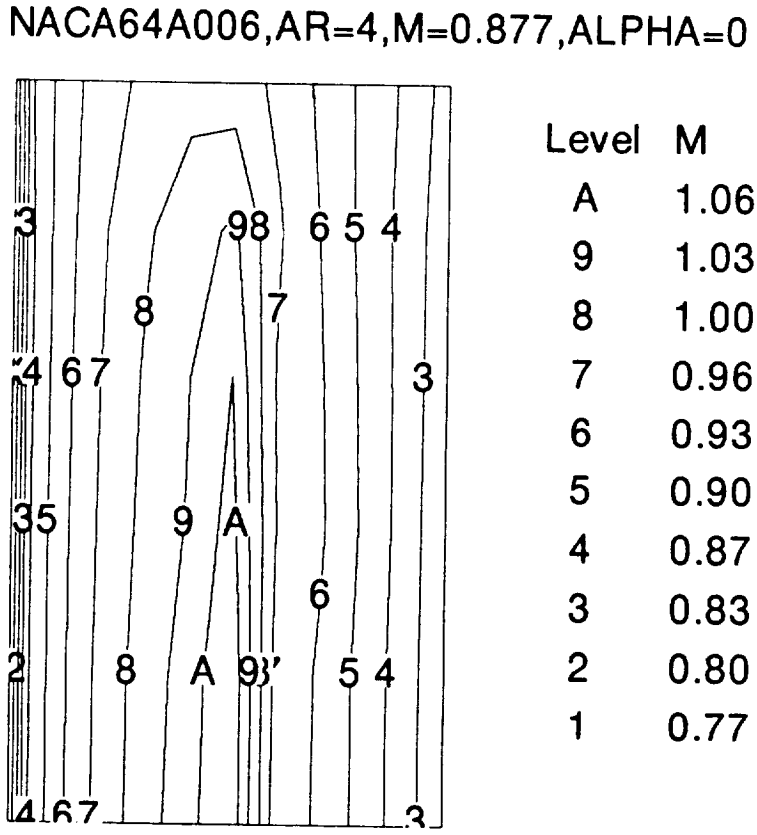


Figure 9a. Transonic flow, surface local Mach contours.

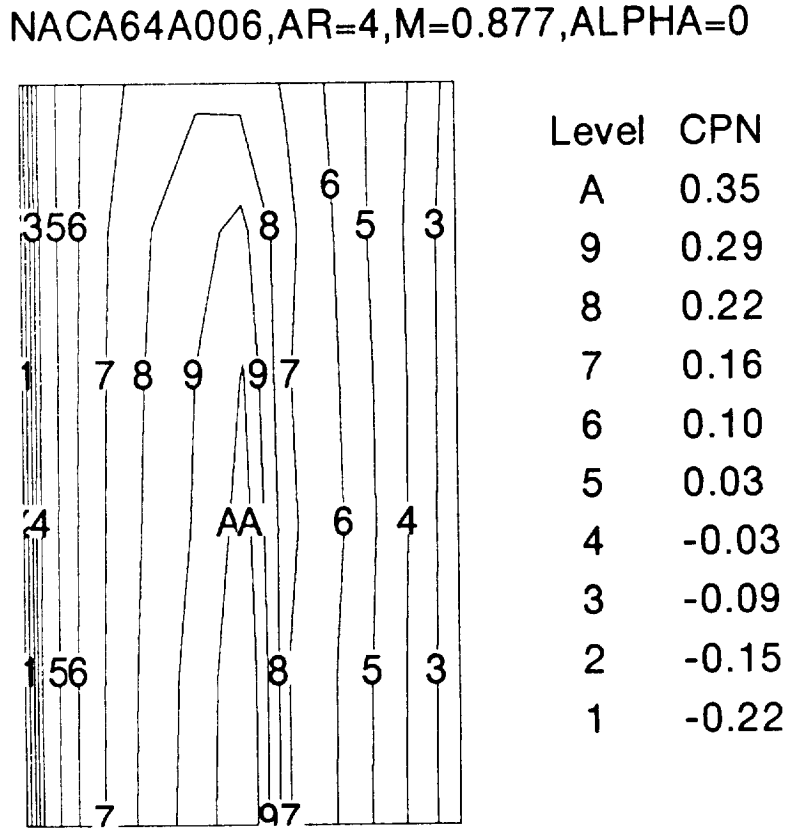


Figure 9b. Transonic flow, surface pressure coefficient contours.

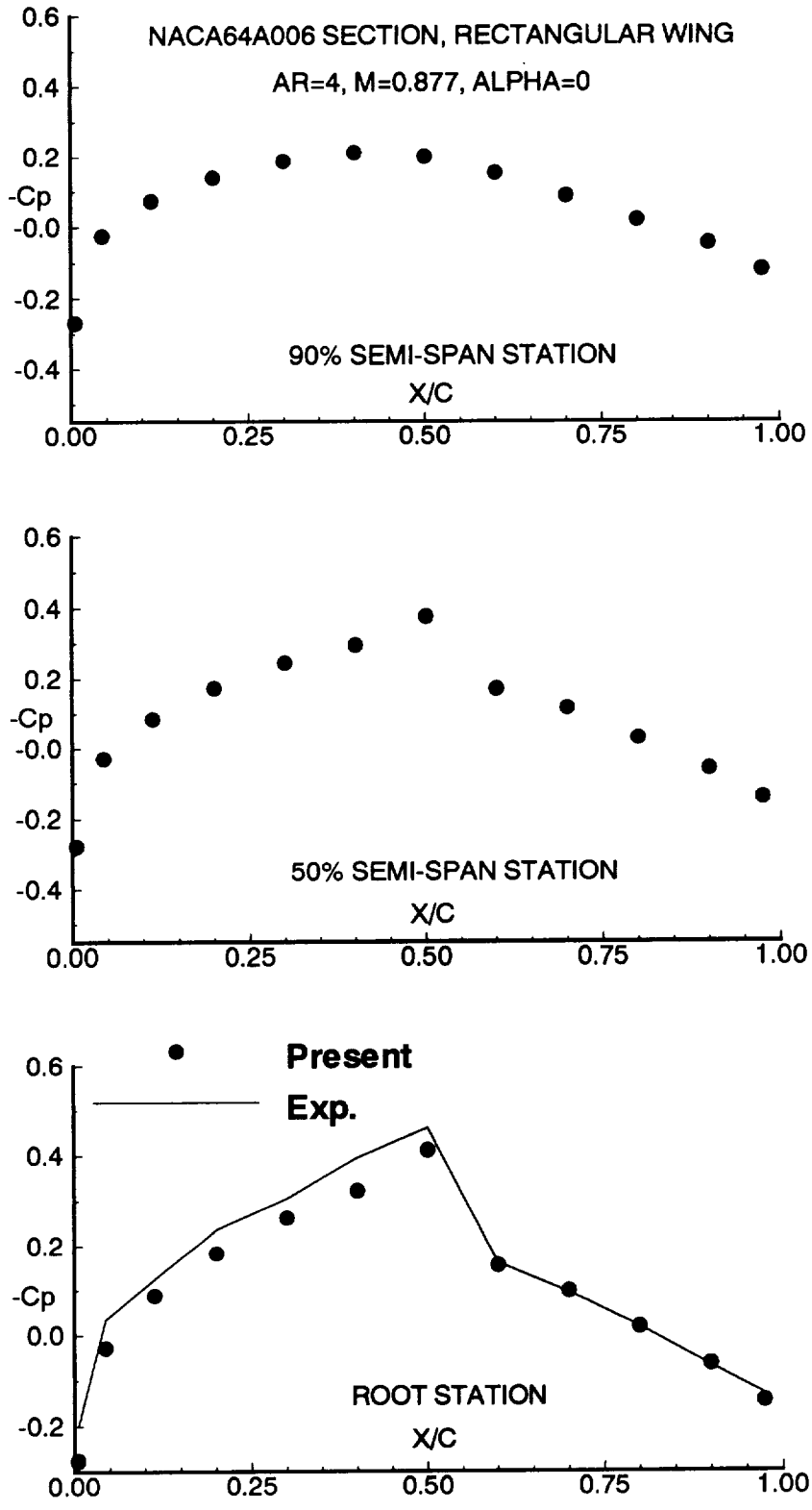
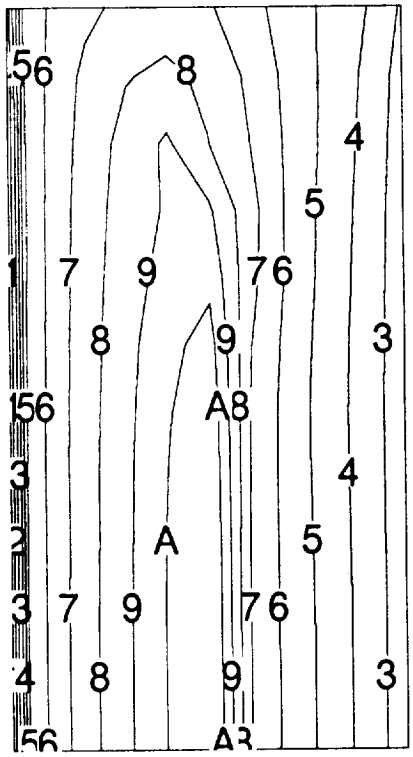


Figure 9c. Transonic flow, surface pressure coefficients.

NACA64A010A(S),AR=4,M=0.80,ALPHA=0



Level	M
A	1.08
9	1.03
8	0.98
7	0.93
6	0.88
5	0.83
4	0.78
3	0.73
2	0.68
1	0.63

Figure 10a. Transonic flow, surface local Mach contours.

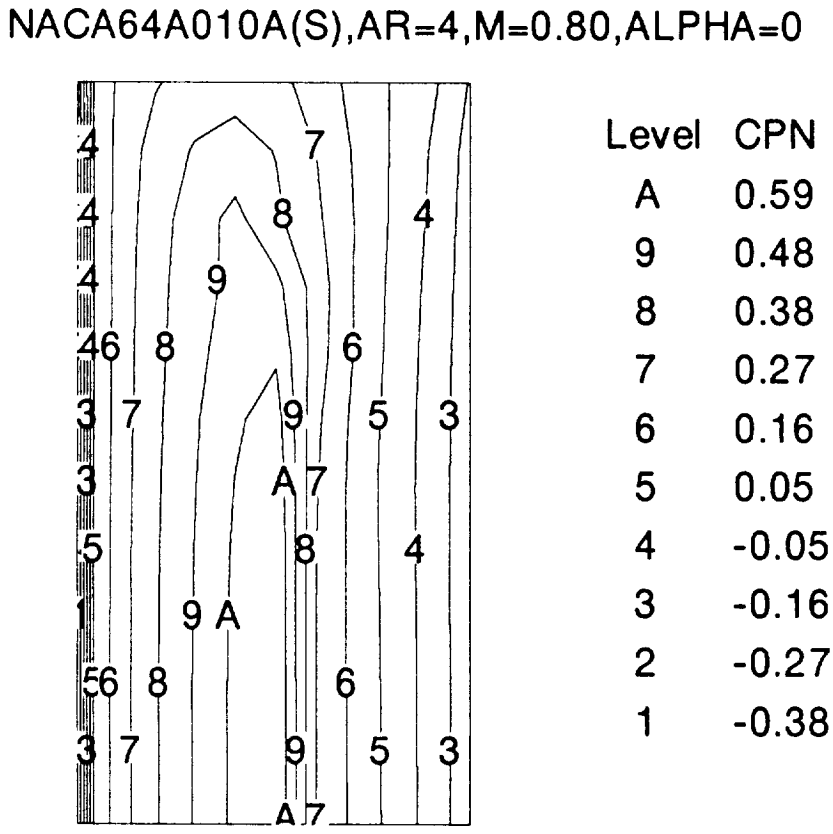


Figure 10b. Transonic flow, surface pressure coefficient contours.

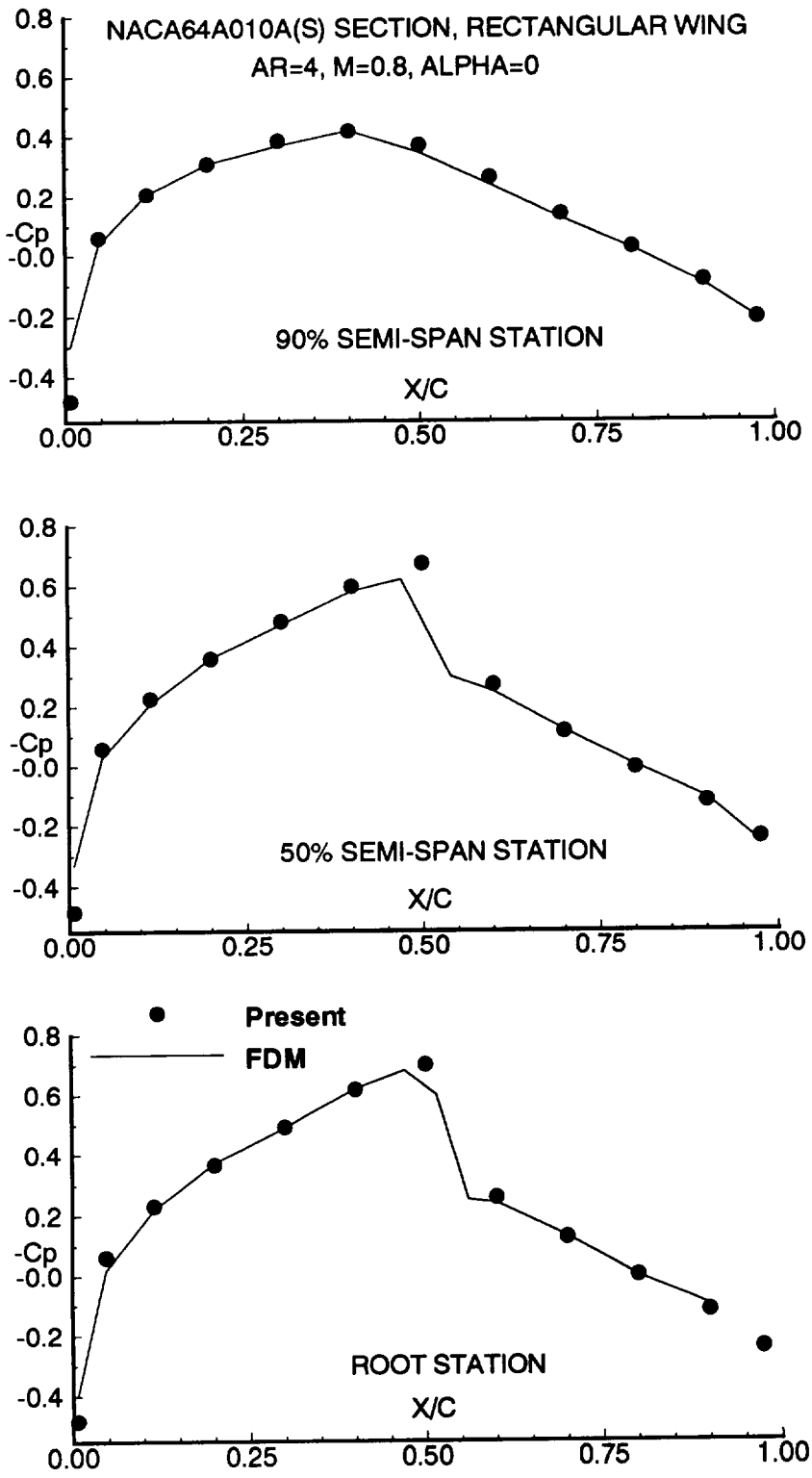


Figure 10c. Transonic flow, surface pressure coefficients.

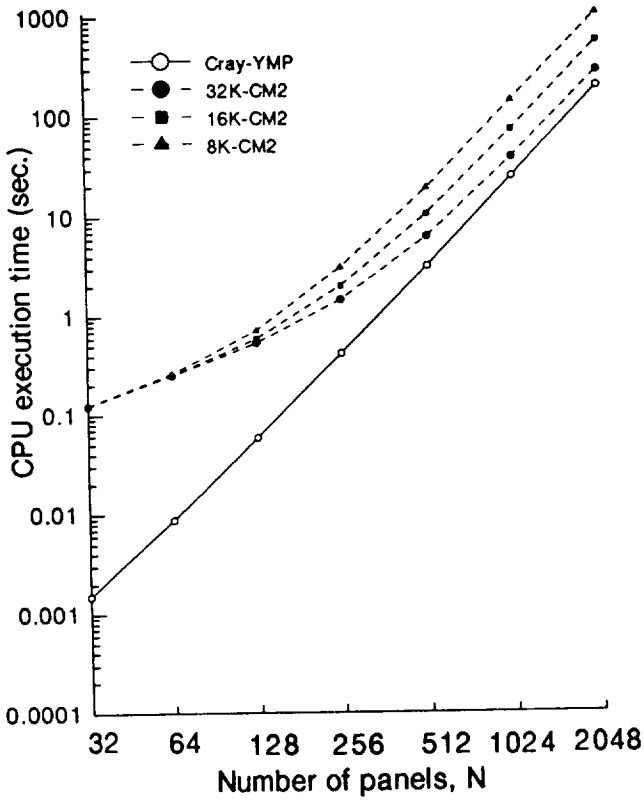


Figure 11. CPU time for solving linear system using Gauss elimination.

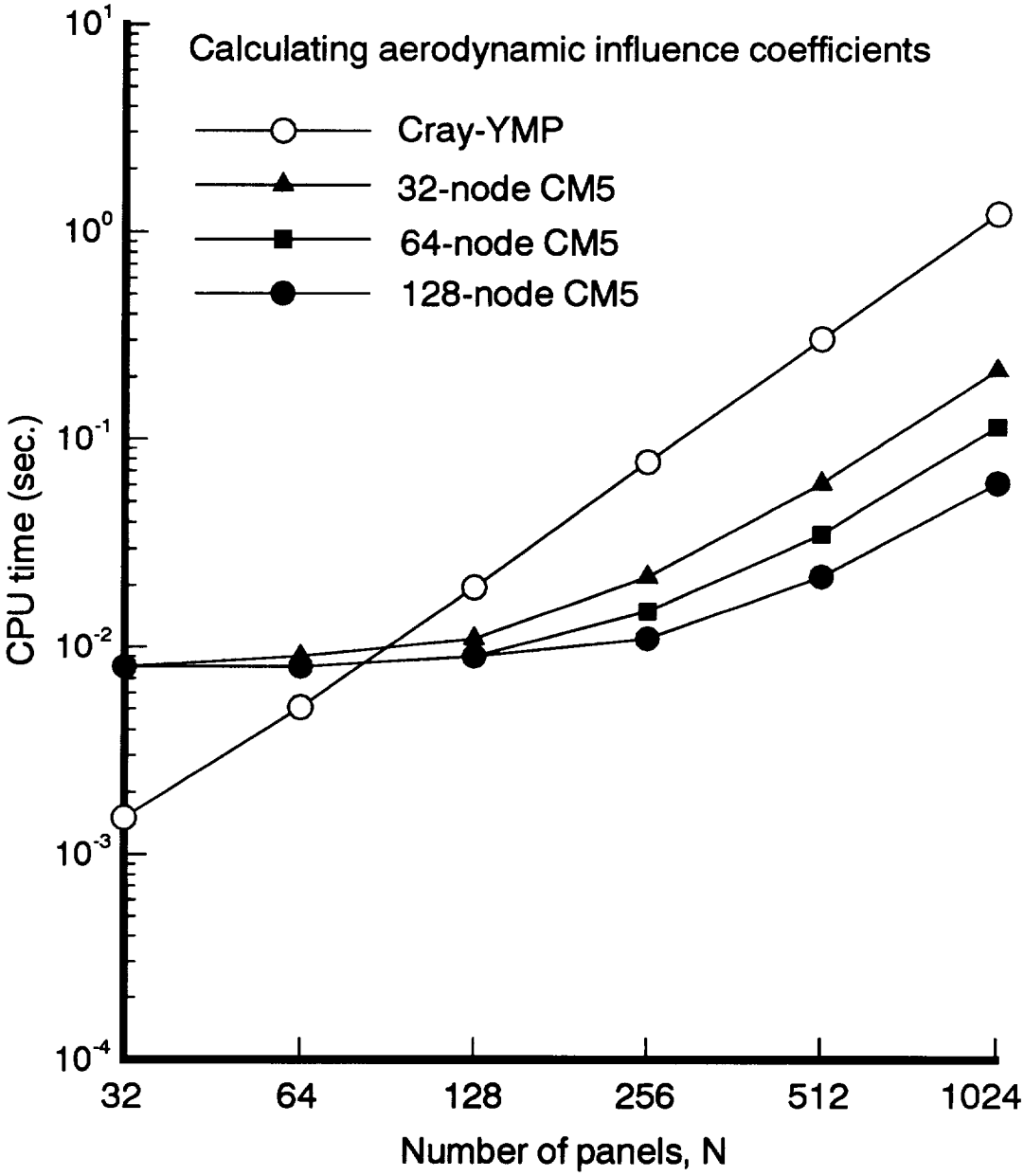


Figure 12. CPU time for evaluating aerodynamic enfluence coefficients.

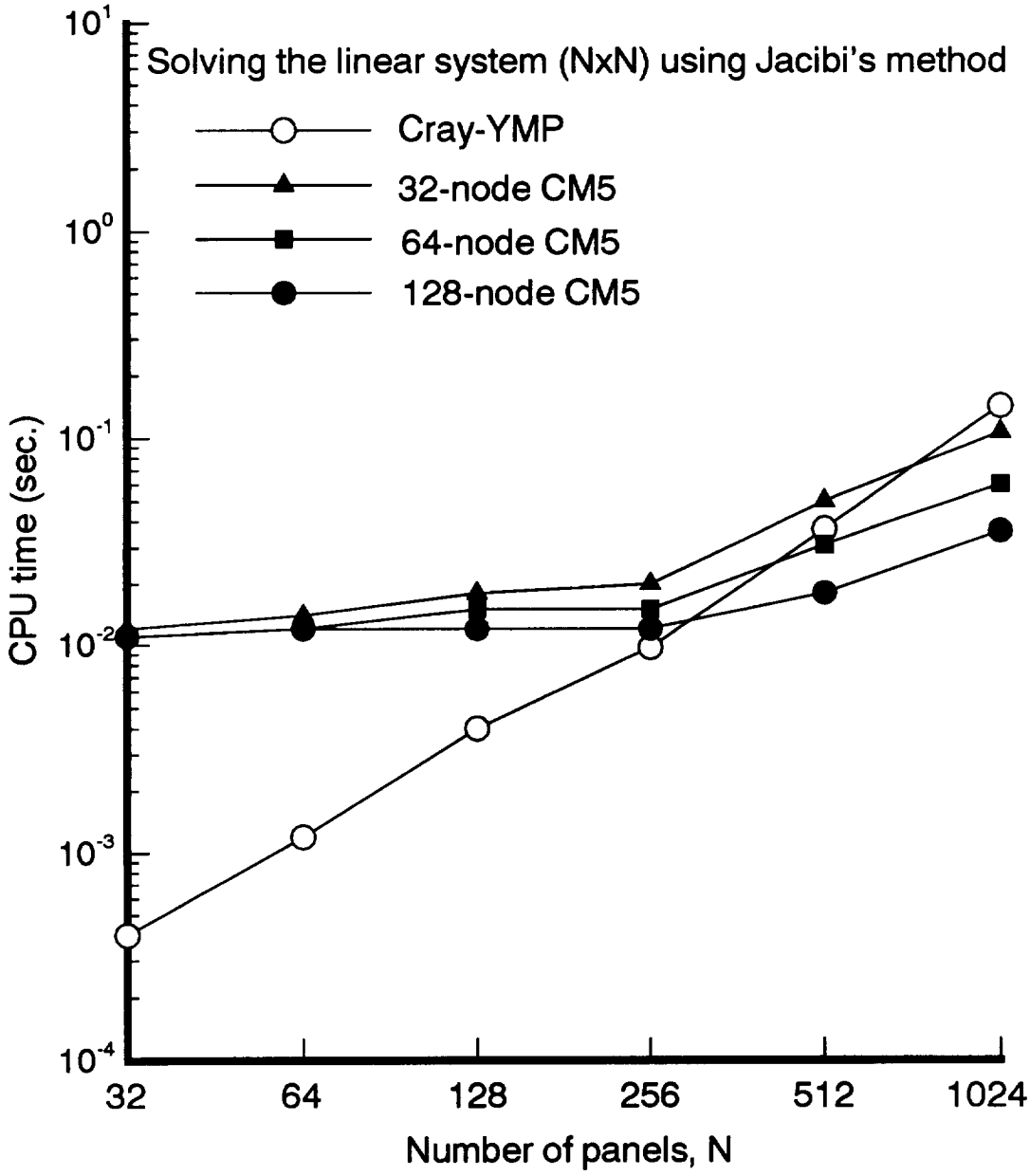


Figure 13. CPU time for solving linear system using Jacobi method.

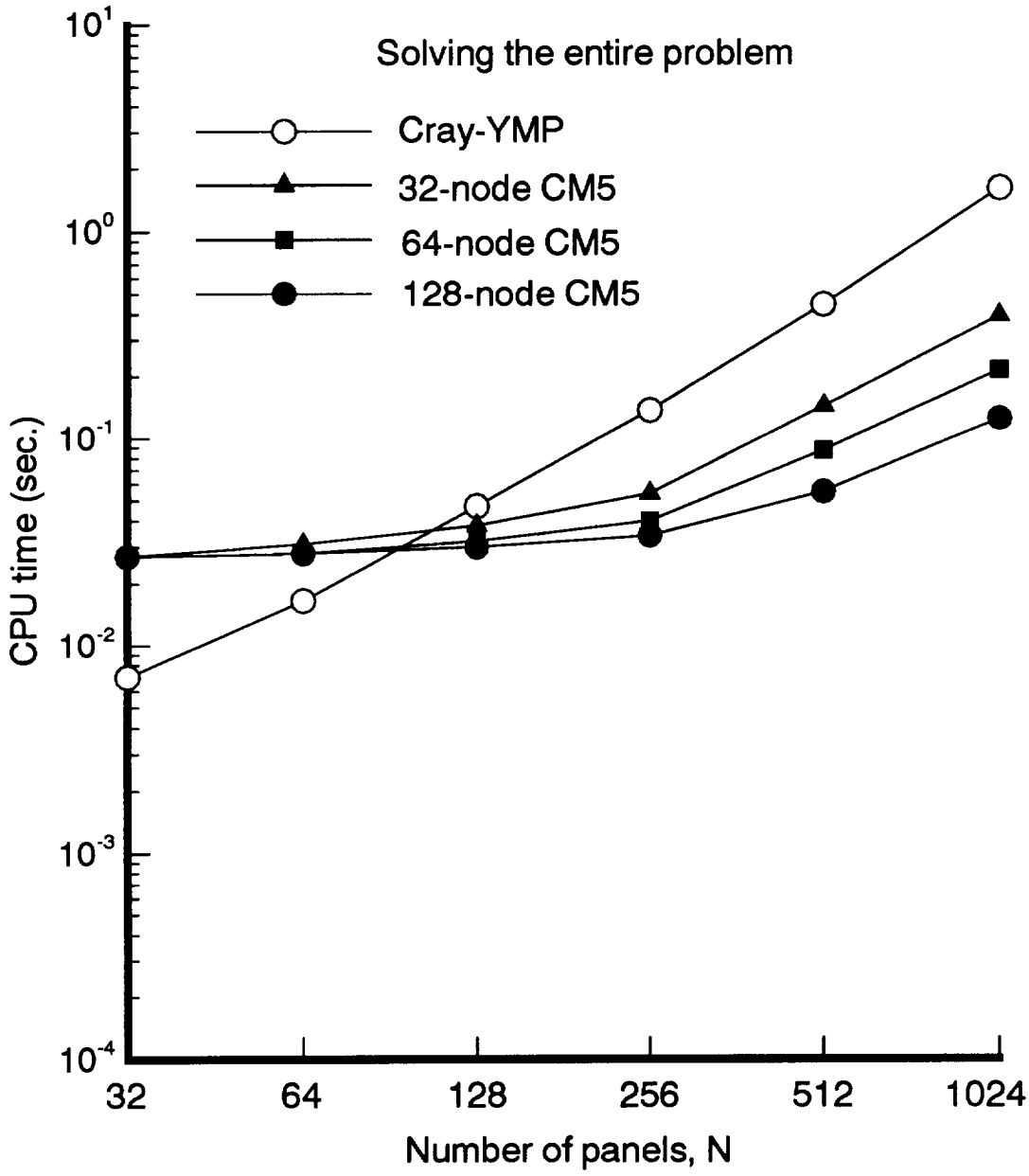


Figure 14. CPU time for solving entire problem.

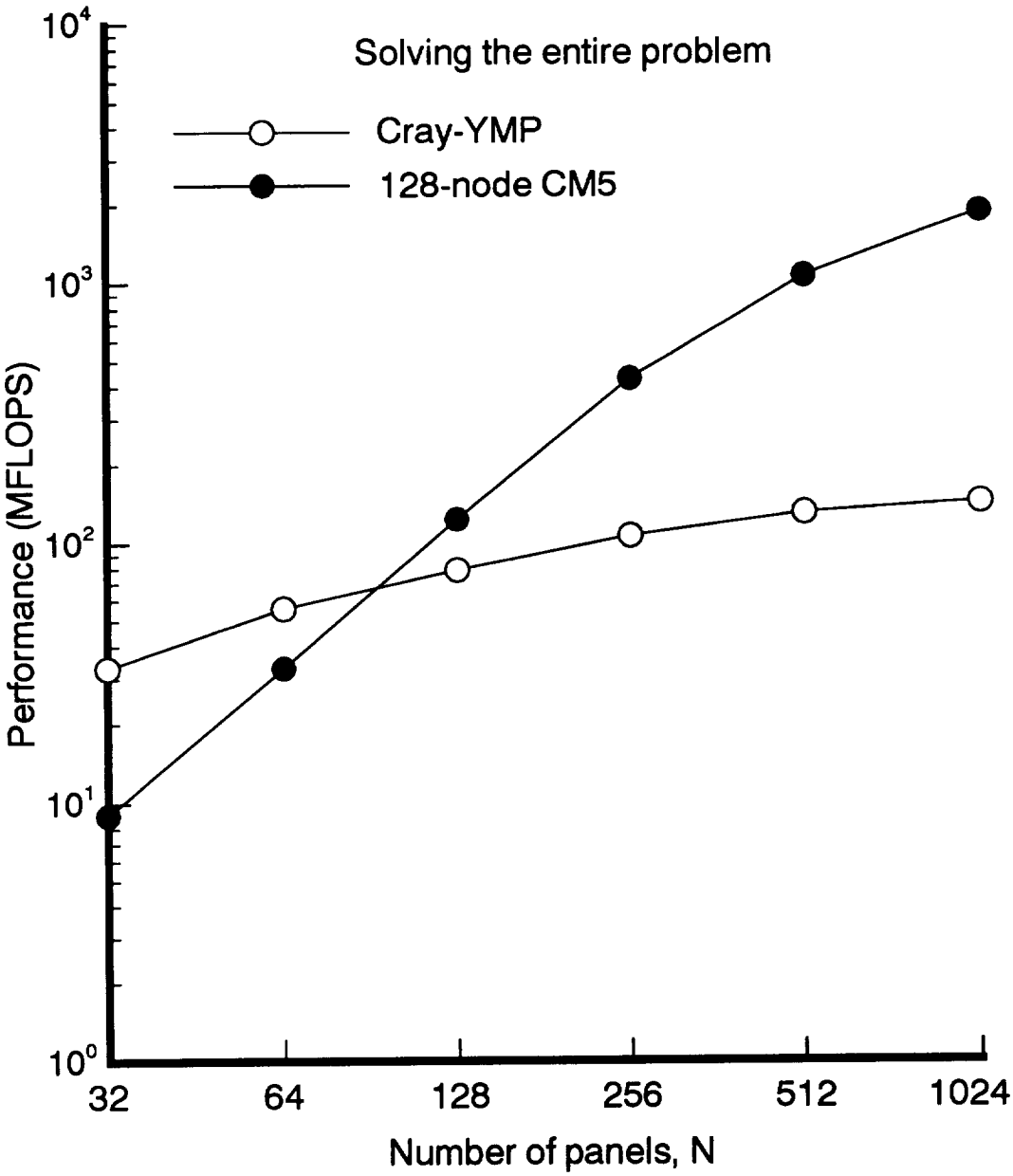


Figure 15. Performance for solving entire problem.

Table 1. The detailed computational performance results

Task/Size(N)		Cray-YMP		32-node CM5		64-node CM5		128-node CM5	
Task	N	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS
Mat Coef	32	0.0015	144	0.008	27	0.008	27	0.008	27
Lin Syst	32	0.0004	31	0.012	1	0.011	1	0.011	1
Total	32	0.0070	33	0.027	9	0.027	9	0.027	9
Mat Coef	64	0.0051	169	0.009	96	0.008	108	0.008	108
Lin Syst	64	0.0012	34	0.014	3	0.012	3	0.012	3
Total	64	0.0166	56	0.031	30	0.028	33	0.028	33
Mat Coef	128	0.0196	177	0.011	315	0.009	385	0.009	385
Lin Syst	128	0.0040	33	0.018	7	0.015	9	0.012	11
Total	128	0.0470	79	0.038	98	0.032	116	0.030	124
Mat Coef	256	0.0773	181	0.022	636	0.015	932	0.011	1272
Lin Syst	256	0.0097	31	0.020	15	0.015	20	0.012	25
Total	256	0.1370	107	0.054	271	0.040	366	0.034	431
Mat Coef	512	0.3050	183	0.061	915	0.035	1595	0.022	2537
Lin Syst	512	0.0368	31	0.050	23	0.031	37	0.018	63
Total	512	0.4450	131	0.143	408	0.087	670	0.055	1060
Mat Coef	1024	1.2200	184	0.216	1039	0.114	1969	0.061	3680
Lin Syst	1024	0.1420	31	0.106	42	0.060	73	0.036	122
Total	1024	1.6100	144	0.391	593	0.214	1083	0.124	1870

Table 2. CPU time in seconds for constructing matrices.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.44	8.25
32-node CM5	0.24	2.34
64-node CM5	0.17	1.20
128-node CM5	0.12	0.65

Table 3. CPU time in seconds for Gaussian elimination.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.58	33.85
32-node CM5	1.43	8.57
64-node CM5	2.11	7.05
128-node CM5	1.65	7.66

Table 4. CPU time in seconds for post processing.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.24	4.30
32-node CM5	2.18	9.45
64-node CM5	2.21	9.61
128-node CM5	2.18	9.61

Table 5. Total CPU time in seconds for incompressible flow.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	1.33	46.60
32-node CM5	3.86	20.55
64-node CM5	4.53	18.01
128-node CM5	4.00	18.08

```
      SUBROUTINE MATELM
      PARAMETER(N=32,M=33)
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      DO 2 K=1,N
      DO 1 J=1,N
      IF (K .EQ. J) FN(K,J)=2.0*PI
      IF (K. EQ. J) FT(K,J)=0.0
      IF (K. EQ. J) GOTO 1
      DYJ=SI(J)*DS(J)
      DXJ=CI(J)*DS(J)
      SPH=DS(J)*0.5
      XD=XC(K)-XC(J)
      YD=YC(K)-YC(J)
      RKJ=SQRT(XD*XD+YD*YD)
      BKJ=ATAN2(YD,XD)
      ALJ=ATAN2(DYJ,DXJ)
      GKJ=ALJ-BKJ
      ZIK=RKJ*COS(GKJ)
      ETK=-RKJ*SIN(GKJ)
      R1S=((ZIK+SPH)**2)+ETK*ETK
      R2S=((ZIK-SPH)**2)+ETK*ETK
      QT=ALOG(R1S/R2S)
      DEN=ZIK*ZIK+ETK*ETK-SPH*SPH
      GNM=ETK*DS(J)
      QN=2.0*ATAN2(GNM,DEN)
      UKJ=QT*CI(J)-QN*SI(J)
      VKJ=QT*SI(J)+QN*CI(J)
      FN(K,J)=-UKJ*SI(K)+VKJ*CI(K)
      FT(K,J)=UKJ*CI(K)+VKJ*SI(K)
    1 CONTINUE
      RHS(K)=UINF*SI(K)-VINF*CI(K)
    2 CONTINUE
      RETURN
      END
```

List 1a. Constructing matrices, 2D problem, in serial version.

```
 SUBROUTINE MATELM
 PARAMETER(N=32,M=33)
 DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
2,DYJ(N,N),DXJ(N,N),SPH(N,N),XD(N,N),YD(N,N)
3,RKJ(N,N),BKJ(N,N),ALJ(N,N),GKJ(N,N),ZIK(N,N)
4,ETK(N,N),R1S(N,N),R2S(N,N),QT(N,N),DEN(N,N)
5,GNM(N,N),QN(N,N),UKJ(N,N),VKJ(N,N)
6,DS2(N,N),CI2(N,N),SI2(N,N)
7,XC2(N,N),YC2(N,N),XC3(N,N),YC3(N,N)
8,SI3(N,N),CI3(N,N)
 LOGICAL MAIN_DIAG(N,N)
 COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
1,UINF,VINF,SDE
 XC2 = SPREAD(XC,DIM=1,NCOPIES=N)
 YC2 = SPREAD(YC,DIM=1,NCOPIES=N)
 XC3 = SPREAD(XC,DIM=2,NCOPIES=N)
 YC3 = SPREAD(YC,DIM=2,NCOPIES=N)
 SI2=SPREAD(SI,DIM=1,NCOPIES=N)
 CI2=SPREAD(CI,DIM=1,NCOPIES=N)
 SI3=SPREAD(SI,DIM=2,NCOPIES=N)
 CI3=SPREAD(CI,DIM=2,NCOPIES=N)
 DS2=SPREAD(DS,DIM=1,NCOPIES=N)
 MAIN_DIAG=DIAGONAL(SPREAD(.TRUE.,1,N),.FALSE.)
 WHERE(MAIN_DIAG)
   FN = 2.0 * PI
   FT = 0.0
 ELSEWHERE
   DYJ = SI2 * DS2
   DXJ = CI2 * DS2
   SPH = DS2 * 0.5
   XD = XC3 - XC2
   YD = YC3 - YC2
   RKJ=SQRT(XD*XD +YD*YD)
   BKJ=ATAN2(YD,XD)
   ALJ=ATAN2(DYJ,DXJ)
   GKJ=ALJ-BKJ
   ZIK=RKJ*COS(GKJ)
   ETK=-RKJ*SIN(GKJ)
   R1S=((ZIK+SPH)**2) + ETK*ETK
   R2S=((ZIK-SPH)**2) + ETK*ETK
   QT=ALOG(R1S/R2S)
   DEN=ZIK*ZIK + ETK*ETK - SPH*SPH
   GNM=ETK*DS2
   QN=2.0*ATAN2(GNM,DEN)
   UKJ=QT*CI2-QN*SI2
   VKJ=QT*SI2+QN*CI2
   FN=-UKJ*SI3+VKJ*CI3
   FT=UKJ*CI3+VKJ*SI3
 ENDWHERE
 RHS=UINF*SI-VINF*CI
 RETURN
 END
```

List 1b. Constructing matrices, 2D problem, in parallel version.

```
      SUBROUTINE JBINT(A,B)
      PARAMETER (N=128,M=128)
C -----------------------------------------------
C SOLVE AX=B USING APPROXIMATE JACOBI ITERATIONS
C ----- SERIAL FORTRAN VERSION
C -----------------------------------------------
      DIMENSION A(N,N),B(N),X(N,100)
      INTEGER VAR
      MAXITER=50
      TOL=0.001
      AX0=0.0
      XMAXDIF=0.0
      DO 1000 I = 1,N
      X(I,1) = 0.0
1000  CONTINUE
      K = 1
66    K= K+ 1
      DO 200 I = 1,N
      DO 300 J=1,N
      VAR = K-1
      IF (J.EQ.I) GOTO 300
      AX0=A(I,J)*X(J,VAR) + AX0
300   CONTINUE
      X(I,K) = 1/A(I,I)*(B(I)-AX0)
      XDIF = ABS(X(I,K)-X(I,K-1))
      IF (XDIF.GT.XMAXDIF) XMAXDIF=XDIF
      AX0=0.0
200   CONTINUE
      IF (XMAXDIF.LT. TOL) THEN
      MAXK = K
      GOTO 99
      ENDIF
      XMAXDIF = 0.0
      IF (K.LT. MAXITER) GOTO 66
      PRINT*,'NOT CONVERGENT YET AFTER ITERATIONS:',MAXITER
      RETURN
99    CONTINUE
      DO 400 I=1,N
      B(I)=X(I,MAXK)
400   CONTINUE
      RETURN
      END
```

List 2a. Jacobi method in serial version.

```
      SUBROUTINE jbite
      PARAMETER (N=128,M=128)
C -----------------------------------------------
C SOLVE AX=B USING APPROXIMATE JACOBI ITERATIONS
C ----- PARALLEL CM-FORTRAN VERSION
C -----------------------------------------------
      DIMENSION A(N,N),B(N),X(N,100),ax0(N),c(N)
      INTEGER VAR
      REAL  XDIF100(n)
      COMMON/BLK2/A
      COMMON/BLK3/B
      maxiter = 50
      tol = 0.001
      x(:,1) = 0.0
      k = 1
   66 k = k + 1
      var = k - 1
      FORALL (I=1:N) AX0(I)=DOTPRODUCT(A(I,:),X(:,VAR))-A(I,I)*X(I,VAR)
      forall (i=1:n) c(i)=a(i,i)
      x(:,k)=1.0/c *(b-ax0)
      xdif100(1:n) = abs(x(:,k) - x(:,k - 1))
      XMAXDIF=MAXVAL(XDIF100)
      IF (xmaxdif .LT. tol) THEN
        maxk = k
        GOTO 99
      ENDIF
      xmaxdif = 0.0
      IF (k .LT. maxiter) GOTO 66
      PRINT *, 'NOT CONVERGENT YET AFTER ITERATIONS:',maxiter
      RETURN
   99 CONTINUE
      b = x(:,maxk)
      RETURN
      END
```

List 2b. Jacobi method in parallel version.

```
      SUBROUTINE VELWING(IVELCT,IWG,IG,JG,KG)
      COMMON/BLK01/X(25,13),Y(25,13),Z(25,13)
      ......
      DO 1 JS=1,NC
      JS1=JS+1
      DO 2 IS=1,NR
      IS1=IS+1
      X1=X(IS,JS)
      ......
      XC=(X1+X2+X3+X4)/4.0
      ......
      DO 11 JR=1,NC/2
      JR1=JR+1
      DO 12 IR=1,NR
      ......
      XF=0.25*(X(IR,JR1)+X(IR1,JR)+X(IR,JR)+X(IR1,JR1))
      ......
      DX=XF-XC
      ......
      DIST=SQRT(DX*DX+DY*DY+DZ*DZ)
        IF(DIST.LT.FARFD) THEN
          IF(DIST.LT.0.0001) THEN
            UC=0.5*UNX(IS,JS)
            VC=0.5*UNY(IS,JS)
            WC=0.5*UNZ(IS,JS)
          ELSE
            CALL VWS(X1,X2,Z1,Z3,Y1,XF,YF,ZF,IS,JS,UC,VC,WC)
          END IF
        ELSE
          AREAXZ=ABS((X2-X1)*(Z3-Z1))
          XYN=UNX(IS,JS)/UNY(IS,JS)
          ZYN=UNZ(IS,JS)/UNY(IS,JS)
          FACXZS=SQRT(1.0+XYN*XYN+ZYN*ZYN)
          AREAS=FACXZS*AREAXZ
          CONSTFF=OPI4*AREAS/(DIST*DIST*DIST)
          UC=CONSTFF*DX
          VC=CONSTFF*DY
          WC=CONSTFF*DZ
        END IF
        ......
        NBA=(JS-1)*NR+IS
        A(KEQ,NBA)=A(KEQ,NBA)-(UC*UNX(IR,JR)
     +                        +VC*UNY(IR,JR)
     +                        +WC*UNZ(IR,JR))
     12 CONTINUE
     11 CONTINUE
      2 CONTINUE
      1 CONTINUE
      RETURN
      END
```

List 3a. Constructing matrices, 3D problem, in serial version.

```
      SUBROUTINE velwing(ivelct,iwg,ig,jg,kg)
      include '/usr/cm/include/cm/CMF_defs.h'
      COMMON/BLK01/X(25,13),Y(25,13),Z(25,13)
      ......
      real unxm3(24,12,24),vnxm3(24,12,24),wnxm3(24,12,24)
      ......
      unxm3 = spread(unx(:nr,:nc),dim=3,ncopies=24)
      ......
      xcm3 = spread(xcm,dim=3,ncopies=24)
      ......
      xcm4 =spread(xcm3,dim=4,ncopies=6)
      ......
      xfm3 = spread(xfm,dim=1,ncopies=12)
      ......
      xfm4 = spread(xfm3,dim=1,ncopies=24)
      ......
      dxm4 = xfm4 -xcm4
      ......
      distm4 = sqrt(dxm4*dxm4+dym4*dym4+dzm4*dzm4)
      ......
      where (distm4.lt.farfd)
        where (distm4 .LT. 0.0001)
          ucm4 = 0.5 * unxm4
          ......
        elsewhere
          xynm4 = unxm4/vnxm4
          zynm4 =   wnxm4/vnxm4
          facxzsm4 = sqrt(1.0 + xynm4 * xynm4 + zynm4 * zynm4)
          ddxm4 = ddxm4
          ddzm4 = ddzm4
          fac4 = opi4*facxzsm4*abs(ddxm4*ddzm4)/6.0
           vwx4 = ffx11+ffx21+ffx31+ffx41+ .....
     &     + ffx44+ffx15+ffx25+ffx35+ffx45
          ......
          ucm4 = vwx4*fac4
          ......
        endwhere
      elsewhere
        zynm4 = wnxm4/vnxm4
        facxzsm4 = sqrt(1+xynm4*xynm4 + zynm4*zynm4)
        contm4 = opi4*areasm4/(distm4*distm4*distm4)
        vcm4 = contm4*dym4
      endwhere
      forall (jr=1:nc/2,ir=1:nr,js=1:nc,is=1:nr)
     & a(ir+(jr-1)*nr,is+(js-1)*nr) = a(ir+(jr-1)*nr,is+(js-1)*nr)
     & -((ucm4(is,js,ir,jr)
     & *unxd4(is,js,ir,jr))+
     & (vcm4(is,js,ir,jr)*vnxd4(is,js,ir,jr))+(wcm4(is,js,ir,jr)
     & *wnxd4(is,js,ir,jr)))
      return
      end
```

List 3b. Constructing matrices, 3D problem, in parallel version.

LIST OF PUBLICATIONS

1. Hu, Hong, "Development of a Shock-Fitting Field Panel Method for 3D Transonic Flows", accepted for publication in the journal, **Computational Mechanics**, July 1994.

2. Logan, Terry and Hu, Hong, "Performance Study of 3D Integral Equation Computations on Massively Parallel Computer", **Boundary Element Methods in Fluid Dynamics II**, Computational Mechanics Publication, 1994, pp. 85-91.

3. Hu, Hong and Paysour, Jada, "Panel Method Computational Performance on CM-5 and Cray-YMP", submitted for publication in the journal, **Boundary Elements Communications**, June 1994.

4. Hu, Hong, "Study of Integral Equation Methods For Transonic Flow Calculations", Journal, **Engineering Analysis with Boundary Elements**, Vol. 11, No. 2, 1993, pp. 101-107.

5. Hu, Hong and Soe, Min, "Panel Method Computational Performance on CM-2 and iPSC/860 MPP Machines", in **Boundary Elements XV**, Computational Mechanics Publications and Elsevier Applied Science, 1993, pp. 67-79.

6. Hu, Hong and Jackson, Isaac T., "CM-2 Performance Evaluation on Panel Method Calculations", in the **Proceedings of The Sixth SIAM Conference on Parallel Processing For Scientific Computing**, 1993, pp. 1005-1010.

7. Hu, Hong, "Unsteady Transonic Wing Flow Computations Using Field-Boundary Element Method", Journal, **Engineering Analysis with Boundary Elements**, Vol. 10, No. 2, 1992, pp. 99-105.

8. Hu, Hong and Jackson, Isaac T., "Comparative Study of Computational Performance of CM-2 and Cray-YMP for Boundary Element Computations", **Boundary Elements Abstracts Journal and Newsletter**, Vol. 3, No. 5, 1992, pp. 185-191.

9. Hu, Hong, "Experience with Transonic Flow IE Computations", in **Boundary Elements in Fluid Dynamics**, Computational Mechanics Publications with Elsevir Applied Science, 1992, pp.35-48.

10. Hu, Hong, "A 3D IEM for Compressible Wing Flows with and without Shocks", in **Boundary Elements in Fluid Dynamics**, Computational Mechanics Publications with Elsevir Applied Science, 1992, pp.49-60.

11. Hu, Hong, "Application of the Integral Equation Method to Flows Around a Wing with Circular-Arc Section", in **Boundary Elements XIII**, Elsevier Applied Science, 1991, pp.209-217.

No. 1

DEVELOPMENT OF A SHOCK-FITTING FIELD-PANEL METHOD FOR 3D TRANSONIC FLOWS

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Accepted for Publication in Journal
Computational Mechanics
July 1994

Computational Mechanics

Springer International

Editors-in-Chief:

Satya N. Atluri, D. Sc. Institute Professor & Regents' Professor of Engineering
Computational Modeling Center, Georgia Institute of Technology,
Atlanta, GA 30332-0356, USA. Tel.: (404) 894-2758; Fax: (404) 894-2299

Genki Yagawa, D. Eng. Professor of Nuclear Engineering
Dept. of Nuclear Engineering, The Faculty of Engineering,
The University of Tokyo, Bunkyo-ku, Tokyo, 113 Japan. Tel.: (03) 812-2111

July 20, 1994

Professor Hong Hu
Hampton University
Department of Mathematics
Hampton, Virginia 23668, USA

Re: *"Development of a Shock-Fitting Field-Panel Method for 3D Transonic Flows"*

Dear Professor Hu,

Thank you for your revised copy of the above mentioned manuscript in such a expeditious manner. I am pleased to accept your revised manuscript for publication in *"Computational Mechanics."* Your paper is being forwarded to the printers, from whom you should hear in due course.

Congratulations on a fine publication.

Sincerely,

S.N. Atluri

Development of A Shock-Fitting Field-Panel Method for 3D Transonic Flows

Hong Hu
Hampton University
Department of Mathematics
Hampton, Virginia 23668, USA

ABSTRACT

The paper presents the development of a shock-fitting field-panel method for three-diemnsion (3D) transonic flows. In this method, the full-potential equation, written in the form of the Poisson's equation, is solved by integral equation field-panel method. The solution consists of a wing surface source panel integral term, a field-volume panel integral term of compressibility over a small limited domain, and a shock panel integral term. Due to the non-linearity of flows, solutions are obtained through an iterative procedure. Instead of using a field-panel refinement procedure, a shock-fitting technique is used to fit the shock. Finally, numerical examples are provided to demonstrate the accuracy of the method.

1. INTRODUCTION

The finite-difference method (FDM) and finite-volume method (FVM) for solving transonic flows have been well developed during the past twenty years. Although the Navier-Stokes equation formulation for the transonic flow computations has been understood as the best model and the FDM and FVM are successful in dealing with transonic flows, the computation of the unsteady Navier-Stokes equations over complex three-dimensional configurations is very expensive, particularly for time-accurated unsteady flow computations. There are also major technical difficulties in FDM and FVM for generating suitable grids for complex three-dimensional aerodynamic configurations.

The experience has shown that rather accurate solutions can been obtained for certain transonic flows using the inviscid modeling of the full-potential equation. For transonic flows without strong shocks and without massive separations, the full-potential equation is an adequate approximation to the Navier-Stokes equations. The integral equation method (IEM) for the potential equation is an alternative to the FDM and FVM. Moreover, the IEM has several advantages over the FDM and FVM. The IEM involues evaluation of integrals, which is more accurate than the FDM and FVM, and hence a coarse grid can been used in IEM. The IEM automatically satisfies the far-field boundary conditions and therefore only a small limited computational domain is needed. The generation of the three-dimensional grid for complex configuration is not difficulty in the IEM, since the mapping from physical plane to computational plane is not required.

During the past few years integral equation methods for transonic flows have been developed by Piers and Sloof (1979), Tseng (1984), Erickson and Strande (1985), Kandil and Hu (1988) and Ogana (1989) for steady airfoils, by Tseng (1984), Kandil and Yates (1986), Madson (1987) and Sinclair (1988) for steady wing and aircraft configurations, and by Hounjet (1981) and Kandil and Hu (1990) for unsteady airfoils by solving either full-potential or transonic small disturbance equations using both surface and field panels. The shock capturing technique was applied in these methods. The method of Kandil

and Hu (1988) solves the full-potential equation for two-dimensional transonic flows, where both shock-capturing and shock-fitting techniques are applied. The capability of capturing shocks with shock-capturing technique and improvement of the shock with shock-fitting technique was presented by Kandil and Hu (1988). The method is efficient and engineering accurate. In the present paper a method for computing steady 3-D flows is presented along with numerical examples to demonstrate the capability, accuracy and the potential of the present IE scheme for subsonic and transonic flow computations. The method is the extension of the steady 2D method of Kandil and Hu (1988) and is being extented to unsteady 3D transonic flow computations. In fact, the present method has been applied to unsteady transonic flows around a zero-thickness wing by Hu (1992, 1993) using vortex panel method. In order to use a coarse grid, which is particularly important in 3D calculations, the shock-fitting technique is applied to the present transonic flow calculations.

2. FIELD-PANEL FORMULATION

2.1 Governing Full-Potential Equation

The non-dimensional steady full-potential equation is given by:

$$\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} = G \tag{1}$$

with

$$G = -\frac{1}{\rho}(u\frac{\partial \rho}{\partial x} + v\frac{\partial \rho}{\partial y} + w\frac{\partial \rho}{\partial z}) \tag{2}$$

and

$$\rho = [1 + \frac{\kappa - 1}{2}(1 - u^2 - v^2 - w^2)]^{\frac{1}{\kappa - 1}} \tag{3}$$

where the characteristic parameters, ρ_∞, a_∞ and c have been used; a is the speed of the sound, ρ the density, and c the wing root-chord length; and Φ is the velocity potential ($\nabla\Phi = \vec{V} = (u, v, w)$), G the compressibility, and κ the gas specific heat ratio.

Equation (1) is not in the conservative form but in the form of the Poisson's equation. By writting the full-potential equation in the Poisson's form, the nonlinearity of the transonic flows can be treated as non-homogeneity and in terms of the IE solution, this non-linearity is represented by field volume integral term. And hence the classical surface- panel method can be extended into field-panel method for non-linaer flows. The experience has shown that such non-conservative formulation has produce accurate solutions as long as the shock is not very strong.

2.2 Boundary Conditions

The general boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, and wake kinematic and dynamic conditions. For the present non-lifting flows, the only surface no-penetration condition and infinity condition are needed and they are given by:

$$\vec{V} \cdot \vec{n}_g = 0 \qquad \text{on} \quad g(\vec{r}) = 0 \tag{4}$$

and

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r}) = 0 \quad \text{and} \quad w(\vec{r}) = 0 \tag{5}$$

where $\vec{n}_g$ is the unit normal vector of the wing surface, $g(\vec{r}) = 0$.

2.3 IE Solution

By using the Green's theorem, the integral equation solution of Eq. (1) in terms of the velocity field is given by

$$\begin{aligned} \vec{V}(x,y,z) = \vec{V}_\infty & \\ & - \frac{1}{4\pi} \int\int_g \frac{q_g(\xi,\eta,\zeta)}{d^2} \vec{e}_d ds(\xi,\eta,\zeta) \\ & + \frac{1}{4\pi} \int\int\int_V \frac{G(\xi,\eta,\zeta)}{d^2} \vec{e}_d d\xi d\eta d\zeta \\ & + \frac{1}{4\pi} \int\int_S \frac{q_S(\xi,\eta,\zeta)}{d^2} \vec{e}_d ds(\xi,\eta,\zeta) \end{aligned} \tag{6}$$

where $\vec{V}_\infty$ is the free-stream velocity; q is the surface source distribution; the subscript, S, refers to the shock surface; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$. It can be seen that the infinity condition, Eq. (5), is automatically satisfied by the integral equation solution, since the integrals become zero when d is large enough.

2.4 Field-Panel Discretisation

The formulation presented here can be easily extended to general lifting flows by including surface and wake vortex-panel integral terms, although the present computations are only made to symmetric non-lifting flows. In this non-lifting computational model, the wing surface is represented by a number of uniform rectangular source panels. A uniform rectangular parallelopiped type of field-volume panels are also used throughout the flow field. Constant surface and volume source (q and G) distributions are assumed over wing / shock surface panels and field volume panels. The discretized integral equation solution in terms of surface and field-volume panels then becomes

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty & \\ & - \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG} q_{g_{i,k}} \int\int_{g_{i,k}} \frac{1}{d^2}\vec{e}_d ds(\xi,\eta,\zeta) \\ & + \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV} G_{i,j,k} \int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{e}_d d\xi d\eta d\zeta \\ & + \frac{1}{4\pi}\sum_{j=1}^{MS}\sum_{k=1}^{NS} q_{S_{j,k}} \int\int_{S_{j,k}} \frac{1}{d^2}\vec{e}_d ds(\xi,\eta,\zeta)\end{aligned} \tag{7}$$

where the indices, i, j and k refer to the surface and field panels; $LG \times NG$ is the total number of wing surface panels; $LV \times MV \times NV$ is the total number of field panels; and $MS \times NS$ is the total number of shock surface panels. A sketch of the computational model is given in Figure 1, while the detailed wing surface panelling is given in Figure 2

where the exact number of wing surface panels is shown.

3. COMPUTATIONAL SCHEME

3.1 Iterative Scheme

Due to the nature of the non-linearity of transonic flows, solutions are obtained through an iterative procedure, where the wing surface source strength and the compressibility over selected volume elements are updated through each iteration. The solution procedure follows the successful form of Kandil and Hu (1988) of two-dimensional computations. Here only the treatment of shocks for transonic flow is described.

3.2 Shock-Fitting Technique

It should be mentioned that mathematically the second (volume) integral term of Eq. (7) includes all compressibility effects including shock discontinuity. Since a relative coarse grid is used in the present IE computational domain where only 10 field panels are used over the wing chord, the contribution of the shock discontinuity is extracted from this volume integral term and it is represented explicitly by the third integral term of Eq. (7). It is very important to use coarse grid in 3D calculations, since the integral calculations over 3D field panels are very expensive. The strength of shock panels, q_S,is equal to the difference of normal velocity across the shock. This can be shown by integrating Eq. (1) over an infinitesimal volume around an infinitesimal area of the shock surface and applying the divergence theorem, one gets

$$\Delta(\nabla\Phi) = V_{2n} - V_{1n} = G\epsilon \tag{8}$$

where ϵ is the infinitesimal thickness normal to the shock surface. By letting $G\epsilon = q_S$ and using Rankine-Hugoniot relation, one finally obtains

$$q_S = \left[\frac{(\kappa-1)M_{1n}^2+2}{(\kappa+1)M_{1n}^2} - 1\right]V_{1n} \tag{9}$$

where the subscripts 1 and 2 refer to the conditions ahead and behind of the shock, respectively; and the subscript n refers to the normal component to the shock. The

purpose to use Rankine-Hugoniot relation is to introduce the effect of entropy change across the shock sinec the full- potential formulation uses isentropic flow assumption which is not true in the shock region.

The constantly distributed, piece-wise continuous (in flow direction) oblique shock panels are used. The slope of shock panels is determined by the so called $\theta - \beta - M$ relation as given by

$$tan\theta = 2cot\beta\left[\frac{M_1^2 sin^2\beta - 1}{M_1^2(\gamma + cos2\beta) + 2}\right] \tag{10}$$

where θ is the flow-deflection angle, and β is the shock angle.

In the present calculation, the shock panel term, the last term of Eq. (7), becomes active only after the sonic line (and hence the shock location) is fixed. In other words, the shock-capturing technique is first used to locate the shock, where the Murman-Cole type-difference scheme is used in consistent with the mixed-nature of transonic flows. The use of the Murman-Cole scheme is equivelent to the introducing of the artificial dissipation. The use of this artificial dissapation scheme within a shock-fitting scheme seems contradictory since some of their effects will cancel each other. But if we consider the shock-fitting as the way to give a correct inviscid shock and the Murman-Cole scheme as the way to give the artificial viscous effect, then the use of the Murman-Cole scheme with shock-fitting scheme will give a correct viscous shock, this is what it should be.

4. NUMERICAL EXAMPLES

The presesnt scheme is applied to rectangular wings of symmetric circular arc sections with different aspect ratios (AR) at different free-stream Mach numbers, one for shock-free subsonic flow and one for transonic flow with a moderate shock. The half-span of the wing surface (including upper and lower surfaces) is divided into 20×6 quadrilateral panels as shown in Figure 2. The one-half of the computational domain is divided into $20 \times 16 \times 9$ field volume elements in chord, normal and span directions, respectively. The size of the computational domain is $2c \times 1.5c \times 2.25c$ and $2c \times 1.5c \times 3c$ for two AR values in the chord, normal and span directions, respectively. It should be noted that the both

surface- and field-panel sizes in chord (flow) direction are as large as 10% of chord length.

The first numerical example is made to the flow around a wing with a 5% thick circular arc section of $AR = 3$ at free-stream Mach number of 0.7, a shock-free subsonic flow, where the non-linearity effect is small. Figure 3a is the calculated surface local Mach contoures which shows that the flow is purly subsonic. The calculated surface pressure coefficients are presented in Figure 3b in terms of contours and Figure 3c in terms of line plot, along with the computational results obtained by the non-linear LTRAN3 TSD FD code of Guruswang and Goorjian (1982) and by the linear SOUSSA IE code of Yates, Cunningham, Desmarais, Silva and Drobenko (1982) at three span stations located at 0%, 50% and 90% of semi-span. As the figure shows, the presently calculated pressure distributions are in close agreement with the non-linear LTRAN3 results over the entire wing surface and agreement with the linear IE SOUSSA results except the discrepancy over leading and trailing edges. The convergence of the solution is obtained by checking the relative error of surface pressure distribution over each iteration, and for this shock-free flows, the number of iterations for a convergent solution is 6.

The second numerical example is made to a transonic flow around a wing with a 6% thick circular arc section of $AR = 4$ at a free-stream Mach number of 0.908. In order to show the capability of shock- fitting, the solutions obtained with and without shock-fitting are presented in Figure 4a through Figure 4d. Figures 4a and 4b are the surface Mach contours and surface pressure coeffcient contours without shock-fitting, where the shock is diffused but the supersonic flow region is clearly seen in the Figure 4a. Figures 4c and 4d are the solutions with shock-fitting, where the shock is clearly predicted. The effect of shock-fitting is self-explanatory from these figures. In order to verify the accuracy of the shock-fitting, the calculated results are ploted in Figure 4e along with the other reference solution. The calculated pressure distributions compare very well with a TSD FD result of Bailey and Steger (1972) and another IE result of Tseng (1984) except the discrepancy at the station near the wing tip. The location and the strength of the shock are correctly predicted by the present method. For the present transonic flow case, 16 iterations are

used to get a convergent solution, where the first 10 iterations are used to locate shock and additional iterations are used to fit shock.

The discrepancy near wing tip may be caused by the different tip shapes used in different computational models, and hence to have different tip- release effects. To investigate this effect, the present computation is made for this case with different wing tip thickness. Figures 5a through 5c are the results obtained by tapering off wing tip to 75%, 50% and 0% of the value at root section, respectively. Figures 5a - 5c show the variation of the surface pressure coefficients at the station near tip due to the tip- release effect.

5. CONCLUDING REMARKS

An integral equation field-panel method based on the full-potential equation formulation for transonic flows is presented. The method can be extended for handling flows around general three dimensional configurations, although only non-lifting cases are tested. The calculated wing surface pressure distribution is reasonably correct including the location and the strength of the shock. As an alternative to the grid refinement, the shock-fitting technique applied here does give a correct shock both in location and in strength. The present IEM is effective in terms of the number of iterations compared with those of FDM and FVM, althought the computational cost per IE iteration is more expensive than those of FDM and FVM. The large grid size (for example, $\Delta x = 0.1c, \Delta y = 0.1c, \Delta z = 0.25c$) used here makes the scheme even more efficient. The CPU time for a typical transonic flow case is around 165 seconds on Cray-YMP computer. Currently the lifting as well as unsteady effects are being added into the computation.

ACKNOWLEDGEMENT

This work is supported by NASA-Langley Research Center under the Grant No.

NAG-1-1170. Mr. Walter Silva is the technical monitor.

REFERENCES

Bailey, F. R.; Steger, J. L. (1972): Relaxation techniques for three-dimensional transonic flow about wings. AIAA Paper 72-189.

Erickson, L. L.; Strande, S. M. (1985): A theoretical basis for extending surface-paneling methods to transonic flow. AIAA J.. 23(12), 1860-1867.

Guruswamy, P.; Goorjian, P. M. (1982): Comparisons between computations and experimental data in unsteady three-dimensional transonic aerodynamics, including aeroelastic applications. AIAA Paper 82-0690-CP.

Hounjet, M. H. L. (1981): Transonic panel method to determine loads on oscillating airfoils with shocks. AIAA J.. 19(5), 559- 566.

Hu, H. (1992): Unsteady transonic wing flow computations using field- boundary element methods. Engineering Analysis with Boundary Elements. 10(2), 99-105.

Hu, H. (1993): Study of integral equation methods for transonic flow calculations. Engineering Analysis with Biundary Elements. 11(2), 101-107.

Kandil, O. A.; Yates, E. C., Jr. (1986): Computation of transonic vortex flows past delta wings-integral equation approach. AIAA J.. 24(11), 1729-1736.

Kandil, O. A.; Hu, H. (1988): Full-potential integral solution for transonic flows with and without embedded Euler domains. AIAA J.. 26(9), 1074-1086.

Kandil, O. A.; Hu, H. (1990): Integral solution of unsteady full- potential equation for a transonic pitching airfoil. J. of Aircraft. 27(2), 123-130.

Madson, M. D. (1987): Transonic analysis of the F-16A with under-wing fuel tanks: an application of the TranAir pull-potential code. AIAA Paper 87-1198.

Ogana, W. (1989): Transonic integro-differential and integral equations with artificial viscosity. Engineering Analysis with Boundary Elements. 6(3), 129-135.

Piers, W. J.; Sloof, J. W., (1979): Calculation of transonic flow by means of a shock-capturing field panel method. AIAA Paper 79-1459.

Sinclair, P. M. (1988): A three-dimensional field-integral method for the calculation of transonic flow on complex configurations - theory and preliminary results. Aeronautical J.. 92(916), 235-241.

Tseng, K. (1984): Application of The Green's Function Method for 2- and 3-Dimensional Steady Transonic Flows. AIAA Paper 84-0425.

Yates, E . C. Jr.; Cunningham, H. J.; Desmarais, R. N.; Silva, W. A.; Drobenko, B. (1982): Subsonic aerodynamic and Flutter characteristics of several wings calculated by the SOUSSA P1.1 panel method. AIAA Paper 82-0727.

List of Figures:

Figure 1. Computational model.

Figure 2. Surface panels.

Figure 3. Shock-free subsonic flow results.

3a. Surface local Mach contours.

3b. Surface pressure coefficient contours.

3c. Surface pressure coefficients.

Figure 4. Effect of shock-fitting.

4a. Surface local Mach contours without shock-fitting.

4b. Surface pressure coefficient contours without shock-fitting.

4c. Surface local Mach contours with shock-fitting.

4d. Surface pressure coefficient contours with shock-fitting.

4e. Surface pressure coefficients.

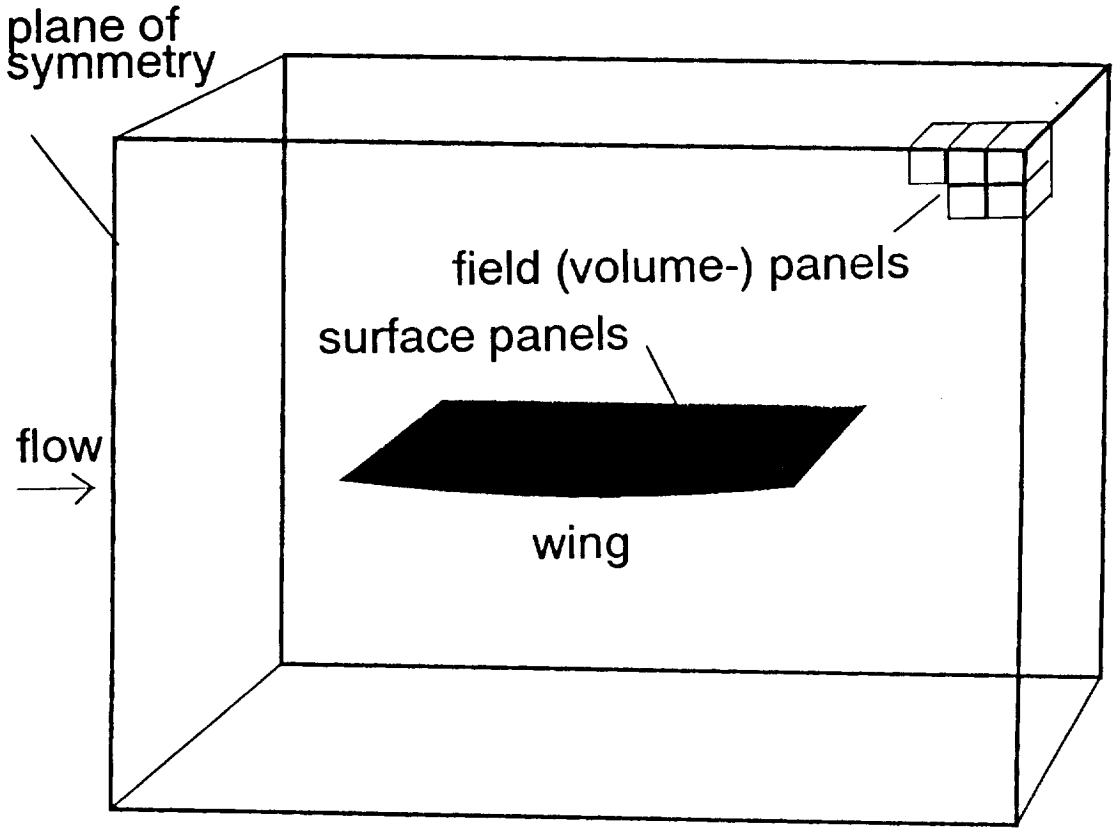
Figure 5. Tip-release effect.

5a. Tip thickness = 75% root thickness.

5b. Tip thickness = 50% root thickness.

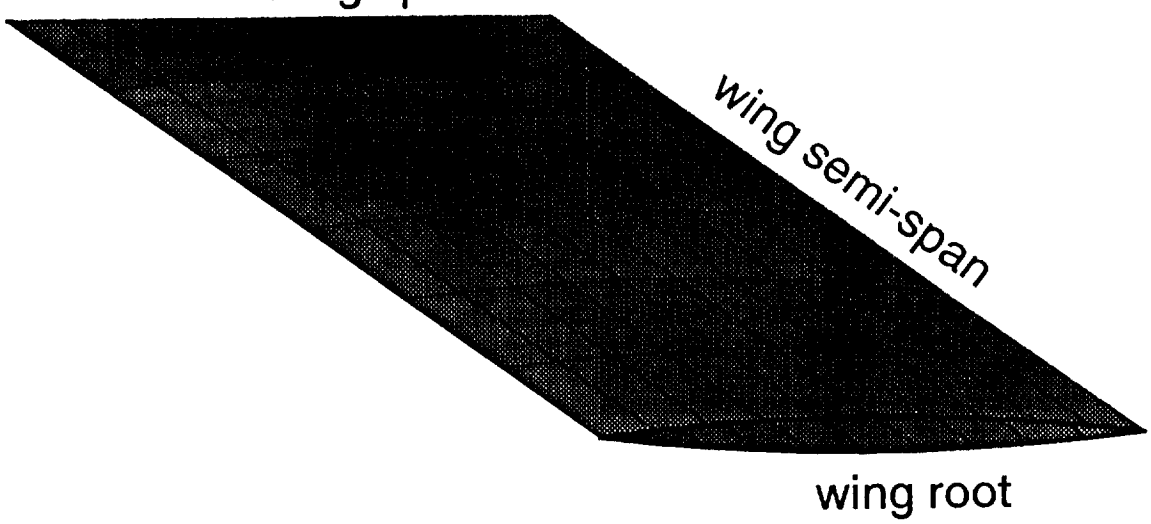
5c. Tip thickness = 0% root thickness.

computational domain - surfaec and field panels



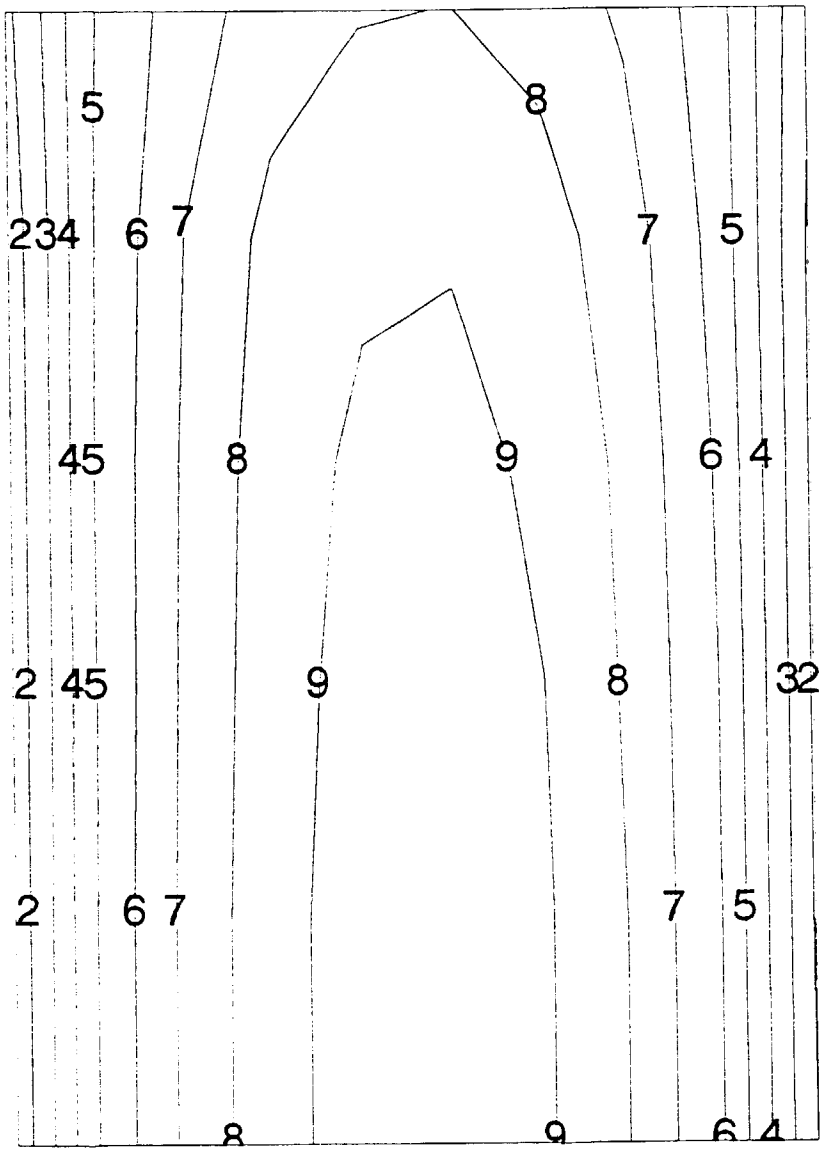
surface panels

6% circular arc section rectangular wing, AR=4



5% circular arc rectangular wing

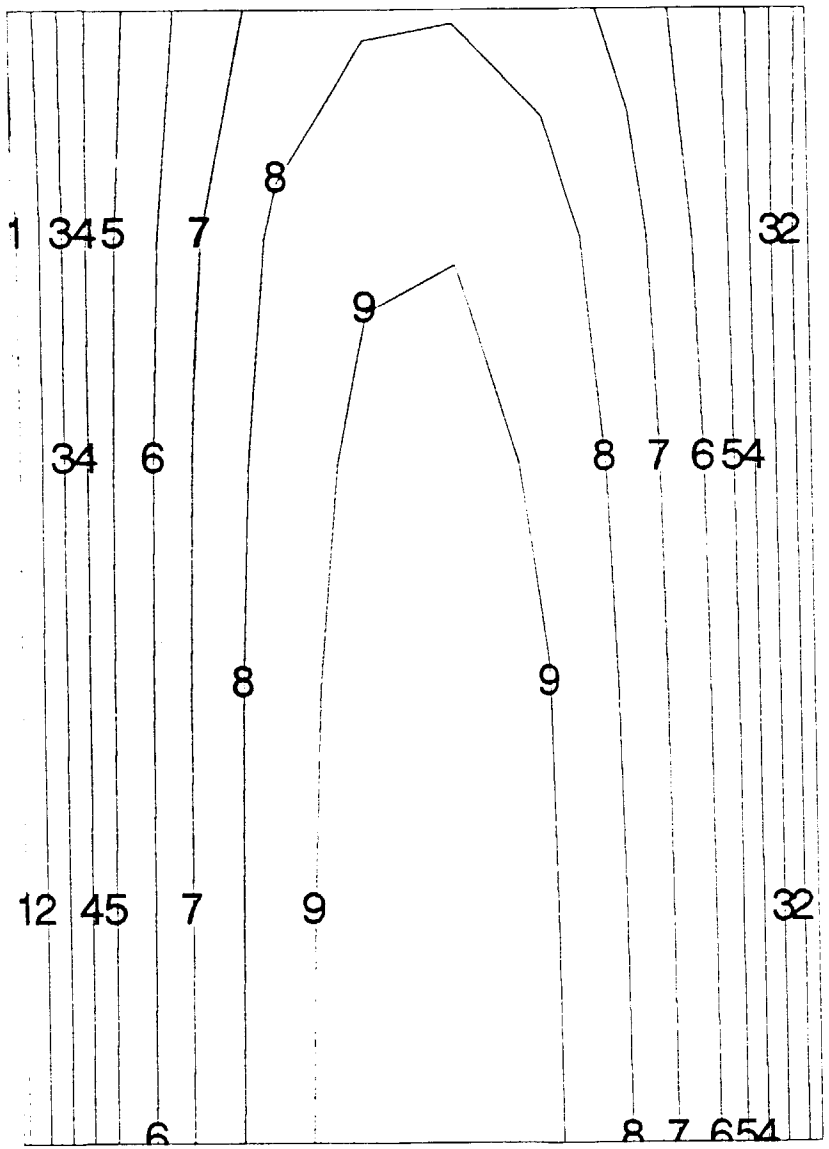
AR=3, M=0.7, ALPHA=0



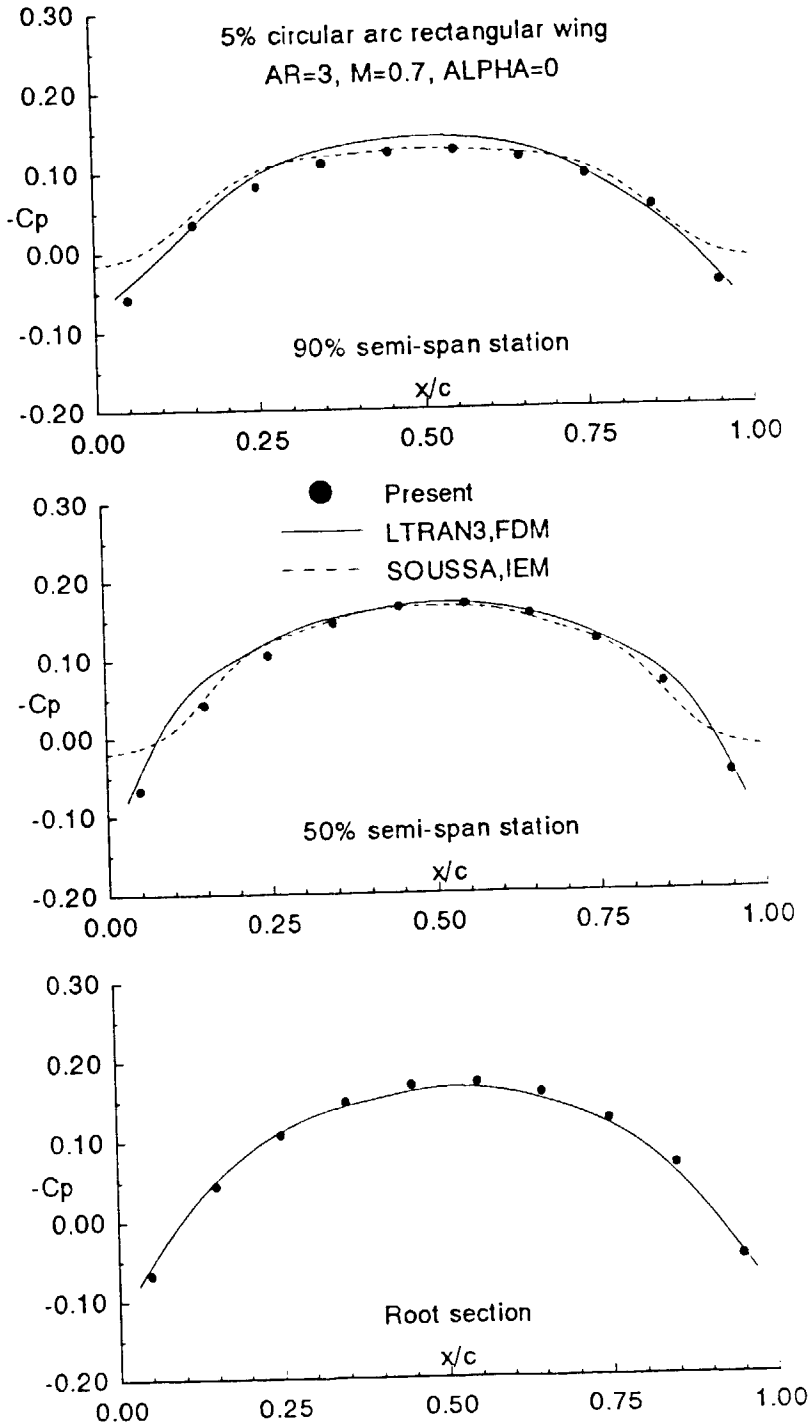
Level	M
A	0.77
9	0.76
8	0.75
7	0.74
6	0.73
5	0.71
4	0.70
3	0.69
2	0.68
1	0.67

5% circular arc rectangular wing

AR=3, M=0.7, ALPHA=0



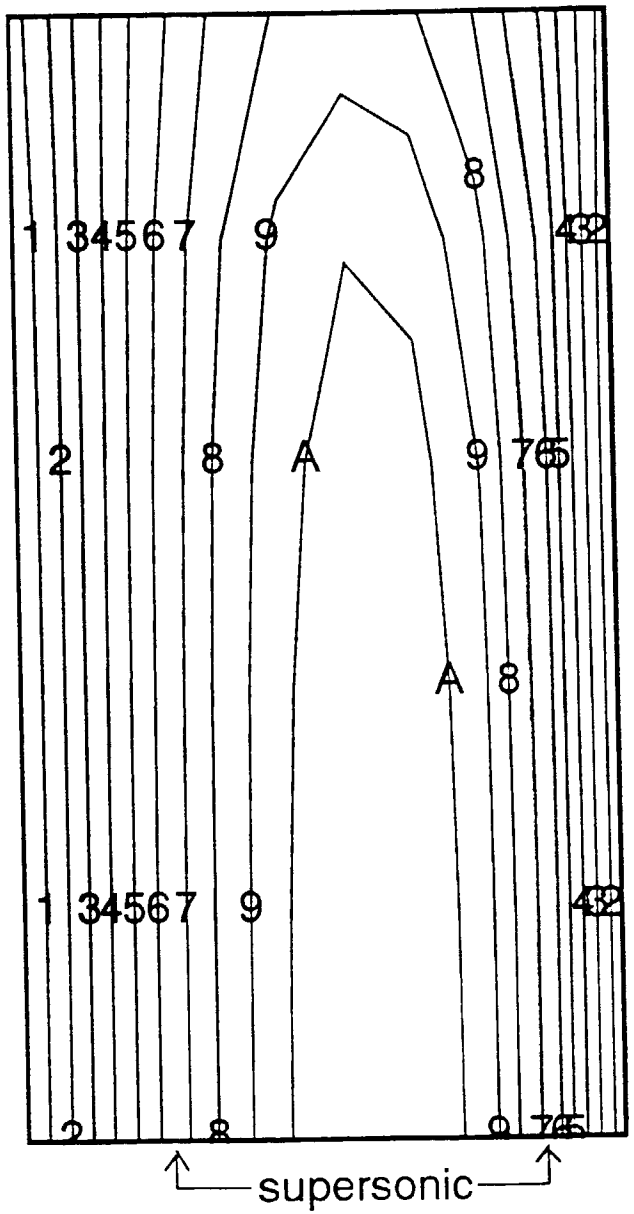
Level	CPN
A	0.18
9	0.15
8	0.13
7	0.10
6	0.07
5	0.05
4	0.02
3	-0.01
2	-0.03
1	-0.06


5% circular arc rectangular wing
AR=3, M=0.7, ALPHA=0
-Cp
x/c
90% semi-span station
50% semi-span station
Root section
Present
LTRAN3,FDM
SOUSSA,IEM
0.30
0.20
0.10
0.00
-0.10
-0.20
0.00
0.25
0.50
0.75
1.00

without shock-fitting

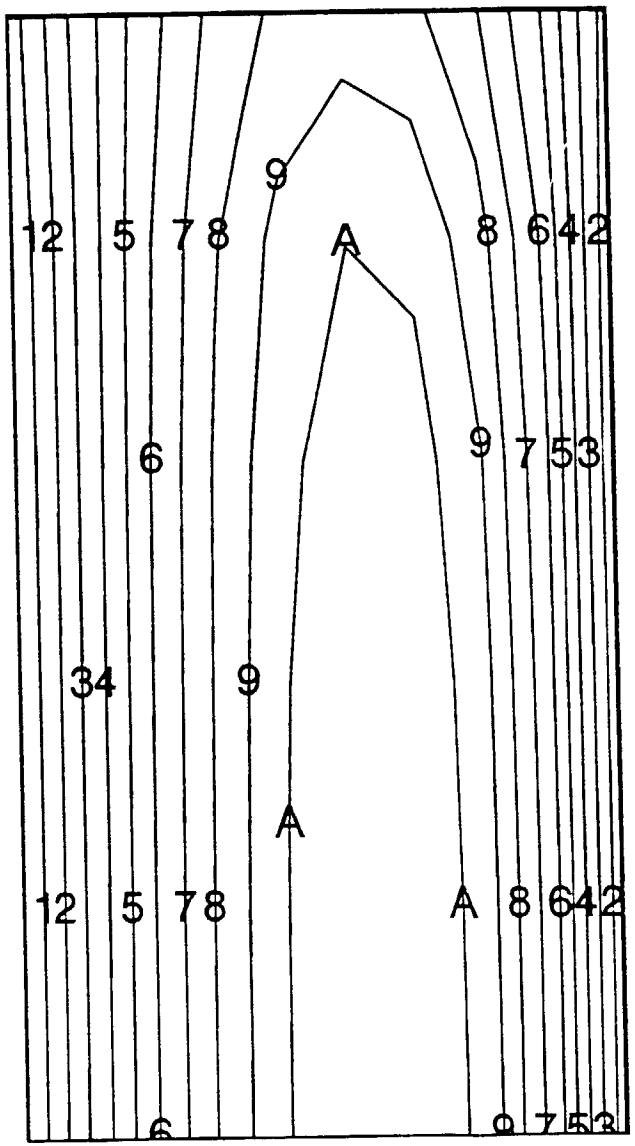
6% circular arc rectangular wing

AR=4, M=0.908, ALPHA=0



Level	M
A	1.06
9	1.04
8	1.02
7	1.00
6	0.97
5	0.95
4	0.93
3	0.91
2	0.89
1	0.87

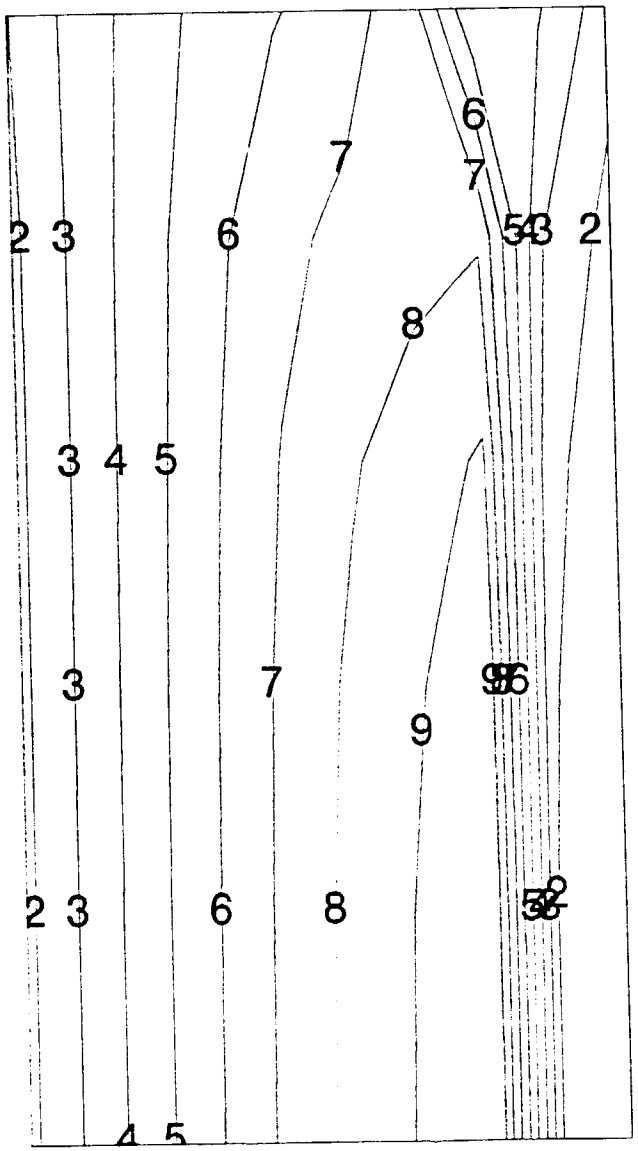
without shock-fitting
6% circular arc rectangular wing
AR=4, M=0.908, ALPHA=0



Level	CPN
A	0.27
9	0.23
8	0.20
7	0.16
6	0.12
5	0.08
4	0.04
3	0.01
2	-0.03
1	-0.07

6% circular arc rectangular wing

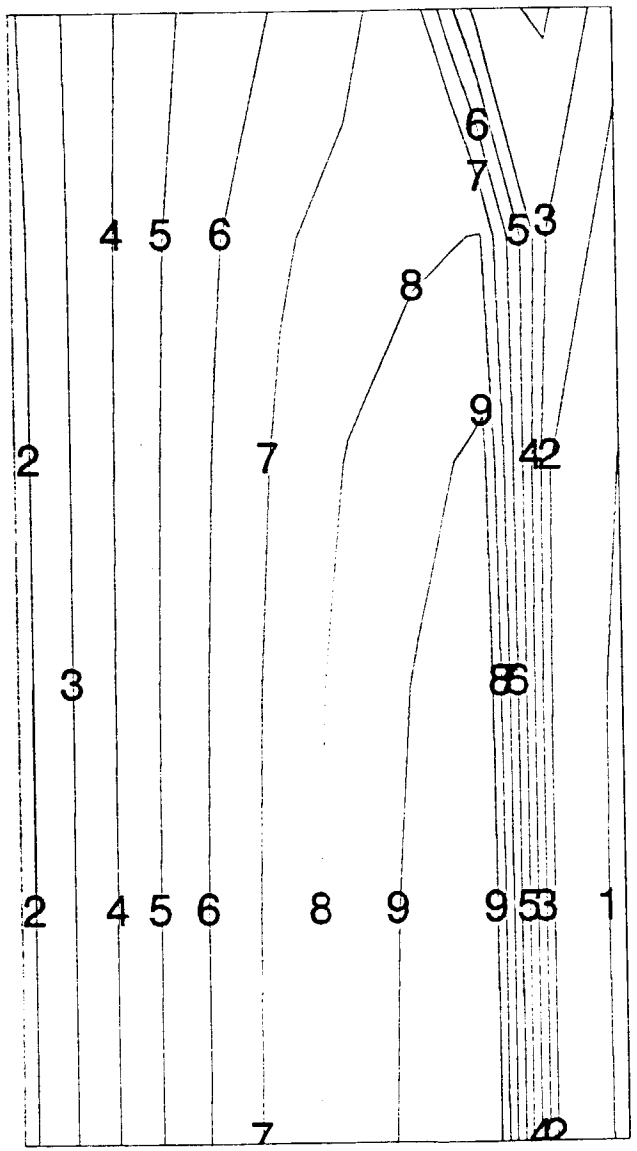
AR=4, M=0.908, ALPHA=0



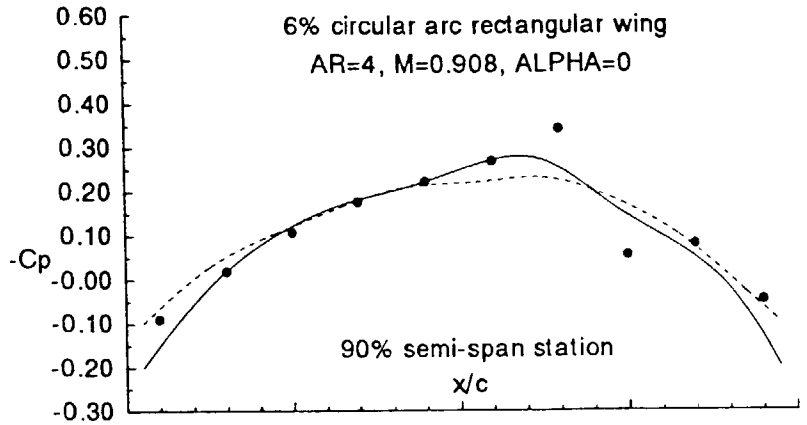
Level	M
A	1.20
9	1.16
8	1.12
7	1.07
6	1.03
5	0.99
4	0.95
3	0.90
2	0.86
1	0.82

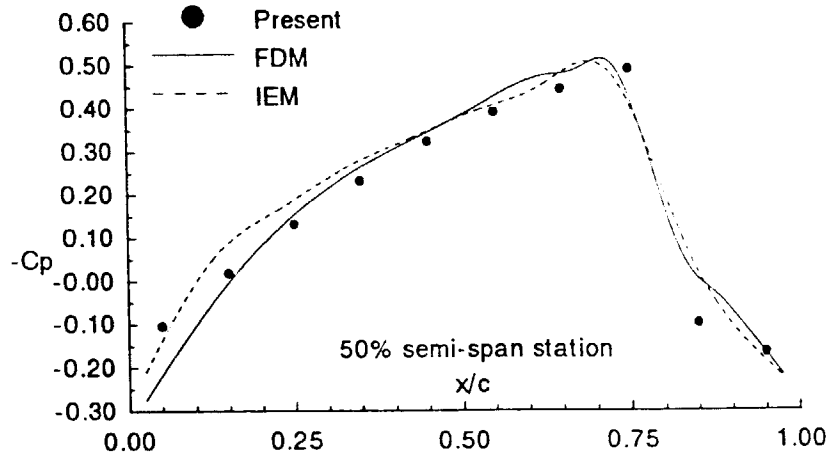
6% circular arc rectangular wing

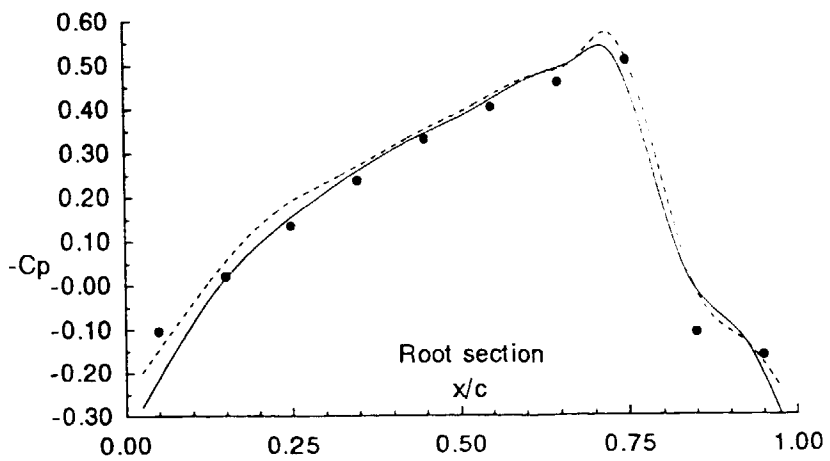
AR=4, M=0.908, ALPHA=0

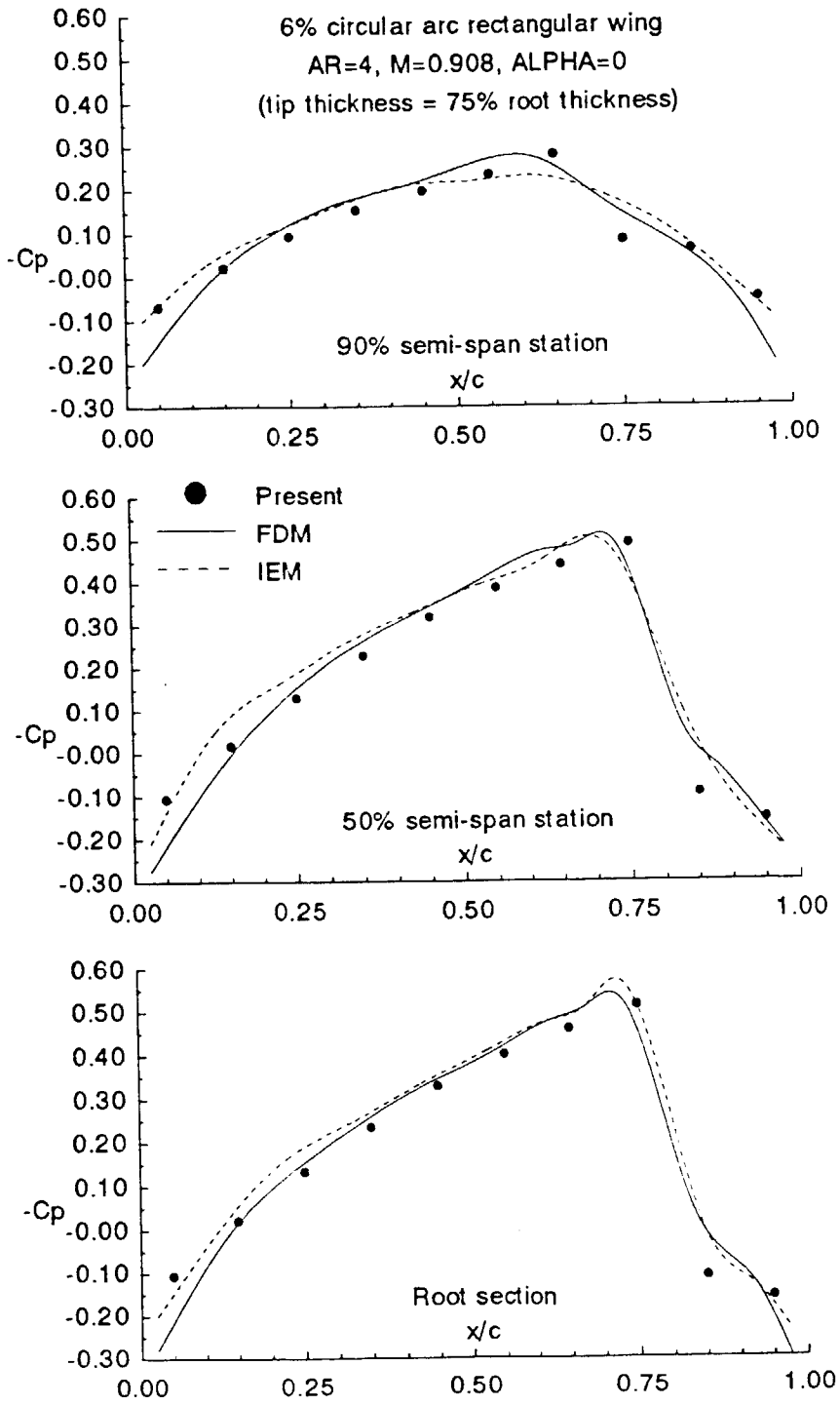


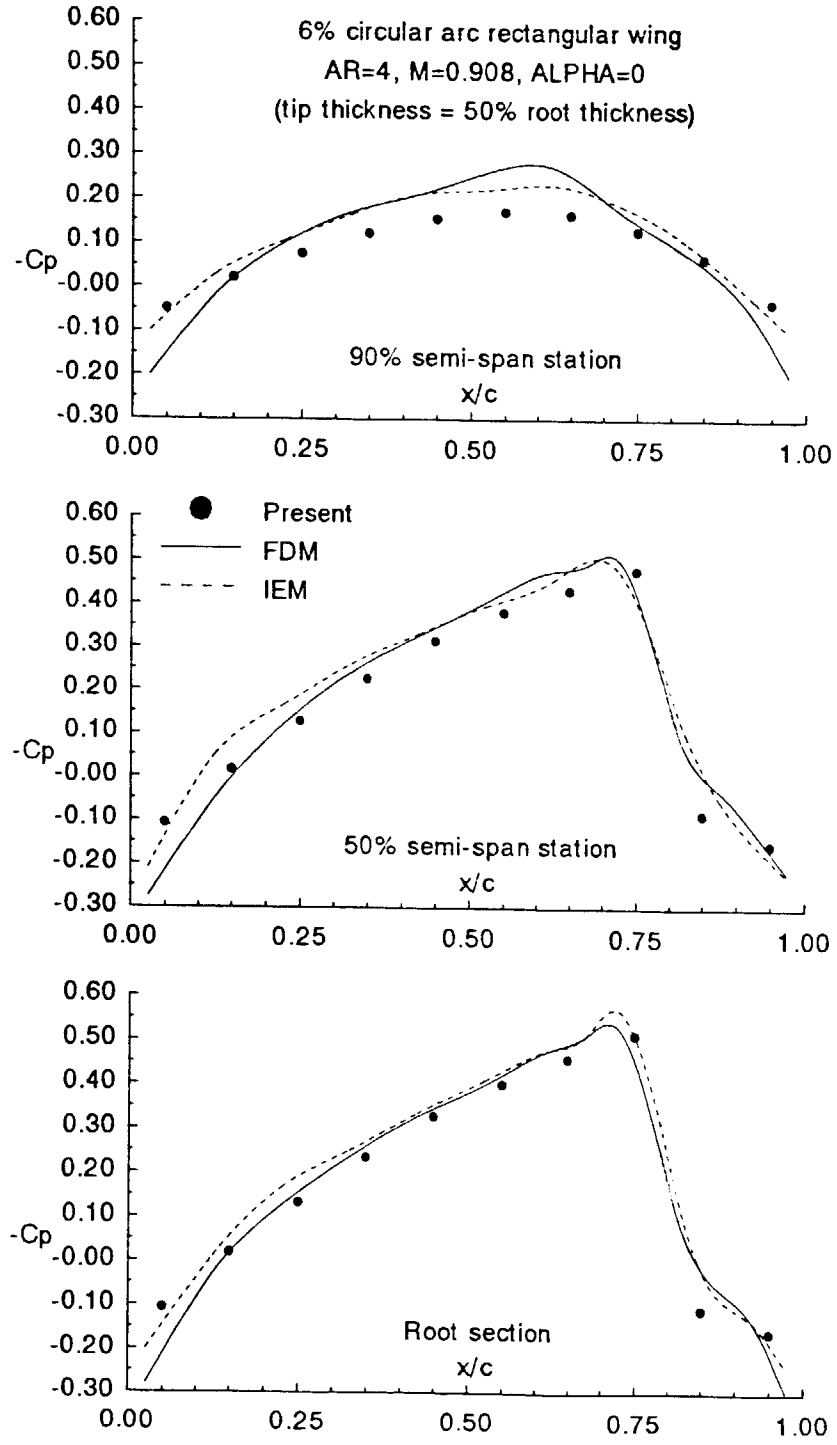
Level	CPN
A	0.51
9	0.44
8	0.36
7	0.29
6	0.22
5	0.14
4	0.07
3	-0.00
2	-0.08
1	-0.15

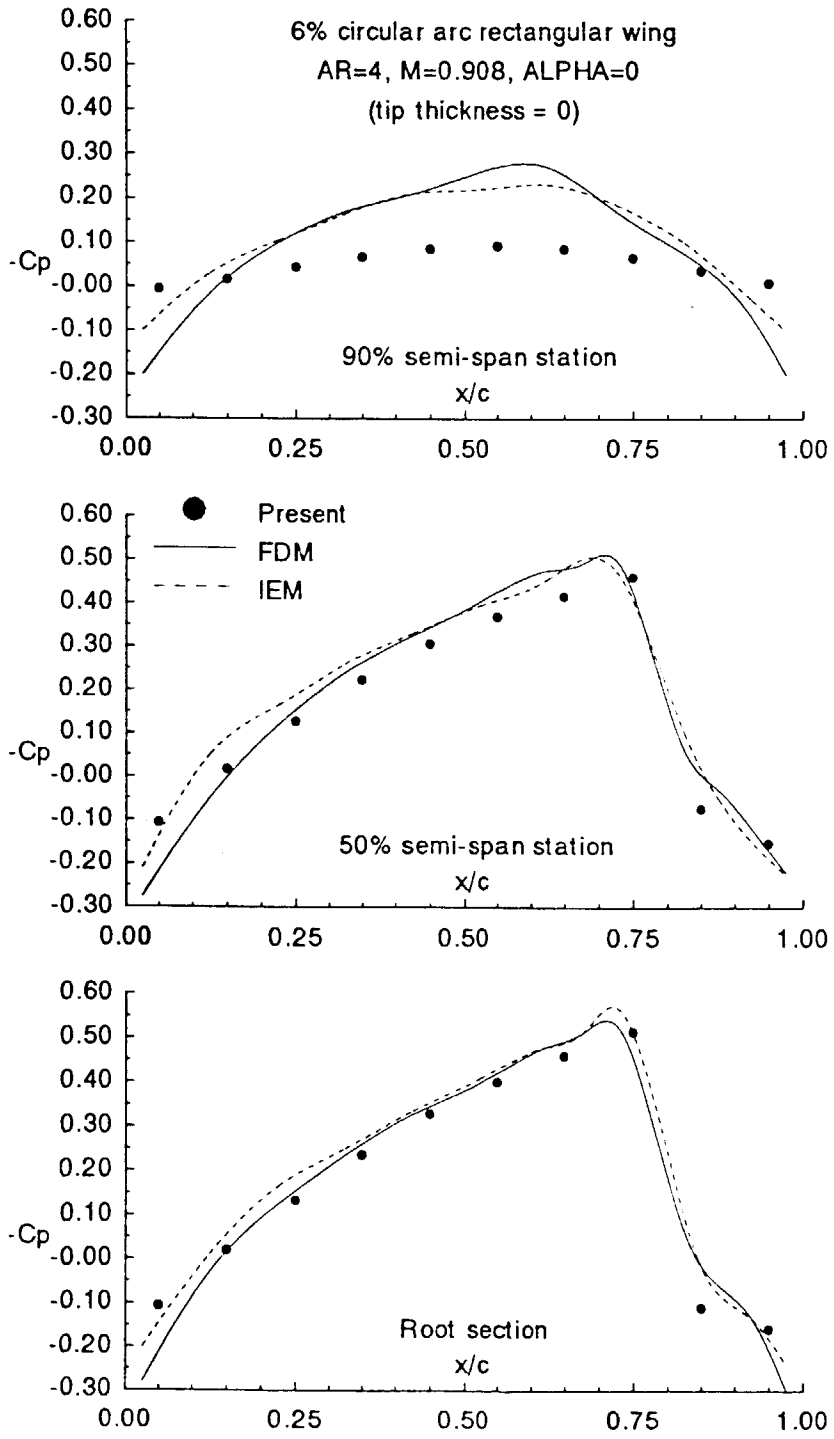

6% circular arc rectangular wing
AR=4, M=0.908, ALPHA=0
-Cp
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
90% semi-span station
x/c


Present
FDM
IEM
-Cp
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
50% semi-span station
x/c
0.00
0.25
0.50
0.75
1.00


-Cp
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
Root section
x/c
0.00
0.25
0.50
0.75
1.00


6% circular arc rectangular wing
AR=4, M=0.908, ALPHA=0
(tip thickness = 75% root thickness)
-Cp
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
0.00
0.25
0.50
0.75
1.00
90% semi-span station
x/c
Present
FDM
IEM
50% semi-span station
Root section


6% circular arc rectangular wing
AR=4, M=0.908, ALPHA=0
(tip thickness = 50% root thickness)
-Cp
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
0.00
0.25
0.50
0.75
1.00
90% semi-span station
x/c
Present
FDM
IEM
50% semi-span station
Root section


6% circular arc rectangular wing
AR=4, M=0.908, ALPHA=0
(tip thickness = 0)
-Cp
x/c
90% semi-span station
50% semi-span station
Root section
Present
FDM
IEM
0.60
0.50
0.40
0.30
0.20
0.10
-0.00
-0.10
-0.20
-0.30
0.00
0.25
0.50
0.75
1.00

No. 2

PERFORMANCE STUDY OF 3D INTEGRAL EQUATION COMPUTATIONS ON MASSIVELY PARALLEL COMPUTER

Terry G. Logan and Hong Hu
(Mr. Logan is a graduate student)
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Element Methods in Fluid Dynamics II
July 1994

C-2

SECOND INTERNATIONAL FLUID DYNAMICS
WORKSHOP
SOUTHAMPTON, UK, 12-13 JULY 1994

Conference Chairmen

D.B. Ingham
S. Kim
H. Power
L. Skerget
N. Tosaka

International Scientific Advisory Committee

A. Acrivos
A.H.-D. Cheng
G. Graziani
G.C. Hsiao
H. Hu
M.S. Ingber
N. Liron
R.M.M. Mattheij
N. Phan-Thien
H.L. Pina
C. Pozrikidis
L.C. Wrobel
J. Wu

Thanks are due to G.A.L. van de Vorst for the use of figure 2 on page 16 which appears on the front cover of this book

Boundary Element Methods in Fluid Dynamics II

Editors:

H. Power
Wessex Institute of Technology, UK

C.A. Brebbia
Wessex Institute of Technology, UK

D.B. Ingham
University of Leeds, UK

Computational Mechanics Publications
Southampton Boston

Performance study of 3D integral equation computations on massively parallel computer

T.G. Logan & H. Hu
Department of Mathematics, Hampton University, Hampton, Virginia 23668, USA

ABSTRACT

A comparative study of computational performance of CM-5 and Cray-YMP computers for a three-dimensional numerical source panel method calculation is made. A serial FORTRAN code is converted into a parallel CM-FORTRAN code. The performance results are obtained on CM-5 with 32, 64 and 128 nodes along with those on Cray-YMP with a single processor. The comparison of the performance indicates that the parallel CM-FORTRAN code out-performs the equivalent serial FORTRAN code, except for the post-aerodynamic calculations where the "parallel" code is not fully parallelized.

INTRODUCTION

In recent years, the processors of conventional vector supercomputers seem to be approaching the limit in computational speed inherent in their technology. However, the need for even faster computations continues to grow. As a consequence, massively parallel computers are being developed as a possible solution. Massively parallel computers, such as CM-5, are families of parallel computer architectures which may provide orders of magnitude improvement in computation performance in the near future over today's fastest conventional supercomputer. In fact the CM-5 computer with a maximum 16k nodes installed is a 2 TFLOPS computer in theory.

Computational fluid dynamics (CFD) is one of the areas which need super-fast computational power. The massively parallel computers has potential to become the main computational tool for CFD; it may replace the conventional supercomputers in the near future. The second author and his co-workers have made the performance study for a two-dimensional source panel method calculation[1], and the study shows that the parallel code achieved a high performance. The purpose of this paper is to present a similar study for a three-dimensional source panel method

calculation.

CM-5 AND CM-FORTRAN

The Connection Machine CM-5 system is a scalable distributed-memory multiprocessor system. The major hardware elements of the system include front-end computers to provide developing and execution environments and a parallel processing unit, which consists of multiple nodes, to execute parallel operations. It supports both the SIMD (Single Instruction Multiple Data) data parallel and MIMD (Multiple Instruction Multiple Data) message passing programming models. The maximum possible configuration for a CM-5 system is $16k$ nodes, where $k = 1024$. The CM-5 used under the present study has 128 nodes installed. Each node has 32 MB of memory, one SPARC processor and four vector processors for a theoretical peak performance of 128 MFLOPS. Therefore the CM-5 with 128 nodes has a theoretical peak performance of 16 GFLOPS.

The CM-FORTRAN language is an implementation of FORTRAN 77 supplemented with array-processing extensions from the standard FORTRAN 90. These array-processing features map naturally onto the data parallel (for SIMD model of parallel pregramming) architecture of the CM-5 system, since the CM- FORTRAN allows array elements to be evaluated simultaneously. The most important difference of CM-FORTRAN from FORTRAN 77 is the treatment of entire arrays as objects, thus explicit indexing in CM-FORTRAN is not always necessary. For example, it is not necessary to write Do-Loops or other such control constructs to have the operation repeated for each element of arrays. On the other hand for message passing models of parallel programming (MIMD), the program may be written in FORTRAN 77 along with message passing routines.

NUMERICAL SOURCE PANEL METHODS

The second author has been working on developing the field-panel method for compressible flows, including transonic flows. The computer code written in serial FORTRAN was developed[2] using the source field-panel method for calculating incompressible and compressible flows, including transonic flows, around a non-zero thickness three-dimensional wing configuration. The code was developed by modifying a zero-thickness configuration, vortex panel method[3]. As mentioned earlier, the present paper presents the parallel performance study for incompressible flows; while the parallel performance for compressible flows, including transonic flows, is under investigation and to be reported in the near future.

For incompressible flows, the governing equation is given by the Laplace equation,

$$\nabla^2 \Phi = 0 \tag{1}$$

where Φ is the velocity potential, $\vec{V} = \nabla\Phi$.

The integral equation solution of Eq. (1) for source panel method, in terms of velocity field ($\vec{V} = \nabla\Phi$), is given by

$$\vec{V}(x,y,z) = \vec{V}_\infty - \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{q_{i,j}(\xi,\eta,\zeta)}{d^2}\vec{e}_d\, ds(\xi,\eta,\zeta) \tag{2}$$

where the subscript ∞ refers to the free-stream condition, $LG \times NG$ is the total number of panels; $q_{i,j}$ is the wing-surface source distribution, which is unknown to be determined by applying boundary condition; ds is the infinitesimal surface area; the $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$, and where $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$.

The wing-surface zero-normal-velocity boundary condition is applied at each control point of all panels,

$$\vec{V}(x,y,z)\cdot\vec{n} = 0 \qquad \text{on} \quad CP_{I,K} : I = 1, NL; K = 1, NG \tag{3}$$

Applying Eq. (2) to Eq. (3), a system of equation is obtained,

$$[A]\{q\} = \{B\} \tag{4}$$

where $[A]$ is $N \times N$ aerodynamic influence coefficient matrix, and $N = LG \times NG$; $\{q\}$ is a $N \times 1$ unknown vector matrix containing q_j for $j = 1$ to N; and $\{B\}$ is a $N \times 1$ known vector matrix which is contributed from $\vec{V}_\infty$.

The solution procedure of the problem using source panel method involves major three steps: (1) evaluation of integrals for N^2 times to construct matrices $[A]$ and also $[B]$; (2) solving the resulting dense linear system of Eq. (4); (3) post-processing of acrodynamic calculations. It should be noted that the Step (1) involves evaluating a large number of integrals. The total number of integrals can be very large for acrodynamic problems, for example, it can be in the order of 10^8 if $LG \times NG = 100 \times 100$. An important feature of the Step (1) is that the calculation for each (x,y,z) and each (ξ,η,ζ) can be performed simultaneously for all (x,y,z) and all (ξ,η,ζ). This feature of panel method calculation leads itself in a natural way for processing data in a parallel computing environment.

PARALLEL FORTRAN IMPLEMENTATION

Before the manual conversion, the CMAX translator is used to partially convert the incompressible flow part of serial FORTRAN code of Ref. 2 into parallel CM-FORTRAN code under the data parallel (SIMD) programming model. But most of the conversion is done manually.

The serial FORTRAN routine for evaluation of integrals to construct $[A]$ and $\{B\}$ matices is a basically four-nested Do-loop along with a

double-nested If- statement where self-induced velocity, near-field and far-field induced velocity are calculated separately within the loop. When the code is executed on Cray-YMP, the vectorization of the most inner Do-Loop is automatically done through the vectorization capability of the FORTRAN 77 compiler. This part of the code is fully converted into parallel CM-FORTRAN where Do-Loop no longer exists instead, four-dimensional arrays are used. The double-nested If-statement is replaced by a double-nested Where-Elsewhere-Endwhere parallel statement.

For solving the matrix system, Eq. (4), both serial and parallel versions of the code use the Gaussian elimination scheme with partial pivoting. For the parallel version, the CMSSL(Connection Machine Scientific Software Library) is called for the Gaussian elimination with different numbers of blocks.

For the post-processing aerodynamic calculations, the code is partially converted. The two most inner loops are fully parallelized, while the two outer loops remains serial.

PERFORMANCE STUDY

The serial code with two different numbers of panels (N) is executed on Cray-YMP supercomputer using single processor to provide the basis for performance comparison. The parallel CM-FORTRAN code is then executed on CM-5 with 32, 64 and 128 nodes under slicewise model for selected runs. The same computational results are obtained using both codes.

Table 1 gives the detailed CPU time results on the Cray-YMP and the CM-5 to construct the matrices. It is founded that the CM-5 out-performs the Cray-YMP by small margin for $N = 24 \times 12$. When the size is increased to $N = 48 \times 24$ the CM-5 out-performs the Cray-YMP by a much larger margin, even with only 32 nodes.

Table 2 displays CPU time for the Gaussian elimination routine, and it tells us that the Cray-YMP out-performs the CM-5 for $N = 24 \times 12$. However its performance relative to the CM-5 decreases drastically when N increases; and in fact we can see that the CM-5 out-performs the Cray-YMP for $N = 48 \times 24$.

Table 3 lists the CPU time for post processing calculations and Table 4 gives total CPU times. The results are self-explanatory.

Figure 1 shows the effect of the block sizes used in the CMSSL Gaussian elimination for $N = 24 \times 12$, while Figure 2 shows a similar result for $N = 48 \times 24$.

CONCLUDING REMARKS

The study of the performance of CM-5 with the comparison of Cray-YMP computer for the three-dimensional source panel method calculations is

Table 1. CPU time in seconds for constructing matrices.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.44	8.25
32-node CM5	0.24	1.53
64-node CM5	0.17	
128-node CM5	0.12	

Table 2. CPU time in seconds for Gaussian elimination.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.58	33.85
32-node CM5	1.43	6.62
64-node CM5	2.23	
128-node CM5		

Table 3. CPU time in seconds for post-processing.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	0.24	4.30
32-node CM5	2.42	11.55
64-node CM5	2.49	
128-node CM5		

Table 4. Total CPU time in seconds.

size, N=	288(=24x12)	1152(=48x24)
Cray-YMP	1.33	46.60
32-node CM5	3.92	18.55
64-node CM5	3.96	
128-node CM5		

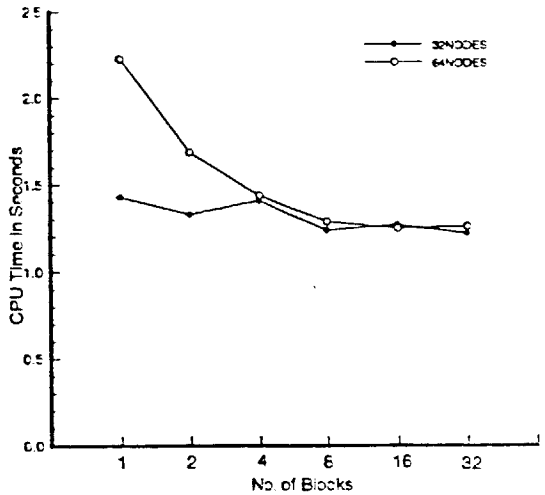


Fig. 1 Effect of number of blocks in Gaussian elimination, N=24x12.

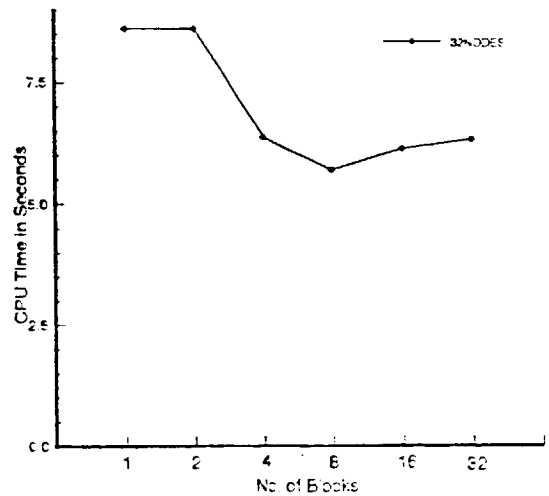


Fig.2. Effect of number of blocks in Gaussian elimination, N=48x24.

made. The high performance results obtained on CM-5 are encouraging. From this study, the following remarks can be made: (1) For calculating integrals to construct $[A]$ and also $\{B\}$ matrices, the CM-5 out-performs the Cray-YMP even with 32 nodes; (2) For solving a small matrix system, such as 24 x 12, Cray-YMP out-performs CM-5, but on the other hand, when the size of the matrix system increases to 48 x 24, the CM-5 near-performs the Cray-YMP; (3) Since the post- processing acrodynamic calculation part is not fully parallelized, the CM-5 under-performs the Cray-YMP on this part of calculation; (4) For the entire code with 48 x 24 source panels, the CM-5 with 32 nodes out-performs the Cray-YMP; (5) the optimal number of blocks used in Gaussian elimination on the CM-5 varies with the different numbers of nodes and the size of the matrix system. Currently the performance study for three-dimensional compressible flows, including transonic flows, is being made.

ACKNOWLEDGEMENTS

This work has been supported by NASA LaRC under the Grant No. NAG-1-1170 with Mr. Walter Silva as the technical monitor. The CM-5 computation is supported by NASA-ARC under the NAS Program.

REFERENCES

1 Hu, H. and Jackson, I. T. Comparative study of computational performance of CM-2 and Cray-YMP for boundary element computations. *Boundary Elements Abstracts Journal and Newsletter*, Vol. 3, No. 5, 1992, pp.185-191.

2 Hu, H. A 3D IEM for compressible wing flows with and without shocks, *Boundary Elements in Fluid Dynamics*, Computational Mechanics Publications and Elsevir Applied Science, 1992, pp. 49-60.

3 Chu, L-C. Integral equation solution of the full-potential equation for three-dimensional, steady, transonic wing flows, Ph.D. Dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March 1988.

No. 3

PANEL METHOD COMPUTATIONAL PERFORMANCE ON CM-5 AND CRAY-YMP

Hong Hu and Jada M. Paysour
(Ms. Paysour is an undergraduate in engineering)
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Submitted for Publication in the Journal
Boundary Elements Communications
June 1994

Panel Method Computational Performance on CM-5 and Cray-YMP

Hong Hu
Department of Mathematics, Hampton University, Hampton, Virginia, USA

Jada M. Paysour
Department of Engineering, Hampton University, Hampton, Virginia, USA

ABSTRACT

The study of computational performance of a two-dimensional source panel method code on the massively parallel computer, CM-5, is made. A serial FORTRAN code running on Cray-YMP supercomputer is converted into a parallel CM-FORTRAN code for running on CM-5. Detailed performance results are obtained for CM-5 with 32 nodes, 64 nodes and 128 nodes and for Cray-YMP with a single processor. The comparison of the performance indicates that CM-5 out-performs Cray-YMP by a factor of 13 for the largest problem tested and achieves a speed of about 2 GFLOPS.

1. INTRODUCTION

Computer with massively parallel processors (MPPs) may provide orders of magnitude improvement in computational performance in a near future over today's fastest conventional vector supercomputer. The MPP computers employ a large number of small processors, which are much less expensive to produce than the vector supercomputer processors, and connect them together such that the computations can be done in parallel to achieve extrem high performance. Computational fluid dynamics (CFD) is one of the areas which need super- fast computational power. The integral equation panel method is the one of the CFD methods, which seems appropriate for parallel processing. The first author

and his co-worker[1] did the performance study of an source panel method computational code on CM-2, a MPP computer. In that work[1], the resulting linear system was solved by a direct method, which was found to be inefficient and hence expensive for large system. This short paper presents the recent work based on Ref. [1], and here the resulting linear system is solved by a much more efficient iterative method.

2. ABOUT CM-5

The Connection Machine CM-5 system is a scalable distributed-memory multiprocessor system. The major hardware elements include front-end computer to provide developing and execution environments and a parallel processing unit to execute parallel operations. The system support both the SIMD (Single Instruction Multiple Data) data parallel and MIMD (Multiple Instruction Multiple Data) message passing programming models. The maximun possible configuration for the system is $16k$ nodes, where $k = 1024$. Each node has one SPARC processor and four vector processors for a therotical peak performance of 128 MFLOPS. Therefore the CM-5 with maximun $16k$ nodes installed would be a 2 TFLOPS machine theoretically.

3. CFD-PANEL METHOD

The physical problems considered here are potential flows around any non-lifting two-dimensional configurations. The governing equation to this type of problem is given by the Laplace equation,

$$\nabla^2 \Phi = 0 \tag{1}$$

where Φ is the velocity potential, $\vec{V} = \nabla\Phi$. This type of problems can be solved by integral equation source panel methods (or called boundary element method). The integral

equation solution in terms of velocity field is given by

$$\vec{V}(x,y) = \vec{V}_\infty - \frac{1}{2\pi}\sum_{i=1}^{N}\int_{g_i} q_{g_i}(\xi,\eta)\frac{(x-\xi)\vec{i}+(y-\eta)\vec{j}}{(x-\xi)^2+(y-\eta)^2}ds(\xi,\eta) \tag{2}$$

where $q_i's$ are surface source distributions, which are unknowns to be determined by applying boundary conditions; the subscripts g refers to the body surfaces; and N is the total number of surface panels.

It should be noted that Eq. (2) involves evaluating a large number of integrals over body surface if the value of N is large. The total number of panels can be very large for three-dimensional aircraft configurations, and it can be, for example, in the order of 10^4. An important feature of Eq. (2) is that the calculation for each (x,y) and each (ξ,η) can be performed simultaneously for all (x,y) and all (ξ,η) with a single instruction. This feature of panel method calculation leads itself in a natural way for processing data in a SIMD parallel computing environment.

By applying body surface zero-normal-velocity condition at each $(x,y) = (x_j,y_j)$ for $j = 1$ to N, a $N \times N$ linear system of equations is obtained as

$$[A][q] = [B] \tag{3}$$

where $[A]$ is $N \times N$ aerodynamic influence coefficient matrix; $[q]$ is a $N \times 1$ unknown vector matrix containing q_i for $i = 1$ to N; and $[B]$ is a $N \times 1$ known vector matrix which is contributed from the free-stream velocity, $\vec{V}_\infty$.

The solution procedure for this problem using panel method involves four steps: (a) generating body geometry information; (b) evaluating integrals of Eq. (2) for $i = 1$ to N and for $(x,y) = (x_j,y_j)$ with $j = 1$ to N to construct $[A]$ matrix; (c) solving resulting linear system of Eq. (3); and (d) post-processing, aerodynamic calculations using Eq. (2). The experience shows that the Steps b and c takes most computational time, which is

partically true for three-dimensional computations[2].

4. PERFORMANCE

As mentioned earlier, this work is based on that of Ref. [1]. Due to the inefficiency of the direct solver[1], the resulting linear system is solved by an indirect, iterative method. The Jacobi mathod is employed, since the method is very much appropriate for parallel environment. List 1 and List 2 are the subroutines for Jacobi iterations in serial FORTRAN and parallel CM- FORTRAN versions, respectively.

The serial code with different numbers of panels (N) is first executed on Cray-YMP supercomputer using single processor to provide the basis for performance comparison. The computational performance in terms of MFLOPS is obtained using Cray-YMP's PERFTRACE utility. The parallel CM-FORTRAN code is then executed on CM-5 with 32 nodes, 64 nodes and 128 nodes under slicewise model.

Table 1 gives the detailed performance results for Cray-YMP and CM-5 computers with varying size of the problem. The performance of CM-5 in terms of MFLOPS is the equivalent Cray-YMP performance. In Table 1, "Mat Coef" refers to evaluating aerodynamic enfluence coefficients; "Lin Syst" refers to solving linear system using Jacobi method; and "Total" refers to solving entire code. The sets of results from Table 1 have been extracted, and are presented in Figs. 1-4.

Fig.1 shows execution time for evaluating aerodynamic influence coefficient matrix, $[A]$, and the matrix $[B]$ on Cray-YMP and CM-5 computers for different numbers of panels. It can be seen that the CPU execution time decreases with the increase of the number of CM-5 nodes after the size of the problem is large enough to fully use all nodes. For example when $N = 1024$, the CPU time of 0.216 seconds with 32 nodes is reduced to 0.114 seconds with 64 nodes, and then is further reduced to 0.061 seconds with 128 nodes. That is to say that whenever the number of nodes used is doubled, the CPU time is almost reduced

by a factor of 2 - a near-perfect parallelization. It is also seen that when the problem size is large enough the CPU time required on CM-5, even with 32 nodes, is significantly (note that Log_{10}-axis is used for execution time !) less than that required on Cray-YMP.

Fig. 2 is the CPU time for solving linear system using Jacobi iterations. The results tell us that, when the N is large enough, CM-5 out-performs Cray- YMP and the Jacobi method is very efficient.

Fig. 3 shows the total CPU time for solving entire problem. The results are self-explanatory. Fig. 4 is a partial reproduction of Fig. 3 for performance results on Cray-YMP and CM-5 with 128 nodes, and it is represented in terms of MFLOPS. From this figure it is clearly seen that the CM-5 performs at about 2 GFLOPS when $N = 1024$. The speed achieved here is very encouraging, which is much faster than that achieved on Cray-YMP.

5. CONCLUDING REMARKS

A source panel method code is successfully implemented on the MPP computer CM-5 using CM-FORTRAN language. The linear system is solved by the efficient iterative Jacobi method. The detailed performance results are obtained and analysed. The parallel CM-FORTRAN code achieves a very high performance and for most of the cases tested here it out-performs Cray-YMP supercomputer. The highest speed achieved in this investigation is about 2 GFLOPS which is very encouraging.

ACKNOWLEDGEMENTS

This work is supported by NASA Langley Research Center under the Grant No. NAG-1-1170. Mr. Walter Silva is the technical monitor. The CM-5 computation is supported

by NASA under NAS Program.

REFERENCES

1. Hu, H. and Jackson, I. T., "Comparative study of computational performance of CM-2 and Cray-YMP for boundary element computations", *Boundary Elements Communications*, (formly Boundary Elements Abstracts Journal and Newsletter), 1992, Vol. 3, No. 5, pp.185-191.
2. Hu. H., "Study of integral equation methods for transonic flow calculations", *Engineering Analysis with Boundary Elements*, 1993, Vol. 11, No. 2, pp. 101-107.

LIST OF FIGURES:

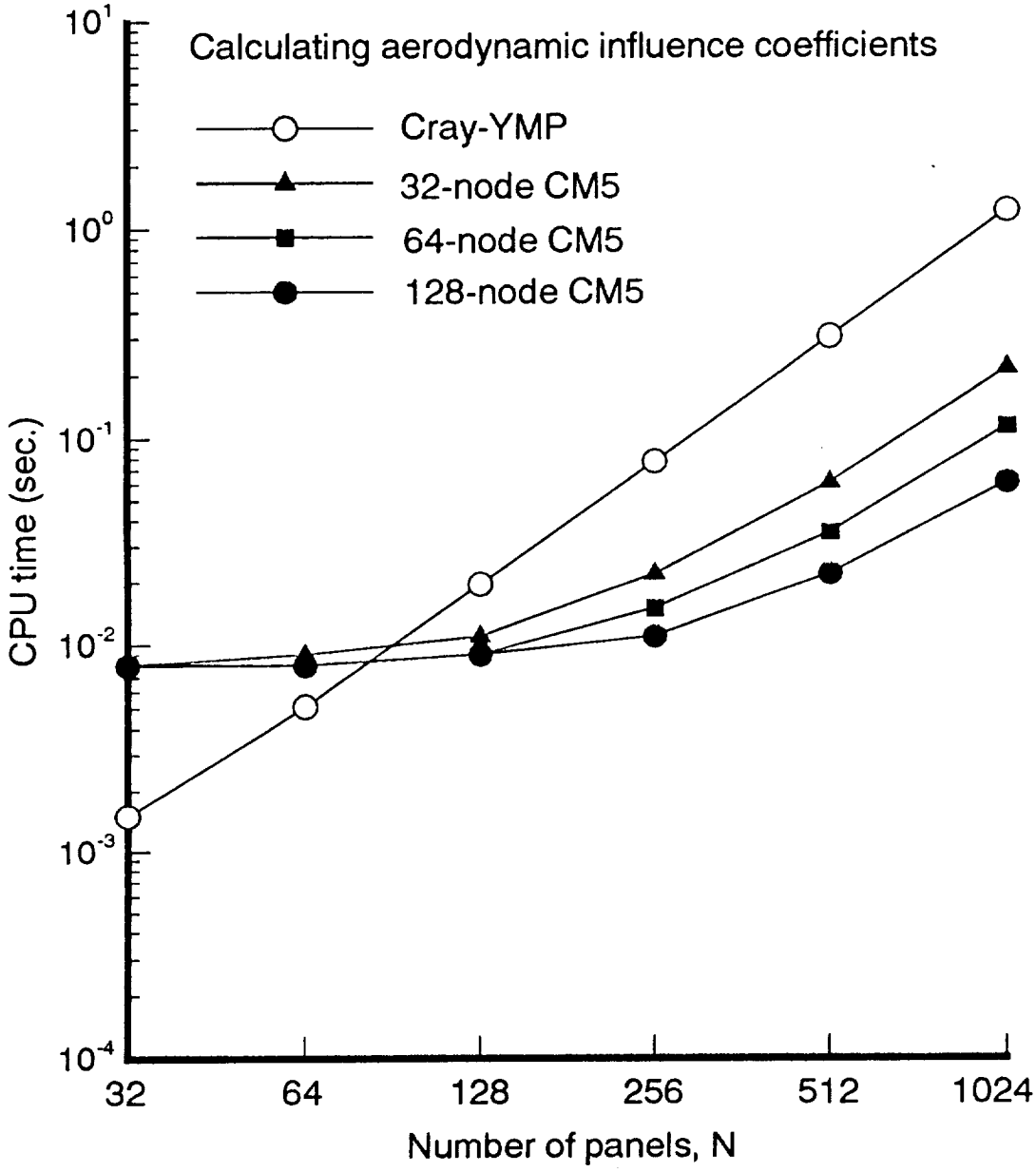
List 1. Jacobi method in serial version.
List 2. Jacobi method in parallel version.
Table 1. The detailed computational performance results.
Figure 1. CPU time for evaluating aerodynamic enfluence coefficients.
Figure 2. CPU time for solving linear system using Jacobi method.
Figure 3. CPU time for solving entire problem.
Figure 4. Performance for solving entire problem.

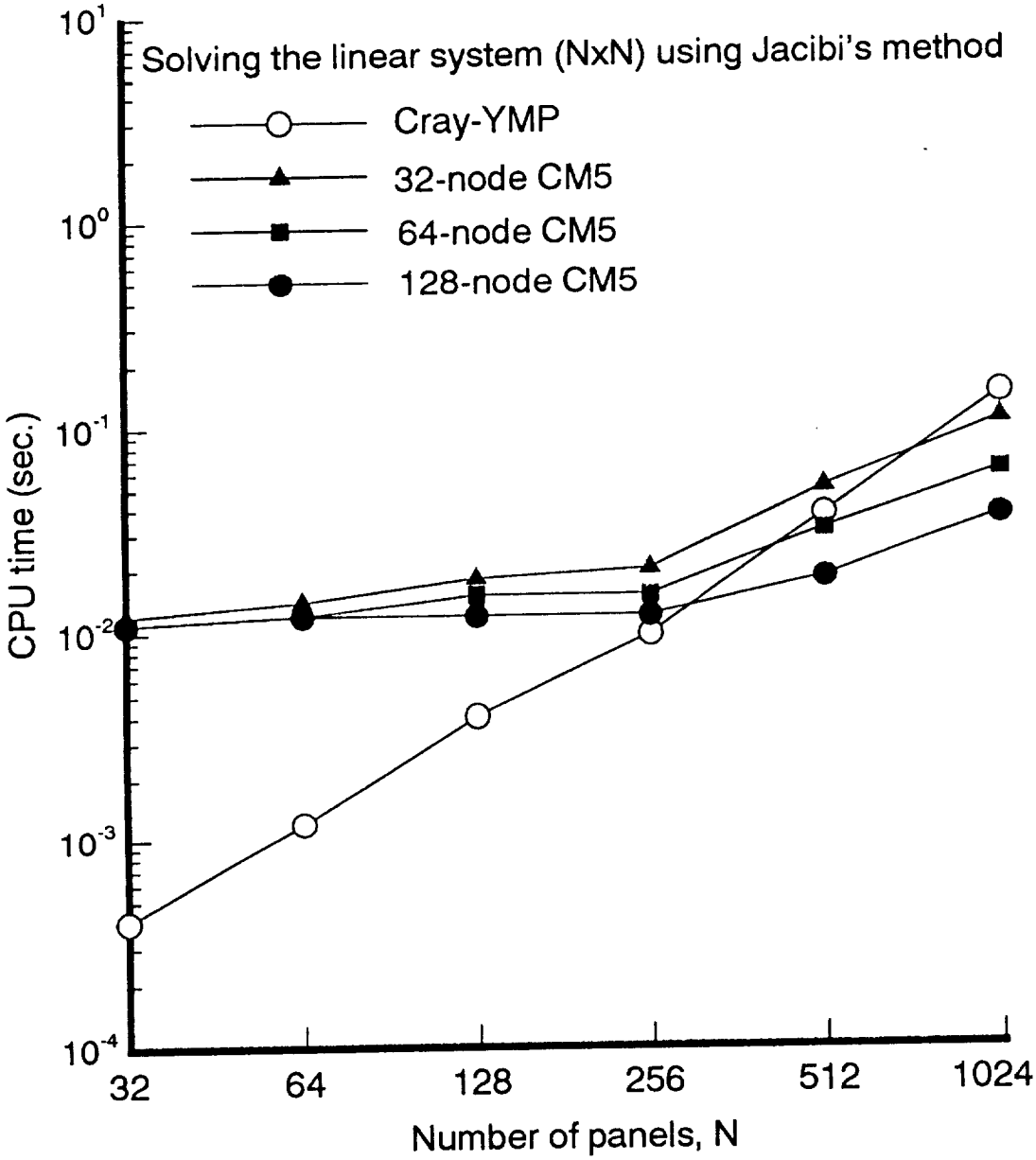
```
      SUBROUTINE JBINT(A,B)
      PARAMETER (N=128,M=128)
C -----------------------------------------------
C SOLVE AX=B USING APPROXIMATE JACOBI ITERATIONS
C ----- SERIAL FORTRAN VERSION
C -----------------------------------------------
      DIMENSION A(N,N),B(N),X(N,100)
      INTEGER VAR
      MAXITER=50
      TOL=0.001
      AX0=0.0
      XMAXDIF=0.0
      DO 1000 I = 1,N
      X(I,1) = 0.0
 1000 CONTINUE
      K = 1
 66   K= K+ 1
      DO 200 I = 1,N
      DO 300 J=1,N
      VAR = K-1
      IF (J.EQ.I) GOTO 300
      AX0=A(I,J)*X(J,VAR) + AX0
 300  CONTINUE
      X(I,K) = 1/A(I,I)*(B(I)-AX0)
      XDIF = ABS(X(I,K)-X(I,K-1))
      IF (XDIF.GT.XMAXDIF) XMAXDIF=XDIF
      AX0=0.0
 200  CONTINUE
      IF (XMAXDIF.LT. TOL) THEN
      MAXK = K
      GOTO 99
      ENDIF
      XMAXDIF = 0.0
      IF (K.LT. MAXITER) GOTO 66
      PRINT*,'NOT CONVERGENT YET AFTER ITERATIONS:',MAXITER
      RETURN
 99   CONTINUE
      DO 400 I=1,N
      B(I)=X(I,MAXK)
 400  CONTINUE
      RETURN
      END
```

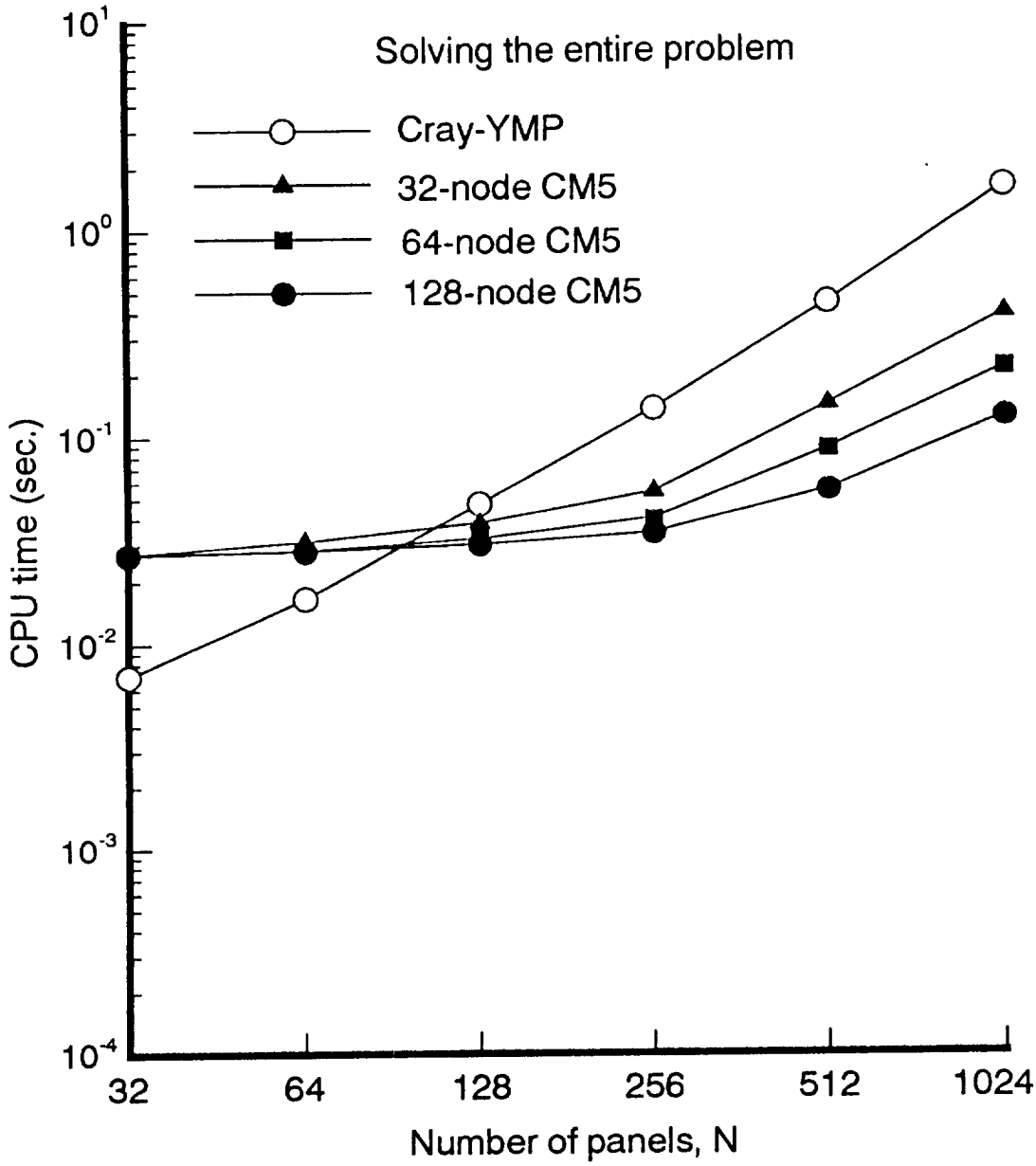
```
      SUBROUTINE jbite
      PARAMETER (N=128,M=128)
C -----------------------------------------------
C SOLVE AX=B USING APPROXIMATE JACOBI ITERATIONS
C ----- PARALLEL CM-FORTRAN VERSION
C -----------------------------------------------
      DIMENSION A(N,N),B(N),X(N,100),ax0(N),c(N)
      INTEGER VAR
      REAL  XDIF100(n)
      COMMON/BLK2/A
      COMMON/BLK3/B
      maxiter = 50
      tol = 0.001
      x(:,1) = 0.0
      k = 1
   66 k = k + 1
      var = k - 1
      FORALL (I=1:N) AX0(I)=DOTPRODUCT(A(I,:),X(:,VAR))-A(I,I)*X(I,VAR)
      forall (i=1:n) c(i)=a(i,i)
      x(:,k)=1.0/c *(b-ax0)
      xdif100(1:n) = abs(x(:,k) - x(:,k - 1))
      XMAXDIF=MAXVAL(XDIF100)
      IF (xmaxdif .LT. tol) THEN
        maxk = k
        GOTO 99
      ENDIF
      xmaxdif = 0.0
      IF (k .LT. maxiter) GOTO 66
      PRINT *, 'NOT CONVERGENT YET AFTER ITERATIONS:',maxiter
      RETURN
   99 CONTINUE
      b = x(:,maxk)
      RETURN
      END
```

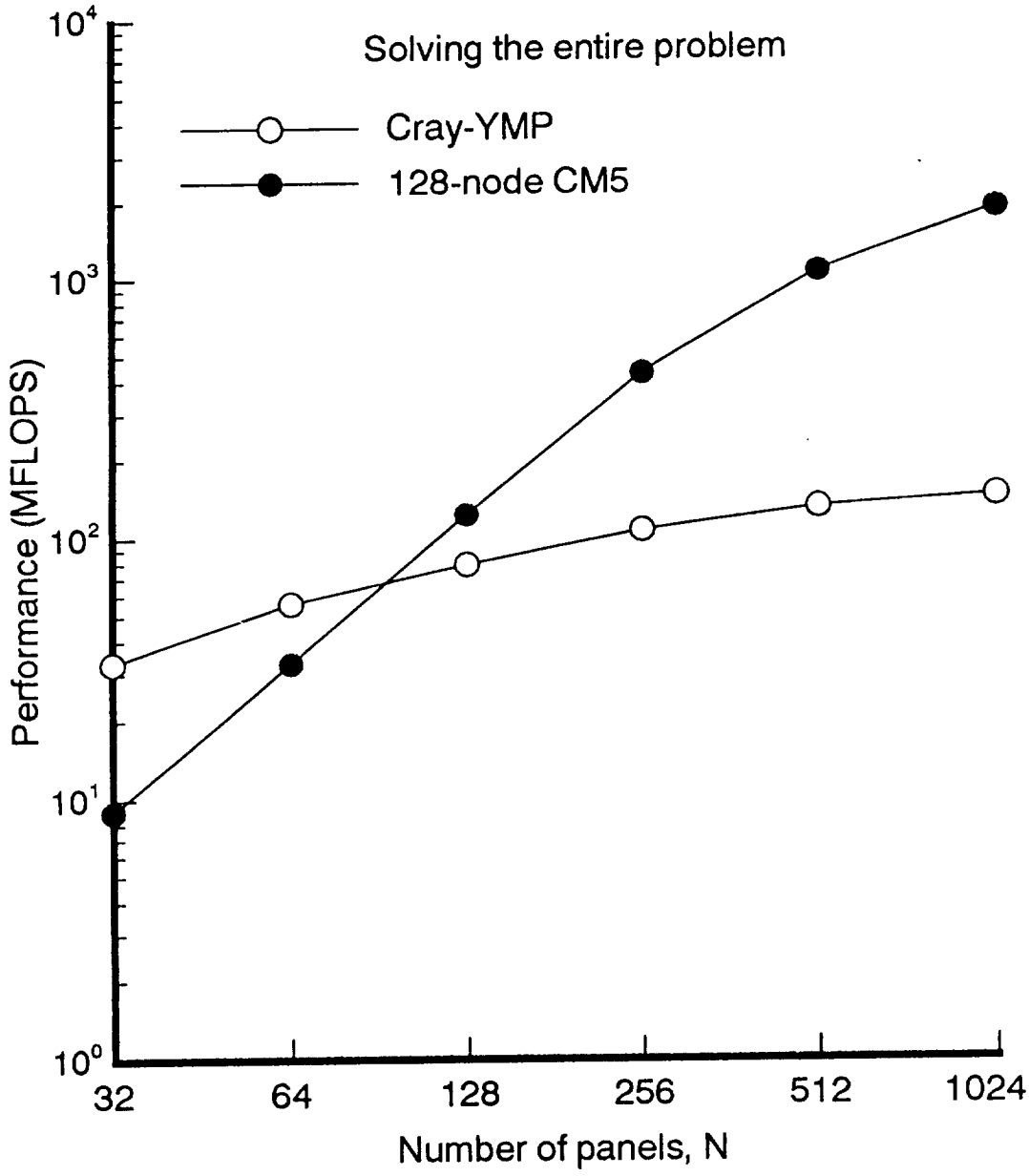
Table 1. The detailed computational performance results

Task/Size(N)		Cray-YMP		32-node CM5		64-node CM5		128-node CM5	
Task	N	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS
Mat Coef	32	0.0015	144	0.008	27	0.008	27	0.008	27
Lin Syst	32	0.0004	31	0.012	1	0.011	1	0.011	1
Total	32	0.0070	33	0.027	9	0.027	9	0.027	9
Mat Coef	64	0.0051	169	0.009	96	0.008	108	0.008	108
Lin Syst	64	0.0012	34	0.014	3	0.012	3	0.012	3
Total	64	0.0166	56	0.031	30	0.028	33	0.028	33
Mat Coef	128	0.0196	177	0.011	315	0.009	385	0.009	385
Lin Syst	128	0.0040	33	0.018	7	0.015	9	0.012	11
Total	128	0.0470	79	0.038	98	0.032	116	0.030	124
Mat Coef	256	0.0773	181	0.022	636	0.015	932	0.011	1272
Lin Syst	256	0.0097	31	0.020	15	0.015	20	0.012	25
Total	256	0.1370	107	0.054	271	0.040	366	0.034	431
Mat Coef	512	0.3050	183	0.061	915	0.035	1595	0.022	2537
Lin Syst	512	0.0368	31	0.050	23	0.031	37	0.018	63
Total	512	0.4450	131	0.143	408	0.087	670	0.055	1060
Mat Coef	1024	1.2200	184	0.216	1039	0.114	1969	0.061	3680
Lin Syst	1024	0.1420	31	0.106	42	0.060	73	0.036	122
Total	1024	1.6100	144	0.391	593	0.214	1083	0.124	1870


Calculating aerodynamic influence coefficients
Cray-YMP
32-node CM5
64-node CM5
128-node CM5
CPU time (sec.)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
32
64
128
256
512
1024
Number of panels, N


Solving the linear system (NxN) using Jacibi's method
Cray-YMP
32-node CM5
64-node CM5
128-node CM5
CPU time (sec.)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
32
64
128
256
512
1024
Number of panels, N


Solving the entire problem
Cray-YMP
32-node CM5
64-node CM5
128-node CM5
CPU time (sec.)
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
32
64
128
256
512
1024
Number of panels, N


Solving the entire problem
Cray-YMP
128-node CM5
Performance (MFLOPS)
10^4
10^3
10^2
10^1
10^0
32
64
128
256
512
1024
Number of panels, N

No. 4

STUDY OF INTEGRAL EQUATION METHODS FOR TRANSONIC FLOW CALCULATIONS

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

(Partially Resulted From This Grant)

Published in Journal
Engineering Analysis with Boundary Elements
Vol. 11, No. 2, 1993, pp. 101-107

Engineering Analysis with Boundary Elements **11** (1993) 101–107

Study of integral equation methods for transonic flow calculations

Hong Hu

Department of Mathematics, Hampton University, Hampton, Virginia 23668, USA

(Received 24 September 1991; accepted 17 December 1992)

An integral equation method based on the full-potential equation for transonic flow calculations is presented. The full-potential equation is written in the moving frame of reference, in the form of the Poisson's equation. The integral equation solution in terms of the velocity field is obtained by the Green's theorem. The numerical solutions are obtained by a time-marching (if unsteady flows), iterative procedure. The computational examples presented in the present paper include steady and unsteady, two-dimensional (airfoil) and three-dimensional (wing) flows. The method of combining the integral equation solution with the finite-volume Euler solution is also presented. Through studying the method and computational examples, the capabilities and limitations of the transonic integral equation method are discussed. Finally, the need for further research is addressed.

Key words: integral equation, field/boundary elements, full-potential equation, transonic flow.

INTRODUCTION

Starting in the 1970's a great deal of progress has been made in solving transonic flow using the finite-difference method (FDM) and finite-volume method (FVM). Although the FDM and FVM are successful in dealing with transonic flows, there are several drawbacks associated with these methods. In the FDM and FVM, fine grid points are needed over a large computational domain. Moreover, there are major technical difficulties in generating suitable grids for complex three-dimensional aerodynamic configurations.

On the other hand, the integral equation method (IEM, or called field-boundary element method, field-panel method) has several advantages over the FDM and FVM. The IEM involves the evaluation of integrals, which is more accurate and simpler than the FDM and FVM. The IEM automatically satisfies the far-field boundary conditions and hence only a small limited computational domain is needed. The IEM does not suffer from the artificial viscosity effects, as compared to FDM and FVM for shock capturing in transonic flow computations. Morevover, the generation of the three-dimensional grid (field-elements) for complex configuration is not difficult in the IEM, since the surface fitted grid is not required.

Because of these advantages, it is highly desirable to fully develop the IEM to treat transonic flows. Integral equation methods for transonic flows have been developed by several investigators.[1–8] The author and his co-workers have been devoted to the development of the IEM, for steady and unsteady transonic airfoil and wing flow computations, during the past several years.[9–15] In the present paper, the recent development, along with the computational examples, are presented. Through studying the method and the numerous computational examples, the capabilities and limitations of the transonic integral equation method are discussed. Finally, the needs for further research are addressed.

FORMULATION

Full-potential equation

In the space-fixed frame of reference, the continuity and momentum equations for unsteady, inviscid compressible flows with negligible body forces are given by

$$\frac{D\rho}{Dt} + \rho\nabla \cdot \mathbf{V} = 0 \tag{1}$$

Engineering Analysis with Boundary Elements 0955-7997/93/$05.00

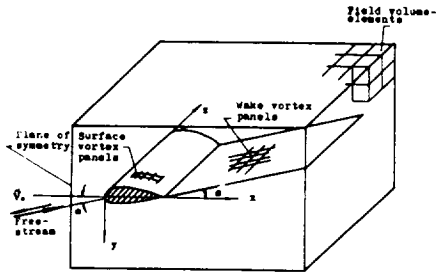


Fig. 2. IE computational domain.

$$\frac{1}{|\nabla w|}\frac{\partial' w}{\partial t} + \mathbf{V}_r \cdot \mathbf{n}_w = 0 \quad \text{on} \quad w(\mathbf{r}, t) = 0 \tag{16}$$

and

$$\nabla C_p = 0 \quad \text{on} \quad w(\mathbf{r}, t) = 0 \tag{17}$$

where $\mathbf{n}$ is the surface unit normal vector; the subscripts g and w refer to the body (wing or airfoil) and wake surface of $g(\mathbf{r}) = 0$ and $w(\mathbf{r}, t) = 0$, respectively; ΔC_p is the pressure jump across the surface; and the subscript sp refers to edge of separation.

Integral equation solution

By using the Green's theorem, the integral equation solution of eqn (8), in terms of the relative velocity field, is given by

$$\begin{aligned}\mathbf{V}_r(x,y,z,t) = &- \mathbf{V}_o(t) - \boldsymbol{\Omega}(t) \times (\mathbf{r} - \mathbf{r}_p) \\ &- \frac{1}{4\pi}\int\int_g \frac{q_g(\xi,\eta,\zeta,t)}{d^2}\mathbf{e}_d \, ds \\ &+ \frac{1}{4\pi}\int\int_g \frac{\gamma_g(\xi,\eta,\zeta,t) \times \mathbf{d}}{d^3} ds \\ &+ \frac{1}{4\pi}\sum_{iw=1}^{NW}\int\int_w \frac{\gamma_w(\xi,\eta,\zeta,t) \times \mathbf{d}}{d^3} ds \\ &- \frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta,t)}{d^2}\mathbf{e}_d \, d\xi \, d\eta \, d\zeta \\ &- \frac{1}{4\pi}\int\int_S \frac{qs(\xi,\eta,\zeta,t)}{d^2}\mathbf{e}_d \, ds\end{aligned} \tag{18}$$

where q is the surface source distribution; γ is the surface vorticity distribution; the subscript S refers to the shock surface; the index NW is the total number of wake surfaces; ds is the infinitesimal surface area; the vector $\mathbf{d}$ is given by $\mathbf{d} = (x - \xi)\mathbf{i} + (y - \eta)\mathbf{j} + (z - \zeta)\mathbf{k}$; and $\mathbf{e}_d$ is defined by $\mathbf{e}_d = \mathbf{d}/|\mathbf{d}|$.

It should also be noticed that eqn (18) has been written for three-dimensional flows. For two-dimensional flows, the above surface integrals become line integrals and the volume integrals become field surface integrals; the coordinates, z and ζ, are not used; and the coefficients of $1/4\pi$ are replaced by $1/2\pi$. The last integral term in eqn (18) is used only for the shock-fitting solutions in steady two-dimensional flows.

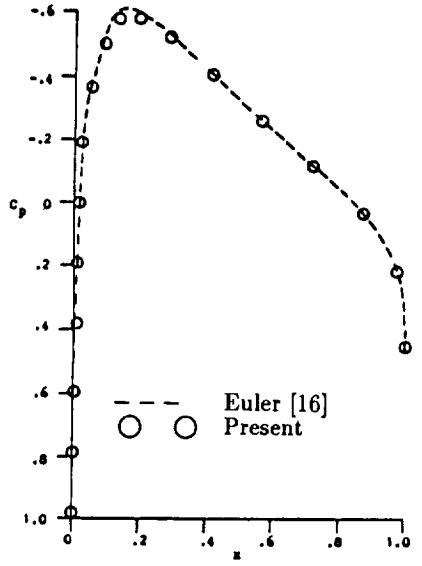


Fig. 3. C_p distribution, NACA0012, $M_\infty = 0{\cdot}72$, $\alpha = 0°$.

COMPUTATIONAL SCHEME

A sketch of the IE computational domains is shown in Fig. 2 for three-dimensional flows. due to the nature of the nonlinearity of the flow, the solutions are obtained through a time-marching (if unsteady flows), iterative procedure, where the compressibility (G_1), unsteadiness (G_2), and the wake shape and its strength are updated through each iteration. The details of the solution procedure can be found in Refs 9 and 14.

NUMERICAL EXAMPLES

The integral equation method has been applied to steady and unsteady, two-dimensional and three-dimensional transonic flows. The computational results, along with the experimental data and other computational results, are presented in the following sub-sections.

Steady subsonic airfoil flow

The computational results for a steady compressible shock-free flow at a high subsonic Mach number are presented here as the first numerical example. The purpose is to validate the IEM for nonlinear compressible flows. The surface pressure distribution[9,10] is shown in Fig. 3, along with the finite-difference (FD) Euler

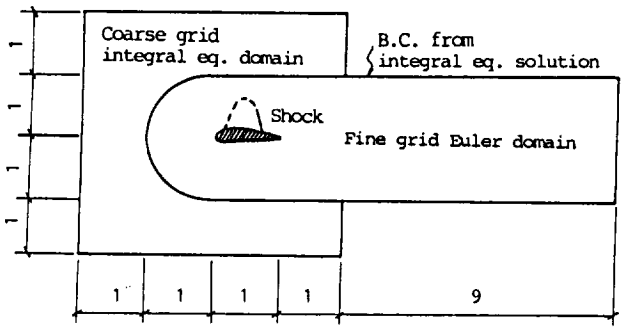


Fig. 7. Hybrid IE-FV computational domain for unsteady flows.

may be required to narrow the shock region, and to predict the shock motion as accurately as possible.

CAPABILITIES AND LIMITATIONS

The steady and unsteady integral equation methods for nonlinear compressible flows have been developed. The methods have been applied to steady airfoil, unsteady airfoil and unsteady wing flows, with or without shocks. The comparison of the present solutions with experimental data and FD or FV solutions shows that the integral equation methods, based on the linear theorem, can handle nonlinear flow problems accurately. For transonic flows with shocks of weak to moderate strength, IEM predicts shocks correctly, with the exception of slight underprediction of the shock strength (Fig. 4). For unsteady flows, the motion of the shock agrees with that predicted by FV Euler solutions (see Refs 9 and 11), and the predicted lifting coefficient agrees with one obtained by FV Euler computation (Fig. 5).

The advantages of the small computational domain and coarse grid have been utilised in the present IEM. For airfoil flow computations, a computational domain of $2 \times 1{\cdot}5$ airfoil chord length with 64×60 field-elements has been used. The application of a smaller number of field-elements, with larger field-elements over the outer region inside the domain, is possible. For wing flow computations, a computational domain of $2{\cdot}3 \times 0{\cdot}75 \times 1{\cdot}5$ wing root chord length with $23 \times 9 \times 9$ field-elements has been used, although finer field-elements around the shock region may be required to accurately predict the shock location and its strength.

The number of iterations used for steady flow computations are approximately 25 for airfoil and five for wing flows, respectively. This is much less than those used in FD and FV computations (usually of the order of 10^3). For unsteady flows, the number of iterations used in each time step range from one to three. Large time steps have also been used in the present unsteady flow computations. This is also one of the advantages of IEM. For a whole cycle of pitching oscillation, for example, a total

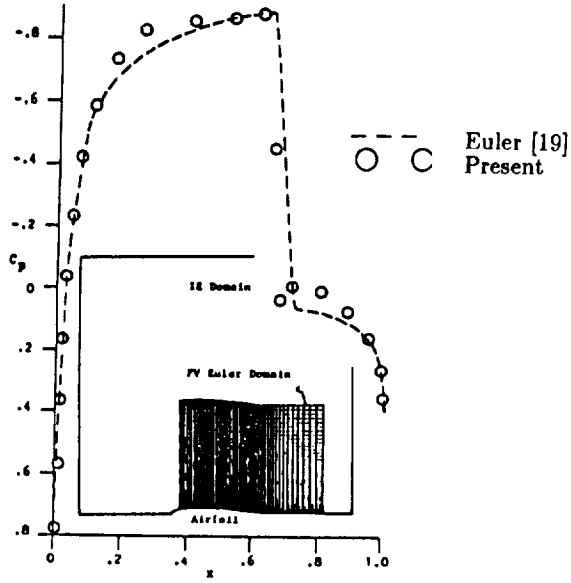


Fig. 8. Euler domain and C_p distribution, a strong shock case, steady flow, $M_\infty = 0{\cdot}84$, $\alpha = 0$.

of 36 time steps have been used; while a typical implicit FD or FV computation needs about 500 time steps for the same case. Therefore, IEM is nevertheless efficient in terms of the number of interations and the time step size, as compared to existing FDM and FVM.

By examining the numerical examples presented here it is found that, on the other hand, all these computations are restricted to the flows with shocks of weak to moderate strength. As the best of the author's knowledge, all existing integral equation solutions, based on potential flow formulation for transonic flows, are restricted to flows without strong shocks. The potential flow assumption neglects the effects due to viscosity, vorticity and entropy production. For transonic flows with strong shocks and massive separation, the potential flow assumption is not an adequate approximation to the real flow.

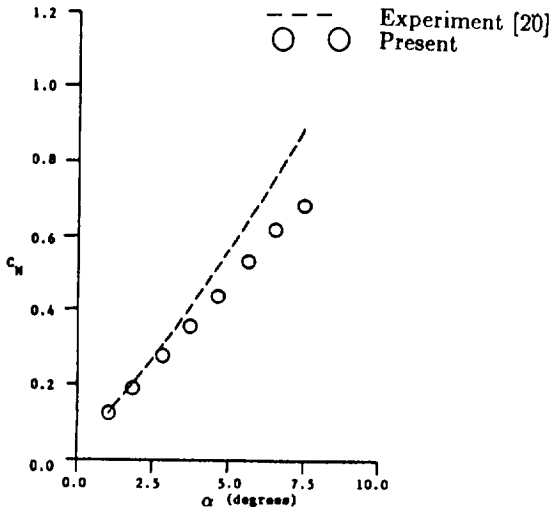


Fig. 9. Lifting coefficients, ramp motion, $M_0 = 0{\cdot}56$, $\alpha(t) = -0{\cdot}01^\circ + 0{\cdot}855^\circ t$.

12. Hu, H. & Chu, L-C. Unsteady three-dimensional transonic flow computations using field element method, *Boundary Element Methods in Engineering*, ed. B.S. Annigeri & K. Tseng, Springer-Verlag, 1990, pp. 140–146.
13. Hu, H. & Kandil, O.A. Unsteady field element method with embedded Euler domain for transonic pitching airfoil flows, *Boundary Element Methods in Engineering*, ed. B.S. Annigeri & K. Tseng, Springer-Verlag, 1990, pp. 161–168.
14. Hu, H. Unsteady transonic wing flow computations using field-boundary element methods, *Engineering analysis with Boundary Elements*, 1992, **10** (2), 99–105.
15. Hu, H. Application of integral equation method to flows around a wing with circular-arch section, *Boundary Elemements XIII*, ed. C.A. Brebbia & G.S. Gipson, Computational Mechanics Publications, Elsevier Applied Science, 1991, pp. 209–217.
16. Sells, C.L. Plane subcritical flow past a lifting airfoil, *Proceedings of the Royal Society of London*, Series A, Vol. 308, No. 1494, 1969, pp. 377–401.
17. Edwards, J.W., Bland, S.R. & Seidel, D.A. Experience with transonic unsteady aerodynamic calculations, NASA TM-86278, 1984.
18. Kandil, O.A. & Chuang, H.A. Unsteady transonic airfoil computation using implicit Euler scheme on body-fixed grid, *AIAA Journal*, 1989, **27** (8), 1031–1037.
19. Jameson, A., Schmidt, W. & Turkel, E. Numerical solutions of the Euler equations by finite-volume methods using Runge Kutta time-stepping scheme, AIAA Paper 81-1259, 1981.
20. London, R.H. NACA0012 oscillatory and transient pitching, AGARD R-702, 1982, pp. 3.1–3.25.

No. 5

PANEL METHOD COMPUTATIONAL PERFORMANCE ON CM-2 AND iPSC/860 MPP MACHINES

Hong Hu and Min Soe
(Mr. Soe is a graduate student)
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Elements XV
August 1993

FIFTEENTH INTERNATIONAL CONFERENCE
ON
BOUNDARY ELEMENT METHODS
BEM XV

Conference Chairmen

C.A. Brebbia and J.J. Rencis

Conference Sponsors

National Science Foundation (NSF)
Mechanical Engineering Department at Worcester Polytechnic Institute (WPI)
Worcester Section of American Society of Mechanical Engineers (ASME)
International Society for Boundary Elements (ISBE)

Local Organizing Committee

R.D. Ciskowski (IBM)
J.J. Connor (MIT)
S. Grilli (URI)
Q-P. Huang (ABAQUS)
B.V. Kiefer (Morgan Construction)
M.N. Noori (WPI)
T.J. Urekew (WPI)
A. Wanderlingh (Hamilton Standard)

International Scientific Advisory Committee

M.H. Aliabadi (UK)
D.J. Cartwright (USA)
R.S. Cheng (USA)
Q. Du (China)
C.T. Dyka (USA)
G.S. Gipson (USA)
L.J. Gray (USA)
M. Guiggiani (Italy)
W.S. Hall (UK)
G.C. Hsiao (USA)
M.S. Ingber (USA)
M.A. Jaswon (UK)
N. Kamiya (Japan)
S. Kobayashi (Japan)
F-B. Lin (USA)
A.K. Mitra (USA)
R.L. Mullen (USA)
D. Ouazar (Morocco)
P. Parreira (Portugal)
T.J. Rudolphi (USA)
R. Shaw (USA)
E.P. Stephan (Germany)
M. Tanaka (Japan)
M. Vable (USA)
L.C. Wrobel (UK)
N.G. Zamani (Canada)
J.Q. Zhen (USA)

Acknowledgement is made to A-M. Giroux and B. Massé for the use of Figure 5(a) on page 13, which appears on the front cover of this book.

Boundary Elements XV

Vol. 1: Fluid Flow and Computational Aspects

Editors:

C.A. Brebbia
Wessex Institute of Technology, Southampton, UK
J.J. Rencis
Worcester Polytechnic Institute, Massachusetts, USA

Computational Mechanics Publications
Southampton Boston

Co-published with

Elsevier Science Publishers
London New York

Panel method computational performance on CM-2 and iPSC/860 MPP machines

H. Hu, M. Soe

Department of Mathematics, Hampton University, Hampton, Virginia 23668, USA

ABSTRACT

A comparative study of computational performance of CM-2, iPSC/860 and Cray-YMP computers for calculating matrix coefficients (aerodynamic influence coefficients) using a two-dimensional source panel method (boundary elements) code has been made. Detailed performance results are obtained for CM-2 and iPSC/860 with different numbers of processors / nodes and for Cray-YMP with a single processor. The comparison of the performance indicates that the aerodynamic influence coefficient calculations on CM-2 and iPSC/860 with a large number of processors / nodes near- or outperforme the equivalent Cray-YMP code and achieve a high performance. Since the aerodynamic influence coefficient calcultaions take over 80% of the total computational time on Cray-YMP when the problem is large enough, it is very important to study the computational performance of this part of calculation.

Key Words: computer performance, CM-2, iPSC/860, Cray-YMP, panel methods, boundary elements, aerodynamics.

INTRODUCTION

In recent years, the processors of conventional vector supercomputers, such as Cray-YMP, seem to be approaching the limit in computational speed inherent in their technology. However, the need for even faster computations continueus to grow. As a consequence, parallel computers are being developed as a possible solution. Massively parallel computers, such as CM-2 and iPSC/860, are families of parallel computer architectures which may provide orders of magnitude improvement in computation performance in a near future over today's fastest supercomputer. In fact the CM-5 computer with a maximum 16k nodes installed has already been a 2 TFLOPS computer in theory.

Computational fluid dynamics (CFD) is one of the areas which need super- fast computation power. The massively parallel computers has

potential to become the main computational tool for CFD to replace the conventional supercomputers in a near future. This paper presents the performance comparison of the conventional supercomputer, Cray-YMP, and the massively parallel computer, CM-2 and iPSC/860, for calculating aerodynamic influence coefficients using a panel method fluid dynamics computational code.

CM-2 AND CM-FORTRAN

The Connection Machine CM-2 system is an integrated combination of hardware and software designed for high speed, large problem parallel computation. The major hardware elements of the system include front-end computers to provide developing and execution environments and a SIMD (Single Instruction Multiple Data) parallel processing unit to execute the data parallel operations. The SIMD parallel processing unit is the heart of CM-2 system, which contains up to 65536 single-bit physical processors (64k) in blocks of 8k (k=1024). The CM-2 used in the present work has 32k processors installed. The 32k single-bit processors on this 32k-CM2 are grouped in 1024 nodes of 32 processors each. Each node also has 64-bit Weitek floating point co- processors, 4MB of local memory, and hardware for interprocessor communication. Parallel data structures are spread across the data processors, with a single element stored in each processor's memory. If the number of the parallel data elements exceeds the total number of physical processors, the connection machine creates virtual processors by dividing the memory of each physical processor. The ratio of virtual to physical processors is known as the VP-ratio, R_{vp}. In general, floating-point performance is maximum when R_{vp} is as large as possible, since the communication overhead is reduced as R_{vp} increases.

The CM-FORTRAN language is an implementation of FORTRAN77 supplemented with array-processing extensions from the ANSI draft and ISO standard FORTRAN 90. These array-processing features map naturally onto the data parallel architecture of the CM-2 system, since the CM- FORTRAN allows array elements to be evaluated simultaneously. The most important difference of CM-FORTRAN from FORTRAN 77 is the treatment of entire arrays as objects, thus explicit indexing in CM-FORTRAN is not always necessary. For example, it is not necessary to write Do-Loops or other such control constructs to have the operation repeated for each element of arrays. This feature maps the problem directly to the CM-2 with minimal programming effect.

iPSC/860 AND INTEL-FORTRAN

Like the CM-2 system, the iPSC/860 is another integrated combination of hardware and software designed for high speed, large problem parallel computation. The major hardware elements of the system include front-end computers and a parallel processing unit. The front-end computer called the "host" is used to communicate with the nodes in the parallel processing unit. The parallel processing unit called the "cube" consists of 128 i860 computer nodes with 8MB of memory per node for a total of one GB of memory. Each node in the cube are fully connected through a

special hardware called the Direct-Connect- Module.

Unlike the CM-2 system however, the iPSC/860 is a MIMD (Multiple Instruction Multiple Data) distributed memory computer and therefore the nodes need a method to communicate. Intel provides massage passing routines to let the nodes to communicate. The Intel-FORTRAN is the FORTRAN 77 along with these message passing routines plus other routines for determining node information, performing global operations and reading / writing to concurrent file system.

CFD - PANEL METHODS

The physical problems considered here are potential flows around any arbitrary complex configurations (let us call it 'body'), including incompressible and compressible flows with and without separations. The governing equation of this type of problem can be written in the form of Poisson's equation given by

$$\nabla^2\Phi = G \tag{1}$$

where G represents full linear or non-linear compressibility and is a function of Φ in general. This type of problems can be solved by panel (boundary element) methods. The method is based on Green's theorem, which represents the solution in terms of integrals over body surfaces, separated surfaces and volume around the body. The surfaces and volume are then divided into a large number of panels, where the integrals are evaluated.

The solution in terms of velocity field ($\vec{V} = \nabla\Phi$) is given by[1,2]

$$\begin{aligned}
\vec{V}(x,y,z) &= \vec{V}_\infty \\
&- \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d\, ds(\xi,\eta,\zeta) \\
&+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{\vec{\gamma}_g(\xi,\eta,\zeta)\times\vec{d}}{d^3} ds(\xi,\eta,\zeta) \\
&+ \frac{1}{4\pi}\sum_{ms=1}^{MS}\sum_{i=1}^{LS}\sum_{k=1}^{NS}\int\int_{w_{ms,i,k}} \frac{\vec{\gamma}_w(\xi,\eta,\zeta)\times\vec{d}}{d^3} ds(\xi,\eta,\zeta) \\
&+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV}\int\int\int_{V_{i,j,k}} \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d\, d\xi d\eta d\zeta
\end{aligned} \tag{2}$$

where q and $\vec{\gamma}$ is the surface source and vorticity distribution, respectively, which are unknowns to be determined by applying boundary conditions; the subscripts g and w refer to the body and separated surfaces, respectively, where MS is the total number of separations; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

It should be noted that Eq. (2) involves evaluating a large number of integrals (in fact they are vectors also !) over body surface (total of $2 \times LG \times NG$), separated surface (total of $MS \times LS \times NS$) and volume (total of $LV \times MV \times NV$). The total number of elements can be very large for aerodynamics problems, and for aircraft configurations it can be, for example, in the order of 10^4. An important feature of Eq. (2) is that the calculation for each (x, y, z) and each (ξ, η, ζ) can be performed simultaneously for all (x, y, z) and all (ξ, η, ζ). This feature of panel method calculation leads itself in a natural way for processing data in a parallel computing environment.

For simplicity in the present work, an incompressible flow ($G = 0$) past a two-dimensional symmetric configuration at zero incidence is considered. At this simple flow condition Eq. (2) reduces to a much simpler form, or

$$\vec{V}(x,y) = \vec{V}_\infty - \frac{1}{2\pi}\sum_{i=1}^{N}\int_{s_i} q_s(\xi,\eta)\frac{(x-\xi)\vec{i}+(y-\eta)\vec{j}}{(x-\xi)^2+(y-\eta)^2}ds(\xi,\eta) \tag{3}$$

By applying the body surface zero-normal-velocity condition at each $(x, y) = (x_j, y_j)$ for $j = 1$ to N (here $N \equiv LG$), a $N \times N$ system of equations is obtained as

$$[A][q] = [B] \tag{4}$$

where $[A]$ is $N \times N$ aerodynamic influence coefficient matrix; $[q]$ is a $N \times 1$ unknown vector matrix containing q_j for $j = 1$ to N; and $[B]$ is a $N \times 1$ known vector matrix which is contributed from $\vec{V}_\infty$ in this two-dimensional flow case.

The solution procedure of above two-dimensional problem using panel method involves four steps: (1) generation of body geometry information; (2) evaluation of integrals of Eq. (3) for $i = 1$ to N and for $(x, y) = (x_j, y_j)$ with $j = 1$ to N to construct $[A]$ matrix (and also $[B]$ matrix); (3) solution of resulting dense linear system given by Eq. (4); and (4) post-processing, aerodynamic calculations. The most important difference between the present simple flow governed by Eq. (3) and general complex flows governed by Eq. (2) is that, $[A]$ and $[B]$ matrix calculations are much more expensive in general three-dimensional computations than in the present simple flow computations. The experiences have shown that the $[A]$ and $[B]$ matrix calculations for large problems usually take over 80 to 90% of total computational time. Figure 1 gives the percentage of CPU time for calculating $[A]$ and $[B]$ matrices over the total CPU time for solving the entire problem. The results were obtained using PERFTRACE utility on Cray-YMP where the system of equations were solved using approximate Jacobi iterations. It is seen that when the problem is large enough this percentage is over 80% even for this 2-D flow.

PARALLEL FORTRAN IMPLEMENTATION

Serial FORTRAN codes of two-dimensional source panel methods described by Eqs. (3) and (4) are available in some references, such as in Ref. [3]. The serial FORTRAN code of Ref. [3] is modified for efficient executing on Cray-YMP8/5128 supercomputer. The code consists of one main routine and four subroutines to perform the four steps mentioned above. Part of the FORTRAN 77 code is then converted to parallel CM-FORTRAN code[4] and Intel-FORTRAN code for executing on CM-2 and iPSC/860 computers, respectively, for calculating $[A]$ and $[B]$ matrices.

Although the panel method solution procedure consists of four steps discussed earlier, the Step 2 for evaluating integrals to construct the matrics takes most part of computational time as shown in the figure 1 eralier. Therefore, the present interest on the performance analysis is focused on the computation of Step 2, although the entire code is being converted into CM-FORTRAN and Intel-FORTEAN to solve entire problem.

List 1. lists part of the program for calculating matrices. List 1(a). is in the serial FORTRAN version. It is noted that this listing is nothing but a two-level Do-Loop, which provides for evaluating each element of $[A]$ and $[B]$. When the code is in execution on Cray-YMP, the vectorization of the inner Do-Loop is automatically done through the vectorization capability of the FORTRAN 77 compiler. List 1(b). is in the parallel CM-FORTRAN version. No Do-Loop is seen here since in CM-FORTRAN entire arrays are treated as objects and array elements are evaluated simultaneously. List 1(c). is in Intel- FORTRAN version, where the routine "MYNODE()" gives the node number and the routine "NUMNODES()" gives the number of nodes in the cube. It is seen that each partion of the outer loop is calculated by one node simultaneously.

PERFORMANCE ANALYSIS

The serial code with different numbers of panels (N) is executed on Cray-YMP supercomputer using single processor to provide the basis for performance comparison. The Cray-YMP used here has a total of 5 processors (CPUS) with 128 MW SRAM central memory. Each CPU is a register-to-register vector processor with peak performance at 150-300MFLOPS. The computational performance in terms of MFLOPS is obtained using Cray-YMP's PERFTRACE utility. The parallel CM-FORTRAN code is executed on CM-2 with 8k, 16k and 32k processors under slicewise model, while the parallel Intel-FORTRAN code is executed on iPSC/860 with 1, 4, 8, 16, 32, 64 and 128 nodes.

Table 1 gives the detailed performance results on Cray-YMP, CM-2 and iPSC/860 computers with varying size of the problem. Selected sets of data from the table are presented in Figs. 2-4.

Figure 2 shows execution time for evaluating aerodynamic influence coefficient matrix $[A]$, and the matrix $[B]$ obtained on Cray-YMP and CM-2 computers for different numbers of panels. The CM-2 performance results at each number of panels, as shown in Table 1, are represented by solid points which is connected by dashed lines since these lines do not represent the actual variation of execution time between each point. It can be seen that the CPU execution time decreases with the increase

of the number of CM-2 processors after the size of the problem is large enough to fully use all processors. For example when $N = 512$, the CPU time of 0.205 seconds with 8k processors is reduced to 0.115 seconds with 16k processors, and then is further reduced to 0.0685 seconds with 32k processors. That is to say that whenever the number of processors used is doubled, the CPU time is reduced by a factor of 2 — a near-perfect parallel computation. It is also seen that when the problem size is large enough the CPU time required on CM-2, even with 8k processors, is significantly (note that Log_{10}-axis is used for execution time !) less than that required on Cray-YMP. When $N = 512$, the CPU time required on CM-2 with 32k processors is about 1/5 of that required on Cray-YMP. This is a very encouraging result.

Figure 3 is similar to Fig.2 but for the comparison of Cray with iPSC/860 performance. The figure shows that the iPSC/860 with 128 nodes outperforms Cray-YMP. It is also seen that the execution time is reduced by a factor of 4 each time when the number of nodes increased from 8 to 32 and then to 128 — another near-perfect parallel computation. For all the numbers of panels shown in the figure, the iPSC/860 with 128 nodes constantly outperforms Cray- YMP by a factor of 2. This is also very encouraging.

Figure 4 is a partial reproduction of Figs. 2-3 for performance results on Cray-YMP, CM-2 with 32k processors and iPSC/860 with 128 nodes, and it is represented in terms of MFLOPS. The performance of CM-2 and iPSC in terms of MFLOPS is the equivalent Cray-YMP performance. The results presented are self-explanatory.

CONCLUDING REMARKS

The study of the performance of CM-2 and iPSC/860 MPP computers with the comparison of Cray-YMP computer for calculating panel method aerodynamic influence coefficients is made. The high performance results obtained on CM-2 and iPSC/860 are very encouraging. Currently the performance analysis for entire computer code and for more complex three-diemnsional problems is being made.


```
      DO 2 K=1,N
      DO 1 J=1,N
      IF (K .EQ. J) FN(K,J)=2.0*PI
      IF (K. EQ. J) FT(K,J)=0.0
      IF (K. EQ. J) GOTO 1
      DYJ=SI(J)*DS(J)
      DXJ=CI(J)*DS(J)
      SPH=DS(J)*0.5
      XD=XC(K)-XC(J)
      YD=YC(K)-YC(J)
      RKJ=SQRT(XD*XD+YD*YD)
      BKJ=ATAN2(YD,XD)
      ALJ=ATAN2(DYJ,DXJ)
      GKJ=ALJ-BKJ
      ZIK=RKJ*COS(GKJ)
      ETK=-RKJ*SIN(GKJ)
      R1S=((ZIK+SPH)**2)+ETK*ETK
      R2S=((ZIK-SPH)**2)+ETK*ETK
      QT=ALOG(R1S/R2S)
      DEN=ZIK*ZIK+ETK*ETK-SPH*SPH
      GNM=ETK*DS(J)
      QN=2.0*ATAN2(GNM,DEN)
      UKJ=QT*CI(J)-QN*SI(J)
      VKJ=QT*SI(J)+QN*CI(J)
      FN(K,J)=-UKJ*SI(K)+VKJ*CI(K)
      FT(K,J)=UKJ*CI(K)+VKJ*SI(K)
    1 CONTINUE
      RHS(K)=UINF*SI(K)-VINF*CI(K)
    2 CONTINUE
```

1(a). Serial FORTRAN version.

```
XC2 = SPREAD(XC,DIM=1,NCOPIES=N)
YC2 = SPREAD(YC,DIM=1,NCOPIES=N)
XC3 = SPREAD(XC,DIM=2,NCOPIES=N)
YC3 = SPREAD(YC,DIM=2,NCOPIES=N)
SI2=SPREAD(SI,DIM=1,NCOPIES=N)
CI2=SPREAD(CI,DIM=1,NCOPIES=N)
SI3=SPREAD(SI,DIM=2,NCOPIES=N)
CI3=SPREAD(CI,DIM=2,NCOPIES=N)
DS2=SPREAD(DS,DIM=1,NCOPIES=N)
MAIN_DIAG=DIAGONAL(SPREAD(.TRUE.,1,N),.FALSE.)
WHERE(MAIN_DIAG)
  FN = 2.0 * PI
  FT = 0.0
ELSEWHERE
  DYJ = SI2 * DS2
  DXJ = CI2 * DS2
  SPH = DS2 * 0.5
  XD = XC3 - XC2
  YD = YC3 - YC2
  RKJ=SQRT(XD*XD +YD*YD)
  BKJ=ATAN2(YD,XD)
  ALJ=ATAN2(DYJ,DXJ)
  GKJ=ALJ-BKJ
  ZIK=RKJ*COS(GKJ)
  ETK=-RKJ*SIN(GKJ)
  R1S=((ZIK+SPH)**2) + ETK*ETK
  R2S=((ZIK-SPH)**2) + ETK*ETK
  QT=ALOG(R1S/R2S)
  DEN=ZIK*ZIK + ETK*ETK - SPH*SPH
  GNM=ETK*DS2
  QN=2.0*ATAN2(GNM,DEN)
  UKJ=QT*CI2-QN*SI2
  VKJ=QT*SI2+QN*CI2
  FN=-UKJ*SI3+VKJ*CI3
  FT=UKJ*CI3+VKJ*SI3
ENDWHERE
RHS=UINF*SI-VINF*CI
```

1(b). Parallel CM-FORTRAN version.


```
ME=MYNODE()
NN=NUMNODES()
.
.
.
DO 2 K=ME+1,N,NN
DO 1 J=1,N
IF (K .EQ. J) FN(K,J)=2.0*PI
IF (K. EQ. J) FT(K,J)=0.0
IF (K. EQ. J) GOTO 1
DYJ=SI(J)*DS(J)
DXJ=CI(J)*DS(J)
SPH=DS(J)*0.5
XD=XC(K)-XC(J)
YD=YC(K)-YC(J)
RKJ=SQRT(XD*XD+YD*YD)
BKJ=ATAN2(YD,XD)
ALJ=ATAN2(DYJ,DXJ)
GKJ=ALJ-BKJ
ZIK=RKJ*COS(GKJ)
ETK=-RKJ*SIN(GKJ)
R1S=((ZIK+SPH)**2)+ETK*ETK
R2S=((ZIK-SPH)**2)+ETK*ETK
QT=ALOG(R1S/R2S)
DEN=ZIK*ZIK+ETK*ETK-SPH*SPH
GNM=ETK*DS(J)
QN=2.0*ATAN2(GNM,DEN)
UKJ=QT*CI(J)-QN*SI(J)
VKJ=QT*SI(J)+QN*CI(J)
FN(K,J)=-UKJ*SI(K)+VKJ*CI(K)
FT(K,J)=UKJ*CI(K)+VKJ*SI(K)
1 CONTINUE
RHS(K)=UINF*SI(K)-VINF*CI(K)
2 CONTINUE
```

1(c). Parallel Intel-FORTRAN version.

List 1. Program listing for matrix calculation.

Size, N=	32	64	128	256	512
Cray-YMP	0.00148	0.00510	0.0106	0.0773	0.305
8k-CM2	0.0173	0.0194	0.0310	0.0680	0.205
16k-CM2	0.0173	0.0194	0.0224	0.0447	0.115
32k-CM2	0.0173	0.0186	0.0203	0.0323	0.0685
1-node iPSC	0.084	0.328	1.288	5.396	21.884
2-node iPSC	0.044	0.164	0.656	2.700	10.948
4-node iPSC	0.024	0.084	0.328	1.352	5.484
8-node iPSC	0.012	0.040	0.166	0.674	2.740
16-node iPSC	0.006	0.020	0.084	0.336	1.364
32-node iPSC	0.004	0.010	0.044	0.168	0.684
64-node iPSC	—	0.006	0.022	0.084	0.342
128-node iPSC	—	—	0.012	0.044	0.172

Table 1. The CPU time in seconds for matrix coefficients.

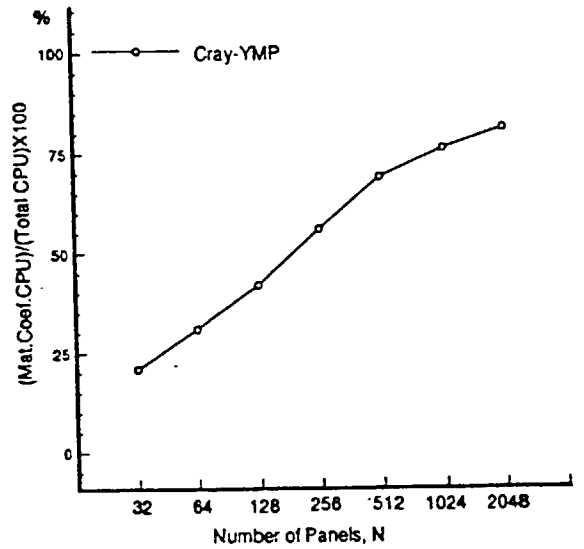


Figure 1. The percentage of CPU time for matrix calculation.

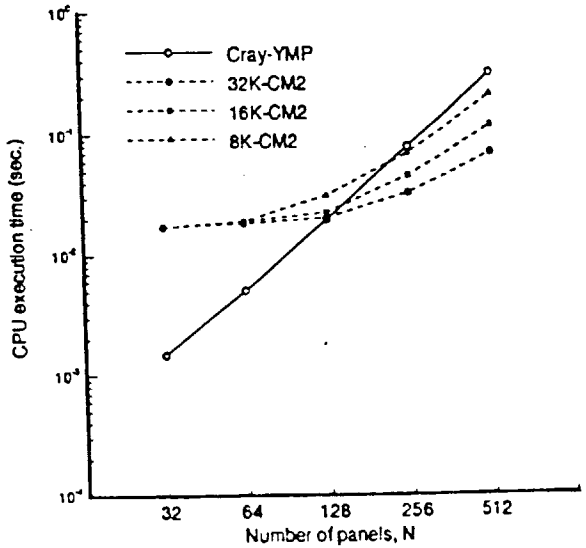


Figure 2. Performance comparison for matrix calculation on CM-2.

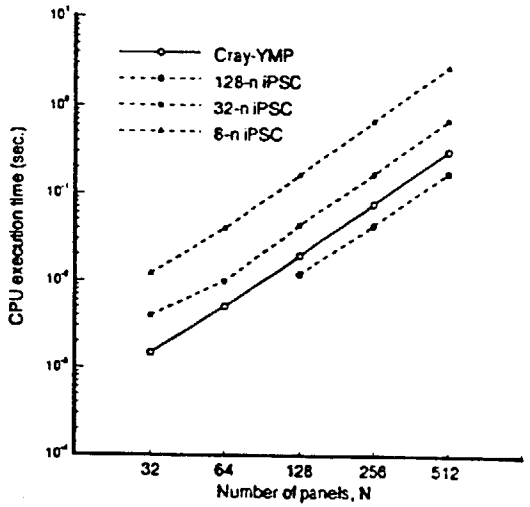


Figure 3. Performance comparison for matrix calculation on iPSC/860.

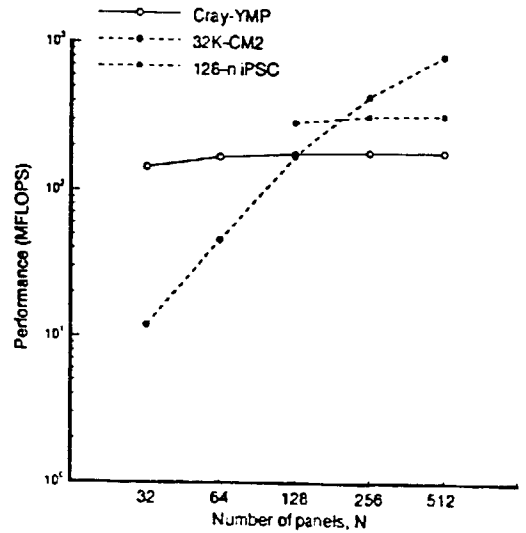


Figure 4. Computational speeds in terms of MFLOPS.

ACKNOWLEDGEMENTS

This work has been supported by NASA LaRC under the Grant No. NAG-1-1170 with Mr. Walter Silva as the technical monitor. The CM-2 and iPSC/860 computation is supported by NASA-ARC under NAS Program. Mr. Jason Bryan, a student of Hampton University supported partially by ONR's SEMS program and partially by the NASA grant mentioned, has been involved and contributed in discussion of this work.

REFERENCES

1 Hu, H. Experience with transonic flow IE computations. *Boundary Elements in Fluid Dynamics*, (eds.: Brebbia, C. A. and Partridge. P. W.), Computational Mechanics Publications and Elsevier Applied Science, 1992, pp. 35-48.

2 Hu, H. A 3D IEM for compressible wing flows with and without shocks. *Boundary Elements in Fluid Dynamics*, (eds.: Brebbia, C. A. and Partridge. P. W.), Computational Mechanics Publications and Elsevier Applied Science, 1992, pp. 49-60.

3 Fletcher, C. A. J. Computational techniques for fluid dynamics. Springer-Verlag Berlin Heidelberg, 1988, chap.14.

4 Hu, H. and Jackson, I. T. Comparative study of computational performance of CM-2 and Cray-YMP for boundary element computations, *Boundary Elements Abstracts Journal and Newsletter*, Vol. 3, No. 5, 1992, pp.185-191.

No. 6

CM-2 PERFORMANCE EVALUATION ON PANEL METHOD CALCULATIONS

Hong Hu and Isaac T. Jackson
(Mr. Jackson is an undergraduate in computer science)
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Proceedings of the 6th SIAM Conference on
Parallel Processing for Scientific Computing
March 1993

Aggarwal, Alok, Borodin, Allan, Gabow, Harold, N., Galil, Zvi, Karp, Richard M., Kleitman, Daniel J., Odlyzko, Andrew M., Pulleyblank, William R., Tardos, Éva & Vishkin, Uzi, Proceedings of the Second Annual ACM-SIAM Symposium on Discrete Algorithms (1990)

Cohen, Gary, Halpern, Laurence & Joly, Patrick, Mathematical and Numerical Aspects of Wave Propagation Phenomena (1991)

Gómez, S., Hennart, J. P., & Tapia, R. A., Advances in Numerical Partial Differential Equations and Optimization: Proceedings of the Fifth Mexico-United States Workshop (1991)

Glowinski, Roland, Kuznetsov, Yuri A., Meurant, Gérard, Périaux, Jacques & Widlund, Olof B., Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations (1991)

Alavi, Y., Chung, F. R. K., Graham, R. L. & Hsu, D. F., Graph Theory, Combinatorics, Algorithms, and Applications (1991)

Wu, Julian J., Ting, T. C. T. & Barnett, David M., Modern Theory of Anisotropic Elasticity and Applications (1991)

Shearer, Michael, Viscous Profiles and Numerical Methods for Shock Waves (1991)

Griewank, Andreas & Corliss, George F., Automatic Differentiation of Algorithms: Theory, Implementation, and Application (1991)

Frederickson, Greg, Graham, Ron, Hochbaum, Dorit S., Johnson, Ellis, Kosaraju, S. Rao, Luby, Michael, Megiddo, Nimrod, Schieber, Baruch, Vaidya, Pravin, & Yao, Frances, Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms (1992)

Field, David A. & Komkov, Vadim, Theoretical Aspects of Industrial Design (1992)

Field, David A. & Komkov, Vadim, Geometric Aspects of Industrial Design (1992)

Bednar, J. Bee, Lines, L. R., Stolt, R. H. & Weglein, A. B., Geophysical Inversion (1992)

O'Malley, Robert E. Jr., Proceedings of the Second International Conference on Industrial and Applied Mathematics (1992)

Keyes, David E., Chan, Tony F., Meurant, Gérard, Scroggs, Jeffrey S., & Voigt, Robert G., Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations (1992)

Dongarra, Jack, Messina, Paul, Kennedy, Ken, Sorensen, Danny C., & Voigt, Robert G., Proceedings of the Fifth SIAM Conference on Parallel Processing for Scientific Computing (1992)

Cornes, James P., Kristensson, Gerhard, Nelson, Paul, & Seth, Daniel L., Invariant Imbedding and Inverse Problems

Sincovec, Richard F., Keyes, David E., Leuze, Michael R., Petzold, Linda R., & Reed, Daniel A., Proceedings of the Sixth SIAM Conference on Parallel Processing for Scientific Computing (1993)

PROCEEDINGS OF THE SIXTH SIAM CONFERENCE ON

PARALLEL PROCESSING FOR SCIENTIFIC COMPUTING

Volume II

March 22-24, 1993

Edited by Richard F. Sincovec
Oak Ridge National Laboratory
David E. Keyes
Yale University
Michael R. Leuze
Oak Ridge National Laboratory
Linda R. Petzold
University of Minnesota
Daniel A. Reed
University of Illinois

siam

Philadelphia

Society for Industrial and Applied Mathematics

[2] BBN Advanced Computer Inc., *Inside the TC2000*, 1989.
[3] E. D. Brooks III, G. A. Darmohray and T. S. Axelrod, *The Cerberus User Manual*, Technical Report, Lawrence Livermore National Laboratory, 1991.
[4] J. M. Mellor-Crummey and M. L. Scott , *Algorithms for scalable synchronization on shared-memory multiprocessors*, ACM Transactions on Computer Systems, 9, (1991), pp.21-65.
[5] J. P. Singh, W-D. Weber and A. Gupta, *SPLASH: Stanford Parallel Applications for Shared-Memory*, Technical Report, Department of Computer Science, Stanford University, 1991.
[6] X. Zhang and W. Wu, *On measuring and evaluating synchronization and virtual memory performance of a MIN-based multiprocessor*, in Proceedings of the 25th Hawaii International Conference of Systems Sciences, 1, IEEE Computer Society Press, 1992, pp. 218-224.

CM-2 Performance Evaluation on Panel Method Calculations*

Hong Hu† Isaac T. Jackson‡

Abstract

A comparative study of computational performance of CM-2 and Cray-YMP computers for a panel method code has been made. A serial FORTRAN 77 code running on Cray-YMP supercomputer has been converted into a parallel CM-FORTRAN code for running in CM-2 massively parallel computer. Detailed performance results are obtained. The comparison of the performance indicates that the influence coefficient calculations on CM-2 with 32k processors outperformed the equivalent Cray-YMP code by factor of 7 for 2048 panels and achieved a speed of 1.2 GFLOPS.

1 Introduction

In recent years, the processors of conventional vector supercomputers seem to be approaching the limit in computational speed inherent in their technology. However, the need for even faster computations continueus to grow. As a consequence, it has been started to develop parallel computers as a possible solution. Massively parallel computers, CM-2, developed by Thinking Machine Corporation are one family of parallel computer architectures which may provide orders of magnitude improvement in computation performance in a near future over today's fastest supercomputer, such as Cray-YMP. This paper presents the performance comparison of the conventional supercomputer, Cray-YMP, and the new massively parallel computer, CM-2, for panel method influence coefficient calculations.

2 CM-2 and CM-FORTRAN

The Connection Machine CM-2 system is an integrated combination of hardware and software designed for high speed, large problem parallel computation. The heart of CM-2 system is the SIMD (Single Instruction Multiple Data) parallel processing unit, which contains up to 65536 single-bit physical processors (64k) in blocks of 8k (k=1024). The CM-2 used in the present work at NASA-Ames Research Center has 32k processors with each

* This work has been supported by NASA-LaRC under the Grant NAG-1-1170 and the CM-2 computation is supported by NASA-ARC under NAS Program.
† Assistant Professor, Dept. of Mathematics, Hampton University, Hampton, VA.
‡ Student, Dept. of Computer Science, Hampton University, Hampton, VA; currently at Defense Information Systems Agency, Arlington, VA.

processor operating at a clock speed of 6.7 MHz. The aggregate peak performance for this 32k-CM2 is about 2 GFLOPS. Parallel data structures are spread across the data processors, with a single element stored in each processor's memory. If the number of the parallel data elements exceeds the total number of physical processors, the connection machine creates virtual processors by dividing the memory of each physical processor. The ratio of virtual to physical processors is known as the VP-ratio, R_{vp}. In general, floating-point performance (usually in terms of MFLOPS or GFLOPS) is maximum when R_{vp} is as large as possible, since the communication overhead is reduced as R_{vp} increases.

The CM-FORTRAN language is an implementation of FORTRAN77 supplemented with array-processing extensions from the ANSI draft and ISO standard FORTRAN 90. These array-processing features map naturally onto the data parallel architecture of the CM-2 system. The most important difference between CM-FORTRAN and FORTRAN 77 is the treatment of entire arrays as objects in CM-FORTRAN, thus explicit indexing in CM-FORTRAN is not always necessary.

3 Panel Method

The physical problems considered here are potential flows around any arbitrary complex configurations (such as airplane wing, but let us call it 'body'), including incompressible and compressible flows with and without separations. This type of problem can be governed by full-potential equation, which can be written in the form of Poisson's equation given by

$$\nabla^2\Phi = G \tag{1}$$

where G represents full linear or non-linear compressibility and is a function of Φ in general. This type of problems can be solved by panel methods (or called boundary element, integral equation methods).

The integral equation solution in terms of velocity field ($\vec{V} = \nabla\Phi$) is given by [1-3]

$$\begin{aligned}\vec{V}(x_j,y_j,z_j) &= \vec{V}_\infty \\ &- \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{S_{i,k}} \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{S_{i,k}} \frac{\vec{\gamma}_g(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{w=1}^{MS}\sum_{i=1}^{LS}\sum_{k=1}^{NS}\int\int_{w_{w,i,k}} \frac{\vec{\gamma}_w(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV}\int\int\int_{V_{i,j,k}} \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,d\xi\,d\eta\,d\zeta\end{aligned} \tag{2}$$

where q and $\vec{\gamma}$ is the surface source and vorticity distribution, respectively, which are unknowns to be determined by applying boundary conditions; the subscripts g and w refer to the body and separated surfaces, respectively, where MS is the total number of separations; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x_j-\xi)\vec{i} + (y_j-\eta)\vec{j} + (z_j-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

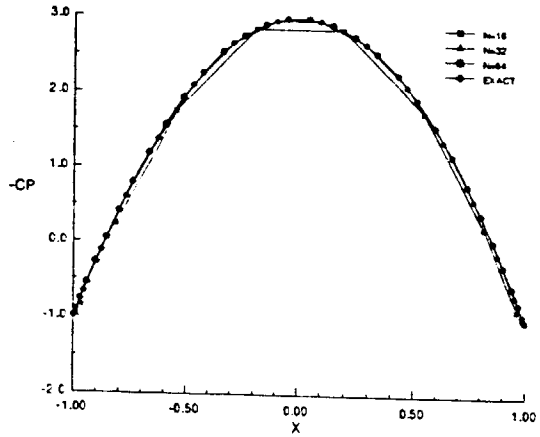


Fig.1. Surface pressure distribution.

It should be noted that Eq. (2) involves evaluating a large number of integrals (in fact they are vectors also !) over body surface (total of $2\times LG\times NG$), separated surface (total of $MS\times LS\times NS$) and volume (total of $LV\times MV\times NV$) for $j = 1$ to N. The total number of panels can be very large for aerodynamics problems, and for aircraft configurations it can be, for example, in the order of 10^6 or higher. The experience [1-3] has shown that evaluating these integrals usually takes above 80% of the total computational time. An important feature of Eq. (2) is that the calculation of these integrals for each (x_j,y_j,z_j) and each (ξ,η,ζ) can be performed simultaneously with a single instruction. This feature of panel method calculation leads itself in a natural way for processing data in a SIMD parallel computing environment. Therefore, it is very interested to study the performance of the calculation on CM-2 computer, a massively parallel SIMD computer.

For simplicity in the present work an incompressible flow ($G = 0$) past a 2-dimensional symmetric configuration at zero incidence is considered. At this simple flow condition, Eq. (2) reduces to a much simpler form, .

$$\begin{aligned}\vec{V}(x_j,y_j) &= \vec{V}_\infty \\ &- \frac{1}{2\pi}\sum_{i=1}^{N}\int_{g_i} q_g(\xi,\eta)\frac{(x_j-\xi)\vec{i}+(y_j-\eta)\vec{j}}{(x_j-\xi)^2+(y_j-\eta)^2}ds(\xi,\eta)\end{aligned} \tag{3}$$

By applying body surface zero normal-velocity condition at each (x_j,y_j) for $j = 1$ to

N, a $N \times N$ system of equations is obtained as

$$[A][q] = [B] \tag{4}$$

where $[A]$ is $N \times N$ influence coefficient matrix, with each element calculated by mainly evaluating the integral of Eq. (3) with a fixed value of i and j. In the next section, the performance for calculating matrices, $[A]$ and $[B]$, is presented.

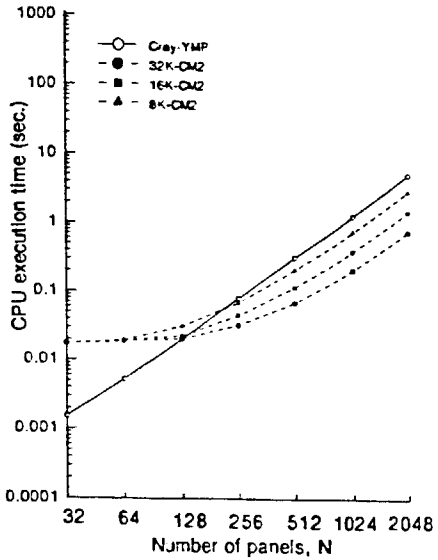


Fig.2. Performance comparison in terms of execution time.

4 Performance Evaluation

A serial FORTRAN 77 code running on Cray-YMP supercomputer at NASA-LaRC is converted into parallel CM-FORTRAN code for running on Connection Machine CM-2 at NASA-ARC. Both codes are executed with different number of panels using one CPU(processor) on Cray-YMP and 8k, 16k and 32k processors on CM-2. When the code is in execution on Cray-YMP, the vectorization of the inner loop is automatically done through the vectorization capability of the FORTRAN 77 compiler at NASA- LaRC. The detailed performance results are obtained.

Fig. 1 presents a calculated, on both Cray and CM-2 computers, surface pressure

distributions over a circular cylinder along with the comparison of the exact solution to show the accuracy of the calculation, although the purpose of this work is not on the method itself but on the performance of massively parallel computations.

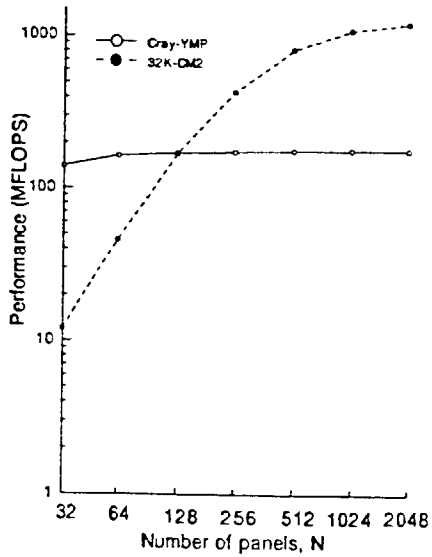


Fig.3. Performance comparison in terms of MFLOPS.

Fig.2 shows execution time for evaluating influence coefficient matrix, $[A]$, and the matrix $[B]$ obtained on Cray-YMP and CM-2 computers for different numbers of panels. It can be seen that the CPU execution time decreases with the increase of the number of CM-2 processors after the size of the problem is large enough to fully use all processors. For example when $N = 2048$, the CPU time of 2.84 seconds with 8k processors is reduced to 1.43 seconds with 16 k processors, and then is further reduced to 0.737 seconds with 32k processors. That is to say that whenever the number of processors used is doubled, the CPU time is reduced by a factor of 2. It is also seen that when the problem size is large enough the CPU time required on CM-2, even with 8k processors, is significantly (note that Log_{10}-axis is used for execution time !) less than that required on Cray-YMP. When $N = 2048$, the CPU time required on CM-2 with 32k processors is about 1/7 of that required on Cray-YMP. This is a very encourage result.

Fig. 3 is a partial reproduction of Fig. 2 for performance results on Cray-YMP and CM-2 with 32k processors, and it is represented in terms of MFLOPS. From this figure it is clearly seen that the CM-2 performed at above 1.2 GFLOPS when $N = 2048$. This speed is about to reach the machine's aggregate peak speed of about 2 GFLOPS. The speed achieved here is the one of the highest speed achieved on CM-2 with 32k processors in comparing with work done by some other investigators [4-6].

5 Concluding Remarks

Performance evaluation for panel method - influence ceofficient calculations has been made. The detailed performance results are obtained. Two concluding remarks can be drawn from this investigation. First, CM-FORTRAN code achieved a very high performance for influence coefficient matrix calculations and for most of the cases tested here it outperformed Cray-YMP supercomputer. The highest speed achieved in this investigation is above 1.2 GFLOPS which is very encourage. Second, the panel method is more appropriate for data parallel processing compared to finite-difference or finite-volume method.

References

[1] O. A. Kandil and H. Hu, *Full-potential integral solution for transonic flows with and without embedded euler domains*, AIAA Journal, Vol. 26, No. 9, 1988, pp. 1079-1086.

[2] O. A. Kandil and H. Hu, *Integral solution of unsteady full-potential equation for a transonic pitching airfoil*, Journal of Aircraft, Vol. 27, No. 2, 1990, pp.123-130.

[3] H. Hu, *A SD IEM for compressible wing flows with and without Shocks*, Boundary Elements in Fluid Dynamics, Computational Mechanics Publications and Elsevier Applied Science, 1992. pp. 49-60.

[4] B. L. Keshava Kumar, J. H. Kane, A. V. Srinivasan and R. B. Wilson, *The influence of massively parallel processing in boundary element computations*, Boundary Element Methods in Engineering, Springer-Verlag Germany, 1990. pp.500-507.

[5] T. A. Egolf, *Helicopter free-wake analysis on a massively parallel computer*, Boundary Element Methods in Engineering, Springer-Verlag Germany, 1990, pp.493-499.

[6] D. C. Jespersen and C. Levit, *A computational fluid dynamics algorithm on a massively parallel computer*, AIAA Paper 89-1936-CP, 1989.

No. 7

UNSTEADY TRANSONIC WING FLOW COMPUTATIONS USING FIELD-BOUNDARY ELEMENT MEHODS

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

(Partially Resulted From This Grant)

Published in Journal
Engineering Analysis with Boundary Elements
Vol. 10, No. 2, 1992, pp. 99-105

Engineering Analysis with Boundary Elements **10** (1992) 99–105

Unsteady transonic wing flow computations using field-boundary element methods

Hong Hu

Department of Mathematics, Hampton University, Hampton, Virginia 23668, USA

An unsteady integral equation (or called field-panel, field-boundary element) scheme for solving the full-potential equation for transonic unsteady wing flows has been developed. The unsteady full-potential equation has been written in a moving frame of reference, in the form of the Poisson's equation. Compressibility and unsteadiness have been treated as non-homogeneity. The integral equation solution in terms of velocity field is obtained by the Green's theorem. The solution consists of a wing surface (boundary elements) integral term of vorticity distribution, a wake surface (boundary elements) integral term of free-vortex sheet and a volume (field-elements) integral term of compressibility and unsteadiness over a small limited domain around the wing. Numerical solutions are obtained by a time-marching, iterative procedure. Time-derivative term is calculated by a second-order backward finite-difference scheme. To be consistent with the mixed-nature of flows, the Murman-Cole type-difference scheme is used to compute the derivatives of the density. The present scheme is applied to flows around a rectangular wing at transonic speed undergoing acceleration motion and transient pitching motion, respectively. The time history of wing surface pressure distributions has been presented.

Key words: unsteady transonic wing flows, full-potential equation, moving frame of reference, field-boundary elements, integral equation.

NOTATION

C_p	Surface pressure coefficient
$\vec{d}$	Distance vector pointed from sender to receiver
$\mathrm{d}s$	Infinitesimal surface area
$\vec{e}_d$	Unit vector of $\vec{d}$
g	Wing surface
G	$G_1 + G_2$
G_1	Compressibility
G_2	Unsteadiness
M_i	Initial value of M_o
M_o	Translation Mach number
$\dot{M}_o$	Rate of change of M_o
$\vec{n}$	Surface normal unit vector
$oxyz$	Moving frame of reference
$OXYZ$	Space-fixed frame of reference
$\vec{r}$	Position vector
$\vec{r}_p$	Pivot point vector
t	Time
$\vec{V}$	Absolute velocity
$\vec{V}_e$	Rotation velocity
$\vec{V}_o$	Translation velocity
$\vec{V}_r$	Relative velocity
w	Wake surface
α	Angle of attack
α_i	Initial value of α
$\dot{\alpha}$	Rate of change of α
$\vec{\gamma}$	Vorticity
κ	Gas specific heat ratio
ρ	Density
Φ	Absolute velocity potential
$\vec{\Omega}$	Angular velocity

INTRODUCTION

Starting in 1970, a great deal of progress has been made in solving transonic flows by using the finite-difference method (FDM) and finite-volume method (FVM). Although the Navier-Stokes equation formulation for the transonic flow computations has been understood as the best model and the FDM and FVM are successful in dealing with transonic flows, the computation of the

Engineering Analysis with Boundary Elements 0955-7997/92/$05.00

The associated boundary conditions are described in the next sub-section.

Boundary conditions

The boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, wake kinematic and dynamic conditions. They are described as follows;

$$\vec{V}_r \cdot \vec{n}_g = 0 \qquad \text{on } g(\vec{r}) = 0 \tag{7}$$

$$\Delta C_p|_{sp} = 0 \tag{8}$$

$$\nabla\Phi \to 0 \text{ away from } g(\vec{r}) = 0 \text{ and } w(\vec{r}, t) = 0 \tag{9}$$

$$\frac{1}{|\nabla w|}\frac{\partial' w}{\partial t} + \vec{V}_r \cdot \vec{n}_w = 0 \qquad \text{on } w(\vec{r}, t) = 0 \tag{10}$$

$$\Delta C_p = 0 \qquad \text{on } w(\vec{r}, t) = 0 \tag{11}$$

where $\vec{n}_g$ is the unit normal vector of the wing surface, $g(\vec{r}) = 0$; C_p is the surface pressure coefficient; the subscript sp refers to the edges of separation, and in the present scheme the only separation from the wing trailing edge is considered; and $w(\vec{r}, t) = 0$ is the wake surface.

IE solution

By using the Green's theorem, the integral equation solution of eqn (2) in terms of the relative velocity field is given by

$$\begin{aligned}\vec{V}_r(x, y, z, t) = &- \vec{V}_o(t) - \vec{\Omega}(t) \times (\vec{r} - \vec{r}_p) \\ &+ \frac{1}{4\pi}\int\int_g \frac{\vec{\gamma}_g(\xi, \eta, \zeta, t) \times \vec{d}}{d^3} \mathrm{d}s(\xi, \eta, \zeta) \\ &+ \frac{1}{4\pi}\int\int_w \frac{\vec{\gamma}_w(\xi, \eta, \zeta, t) \times \vec{d}}{d^3} \mathrm{d}s(\xi, \eta, \zeta, t) \\ &+ \frac{1}{4\pi}\int\int\int_V \frac{G(\xi, \eta, \zeta, t)}{d^2} \vec{e}_d \,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta\end{aligned} \tag{12}$$

where $\vec{\gamma}$ is the surface vorticity distribution; the subscripts g and w refer to the wing and wake surfaces, respectively; $\mathrm{d}s$ is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x - \xi)\vec{i} + (y - \eta)\vec{j} + (z - \zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = d/|d|$.

In eqn (12), the first integral term is the contribution of the wing surface vorticity; the second integral term is the contribution of the wake vorticity; and the third integral term is the contribution of the full compressibility and the unsteadiness. It should be noticed that the infinity condition, eqn (9), is automatically satisfied by the integral equation solution. It should also be noticed that the integrand of the volume integral term, the third integral term in eqn (12), decreases rapidly with increasing distance, d, not only because of the factor of $1/d^2$ but also $G(\xi, \eta, \zeta, t)$ diminishes rapidly with increasing distance. Consequently, for computational purposes, the volume integral term needs to be addressed only within the immediate vicinity of the body. This is the beauty of the IE methods.

COMPUTATIONAL SCHEME

Discretisation

A sketch of the computational model, with the relative size of the computational domain, is shown in Fig. 2. The wing and its wake are represented by triangular vortex panels. A uniform rectangular parallelepiped type of volume elements are used throughout the flow field. The discretised integral equation solution becomes

$$\begin{aligned}\vec{V}_r(x, y, z, t) &= -\vec{V}_o(t) - \vec{\Omega}(t) \times (\vec{r} - \vec{r}_p) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{\vec{\gamma}_{g_{i,k}}(\xi, \eta, \zeta, t) \times \vec{d}}{d^3} \mathrm{d}s(\xi, \eta, \zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LW}\sum_{k=1}^{NW}\int\int_{w_{i,k}} \frac{\vec{\gamma}_{w_{i,k}}(\xi, \eta, \zeta, t) \times \vec{d}}{d^3} \mathrm{d}s(\xi, \eta, \zeta, t) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV} G_{i,j,k}\int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{e}_d \,\mathrm{d}\xi\,\mathrm{d}\eta\,\mathrm{d}\zeta\end{aligned} \tag{13}$$

where the indices, i, j and k refer to the surface panels and field elements; $LG \times NG$ is the total number of wing surface panels; $LW \times NW$ is the total number of wake surface panels; and $LV \times MV \times NV$ is the total number of field elements. A constant G-distribution is used over the small field element, while a linear $\vec{\gamma}$-distribution is used over the small surface panel.

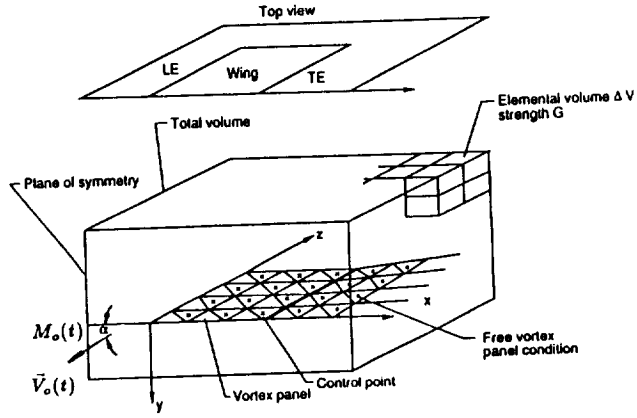


Fig. 2. Computational model, based on Reference 11.

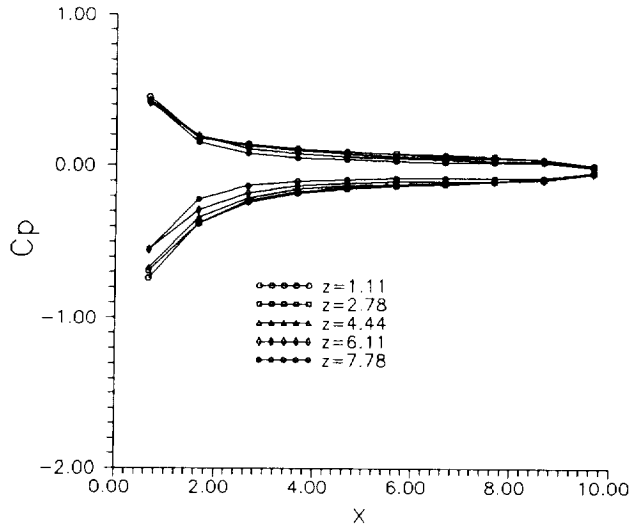


Fig. 6. C_p distribution at $M_o(t) = 0{\cdot}8$ over wing surface, acceleration motion.

The time derivative term of the potential, $(\partial'\Phi/\partial t)^{(n)}$, can be numerically calculated by $\Phi^{(n)}$ and $\Phi^{(n-1)}$, and hence $\Phi^{(n)}$ and $\Phi^{(n-1)}$ must be calculated by integration of the velocity field numerically. In order to avoid numerical error when doing this numerical integration, eqn (6) is used to compute $(\partial'\Phi/\partial t)^{(n)}$ distribution. Thus eqn (6) takes the form

$$\left(\frac{\partial'\Phi}{\partial t}\right)^{(n)} = \frac{1-\rho^{(n-1)^{n-1}}}{\kappa - 1} - \tfrac{1}{2}\vec{V}_r^{(n)^2} + \tfrac{1}{2}(\vec{V}_o^{(n)} + \vec{V}_e^{(n)})^2 \tag{17}$$

With $G_2^{(n)}$ obtained from eqn (16) and $(\partial'\Phi/\partial t)^{(n)}$ obtained from eqn (17), Step 1 is repeated until the solution converges.

Step 3 – at time step $(k = n + 1)$
Step 2 is repeated for time step $(n + 1)$.

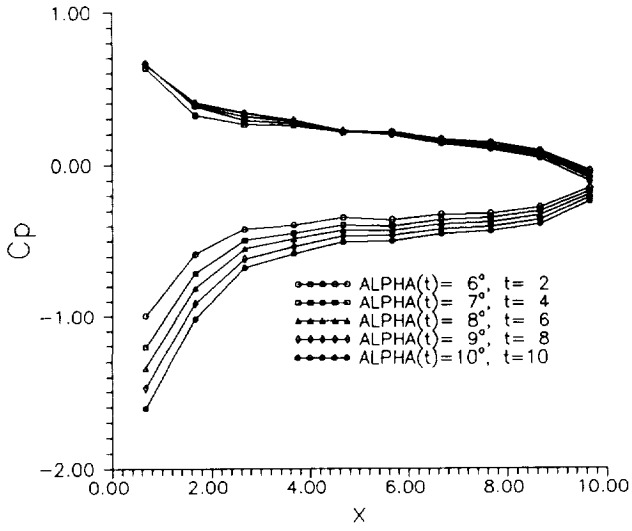


Fig. 7. Unsteady C_p history at $z = 1{\cdot}11$, pitching motion.

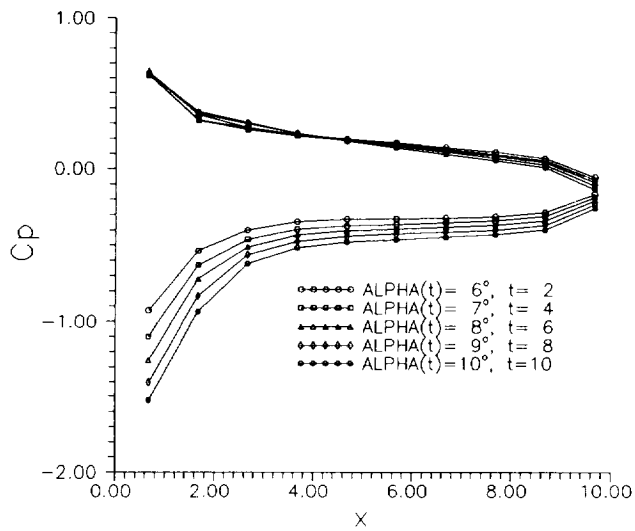


Fig. 8. Unsteady C_p history at $z = 4{\cdot}44$, pitching motion.

NUMERICAL EXAMPLES

The present scheme has been applied to a zero-thickness, rectangular wing with aspect ratio of two. The half-span of the wing and the wake is divided into 10×6 and 10×10 quadrilateral panels, respectively. Each quadrilateral panel consists of two triangular panels. The one-half of the computational domain is divided into $23 \times 9 \times 9$ field volume elements in x, y and z directions, respectively. Two numerical examples are presented as mentioned before. The first one is the acceleration motion and the second one is the pitching motion.

Acceleration motion

In this numerical example, the wing is given an acceleration motion at an angle of attack of 5 degrees.

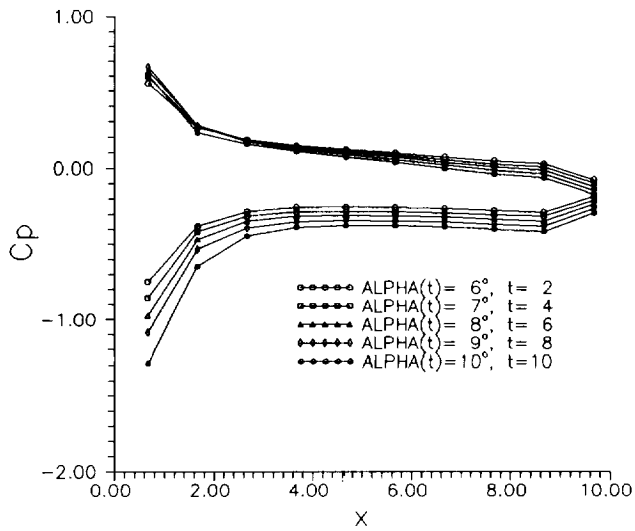


Fig. 9. Unsteady C_p history at $z = 7{\cdot}78$, pitching motion.

7. Kandil, O.A. & Yates, E.C., Jr. Computation of transonic vortex flows past delta wings-integral equation approach, *AIAA Journal*, 1986, **24**(11), 1729–1736.
8. Madson, M.D. Transonic analysis of the F-16A with under-wing fuel tanks: an application of the TranAir full-potential code, *AIAA* Paper 87–1198, 1987.
9. Kandil, O.A. & Hu, H. Transonic airfoil computation using the integral equation with and without embedded Euler domains, *Boundary Elements IX*, **3**, ed. C.A. Brebbia, W.L. Wendland & G. Kuhn, Computational Mechanics Publications, Springer-Verlag, 1987, pp. 553–566.
10. Kandil, O.A. & Hu, H. Full-potential integral solution for transonic flows with and without embedded Euler domains, *AIAA Journal*, 1988, **26**(9), 1074–1086.
11. Chu, L-C. Integral equation solution of the full potential equation for three-dimensional, steady, transonic wing flows, Ph.D. dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March 1988.
12. Sinclair, P.M. A three-dimensional field-integral method for the calculation of transonic flow on complex configuration – theory and preliminary results, *Aeronautical Journal*, June/July 1988, **92**(916), 235–241.
13. Kandil, O.A. & Hu, H. Unsteady transonic airfoil computation using the integral solution of full-potential equation, *Boundary Elements X*, **2**, ed. C.A. Brebbia, Computational Mechanics Publications, Springer-Verlag, 1988, pp. 352–371.
14. Kandil, O.A. & Hu, H. Integral solution of unsteady full-potential equation for a transonic pitching airfoil, *Journal of Aircraft*, 1990, **27**(2), 123–130.
15. Ogana, W. Transonic integro-differential and integral equations with artificial viscosity, *Engineering Analysis with Boundary Elements*, 1989, **6**(3), 129–135.
16. Hu, H. Field-boundary element solution for a transonic pitching wing, *Topics in Engineering, Vol. 7: Advances in Boundary Elements Methods in Japan and USA*, ed. M. Tanaka, C.A. Brebbia & R. Shaw, Computational Mechanics Publications, 1990, pp. 295–305.
17. Hu, H. Field-boundary element computation for transonic flows, *Computation Engineering with Boundary Elements*, Vol 1, ed. S. Grilli, C.A. Brebbia & A.H-D. Cheng, Computational Mechanics Publications, 1990, pp. 319–330.
18. Hu, H. full-potential integral solutions for steady and unsteady transonic airfoils with and without embedded Euler domains, Ph.D. dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March, 1988.

No. 8

COMPARATIVE STUDY OF COMPUTATIONAL PERFORMANCE OF CM-2 AND CRAY-YMP FOR BOUNDARY ELEMENT COMPUTATIONS

Hong Hu and Isaac T. Jackson
(Mr. Jackson is an undergraduate in computer science)
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Elements Abstracts Journal and Newsletter
Vol. 3, No. 5, September, 1992

Boundary Elements

Abstracts and Newsletter

(Sponsored by the International Society for Boundary Elements)
The worldwide Newsletter and abstracting publication for boundary element work published in journals, books, conferences, reports and theses.

Volume 3 No 5 ISSN: 0957-2902 September 1992

The purpose of the *Boundary Elements Abstracts Journal* is to provide rapid access to worldwide literature on boundary element research. The *Boundary Elements Newsletter* contains information on boundary element software, benchmarks, institution profiles, interviews with leading researchers, calendar of events and other information.

With six issues per year, the *Boundary Element Abstracts Journal and Newsletter* provides an essential publication to all researchers and those who are particularly interested in the boundary element method.

Boundary Elements Abstracts Journal and Newsletter is published bimonthly (January, March, May, July, September, November) by Computational Mechanics Publications.

Annual Subscription: £39 for individuals or £83 for organisations. Subscription includes membership of the International Society for Boundary Elements. Orders to Computational Mechanics Publications, Ashurst Lodge, Ashurst, Southampton, SO4 2AA, UK. Tel: 44 (0) 703 293223, Fax: 44 (0) 703 292853, Telex: 47388 Chacom G Attn. Compech.

North American orders to: Computational Mechanics Inc., 25 Bridge Street, Billerica, MA, 01821, USA. Tel: 508 667 5841, Fax 508 667 7582

Printed by Hobbs the Printers of Southampton, UK

Contents

A Guideline to BE Abstracts and Newsletter	161
I. Books on Boundary Elements	163
II. Journal/Conference Proceedings Papers	163
III. State-of-the-Art in Boundary Element Applications to Electrostatics and Electromagnetics	174
IV. Theses and Special Reports	181
Author Index	182
Boundary Elements Newsletter	185
Research Note	185
Short Communication in Software	192
Benchmarks	192
Institute Profile	193
Interview	193
Calendar of Events	195
ISBE Bulletin	196
Book Review	198

Boundary Elements Newsletter

Research Note

Comparative study of computational performance of CM-2 and Cray-YMP for boundary element computations

Hong Hu
Department of Mathematics, Hampton University, Hampton, Virginia, USA

&

Isaac T. Jackson*
Department of Computer Science, Hampton University, Hampton, Virginia, USA

A comparative study of computational performance of CM-2 and Cray-YMP computers for a simple two-dimensional source boundary element (panel) code has been made. A serial FORTRAN 77 code running on cray-YMP supercomputer has been converted into a parallel CM-FORTRAN code for running in CM-2 massively parallel computer. Detailed performance results are obtained for CM-2 with 8k, 16k, and 32k processors and for Cray-YMP. The comparison of the performance indicates that the influence of coefficient calculations on CM-2 with 32k processors outperformed the equivalent Cray-YMP code by a factor of 7 for 2048 boundary elements and achieved a speed of 1.2 GFLOPS. An even higher performance on CM-2 with more processors for larger problems is expected.

Key words: computer performance, massively parallel processors (MPP), connection machine (CM-2), supercomputer (Cray-YMP), boundary elements, aerodynamics

1. Introduction

In recent years, the processors of conventional vector supercomputers seem to be approaching the limit in computational speed inherent in their technology. However, the need for even faster computations continues to grow. As a consequence, parallel computers are being developed as a possible solution. Massively parallel computers, CM-2, developed by Thinking Machine Corporation are one family of paralel computer architectures which may large improvements in computation performance in the near future over today's fastest supercomputer, such as Cray-YMP. It is understood that the CM-2 with 64k processors has an aggregate peak performance of 4 GFLOPS, which is already faster than multi-processor Cray-YMP's aggregate peak performance of around 1 GFLOP.

Computational fluid dynamics (CFD) is one of the areas which need super-fast computation power. The connection machine has the potential to become tha main computational tool for CFD to repace the conventional supercomputers in the near future. This paper presents the performance comparison of the conventional supercomputer, Cray-YMP, and the new massively parallel computer, CM-2, for a simple boundary element fluid dynamics computational code. The present work has three principal objectives: (1) conversion of a simple FORTRAN 77 code running on Cray-YMP machine into parallel CM-FORTRAN code running on CM-2 machine; (2) evaluation of the performance of the codes on Cray-YMP and CM-2; and (3) prepara-

*Currently at Defense Information Systems Agency, Arlington, Virginia USA

tion for further code conversion and performance evaluation of comples three-dimensional boundary element fluid dynamics computational code.

In the next section the basic aspects of the CM-2 machine and CM-FORTRAN are briefly reviewed, which is followed by the description of the physical problem under this investigation in Section 3. the code conversion and performance evaluation are presented in Sections 4 and 5, repectively. Finally in Section 6, the concluding remarks are drawn.

2. CM-2 and CM-FORTRAN

The connection machine CM-2 system is an integrated combination of hardware and software designed for high speed, large problem parallel computation. The hardware elements of the system include front-end computers, a SIMD (single instruction multiple data) parallel processing unit to execute the data parallel operations, and a high performance data parallel operations, and a high performance data parallel I/O system. The SIMD parallel processing unit is the heart of CM-2 system, which contains up to 65536 single-bit physical processors (64k) in blocks of 8k (k=1024). The CM-2 used in the present work at NASA-Ames Research Centre has 32k processors with each processor operating at a clock speed of 6.7 MHz. The aggregate peak performance for this 32k CM-2 is about 2 GFLOPS.

The 32k single-bit processors on this 32k-CM2 are grouped in 1024 nodes of 32 processors each. Each node also has 64-bit Weitek floating point co-processors, 4MB of local memory, and hardware for interprocessor communication. Parallel data structures are spread across the data processors, with a single element stored in each processors's memory. If the number of the parallel data elements exceeds the total number of physical processors, the connection machine creates virtual processors by dividing the memory of each physical processor. The ratio of virtual to physical processors is known as the VP-ratio, R_{vp}. In general, floating-point performance (usually in terms of MFLOPS or GFLOPS) is maximum when Rvp is as large as possible, since the communication overhead is reduced as Rvp increases. The detailed description about CM-2 computer is documented in many places, such as References 1 and 2.

The CM-FORTRAN lagruage is an implementation of FORTRAN 77 supplemented with array-processing extentions from the ANSI draft and ISO standard FORTRAN 90. These array-processing features map naturally onto the data parallel architecture of the CM-2 system, since CM-FORTRAN allows array elements to be evaluated simultaneously. Reference 3 explains these extensions, while Reference 4 gives a full description of CM-FORTRAN. The most important difference between CM-FORTRAN and FORTRAN 77 is the treatment of entire arrays as objects in CM-FORTRAN, thus explicit indexing in CM-FORTRAN is not always necessary. For example, it is not necessary to write Do-Loops or other such control constructs to have the operation repeated for each element of arrays. Therefore, this feature maps the problem directly to the CM-2 with minimal programming effect. The further explanation of CM-FORTRAN will be given along with the code conversion example in Section 4.

3. Fluid Dynamics - BEM

The physical problems considered here are potential flows around any arbitrary complex configurations (let us call it 'body'), including incompressible and compressible flows with and without separations. The governing equation to this type of problem can be written in the form of Poisson's equation given by

$$\nabla^2\Phi = G \tag{1}$$

where G represents full linear or non-linear compressibilty and is a function of Φ in general. This type of problem can be solved by boundary element methods (also called panel methods, integral equation methods). The boundary element method is based on Green's theorem, which represents the solution in terms of integrals over body surfaces, separated surfaces and volume around the body. The surfaces and volume are then divided into a large number of elements, where integrals are evaluated.

The boundary element solution in terms of velocity field ($\vec{V} = \nabla\Phi$) is given by [5,6]

$$\begin{aligned}\vec{V}(x,y,z) &= \vec{V}_\infty \\ &- \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{\vec{\gamma}_g(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{ms=1}^{MS}\sum_{i=1}^{LS}\sum_{k=1}^{NS}\int\int_{w_{ms,i,k}} \frac{\vec{\gamma}_w(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV}\int\int\int_{V_{i,j,k}} \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,d\xi\,d\eta\,d\zeta\end{aligned} \tag{2}$$

where q and $\vec{\gamma}$ are the surface source and vorticity distribution respectively, which are unknowns to be determined by applying boundary conditions; the subscripts g and w refer to the body and separated surfaces, respectively, where MS is the total number of separations; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

It should be noted that eqn (2) involves evaluating a large number of integrals (in fact they are vectors also!) over body surface (total of $2 \times LG \times NG$), separated surface (total of $MS \times LS \times NS$) and volume (total of $LV \times MV \times NV$). The total number of elements can be very large for aerodynamic problems, and for aircraft configurations it can be, for example in the order of 106. An important feature of eqn (2) is that the calculation for each (x,y,z) and each (ξ,η,ζ) can simultaneously for all (x,y,z) and (ξ,η,ζ) with a single instruction. This feature of boundary element calculation leads itself in a natural way for processing data in a SIMD parallel computing environment.

For simplicity in the present work, an incompressible flow (G=0) past a two-dimensional symmetric configuration at zero incidence is considered. At this simple flow condition, eqn (2) reduces to a much simpler form,

$$\begin{aligned}\vec{V}(x,y) &= \vec{V}_\infty \\ &- \frac{1}{2\pi}\sum_{i=1}^{N}\int_{g_i} q_g(\xi,\eta)\frac{(x-\xi)\vec{i} + (y-\eta)\vec{j}}{(x-\xi)^2 + (y-\eta)^2}ds(\xi,\eta)\end{aligned} \tag{3}$$

By applying body surface zero-normal-velocity condition at each $(x,y) = (x_j,y_j)$ for j=1 to N, here $N \equiv LG$, a $N \times N$ system of equations is obtained as

$$[A][q] = [B] \tag{4}$$

where $[A]$ is $N \times N$ influence coefficient matrix; $[q]$ is an $N \times 1$ unknown vector matrix containing q_j for j=1 to N; and $[B]$ is an $N \times 1$ known vector matrix which is contributed from $\vec{V}_\infty$ in this simple two-dimensional flow case.

The solution procedure for the above two-dimensional problem using boundary element method involves four steps: (1) generation of body geometry information; (2) evaluation on integrals of eqn (3) for i=1 to N and for $(x,y) = (x_j,y_j)$ with j=1 to N to construct $[A]$ matrix; (3)

numerical solution of resulting linear system given by eqn (4); and (4) post-processing, aerodynamic calculations. The solution procedure for the general three-dimensional flows governed by eqn (2) is basically the same as these four steps, except that the solutions are usually obtained through an iterative procedure. The most important difference between the present simple flow governed by eqn (2) and general complex flows governed by eqn (3) is that, $[A]$ and $[B]$ matrix calculations are much more expensive in general three-dimensional computations. The experiences have shown that the $[A]$ and $[B]$ matrix calculations for general three-dimensional complex flow computations usually take above 80% of total computational time.[5,6] The detailed discussion about general three-dimensional complex flows by boundary element methods can be found in References 5 and 6.

4. CM-FORTRAN implementation

Serial FORTRAN codes of two-dimensional source panel (boundary element) methods described by eqns (3) and (4) are available in some references, such as Reference 7. The serial FORTRAN code in Reference 7 thus provides the basis for the CM-2 code present here. This serial FORTRAN code has first modified for efficient executing on NASA-Langley Research Center's CM-2 computer. The listing of the CM-FORTRAN code is presented in the Appendix, where several CM-Timing routines have been used to get performance results.

Although the boundary element solution procedure consists of four steps discussed earlier, Step 2 for evaluating integrals and Step 3 for solving linear the system take most part of the computational time. Experience has shown that these two steps usually take more than 95% of total computational time when the problem has a reasonable size. Therefore, the present interest on the performance analysis is first focused on Step 2 computations and then Step 3 computations, although the code is fully converted into CM-FORTRAN.

```
      SUBROUTINE MATELM
      PARAMETER(N=32,M=33)
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      DO 2 K=1,N
      DO 1 J=1,N
      IF (K .EQ. J) FN(K,J)=2.0*PI
      IF (K. EQ. J) FT(K,J)=0.0
      IF (K. EQ. J) GOTO 1
      DYJ=SI(J)*DS(J)
      DXJ=CI(J)*DS(J)
      SPH=DS(J)*0.5
      XD=XC(K)-XC(J)
      YD=YC(K)-YC(J)
      RKJ=SQRT(XD*XD+YD*YD)
      BKJ=ATAN2(YD,XD)
      ALJ=ATAN2(DYJ,DXJ)
      GKJ=ALJ-BKJ
      ZIK=RKJ*COS(GKJ)
      ETK=-RKJ*SIN(GKJ)
      R1S=((ZIK+SPH)**2)+ETK*ETK
      R2S=((ZIK-SPH)**2)+ETK*ETK
      QT=ALOG(R1S/R2S)
      DEN=ZIK*ZIK+ETK*ETK-SPH*SPH
      GNM=ETK*DS(J)
      QN=2.0*ATAN2(GNM,DEN)
      UKJ=QT*CI(J)-QN*SI(J)
      VKJ=QT*SI(J)+QN*CI(J)
      FN(K,J)=-UKJ*SI(K)+VKJ*CI(K)
      FT(K,J)=UKJ*CI(K)+VKJ*SI(K)
    1 CONTINUE
      RHS(K)=UINF*SI(K)-VINF*CI(K)
    2 CONTINUE
      RETURN
      END
```

Fig. 1(a) FORTRAN 77 subroutine for calculating matrices

Figure 1(a) is the list of the serial Cray version FORTRAN subroutine for evaluating influence coefficient matrix, $[A]$, and the matrix $[B]$. It is noted that this subroutine is nothing but a two-level Do-Loop, which provides for evaluating each element of $[A]$ and $[B]$. When the code is in execution on Cray-YMP, the vectorization of the inner Do-Loop is automatically done through the vectorization capability of the FORTRAN 77 compiler.

```
      SUBROUTINE MATELM
      PARAMETER(N=32,M=33)
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
     2,DYJ(N,N),DXJ(N,N),SPH(N,N),XD(N,N),YD(N,N)
     3,RKJ(N,N),BKJ(N,N),ALJ(N,N),GKJ(N,N),ZIK(N,N)
     4,ETK(N,N),R1S(N,N),R2S(N,N),QT(N,N),DEN(N,N)
     5,GNM(N,N),QN(N,N),UKJ(N,N),VKJ(N,N)
     6,DS2(N,N),CI2(N,N),SI2(N,N)
     7,XC2(N,N),YC2(N,N),XC3(N,N),YC3(N,N)
     8,SI3(N,N),CI3(N,N)
      LOGICAL MAIN_DIAG(N,N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      XC2 = SPREAD(XC,DIM=1,NCOPIES=N)
      YC2 = SPREAD(YC,DIM=1,NCOPIES=N)
      XC3 = SPREAD(XC,DIM=2,NCOPIES=N)
      YC3 = SPREAD(YC,DIM=2,NCOPIES=N)
      SI2=SPREAD(SI,DIM=1,NCOPIES=N)
      CI2=SPREAD(CI,DIM=1,NCOPIES=N)
      SI3=SPREAD(SI,DIM=2,NCOPIES=N)
      CI3=SPREAD(CI,DIM=2,NCOPIES=N)
      DS2=SPREAD(DS,DIM=1,NCOPIES=N)
      MAIN_DIAG=DIAGONAL(SPREAD(.TRUE.,1,N),.FALSE.)
      WHERE(MAIN_DIAG)
        FN = 2.0 * PI
        FT = 0.0
      ELSEWHERE
        DYJ = SI2 * DS2
        DXJ = CI2 * DS2
        SPH = DS2 * 0.5
        XD = XC3 - XC2
        YD = YC3 - YC2
        RKJ=SQRT(XD*XD +YD*YD)
        BKJ=ATAN2(YD,XD)
        ALJ=ATAN2(DYJ,DXJ)
        GKJ=ALJ-BKJ
        ZIK=RKJ*COS(GKJ)
        ETK=-RKJ*SIN(GKJ)
        R1S=((ZIK+SPH)**2) + ETK*ETK
        R2S=((ZIK-SPH)**2) + ETK*ETK
        QT=ALOG(R1S/R2S)
        DEN=ZIK*ZIK + ETK*ETK - SPH*SPH
        GNM=ETK*DS2
        QN=2.0*ATAN2(GNM,DEN)
        UKJ=QT*CI2-QN*SI2
        VKJ=QT*SI2+QN*CI2
        FN=-UKJ*SI3+VKJ*CI3
        FT=UKJ*CI3+VKJ*SI3
      ENDWHERE
      RHS=UINF*SI-VINF*CI
      RETURN
      END
```

Fig. 1(b) CM-FORTRAN subroutine for calculating matrices

Figure 1(b) is the list of the parallel CM-FORTRAN subroutine for evaluating coefficient matrix, $[A]$, and the matrix $[B]$. A few things should be mentioned. First, since no Do-Loop has been seen here since in CM-FORTRAN entire arrays are treated as objects and array elements are evaluated simultaneously. Second, the conditional IF-Statement is represented in the form of WHERE-ELSEWHERE-ENDWHERE format which allows the conditional processing to be done in parallel; Third, the CM-FORTRAN intrinsic function SPREAD is used here to create two-dimensional arrays form one-dimensional arrays by duplicating the elements in either row- or column-directions as desired for easy implementation of parallel processing of statements like, $X1(K,J)=X2(K)+X3(J)$; Fourth, temporary scale variables, such as DYJ, DXJ and so on in serial FORTRAN become two-dimensional arrays in CM-FORTRAN in order to implement parallel processing. However, such arrays increase the total memory requirement of CM-code significanly as compared with the serial FORTRAN code. For example, within the present invesitgation it has been found that hte CM-FORTRAN code with 4096 boundary elements exceeded the 2GB memory limit of the CM-2 computer with 16k processors.

The dense linear system of eqn (4) is solved by the Gauss elimination method in Cray serial FORTRAN version. To compare

Table 1. The detailed computational performance results

Task/Size(N)		8K-CM2		16K-CM2		32K-CM2		Cray-YMP	
Task	N	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS	Time(s)	MFLOPS
Mat Coef	32	0.0173	12.3	0.0173	12.3	0.0173	12.3	0.00152	140
Lin Syst	32	0.125	0.210	0.123	0.210	0.123	0.210	0.00152	17
Total	32	0.156	1.64	0.154	1.66	0.154	1.66	0.00851	30
Mat Coef	64	0.0194	44.5	0.0194	44.5	0.0186	46.4	0.00523	165
Lin Syst	64	0.261	0.741	0.252	0.767	0.250	0.774	0.00879	22
Total	64	0.294	3.78	0.286	3.85	0.282	3.90	0.0250	44
Mat Coef	128	0.0310	112	0.0224	155	0.0203	171	0.0202	172
Lin Syst	128	0.717	2.03	0.590	2.46	0.534	2.72	0.0581	25
Total	128	0.764	6.61	0.627	8.05	0.569	8.89	0.101	50
Mat Coef	256	0.0680	205	0.0447	312	0.0323	432	0.0793	176
Lin Syst	256	3.08	3.77	2.00	5.81	1.45	8.01	0.415	28
Total	256	3.17	8.24	2.07	12.6	1.50	17.4	0.544	48
Mat Coef	512	0.205	273	0.115	487	0.0685	818	0.313	179
Lin Syst	512	18.7	4.87	10.4	8.76	6.18	14.7	3.14	29
Total	512	19.0	7.89	10.5	14.3	6.30	23.8	3.57	42
Mat Coef	1024	0.731	308	0.383	587	0.206	1092	1.25	180
Lin Syst	1024	139	5.20	71.3	10.3	37.7	19.4	24.4	30
Total	1024	140	6.80	71.7	13.4	38.0	25.2	25.9	37
Mat Coef	2048	2.84	316	1.43	627	0.737	1216	4.98	180
Lin Syst	2048	1036	5.60	546	10.5	277	21	192	30
Total	2048	1039	6.48	548	12.3	278	24	198	34

computational performance, this linear system is solved by calling a modified Gauss-Jordan function routine from CM Scientific Software Library (CMSSL) when running in CM-2 computer. CMSSL is created for data parallel architectures and is designed to handle concurrent applications. Although the Gauss-Jordan method requires about 50% more operations compared to the Gauss elimination method, the Gauss-Jordan method is more appropriate for data parallel computation. Another reason to use Gauss-Jordan rather the the Gauss elimination method is simply that only the Gauss-Jordan method is available from CMSSL.

5. Performance analysis

The serial code with different numbers of boundary elements (N) is first executed on the Cray-YMP supercomputer using single processor to provide the basis for performance comparison. The Cray-YMP used here has 5 processors (CPUS) with 128 MW SRAM central memory. Each CPU is a register-to-register vector processor with peak performance at 150-300mflops. The computational performance in terms of mflops is obtained using Cray-YMP's PERFTRACE utility. The parallel CM-FORTRAN code with different numbers of boundary elements is then executed on CM-2 with 8k, 16k and 32k processors under a slicewise model.

Table 1 gives detailed performance results for Cray-YMP and CM-2 computers with a varying size of problem. The performance of CM-2 in terms of mflops is the equivalent Cray-YMP performance. In Table 1, 'Mat Coef' refers to the subroutine for evaluating matrices; 'Lin Syst' refers to the routine for solvinf linear system; and 'Total refers to the total computation for entire code. The sets of results form Table 1 have been extracted to be presented in Figs 2-5. Each figure is discussed below.

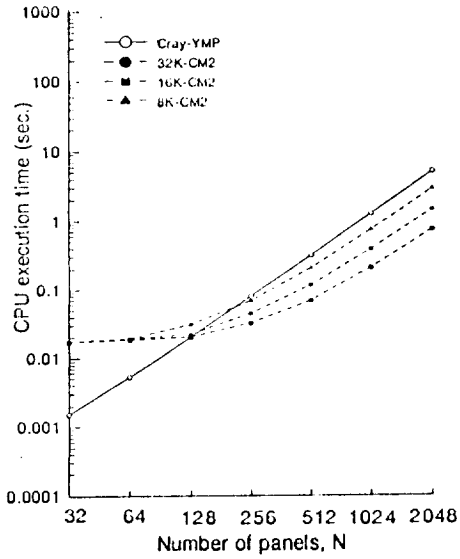


Fig. 2 Performance comparison for calculating matrices

Figure 2 shows execution time for evaluating the influence coefficient matrix, $[A]$, and the matrix $[B]$ obtained on Cray-YMP and CM-2 computers for different numbers of boundary elements. The CM-2 performance at each number of boundary elements, as shown in Table 1, is represented by solid points which are connected by dashed lines since these lines do not represent the actual variation of execution time between each point. It can be seen that the cpu execution time decreases with the increase of the number of CM-2 processors after the size of the problem is large enough to fully use all processors. For example when N=2048, the cpu time of 2.84 seconds with 8k processors is reduced to 1.43 seconds with 16k processors, and then is further reduced to 0.737 seconds with 32k processors. That is to say that whenever the wavenumber of processors used is doubled, the CPU time is reduced by a factor of 2. It is also seen that when the problem size is large enough for the CPU time required on CM-3, even with 8k processors, is significantly (note that Log_{10}-axis is used for

execution time!) less than that required on CM-2 with 32k processors is about 1/7 odf that required on Cray-YMP. This is a very encouraging result.

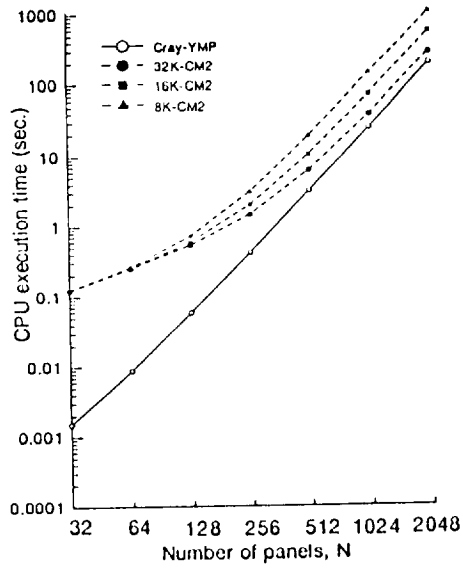


Fig. 3 Performance comparison for solving linear system

Figure 3 shoes the CPU time for a linear system solver. It should be mentioned that again before comparing performance, the Gauss-Jordan solver requires about 50% more operations compared to the Gauss elimination solver. The comparison shows that the cpu time required on CM-2 with 32k processors approaches that required on Cray-YMP with the increase of the problem size. For example, the CPU time required on CM-2 with 32k processors is 0.125 seconds for N=32 which is much larger than that for Cray-YMP of 0.00152 seconds; while this comparison becomes 277 seconds to 192 seconds when N=2048. Therfore it can be expected thet when the problem size becomes large enough the CM-2 with 32k processors will outperform Cray-YMP for solving linear systems. By comparing the CPU time on CM-2 with 8k, 16k and 32k processors it is believed that if 64k processors are used, CM-2 will outperform Cray-YMP even at a not very large N-value, for example at N=512.

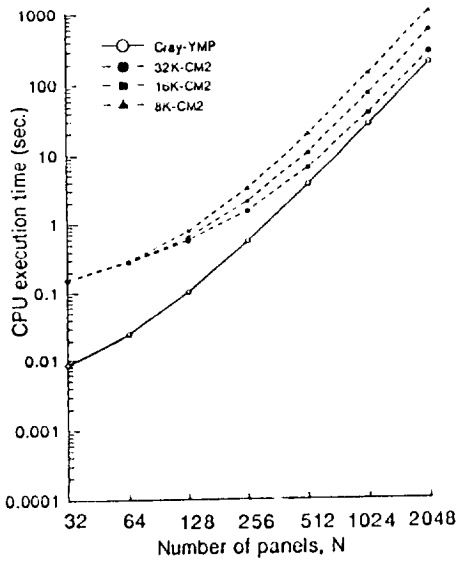


Fig. 4 Performance comparison for entire code computation

Figure 4 shows the total cpu time. It is found that the situation in this figure is similar to that of Fig.3., since in the present physical problem, the solution of the linear system takes most of the computational time when N is large. But it is not true for general three-dimensional complex flows where separation occurs and compressibilty is important, as governed by eqn (2). In such complex flow situations, the experience has shown that the evaluation of coefficient matrix, $[A]$, and the matrix $[B]$ usually take above 80% of the total computational time,[5,6] as mentioned earlier.

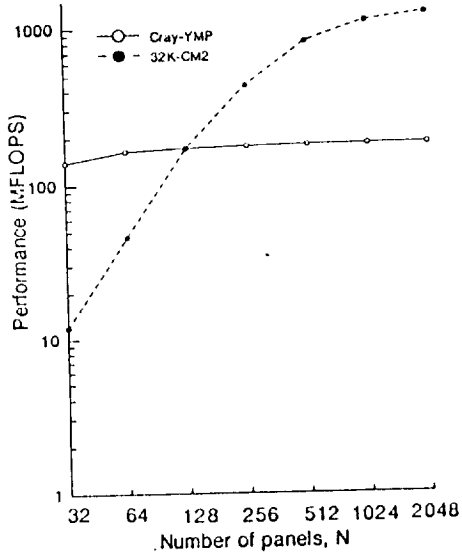


Fig. 5 Computational speeds for calculating matrices in terms of MFLOPS

Figure 5 is a reproduction of Fig. 2 for performance results on Cray-YMP and CM-2 with 32k processors, and is represented in terms of mflops. From this figure it is clearly seen that the CM-2 performed at above 1.2 GFLOPS when N=2048. This speed is about to reach the machine's aggregate peak speed of about 2 gflops. The speed acheived here is one of the highest speeds achieved on CM-2 with 32k processors for fluid dynamics problems done by some investigators.[8-12]

6. Concluding remarks

A simple source panel (boundary element) code has been sucessfully implemented on the massively parallel Connection Machine CM-2 using CM-FORTRAN language. The detailed performance results are obtained and analysed. Some concluding remarks can be drawn from this investigation. First, the conversion of serial fortran code to parallel CM-FORTRAN code is straightforward with little difficulty. Second, CM-FORTRAN code acheived a very high performance and for most of the cases tested here it outperformed or near performed the Cray-YMP supercomputer. The highest speed achieved in this investigation is above 1.2 GFLOPS which is very encouraging. Third, the boundary element method is more appropriate for data parallel processing compared to finite-difference or finite-volume methods. Fourth, further computational performance inverstigation can be made on real life general three-dimensional complex flow problems using boundary element methods where an even higher performance should be expected on CM-2.

Acknowledgements

This work has been supported by NASA Langley Research Center under Grant No. NAG-1-1170. Dr Carson Yates, Jr. was and Mr

Walter Sila is the technical monitor. The CM-2 computation is supported by NASA-Ames Research Center under NAS Program.

References

1. *Connection machine model CM-2 technical summary*, The Thinking Machine Corporation, Cambridge, Massachusetts, May 1989.

2. *CM user's guide*, The Thinking Machine Corporation, Cambridge, Massachusetts, Nov. 1990.

3. *CM FORTAN programming guide*, The Thinking Machine Corporation, Cambridge, Massachesetts, Feb. 1991.

4. *CM FORTRAN reference manual*, The Thinking Machine Corporation, Cambridge, Massachesetts, Feb. 1991.

5. Hu, H., Experience with transonic flow IE computations, *Boundary Elements in Fluid Dynamics*, ed. C.A. Brebbia and P.W. Partridge, Computational Mechanics Publications and Elsevier Applied Science, 1992, 35-48.

6. Hu, H., A 3D IEM for compressible wing flows with and without shocks, *Boundary Elements in Fluid Dynamics*, ed. C.A. Brebbia and P.W. Partridge, Computational Mechanics Publications with Elsevier Applied Science, 1992, 49-60.

7. Fletcher, C.A.J., *Computational techniques for fluid dynamcis*, Springer-Verlag Berlin Heidelberg, 1988, chap. 14.

8. Keshava Kumar, B.L., Kane, J.H., Srinivasan, A.V. & Wilson, R.B., The influence of massively parallel processing in boundary element computations, *Boundary Element Methods in Engineering*, ed. B.C. Annigiri and K. Tseng, Springer-Verlag, Germany, 1990, 500-507.

9. Egolf, T.A., Helicopter free-wake analysis on a massively parallel computer, *Boundary Element Methods in Engineering*, ed. B.C. Annigiri and K. Tseng, Springer-Verlag, Germany, 1990, 493-499.

10. Jespersen, D.C. & Levit, C., A computational fluid dynamics algorithm on a massively parallel computer, *AIAA Paper 89-1936-CP*, 1989.

11. Chriske, D.M. & Bogucz, E.A., Performance evaluation of the connection machine for panel method calculations, *AIAA Paper 91-0439*, 1991.

12. Madavan, N.K., CFD on the CM-2 from RANS to DNS simulation, *NAS user seminar*, NASA-Ames Research Center, April 22, 1992.

Appendix

The following is the list of CM-FORTRAN code for two-dimensional boundary element (panel) method.

```
      PROGRAM BEM2D_CM
      PARAMETER(N=32,M=33)
C -------------------------------------------------
C     BEM CALCULATES VELOCITY AND PRESSURES ABOUT
C     ARBITRARY 2D BODY
C     PARAMETER        DESCRIPTION
C     AA         MATRIX [A]
C     B          MINOR SEMI-AXIS LENGTH OF THE ELLIPSE
C     FMN        FREESTREAM MACH NUMBER Minf
C     FN         INDUCED VELOCITY NORMAL TO PANEL K
C                DUE TO UNIT SOURCE
C     FT         INDUCED VELOCITY TANGENTIAL TO PANEL K
C                DUE TO UNIT SOURCE
C     M          NUMBER OF ELEMENT END POINTS
C     N          NUMBER OF ELEMENTS
C     RHS        VECTOR MATRIX [B]
C     SDE        SOURCE DENSITY VECTOR, [Q]
C     U,VINF     FREESTREAM VELOCITY COMPONENTS
C     X,Y        COORDINATES OF PANEL END POINTS
C     XC,YC      COORDINATES  OF PANEL  CONTROL POINTS
C     QE         EXACT TANGENTIAL VELOCITY AT
C                THE SURFACE OF ELLIPSE
C -------------------------------------------------
      INCLUDE '/usr/include/cm/CMF_defs.h'
      INCLUDE '/usr/include/cm/cmssl-cmf.h'
C -------------------------------------------------
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N),AA(N,N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      DIMENSION BG(N,1),IPIVOT(N),WK(N)
      CALL CM_TIMER_CLEAR(8)
      CALL CM_TIMER_START(8)
      UINF = 1.0
      VINF = 0.0
      FMN = 0.00
      B = 1.00
      NMAX = N
      NRHS = 1
      IFAC = 0
      PI = 3.14159265
      CPI = 2.0/PI
      WRITE(6,2)N,B
      WRITE(6,3)UINF,VINF,FMN
    2 FORMAT(1X,'PANEL METHOD WITH ',I5,'ELEMENTS',5X,
     1'ELLIPSE MINOR SEMI-AXIS = ',F6.3,/)
    3 FORMAT(1X,'ONSET VELOCITY COMPONENTS = ',2F6.3,
     1 2X,' FREESTEAM MACH NUMBER = ',F6.3,//)
C -------------------------------------
C     CALCULATE COORDINATES OF BODY AND
C     CONTROL POINTS - Step 1
C -------------------------------------
      CALL CM_TIMER_CLEAR(0)
      CALL CM_TIMER_START(0)
      CALL BODY(FMN,B)
      CALL CM_TIMER_STOP(0)
      CALL CM_TIMER_PRINT(0)
C -------------------------------------
C     CONSTRUCT THE MATRIX EQUATION - Step 2
C -------------------------------------
      CALL CM_TIMER_CLEAR(1)
      CALL CM_TIMER_START(1)
      CALL MATELM
      CALL CM_TIMER_STOP(1)
      CALL CM_TIMER_PRINT(1)
      CALL CM_TIMER_CLEAR(9)
      CALL CM_TIMER_START(9)
      AA = FN
      SDE = RHS
      CALL CM_TIMER_STOP(9)
      CALL CM_TIMER_PRINT(9)
C -------------------------------------
C     SOLVE LINEAR SYSTEM - Step 3
C     FUNCTION ROUTINE GEN_GJ_SOLVE
C     IS CALLED FROM CMSSL
C -------------------------------------
      CALL CM_TIMER_CLEAR(10)
      CALL CM_TIMER_START(10)
      BG(1:N,1) = RHS(1:N)
      CALL CM_TIMER_STOP(10)
      CALL CM_TIMER_PRINT(10)
      CALL CM_TIMER_CLEAR(2)
      CALL CM_TIMER_START(2)
      PRINT *, 'PIVOT_MIN=',
     +GEN_GJ_SOLVE(AA,BG,N,NRHS,
     +CMSSL_partial_pivoting,IER)
      CALL CM_TIMER_STOP(2)
      CALL CM_TIMER_PRINT(2)
      CALL CM_TIMER_CLEAR(11)
      CALL CM_TIMER_START(11)
      SDE(1:N) = BG(1:N,1)
      CALL CM_TIMER_STOP(11)
      CALL CM_TIMER_PRINT(11)
C -------------------------------------
C     CALCULATE VELOCITY AND PRESSURE
C     AT THE BODY SURFACE - Step 4
C -------------------------------------
      CALL CM_TIMER_CLEAR(3)
      CALL CM_TIMER_START(3)
      CALL SURVL(B,FMN)
      CALL CM_TIMER_STOP(3)
      CALL CM_TIMER_PRINT(3)
      CALL CM_TIMER_STOP(8)
      CALL CM_TIMER_PRINT(8)
      STOP
      END
```

```
      SUBROUTINE BODY(FMN,B)
      PARAMETER(N=32,M=33)
C -------------------------------------------------
C     CALCULATES BODY AND CONTROL POINT COORDINATES
C     FOR AND ELLIPSE WITH MINOR SEMI-AXIS, B
C -------------------------------------------------
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
     2,SX(N),SY(N),TH(M),AI(M)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      NHLFF = N/2 + 1
      NHH = NHLFF + 1
      AN = NHLFF - 1
      DTH = PI/AN
      FORALL(I=1:NHLFF) AI(I)=I-1
      TH(1:NHLFF) = PI - AI(1:NHLFF) * DTH
      X(1:NHLFF) = COS(TH(1:NHLFF))
      Y(1:NHLFF) = B * SIN(TH(1:NHLFF))
      X(NHH:M) = X(NHLFF-1:1:-1)
      Y(NHH:M) = -Y(NHLFF-1:1:-1)
      XC=(X(1:N)+X(2:N+1))*0.5
      YC=(Y(1:N)+Y(2:N+1))*0.5
      SX=X(2:N+1)-X(1:N)
      SY=Y(2:N+1)-Y(1:N)
      DS=SQRT(SX*SX+SY*SY)
      CI=(X(2:N+1)-X(1:N))/DS(1:N)
      SI=(Y(2:N+1)-Y(1:N))/DS(1:N)
      RETURN
      END

      SUBROUTINE MATELM
      PARAMETER(N=32,M=33)
C -------------------------------------
C     CALCULATES MATRIX ELEMENTS AND RHS.
C -------------------------------------
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
     2,DYJ(N,N),DXJ(N,N),SPH(N,N),XD(N,N),YD(N,N)
     3,RKJ(N,N),BKJ(N,N),ALJ(N,N),GKJ(N,N),ZIK(N,N)
     4,ETK(N,N),R1S(N,N),R2S(N,N),QT(N,N),DEN(N,N)
     5,GNM(N,N),QN(N,N),UKJ(N,N),VKJ(N,N)
     6,DS2(N,N),CI2(N,N),SI2(N,N)
     7,XC2(N,N),YC2(N,N),XC3(N,N),YC3(N,N)
     8,SI3(N,N),CI3(N,N)
      LOGICAL MAIN_DIAG(N,N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      XC2 = SPREAD(XC,DIM=1,NCOPIES=N)
      YC2 = SPREAD(YC,DIM=1,NCOPIES=N)
      XC3 = SPREAD(XC,DIM=2,NCOPIES=N)
      YC3 = SPREAD(YC,DIM=2,NCOPIES=N)
      SI2=SPREAD(SI,DIM=1,NCOPIES=N)
      CI2=SPREAD(CI,DIM=1,NCOPIES=N)
      SI3=SPREAD(SI,DIM=2,NCOPIES=N)
      CI3=SPREAD(CI,DIM=2,NCOPIES=N)
      DS2=SPREAD(DS,DIM=1,NCOPIES=N)
      MAIN_DIAG=DIAGONAL(SPREAD(.TRUE.,1,N),.FALSE.)
      WHERE(MAIN_DIAG)
        FN = 2.0 * PI
        FT = 0.0
      ELSEWHERE
        DYJ = SI2 * DS2
        DXJ = CI2 * DS2
        SPH = DS2 * 0.5
        XD = XC3 - XC2
        YD = YC3 - YC2
        RKJ=SQRT(XD*XD +YD*YD)
        BKJ=ATAN2(YD,XD)
        ALJ=ATAN2(DYJ,DXJ)
        GKJ=ALJ-BKJ
        ZIK=RKJ*COS(GKJ)
        ETK=-RKJ*SIN(GKJ)
        R1S=((ZIK+SPH)**2) + ETK*ETK
        R2S=((ZIK-SPH)**2) + ETK*ETK
        QT=ALOG(R1S/R2S)
        DEN=ZIK*ZIK + ETK*ETK - SPH*SPH
        GNM=ETK*DS2
        QN=2.0*ATAN2(GNM,DEN)
        UKJ=QT*CI2-QN*SI2
        VKJ=QT*SI2+QN*CI2
        FN=-UKJ*SI3+VKJ*CI3
        FT=UKJ*CI3+VKJ*SI3
      ENDWHERE
      RHS=UINF*SI-VINF*CI
      RETURN
      END

      SUBROUTINE SURVL(B,FMN)
      PARAMETER(N=32,M=33)
C -------------------------------------------------
C     CALCULATES VELOCITIES AND PRESSURE AT CONTROL
C     POINTS
C -------------------------------------------------
      DIMENSION X(M),Y(M),XC(N),YC(N),DS(N),FN(N,N)
     1,FT(N,N),RHS(N),SDE(N),CI(N),SI(N)
     2,QTS(N),QNS(N),QNK(N),QTK(N),UU(N),VV(N)
     3,SDE2(N,N),FTSDE2(N,N),FNSDE2(N,N),DUM(N)
     4,PP(N),QEX(N),CPN(N)
      COMMON X,Y,XC,YC,DS,FN,FT,RHS,PI,CPI,CI,SI
     1,UINF,VINF,SDE
      SDE2=SPREAD(SDE,DIM=1,NCOPIES=N)
      FTSDE2=FT * SDE2
      FNSDE2=FN * SDE2
      QTS=SUM(ARRAY=FTSDE2,DIM=2)
      QNS=SUM(ARRAY=FNSDE2,DIM=2)
      QNK=QNS + VINF*CI -UINF*SI
      QTK=QTS + VINF*SI +UINF*CI
      UU=UINF - QNS * SI + QTS *CI
      VV=VINF + QNS * CI + QTS *SI
      PP=1. -UU*UU-VV*VV
      CPN=-PP/SQRT(1.0-FMN*FMN)
      DUM=B*B*XC
      DUM=YC*YC+DUM*DUM
      QEX=(1.0 + B)*YC/SQRT(DUM)
      WRITE(6,1) N
      WRITE(6,2) XC
      WRITE(6,3) N
      WRITE(6,2) CPN
    1 FORMAT(/,2X,'XC(I),I=1,',I5)
    2 FORMAT(2X,8F10.3)
    3 FORMAT(/,2X,'CPN(I),I=1,',I5)
      RETURN
      END
```

No. 9

EXPERIENCE WITH TRANSONIC FLOW IE COMPUTATIONS

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Elements in Fluid Dynamics
April 1992

INTERNATIONAL CONFERENCE ON COMPUTER
MODELLING OF SEAS AND COASTAL REGIONS AND BOUNDARY
ELEMENTS AND FLUID DYNAMICS
SOUTHAMPTON, U.K., APRIL 1992

Acknowledgement is made to K. Kitagawa *et al.* for use of Fig. 11, page 19, which appears on the front cover of this book.

Boundary Elements in Fluid Dynamics

Editors: C.A. Brebbia, Wessex Institute of Technology
P.W. Partridge, Wessex Institute of Technology

Computational Mechanics Publications
Southampton Boston

Co-published by

Elsevier Applied Science
London New York

Experience with Transonic Flow IE Computations

H. Hu

Department of Mathematics, Hampton University, Hampton, Virginia 23668, U.S.A.

ABSTRACT

An integral equation method based on the full-potential equation for transonic flow calculations is presented. The full-potential equation is written in the moving frame of reference; in the form of the Poisson's equation. The integral equation solution in terms of the velocity field is obtained by the Green's theorem. The mixed nature of the transonic flow is treated by a type- difference scheme. The numerical solutions are obtained by a time-marching (if unsteady flows), iterative procedure. The computational examples presented in the present paper include steady and unsteady, two-dimensional (airfoil) and three-dimensional (wing) flows. The method of combining the integral equation solution with the finite-volume Euler solution is also presented. Through studying the method and their computational examples, the capabilities and limitations of the transonic integral equation method are discussed. Finally, the needs for further research is addressed.

Key Words: integral equation, field/boundary elements, full-potential equation, transonic flow.

INTRODUCTION

Starting in 1970, a great deal of progress has been made in solving transonic flow by using the finite-difference method (FDM) and finite-volume method (FVM). Although the FDM and FVM are successful in dealing with transonic flows, there are several drawbacks associated with those methods. In the FDM and FVM, fine grid points are needed over a large computational domain. Moreover, there are major technical difficulties in generating suitable grids for complex three-dimensional aerodynamic configurations.

On the other hand, the integral equation method (IEM, or called Field- Boundary Element Method, Field-Panel Method) has several advantages over the FDM and FVM. The IEM involues evaluation of integrals, which is more accurate and simpler than the FDM and FVM. The IEM automatically satisfies the far-field boundary conditions and hence

only a small limited computational domain is needed. The IEM does not suffer from the artificial viscosity effects as compared to FDM and FVM for shock capturing in transonic flow computations. Moreover, the generation of the three-dimensional grid (field-elements) for complex configuration is not difficulty in the IEM, since the surface fitted grid is not required.

Because of these advantages of IEM, it is highly desirable to fully develop the IEM to treat transonic flows. Integral equation methods for transonic flows have been developed by several investigators[1-8]. The author and his co-workers have been devoted to the development of the IEM for steady and unsteady transonic airfoil and wing flow computations during the past several years[9-15]. In the present paper, the recent development along with the computational examples are presented. Through studying the method and the numerous computational examples, the capabilities and limitations of the transonic integral equation method are discussed. Finally, the needs for further research are addressed.

FORMULATION

Full-Potential Equation

In the space-fixed frame of reference, the continuity and momentum equations for unsteady, inviscid compressible flows with negligible body forces are given by

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\vec{V} = 0 \tag{1}$$

and

$$\rho\frac{D\vec{V}}{Dt} + \nabla p = 0 \tag{2}$$

respectively, where ρ is the density; p the pressure; $\vec{V}$ the absolute velocity; and t the time.

However for a general unsteady motion of a body, the governing equations are simple to solve if the moving (body-fixed) frame of reference formulation is used. This formulation does not require the grid-motion calculation since the grid is rigidly fixed in the frame of reference. In addition to the space-fixed frame of reference $OXYZ$, a moving frame of reference $oxyz$ is introduced as shown in Fig. 1. The moving frame of reference $oxyz$ is translating at a velocity of $\vec{V}_o(t)$ and rotating around a pivot point, $\vec{r}_p = (x_p, y_p, z_p)$, at an angular velocity of $\vec{\Omega}(t)$.

The relation for the absolute velocity, relative velocity ($\vec{V}_r$) and transformation velocity $\vec{V}_e = (\vec{V}_o + \vec{\Omega}\times(\vec{r}-\vec{r}_p))$ is given by

$$\vec{V} = \vec{V}_r + \vec{V}_e \tag{3}$$

where $\vec{r}$ is the position vector measured in the moving frame of reference.

The substantial derivative of a scalar quantity like ρ is related to its substantial derivative in the moving frame and to the local derivative in the moving frame by the equation

$$\frac{D\rho}{Dt} = \frac{D'\rho}{Dt} = \frac{\partial'\rho}{\partial t} + \vec{V}_r\cdot\nabla\rho \tag{4}$$

where the $(')$ refers to the derivative with respect to the moving frame of reference. Also, the substantial derivative of a vector quantity like $\vec{V}$ is realted to one in moving frame of reference by

$$\frac{D\vec{V}}{Dt} = \frac{D'\vec{V}}{Dt} + \vec{\Omega}\times\vec{V} = \frac{\partial'\vec{V}}{\partial t} + \vec{V}_r\cdot\nabla\vec{V} + \vec{\Omega}\times\vec{V} \tag{5}$$

By using Eqs. (3)-(5) and assuming that the flow is isentropic and irrotational, Equations (1) and (2) take the form in the moving frame of reference as follows:

$$\nabla^2\Phi = -\frac{\nabla\rho}{\rho}\cdot\vec{V}_r - \frac{1}{\rho}\frac{\partial'\rho}{\partial t} \tag{6}$$

and

$$\frac{\rho}{\rho_\infty} = \left\{1 - \frac{\kappa-1}{2a_\infty^2}\left[(\nabla\Phi)^2 + 2\left(\frac{\partial'\Phi}{\partial t}\right) - 2\nabla\Phi\cdot\vec{V}_e\right]\right\}^{\frac{1}{\kappa-1}} \tag{7}$$

where Φ is the absolute velocity potential given by $\vec{V} = \nabla\Phi = \nabla'\Phi$ and κ is the gas specific heat ratio. Equation (6) is the unsteady full-potential equation in the moving frame of reference with the density given by Eq. (7). After introducing the characteristic parameters of length (such as, airfoil chord length), speed of sound at infinity (a_∞) and density at infinity (ρ_∞) and defining the translation mach number as $M_o = |\vec{V}_o|/a_\infty$ (note that $M_o = M_\infty$, free-stream Mach number), Eqs. (6) and (7) takes dimensionless form as follows:

$$\nabla^2\Phi = G \tag{8}$$

with

$$G = G_1 + G_2 \tag{9}$$

$$G_1 = -\frac{\nabla\rho}{\rho}\cdot\vec{V}_r \tag{10}$$

$$G_2 = -\frac{1}{\rho}\frac{\partial'\rho}{\partial t} \tag{11}$$

and

$$\rho = \left\{1 + \frac{\kappa-1}{2}\left[-V_r^2 + (\vec{V}_o + \vec{V}_e)^2 - 2\left(\frac{\partial'\Phi}{\partial t}\right)\right]\right\}^{\frac{1}{\kappa-1}} \tag{12}$$

where G_1 is the compressibility and G_2 is the unsteadiness. It should be noticed that all the quantities in Eqs. (8)-(12) are dimensionless, although the same notation as ones used for dimensional quantities are used.

For steady flows, $\vec{\Omega}(t)$, G_2 and $\partial'\Phi/\partial t$ are set to be zero; and $\vec{V}_o(t)$ becomes time-independent, which is the negative of the free-stream velocity.

Boundary Conditions

The boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, wake kinematic condition and wake dynamic condition, which are described as follows:

$$\vec{V}_r \cdot \vec{n}_g = 0 \quad \text{on} \quad g(\vec{r}) = 0 \tag{13}$$

$$\Delta C_p \mid_{sp} = 0 \tag{14}$$

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r}) = 0 \quad \text{and} \quad w(\vec{r},t) = 0 \tag{15}$$

$$\frac{1}{|\nabla w|}\frac{\partial' w}{\partial t} + \vec{V}_r \cdot \vec{n}_w = 0 \quad \text{on} \quad w(\vec{r},t) = 0 \tag{16}$$

and

$$\Delta C_p = 0 \quad \text{on} \quad w(\vec{r},t) = 0 \tag{17}$$

where $\vec{n}$ is the surface unit normal vector; the subscripts g and w refer to the body (wing or airfoil) and wake surface of $g(\vec{r}) = 0$ and $w(\vec{r},t) = 0$, respectively; ΔC_p is the pressure jump across the surface; and the subscript sp refers to edge of separation.

Integral Equation Solution

By using the Green's theorem, the integral equation solution of Eq. (8) in terms of the relative velocity field is given by

$$\begin{aligned}\vec{V}_r(x,y,z,t) = &-\vec{V}_o(t) - \vec{\Omega}(t)\times(\vec{r}-\vec{r}_p)\\ &-\frac{1}{4\pi}\int\int_g \frac{q_g(\xi,\eta,\zeta,t)}{d^2}\vec{e}_d ds\\ &+\frac{1}{4\pi}\int\int_g \frac{\vec{\gamma}_g(\xi,\eta,\zeta,t)\times\vec{d}}{d^3} ds\\ &+\frac{1}{4\pi}\sum_{iw=1}^{NW}\int\int_w \frac{\vec{\gamma}_w(\xi,\eta,\zeta,t)\times\vec{d}}{d^3} ds\\ &+\frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta,t)}{d^2}\vec{e}_d d\xi d\eta d\zeta\\ &+\frac{1}{4\pi}\int\int_S \frac{q_S(\xi,\eta,\zeta,t)}{d^2}\vec{e}_d ds\end{aligned} \tag{18}$$

where q is the surface source distribution; $\vec{\gamma}$ is the surface vorticity distribution; the subscript S refers to the shock surface; the index NW is the total number of wake surfaces; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

It should also be noticed that Eq. (18) has been written for three-dimensional flows. For two-dimensional flows, above surface integrals become line integrals and the volume integrals become field surface integrals. The coordinates, z and ζ, are not used. The coefficients of $1/4\pi$ are replaced by $1/2\pi$ for two-dimensional flows. The last integral term in Eq. (18) is used only for the shock-fitting solutions in steady two-dimensional flows.

COMPUTATIONAL SCHEME

A sketch of the IE computational domains is shown in Fig.2 for three-dimensional flows. Due to the nature of the nonlinearity of the flow, the solutions are obtained through a iterative procedure where the compressibility (G_1), unsteadiness (G_2), and the wake shape and its strength are updated through each iteration. Once the solution converges, it marches to the next time step. The details of the solution procedure can be found in Ref. [9,14]. Here only the treatment of the mixed nature of the flow and its shock is described.

Type-Difference Scheme

Transonic flows are characterized by the presence of both subsonic and supersonic regions within the flow field simultaneously. Therefore, transonic flows are described by a mixed elliptic-hyperbolic partial differential equation with the boundary between them unknown apriori. To be consistent with the mixed nature of the transonic flow, the Murman-Cole type-difference scheme is applied to the present IE computation for $\nabla\rho$ calculations. For the subsonic points where the local Mach number is less than unit, central- differencing is used. For supersonic points where the local Mach number is greater than unit, backward-(upstream-) differencing is used. One exception is to use forward-differencing at the first elements after the shock discontinuity. This type-difference scheme is consistent with the nature of the transonic flow, because the local disturbance in a subsonic flow propagates in all directions while in a supersonic flow the local disturbance is confined to the downstream Mach cone of the disturbance.

Shcok-Fitting Technique

It should be mentioned that mathematically the fourth volume integral term of Eq. (18) includes all compressibility effects including shock discontinuity. Since a relative coarse grid has been used in the present IE computational domain, the contribution of the shock discontinuity is extracted from the fourth integral terms and it is represented explicitly by fifth integral term of Eq. (18). The strength of the shock panel (q_S) is equal to the difference of normal velocity across the shock. The slope of the shock panel is determined by the Rankine-Hugoniot relation. The technique to introduce the shock panel term, the fifth integral term of Eq. (18), and to fit the shock using Rankine-Hugoniot relation is called shock-fitting technique, as shown in Fig. 3. This shock-fitting technique is applied to steady 2-D computations.

NUMERICAL EXAMPLES

The integral equation method has been applied to steady and unsteady,

two- dimensional and three-dimensional transonic flows. The computational results along with the experimental data and other computational results are presented in the following sub-sections.

Steady Subsonic Airfoil Flow

The computational results for a steady compressible shock-free flow at high subsonic Mach number are presented here as the first numerical example. The purpose is to validate the IEM for nonlinear compressible flows. The surface pressure distribution[9,10] is shown in Figure 4, along with the finite-difference (FD) Euler solution[16]. In this case, the NACA0012 airfoil has been used at the flow condition of $M_\infty = 0.72$ and $\alpha = 0°$. The results show that the integral equation (IE) solutions for this case agree with the FD Euler solutions.

Steady Transonic Airfoil Flow

The second numerical example is a transonic flow case with shock of moderate strength[9,10]. Figure 5 shows the results for the flow around the NACA64A010A airfoil at $M_\infty = 0.796$ and $\alpha = 0°$. This airfoil has 10% thickness and a small camber. In this case, a computational domain of 2×1.5 airfoil chord length with 64×60 field-elements has been used. The number of iteration used to achieve convergence is about 25. This is much less than that used in FDM and FVM. The results show that the integral equation method can predict the shock correctly.

Unsteady Transonic Airfoil Flow

The third numerical example is the unsteady transonic airfoil flow case. In this case, the unsteady IEM has been applied to the NACA0012 airfoil at a translation Mach number of 0.755 undergoing forced pitching oscillation around a pivot point at the quarter-chord, measured from the leading edge ($\vec{r}_p = (0.25, 0.0)$). The angles of attack are given by: $\alpha(t) = 0.016° + 1.255° sin(0.1632t)$. Figure 6 shows the computed surface lifting coefficients[9,11] along with the finite-volume (FV) Euler solutions produced by Ref. [18]. The comparison shows a good agreement between IE solutions and FV Euler solutions. The shock strength and its motion are also predicted correctly (see Ref. [9,11]).

Unsteady Transonic Wing Flow

Recently, IEM has been applied to three-dimensional unsteady transonic flows. The unsteady transonic flow around a rectangular wing has been computed. In this numerical example, an zero-thickness rectangular wing with aspect ratio of 2 is given an forced transient pitching motion at a transonic translation Mach number of 0.7. To simplify the problem, only a pitching motion in xy-plane is considered; and therefore $\vec{\Omega}$ is given by $\vec{\Omega} = 0\vec{i} + 0\vec{j} + \alpha\vec{k}$, where α is given by $\alpha(t) = 5° + 0.5°t$ with $t = n\Delta t$. n is time-step index (from 1 to 10 in this case) and Δt is time-step size, which is chosen as 1. The half-span of the wing and wake is divided into 10×6 and 10×10 quadrilateral panels, respectively. Each quadrilateral panel consists of two triangle panels. The time history of the surface pressure distributions over the wing root section is shown in Figure 7. The results show that the IEM can capture the unsteady effect correctly, although finer field-elements may be required to narrow the shock region and to predict the shock motion as accurate as possible.

CAPABILITIES AND LIMITATIONS

The steady and unsteady integral equation methods for nonlinear compressible flows have been developed. The methods have been applied to steady airfoil, unsteady airfoil and unsteady wing flows with or without shocks. The comparison of the present solutions with experimental data and FD or FV solutions shows that the integral equation methods based on the linear theorem can handle nonlinear flow problems accurately. For transonic flows with shocks of weak to moderate strength, IEM predicts shocks correctly, with the exception of slight underprediction of the shock strength (Fig. 5). For unsteady flows, the motion of the shock agrees with that predicted by FV Euler solutions (see Ref. [9,11]), and the predicted lifting ceofficient agrees with one obtained by FV Euler computation (Fig. 6).

The advantages of the small computational domain and coarse grid have been utilized in the present IEM. For airfoil flow computations, a computational domain of 2×1.5 airfoil chord length with 64×60 field-elements has been used. The use of less number of field-elements with larger field- elements over the outer region inside the domain is possible. For wing flow computations, a computational domain of $2.3 \times 0.75 \times 1.5$ wing root chord length with $23 \times 9 \times 9$ field-elements has been used, although finer field-elements around the shock region may be required to accurately predict the shock location and its strength.

The numbers of iteration used for steady flow computations are about 25 for airfoil and 5 for wing flows, respectively. This is much less than those used in FD and FV computations (usually order of 10^3). For unsteady flows, the numbers of iteration used in each time step range from 1 to 3. Large time steps have also been used in the present unsteady flow computations. This is also the one of the advantages of IEM. For a whole cycle of pitching oscillation for example, a total of 36 time steps has been used; while a typical implicit FD or FV computation needs about 500 time steps for the same case. Therefore, IEM is nevertheless efficient in terms of the number of iteration and the time step size, as compared to existing FDM and FVM.

By examining the numerical examples presented here it is found that, on the other hand, all these computations are restricted to the flows with shocks of weak to moderate strength. As the best of the author's knowledge, all existing integral equation solutions based on potential flow formulation for transonic flows are restricted to flows without strong shocks. The potential flow assumption neglects the effects due to viscosity, vorticity and entropy production. For transonic flows with strong shocks and masive separation, the potential flow assumption is not an adequate approximation to the real flow.

In order to extend the integral equation method to a wide range of transonic flows, the effects of viscosity, vorticity and entropy production must be considered. The attempt has been made by some inveatigators by using the Navier-Stokes equations in integral equation formulation. On

the other hand, the idea of combining FVM for Euler equations (neglecting viscous effect) with IEM for full-potential equation has been developed[9,10]. In the following section, a brief description of the method with the computational examples will be given.

HYBRID IE-FV METHODS

A hybrid IE-FV method has been developed for both steady and unsteady transonic airfoil flow computations. In this method, unsteady Euler equations are solved in the small inner domain using a FVM, while the full-potential equation is solved over outer domain using IEM to update the outer boundary conditions of the Euler inner domain. A sketch of the computational domain for unsteady flow computation is shown in Fig.8. The method has been applied to both steady and unsteady transonic airfoil flows.

Figure 9 shows the computational domain and numerical results[9,10] for a typical strong shock flow case, the steady flow around NACA0012 airfoil at $M_\infty = 0.84$ and $\alpha(t) = 0°$. For steady flows as shown in Fig.9, the Euler computational domain is smaller than one for unsteady flows, since shock is fixed in location for steady flows. This case took 10 IE-iterations to locate the shock, 300 FV-iterations to achieve a residual error of 10^{-3} and 3 IE-iterations to update the Euler inner domain outer boundary conditions. The present solution compares very well with the FV Euler solution[19] both in strength and location of the shock (a strong shock !).

Figure 10 shows the recently computed lifting coefficients for a transient ramp motion of the NACA0012 airfoil at $M_\infty = 0.56$ and angles of attack given by: $\alpha(t) = -0.01° + 0.855°t$, along with the experimental data of Ref. [20]. The comparison shows a good agreement between the present solution and the experimental data at a range of low to moderate angles of attack (from 0° to 4.7°). As the angle of attack increases, the discrepancy between the present solution and the experimental data increases. This difference may be attributed to the absence of the viscous terms including the turbulence effects from the present inviscid formulation - Euler/potential formulation.

CONCLUDING REMARKS

The integral equation solutions of the full-potential equation for transonic flows have been developed. The numerical examples have been presented. Throught studying the methods and their numerical examples, the capabilities and limitations of the transonic integral equation methods based on full- potential formulation have been discussed. The combination of full-potential IE and Euler FV solutions has benn demonstrated for a strong shock flow case and an unsteady transonic flow case. Further works of extending the present transonic integral equation method may be focused on: (1) whole aircraft computations; (2) three-dimensional, general unsteady motion (including deformation) computations; (3) two-dimensional and three-dimensional, steady and unsteady flow computations by hybrid FV

Navier-Stokes/IE potential method.

ACKNOWLEDGEMENT

This research work has been partially supported by NASA Langley Research Center under the Grant No. NAG-1-1170. Dr. E. Carson Yates, Jr. is the technical monitor.

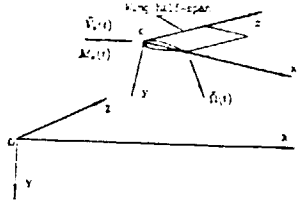


Figure 1. Frames of reference.

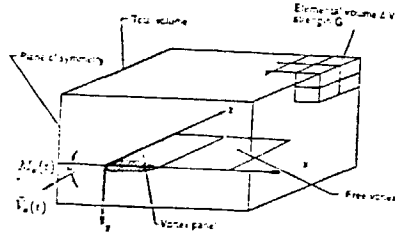


Figure 2. IE Computational domain.

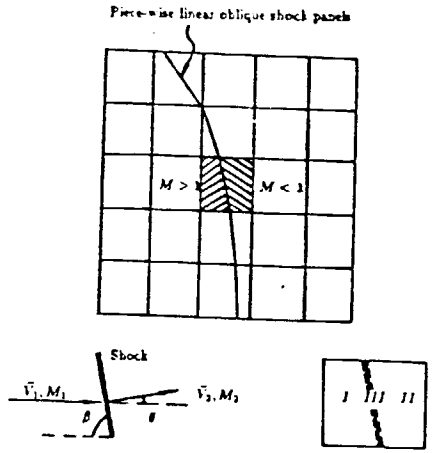


Figure 3. Illustration of shock panels and field-element splitting; Areas I and II: 4th integral term; Area III: 5th integral term of Eq. (18).

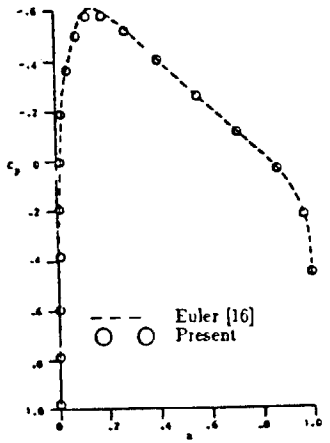


Figure 4. C_p distribution, NACA0012, $M_\infty = 0.72$, $\alpha = 0°$.

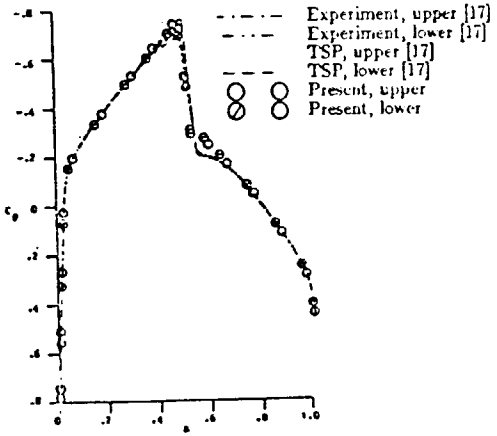


Figure 5. C_p distribution, NACA64A010A, $M_\infty = 0.796$, $\alpha = 0°$.

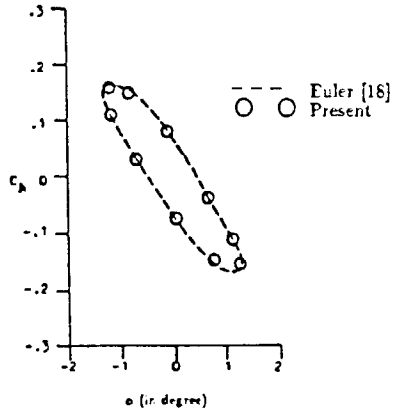


Figure 6. Lifting coefficients, pitching oscillation, $M_o = 0.755$, $\alpha(t) = 0.016° + 1.255° sin(0.1632t)$.

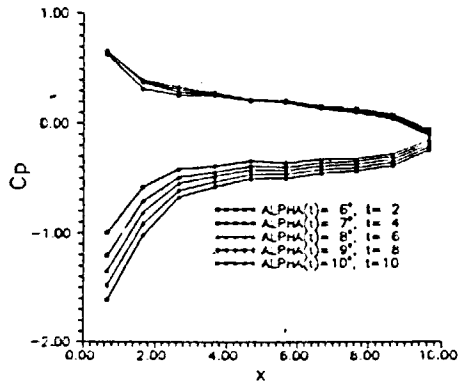


Figure 7. Wing root section C_L history, ramp motion, $M_o = 0.7$, $\alpha(t) = 5° + 0.5°t$.

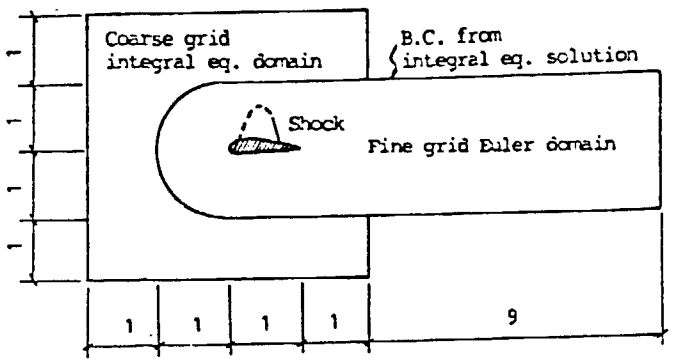


Figure 8. Hybrid IE-FV computational domain for unsteady flows.

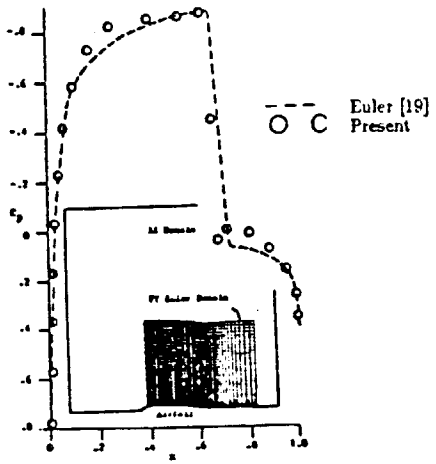


Figure 9. Euler domain and C_p distribution, a strong shock case, steady flow, $M_\infty = 0.84$, $\alpha = 0$.

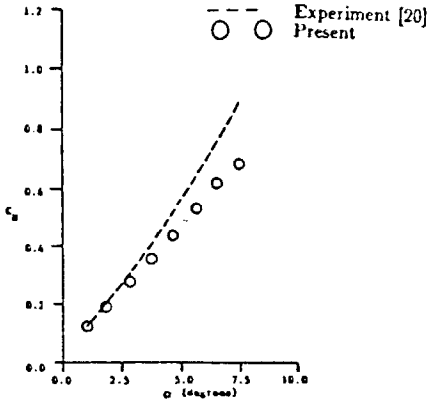


Figure 10. Lifting coefficients, ramp motion, $M_\infty = 0.56$, $\alpha(t) = -0.01° + 0.855°t$.

REFERENCES

1 Piers, W. J. and Sloof, J. W. Calculation of Transonic Flow by Means of a Shock-Capturing Field Panel Method, AIAA Paper 79-1459, 1979.

2 Hounjet, M. H. L. Transonic Panel Method to Determine Loads on Oscillating Airfoils with Shocks, *AIAA Journal*, Vol. 19, No. 5, pp.559-566, 1981.

3 Tseng, K. and Morino, L. Nonlinear Green's Function Methods for Unsteady Transonic Flows, in *Transonic Aerodynamics*, (Ed. Nixon, D), AIAA, pp.565-603, 1982.

4 Oskam, B. Transonic Panel Method for the Full Potential Equation Applied to Multicomponent Airfoils, *AIAA Journal*, Vol. 23, No. 9, pp.1327-1334, 1985.

5 Erickson, L. L. and Strande, S. M. A Theoretical Basis for Extending Surface-Paneling Methods To Transonic Flow, *AIAA Journal*, Vol.23, No. 12, pp.1860-1867, 1985.

6 Sinclair, P. M. An Exact Integral (Field Panel) Method for the Calculation of Two-Dimensional Transonic Potential Flow Around Complex Configurations, *Aeronautical Journal*, Vol. 90, No. 896, pp.227-236, 1986.

7 Kandil, O. A. and Yates, E. C., Jr. Computation of Transonic Vortex Flows Past Delta Wings-Integral Equation Approach, *AIAA Journal*, Vol. 24, No. 11, pp.1729-1736, 1986.

8 Ogana, W. Boundary Element Methods in Two-Dimensional Transonic

Flows, in *Boundary Elements IX, Vol.3: Fluid Flow and Potential Applications*, (Eds. Brebbia, C. A., Wendland, W. L. and Kuhu, G.), Computational Mechanics Publications, Springer-Verlag, pp.567-582, 1987.

9 Hu, H. Full-Potential Integral Solutions for Steady and Unsteady Transonic Airfoils with and without Embedded Euler Domains, Ph.D. Dissertation, Dept. of Mechanical Engineering and Mechanics,Old Dominion University, Norfolk, Virginia, March 1988.

10 Kandil, O. A. and Hu, H. Full-Potential Integral Solution for Transonic Flows with and without Embedded Euler Domains, *AIAA Journal*, Vol. 26, No. 9, pp.1074-1086, 1988.

11 Kandil, O. A. and Hu, H. Integral Solution of Unsteady Full- Potential Equation for a Transonic Pitching Airfoil, *Journal of Aircraft*, Vol. 27, No. 2, pp.123-130, 1990.

12 Hu, H. and Chu, L-C. Unsteady Three-Dimensional Transonic Flow Computations Using Field Element Method, in *Boundary Element Methods in Engineering*. (Eds. Annigeri, B. S. and Tseng, K.), Springer-Verlag, pp. 140-146, 1990.

13 Hu, H. and Kandil, O. A. Unsteady Field Element Method with Embedded Euler Domain for Transonic Pitching Airfoil Flows, in *Boundary Element Methods in Engineering*. (Eds. Annigeri, B. S. and Tseng, K.), Springer-Verlag, pp. 161-168, 1990.

14 Hu, H. Unsteady Transonic Wing Flow Computations Using Field- Boundary Element Methods, to appear in *Journal, Engineering Analysis with Boundary Elements.*

15 Hu, H. Application of Integral Equation Method to Flows Around a Wing with Circular-Arc Section, in *Boundary Elements XIII*, (Eds. Brebbia, C. A. and Gipson, G. S.), Computational Mechanics Publications, Elsevier Applied Science, pp. 209-217, 1991.

16 Sells, C. L. Plane Subcritical Flow Past a Lifting Airfoil, in *Proceedings of the Royal Society of London*, Series A, Vol. 308, No. 1494, pp.377-401, 1969.

17 Edwards, J. W., Bland, S. R. and Seidel, D. A. Experience with Transonic Unsteady Aerodynamic Calculations, NASA TM-86278, 1984.

18 Kandil, O. A. and Chuang, H. A. Unsteady Transonic Airfoil Computation Using Implicit Euler Scheme on Body-Fixed Grid, *AIAA Journal*, Vol. 27, No. 8, pp. 1031-1037, 1989.

19 Jameson, A., Schmidt, W. and Turkel, E. Numerical Solutions of the Euler Equations by Finite-Volume Methods Using Runge Kutta Time-Steping Scheme, AIAA Paper 81-1259, 1981.

20 London, R. H. NACA0012 Oscillatory and Transient Pitching, AGARD R- 702, pp.3.1-3.25, 1982.

No. 10

A 3D IEM FOR COMPRESSIBLE WING FLOWS WITH AND WITHOUT SHOCKS

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Elements in Fluid Dynamics
April 1992

INTERNATIONAL CONFERENCE ON COMPUTER MODELLING OF SEAS AND COASTAL REGIONS AND BOUNDARY ELEMENTS AND FLUID DYNAMICS
SOUTHAMPTON, U.K., APRIL 1992

Acknowledgement is made to K. Kitagawa *et al.* for use of Fig. 11, page 19, which appears on the front cover of this book.

Boundary Elements in Fluid Dynamics

Editors: C.A. Brebbia, Wessex Institute of Technology
P.W. Partridge, Wessex Institute of Technology

Computational Mechanics Publications
Southampton Boston

Co-published by

Elsevier Applied Science
London New York

A 3D IEM for Compressible Wing Flows With and Without Shocks

H. Hu

Department of Mathematics, Hampton University, Hampton, Virginia 23668, U.S.A.

ABSTRACT

An integral equation (or called field-panel, field-boundary element) scheme for solving the full-potential equation for incompressible and compressible flows with and without shocks has been developed. The full-potential equation has been written in the form of the Poisson's equation. Compressibility has been treated as non-homogeneity. The integral equation solution in terms of velocity field is obtained by the Green's theorem. The solution consists of wing (or a general body) surface (boundary elements) integral term(s) of vorticity/source distribution(s), wake surface (boundary elements) integral term(s) of free-vortex sheet(s), a volume (field-elements) integral term of compressibility over a small limited domain around the source of disturbance, and a shock surface (boundary elements) integral term of source distributions. Solution is obtained through an iterative procedure for non-linear compressible flows. To be consistent with the mixed-nature of transonic flows, the Murman-Cole type-difference scheme is used to compute the derivatives of the density. The present scheme is applied to flows around a rectangular wing with circular-arc section at incompressible, high- subsonic and transonic flow conditions.

Key Words: integral equation method, full-potential equation, subsonic and transonic wing flows.

INTRODUCTION

The finite-difference method (FDM) and finite-volume method (FVM) for solving transonic flows have been well developed during the past twenty years. Although the Navier-Stokes equation formulation for the transonic flow computations has been understood as the best model and the FDM and FVM are successful in dealing with transonic flows, the computation of the unsteady Navier-Stokes equations over complex three-dimensional configurations is very expensive, particularly for time-accurated unsteady flow computations. There are also major technical difficulties in FDM and FVM for generating suitable grids for complex three-dimensional aerodynamic configurations.

The experience has shown that rather accurate solutions can been obtained for many transonic flows using the inviscid modeling of the full-potential equation. For transonic flows without strong shocks and massive separations, the full-potential equation is an adequate approximation to the Navier-Stokes equations. The integral equation method (IEM) for the potential equation is an alternative to the FDM and FVM. Moreover, the IEM has several advantages over the FDM and FVM. The IEM involues evaluation of integrals, which is more accurate and simpler than the FDM and FVM, and hence a coarse grid (field-elements) can been used in IEM. The IEM automatically satisfies the far-field boundary conditions and therefore only a small limited computational domain is needed. The IEM does not suffer from the artificial viscosity effects as compared to FDM and FVM for shock capturing in transonic flow computations. The generation of the three-dimensional grid for complex configuration is not difficulty in the IEM, since the mapping from physical plane to computational plane is not required.

Integral equation methods for transonic flows have been developed by several investigators[1-18] during the past few years for steady airfoil, wing and aircraft configurations and unsteady airfoils and wings. In the present paper a method for computing general steady 3-D flows, which is an extension of 2D method of Ref.10, is presented along with simple numerical examples to demonstrate the capability and the potential of the present IE scheme for incompressible, subsonic and transonic flow computations. The shock-fitting technique is applied to the present transonic flow calculations, so that a coarse grid can be used.

FORMULATION

Governing Equations
The non-dimensional steady full-potential equation is given by:

$$\nabla^2\Phi = G \tag{1}$$

with

$$G = -\frac{\nabla\rho}{\rho}\cdot\vec{V} \tag{2}$$

and

$$\rho = [1 + \frac{\kappa-1}{2}(1-|\vec{V}|^2)]^{\frac{1}{\kappa-1}} \tag{3}$$

where the characteristic parameters, ρ_∞, a_∞ and l have been used; a is the speed of the sound, ρ the density, and l the wing surface panel length; and Φ is the velocity potential ($\vec{V} = \nabla\Phi$), G the compressibility, and κ the gas specific heat ratio.

Boundary Conditions
The boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, and wake kinematic and dynamic conditions. They are described as follows:

$$\vec{V}\cdot\vec{n}_g = 0 \qquad \text{on} \quad g(\vec{r}) = 0 \tag{4}$$

$$\Delta C_p|_{sp} = 0 \tag{5}$$

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r}) = 0 \quad \text{and} \quad w(\vec{r}) = 0 \tag{6}$$

$$\vec{V}\cdot\vec{n}_w = 0 \qquad \text{on} \quad w(\vec{r}) = 0 \tag{7}$$

and

$$\Delta C_p = 0 \qquad \text{on} \quad w(\vec{r}) = 0 \tag{8}$$

where $\vec{n}_g$ is the unit normal vector of the wing (or a general body) surface, $g(\vec{r}) = 0$; C_p is the surface pressure coefficient; the subscripts sp refers to the edges of separation; and $w(\vec{r}) = 0$ is wake surface(s).

IE Solution
By using the Green's theorem, the integral equation solution of Eq. (1) in terms of the velocity field is given by

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty &+ \frac{1}{4\pi}\int\int_g \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\int\int_g \frac{\vec{\gamma}_g(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{nw=1}^{NW}\int\int_w \frac{\vec{\gamma}_w(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,d\xi\,d\eta\,d\zeta \\ &+ \frac{1}{4\pi}\int\int_S \frac{q_S(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,ds(\xi,\eta,\zeta)\end{aligned} \tag{9}$$

where $\vec{V}_\infty$ is the free-stream velocity; q is the surface source distribution; $\vec{\gamma}$ is the surface vorticity distribution; the subscript, S, refers to the shock surface; NW is the total number of edges of separation; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i}+(y-\eta)\vec{j}+(z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$. It can be seen that the infinity condition, Eq. (6), is automatically satisfied by the integral equation solution.

COMPUTATIONAL SCHEME

Discretisation
In the present paper, three numerical examples (incompressible, high- subsonic and transonic flows as mentioned early) are considered. To simplify the problem, the computations are made for symmetric (no-lifting) flows only. As a consequence, the second and the third integrals in Eq. (9) are not used. In this computational model, the wing surface is represented by a number of rectangular source panels. A uniform rectangular parallelopiped type of volume elements are used throughout the flow field. A constant surface and volume source (q and G) distributions are used over

small wing and shock surface panels and over small volume elements. The discretized integral equation solution then becomes

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG} q_{g_{i,k}} \int\int_{g_{i,k}} \frac{1}{d^2}\vec{\epsilon}_d ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV} G_{i,j,k} \int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{\epsilon}_d d\xi d\eta d\zeta \\ &+ \frac{1}{4\pi}\sum_{j=1}^{MS}\sum_{k=1}^{NS} q_{S_{j,k}} \int\int_{S_{j,k}} \frac{1}{d^2}\vec{\epsilon}_d ds(\xi,\eta,\zeta)\end{aligned} \tag{10}$$

where the indices, i, j and k refer to the surface panels and field elements; $LG \times NG$ is the total number of wing surface panels; $LV \times MV \times NV$ is the total number of field elements; and $MS \times NS$ is the total number of shock surface panels.

Iterative Scheme

Due to the nature of the non-linearity of high-subsonic and transonic flows, the solutions are obtained through an iterative procedure, where the wing surface source strength and the compressibility are updated through each iteration. The solution procedure follows the successful form of Ref. 10 of two-dimensional computations. Here only the treatment of shocks for transonic flows and their associate mixed nature of flows are described.

Type-Difference Scheme

Transonic flows are characterized by the presence of both subsonic and supersonic regions within the flow field simultaneously. Therefore, transonic flows are described by a mixed elliptic-hyperbolic partial differential equation with the boundary between them unknown apriori. To bo consistent with the mixed nature of the transonic flow, the Murman-Cole type-difference scheme is applied to the present IE computation for $\nabla\rho$ calculations. For subsonic points where the local Mach number is less than unit, central-differencing is used. For supersonic points where the local Mach number is greater than unit, backward-(upstream-) differencing is used. One exception is to use forward-(downstream-) differencing at the first elements after the shock discontinuity. This type-difference scheme is consistent with the nature of the transonic flow, because the local disturbance in a subsonic flow propagates in all directions while in a supersonic flow the local disturbance is confined to the downstream Mach cone of the disturbance.

Shock-Fitting Technique

It should be mentioned that mathematically the fourth (volume) integral term of Eq. (9) includes all compressibility effects including shock discontinuity. Since a relative coarse grid (volume elements) has been used in the present IE computational domain, the contribution of the shock discontinuity is extracted from the forth integral term and it is represented explicitly by the fifth (surface) integral term of Eq. (9). The strength of

shock panels, q_S,is equal to the difference of normal velocity across the shock. This can be shown by integrating Eq. (1) over an infinitesimal volume around an infinitesimal area of the shock surface and applying the divergence theorem, one gets

$$\Delta(\nabla\Phi) = V_{2n} - V_{1n} = G\epsilon \tag{11}$$

where ϵ is the infinitesimal thickness normal to the shock surface. By letting $G\epsilon = q_S$ and using Rankine-Hugoniot relation, one finally obtains

$$q_S = \left[\frac{(\kappa-1)M_{1n}^2+2}{(\kappa+1)M_{1n}^2} - 1\right]V_{1n} \tag{12}$$

where the subscripts 1 and 2 refer to the conditions ahead and behind of the shock, respectively; and the subscript n refers to the normal component to the shock. The technique to introduce the shock panel and to fit the shock is called shock-fitting technique. In the present calculations for the simplicity, the less accurate, lumped shock panels are used and these shock panels are placed at interface of two adjacent volume elements where the local Mach number changes from greater than unit to smaller than unit.

NUMERICAL EXAMPLES

The presesnt scheme has been applied to a rectangular wing of aspect ratio of 3 with a 5%-thick symmetric circular-arc section. The half-span of the wing (including upper and lower surfaces) is divided into 40×6 quadrilateral panels. The one-half of the computational domain is divided into $40 \times 16 \times 9$ field volume elements in x, y and z directions, respectively. Only symmetric flows with zero angle of attack (non-lifting flows) are considered, and attached flow assumption is made also.

The first numerical example is made for the flow at zero Mach number, an incompressible flow, to validate the computational code. The calculated wing surface pressure distributions are presented in Figure 1. Due to the lack of experimental data for this case, a 2D result calculated by a 2D panel method computational code[19] is plotted also in the figure. The present 3D result at root section (0% semi-span station) compares well with the 2D result.

The second numerical example is made for the flow at free-stream Mach number of 0.7, a shock-free flow. The present calculated surface pressure distributions are presented in Figure 2, along with the experimental data of Ref. 20. In Ref. 20, experimental investigations were made in the supersonic wind tunnel at NASA Ames Research Center. In this paper, the calculated wing surface pressure distributions are compared with the experimental data of Ref. 20 at three span stations located at 0%, 50% and 90% semi-span stations. As the figure shows, the calculated pressure distributions are in qualitative agreement with experimental data, but the calculated pressure level are slightly lower than experimental data at 50% semi-span station. This difference may caused by an effective increase in model thickness due to the boundary layer in the experimental

test for which the Reynolds number is 4.0×10^6, as mentioned in Ref.21. The big difference between the calculated values and the experimental data is also observed at the 90% semi-span station, as shown in Figure 2. This difference is mainly due to the fact that in the present computational model the wing tip region is tapered into zero thickness starting at 83% semi-span station and therefore it accelerates the flow in the spanwise direction. In fact the calculated pressure distribution plotted at 90% semi-span station is made at 92% semi-span station where the computational wing model has about zero thickness. Moreover, the difference between the calculated values and the experimental data at 90% semi-span station becomes smaller when the tapering of the computational model wing section thickness starts at 92% semi-span station (the results are not shown here). Although the calculated pressure distributions are not in very good agreement with the experimental data, they are in very good agreement with the computational results obtained by SOUSSA[21] panel method code (see Figure 4 of Ref.21) and by LTRAN3[22] TSD finite- difference method code (see Figure 2 of Ref.22) at 0% ,50% and 70% semi-span station (the comparison is not shown here).

The third numerical example is made for a transonic flow with shock at a free-stream Mach number of 0.9. The calculated pressure distribution compared with the experimental data of Ref.20 at 0%, 50% and 90% semi-span stations, as shown in Figure 3. The set of curves at 50% semi-span station compares well, with the exception that the level of the calculated pressure distribution and the shock strength are lower than the experiment. This is partly due to an effective increase in model thickness in the experiment for which the Reynolds number is 4.4×10^6 as mentioned before, and partly due to the use of the less accurate, or over-simplified, lumped shock panels (in fact they are source points) in the present transonic flow calculation. The difference found at 0% semi-span station can be attributed mainly to the effects of the boundary layer on the wall where the test model was mounted (see Figure 1(a) of Ref. 20).

CONCLUDING REMARKS

An integral equation scheme based on the full-potential equation formulation for transonic flows has been developed. The scheme is capable of handling flows around general three dimensional configurations, although only simple cases are tested in the present paper. The calculated wing surface pressure distribution is reasonably correct including the location and the strength of the shock when using the present simplified computational model. The transonic flow computational results may be improved by using more accurate, distributed shock panels and putting the shock panels in more accurate locations with correct orientations using the Rankine-Hugoniot relations like what was done for airfoil calculations[10]. For the present shock-free flow case, only 6 iterations are used to get a convergent solution, while for the present transonic flow case, 11 iterations are used. The present IEM is effective in terms of the number of iterations compared with those of FDM and FVM, althought the computational cost per IE iteration is more expensive than those of FDM and FVM. Currently, the work has been focused on the improvement of the transonic

flow calculations, and in the near furture effects of lifting, unsteadiness and separations will be considered.

ACKNOWLEDGEMENTS

This work has been supported by NASA Langley Research Center under the Grant No.: NAG-1-1170. Dr. Carson Yates, Jr. is the technical monitor. The discussions with Dr. Carson Yates, Jr. are very helpful during the entire course of this research and are gratefully acknowledged.

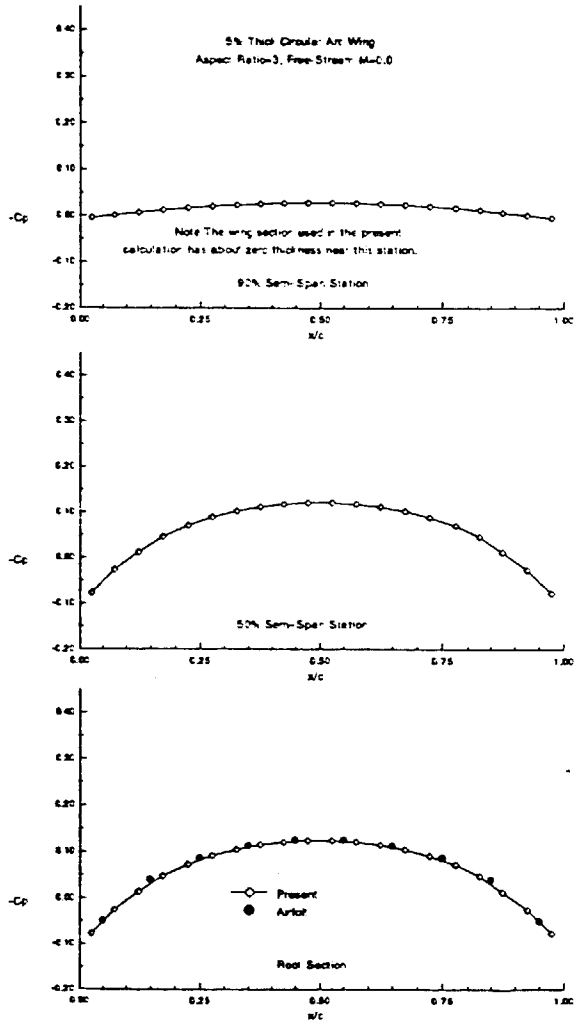


Figure 1. Pressure coefficients for an incompressible flow.

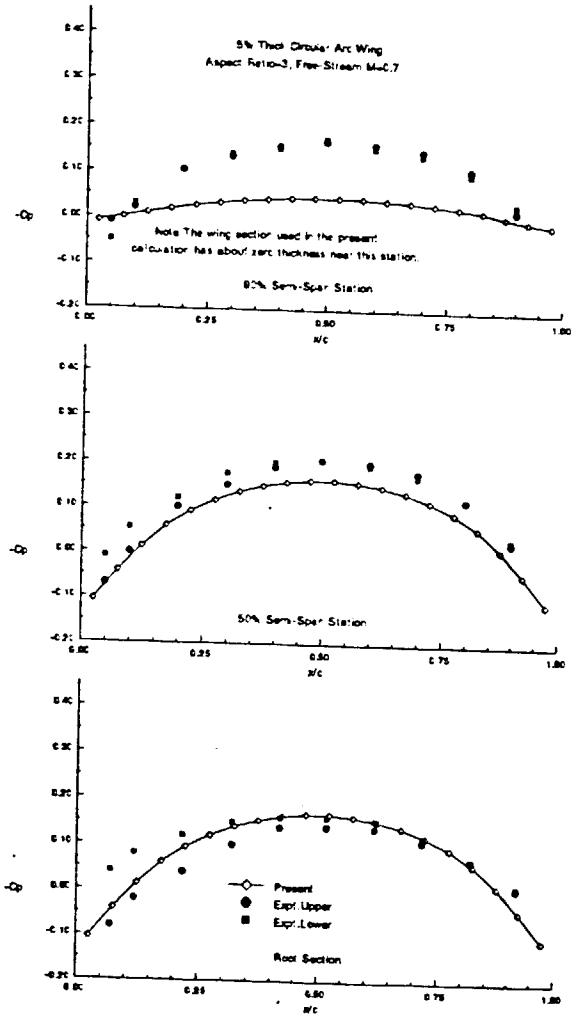


Figure 2. Pressure coefficients for a high-subsonic flow.

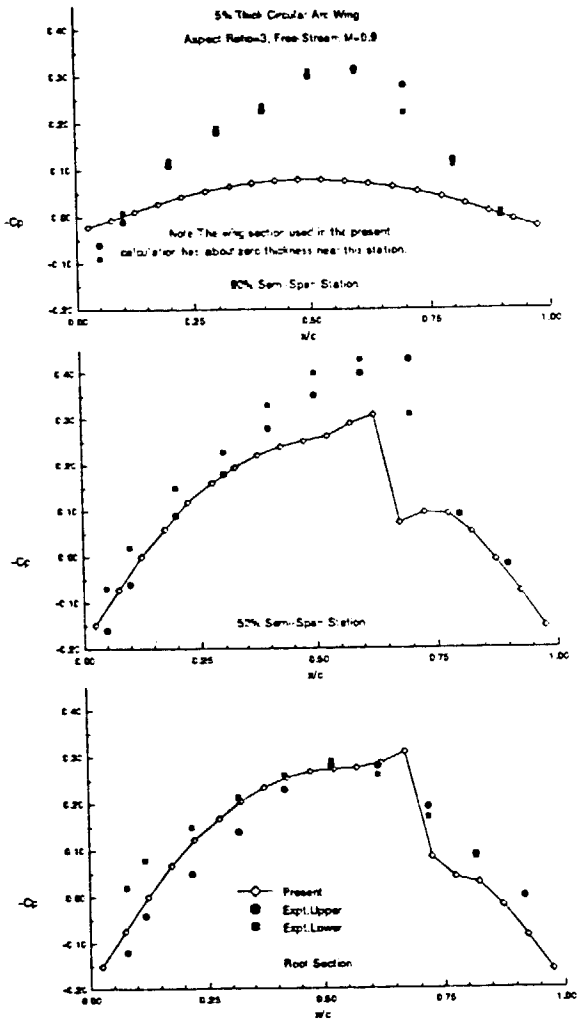


Figure 3. Pressure coefficients for a transonic flow.

REFERENCES

1. Piers, W. J. and Sloof, J. W. Calculation of transonic flow by means of a shock-capturing field panel method, AIAA Paper 79-1459, 1979.

2. Hounjet, M. H. L. Transonic panel method to determine loads on oscillating airfoils with shocks, *AIAA Journal*, 1981, Vol. 19, No. 5, pp.559-566.

3. Tseng, K. and Morino, L. Nonlinear Green's function methods for unsteady transonic flows, *Transonic Aerodynamics*, (Ed. Nixon, D), 1982, pp.565-603.

4. Oskam, B. Transonic panel method for the full potential equation applied to multicomponent airfoils, *AIAA Journal*, 1985, Vol. 23, No. 9, pp.1327-1334.

5. Erickson, L. L. and Strande, S. M. A theoretical basis for extending surface-paneling methods to transonic flow, *AIAA Journal*, 1985, Vol.23, No. 12, pp.1860-1867.

6. Sinclair, P. M. An exact integral (field panel) method for the calculation of two-dimensional transonic potential flow around complex configurations, *Aeronautical Journal*, June/July 1986, Vol. 90, No. 896, pp.227-236.

7. Kandil, O. A. and Yates, E. C., Jr. Computation of transonic vortex flows past delta wings-integral equation approach, *AIAA Journal*, 1986, Vol. 24, No. 11, pp.1729-1736.

8. Madson, M. D. Transonic analysis of the F-16A with under-wing fuel tanks: an application of the TranAir pull-potential code, AIAA Paper 87-1198, 1987.

9. Kandil, O. A. and Hu, H. Transonic airfoil computation using the integral equation with and without embedded Euler domains, *Boundary Elements IX*, Vol. 3, (Eds.: Brebbia, C. A., Wendland, W. L. and Kuhn, G.), Computational Mechanics Publications, Springer-Verlag, 1987, pp. 553-566.

10. Kandil, O. A. and Hu, H. Full-potential integral solution for transonic flows with and without embedded Euler domains, *AIAA Journal*, 1988, Vol. 26, No. 9, pp. 1074-1086.

11. Chu, L-C. Integral equation solution of the full potential equation for three-dimensional, steady, transonic wing flows, Ph.D. dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March 1988.

12. Sinclair, P. M. A three-dimensional field-integral method for the calculation of transonic flow on complex configurations - theory and preliminary results, *Aeronautical Journal*, June/July 1988, Vol. 92, No. 916, pp. 235-241.

13. Kandil, O. A. and Hu, H. Unsteady transonic airfoil computation using the integral solution of full-potential equation, *Boundary Elements X*, Vol. 2, (Ed.: Brebbia, C. A.). Computational Mechanics Publications, Springer-Verlag, 1988, pp. 352-371.

14. Kandil, O. A. and Hu, H. Integral solution of unsteady full- potential equation for a transonic pitching airfoil, *Journal of Aircraft*, 1990, Vol. 27, No. 2, pp. 123-130.

15. Ogana, W. Transonic integro-differential and integral equations with artificial viscosity, *Engineering Analysis with Boundary Elements*, 1989, Vol. 6, No. 3, pp. 129-135.

16. Hu. H. Field-boundary element solution for a transonic pitching wing, *Topics in Engineering, Vol. 7: Advances in Boundary Elements Methods in Japan and USA*, (Eds.: Tanaka, M., Brebbia, C. A. and Shaw, R.), Computational Mechanics Publication, 1990, pp. 295-305.

17. Hu, H. Field-boundary element computation for transonic flows, *Computation Engineering with Boundary Elements*, Vol. 1, (Eds.: Grilli, S., Brebbia, C. A. and Cheng, A. H-D.), Computational Mechanics Publication, 1990, pp. 319-330.

18. Hu, H. Unsteady transonic wing flow computations using field- boundary element methods, to appear in the *Journal, Engineering Analysis with Boundary Elements*.

19. Hu, H. and Simms, A. L. (work in progress.)

20. Lessing, H. C., Troutman, J. L. and Menees, G. P. Experimental determination of the pressure distribution on a rectangular wing oscillating in the first bending mode for Mach numbers from 0.24 to 1.30, NASA TN D-344, 1960.

21. Yates, E . C. Jr., Cunningham, H. J. and Desmarais, R. N. Subsonic aerodynamic and Flutter characteristics of several wings calculated by the SOUSSA P1.1 panel method, AIAA Paper 82-0727, 1982.

22. Guruswamy, P. and Goorjian, P. M. Comparisons between computations and experimental data in unsteady three-dimensional transonic aerodynamics, including aeroelastic applications, AIAA Paper 82-0690-CP, 1982.

No. 11

APPLICATION OF THE INTEGRAL EQUATION METHOD TO FLOWS AROUND A WING WITH CIRCULAR-ARC SECTION

Hong Hu
Department of Mathematics
Hampton University
Hampton, Virginia 23668

Published in
Boundary Elements XIII
August 1991

THIRTEENTH INTERNATIONAL CONFERENCE
ON
BOUNDARY ELEMENT METHODS
BEM 13
1991
SCIENTIFIC COMMITTEE

C.A. Brebbia
G.S. Gipson
E. Alarcon
M.H. Aliabadi
D.E. Beskos
M.B. Bush
D.J. Cartwright
A. Chaudouet-Miranda
A.H.-D. Cheng
R. Ciskowski
J.J. Connor
S. Crouch
T. Cruse
J. Dominguez
G. Qu
L.J. Gray
S. Grilli
W.S. Hall
T. Hromadka
M.A. Jaswon
N. Kamiya
D. Karabalis
P.-F. Liu
S. Mukherjee
G. Novati
K. Onishi
P.D. Panagiotoupolous
F. Paris
H. Pina
G.F. Pinder
H. Power
T.J. Rudolphi
J.J. Rencis
A.P.S. Salvadurai
S. Saigal
R.P. Shaw
P. Skerget
E.P. Stephan
G. Symm
M. Tanaka
J.C.F. Telles
N. Tosaka
M. Vable
W.S. Venturini
J.O. Watson
L.C. Wrobel

Boundary Elements XIII

Editors: C.A. Brebbia, Wessex Institute of Technology, U.K.
G.S. Gipson, Oklahoma State University, U.S.A.

Computational Mechanics Publications
Southampton Boston

Co-published with

Elsevier Applied Science
London New York

Application of the Integral Equation Method to Flows Around a Wing with Circular-Arc Section

H. Hu

Department of Mathematics, Hampton University, Hampton, Virginia 23668, U.S.A.

ABSTRACT

An integral equation (or called field-panel, field-boundary element) scheme for solving the full-potential equation for transonic flows has been developed. The full-potential equation has been written in the form of the Poisson's equation. Compressibility has been treated as non-homogeneity. The integral equation solution in terms of velocity field is obtained by the Green's theorem. The solution consists of surface (boundary elements) integral term(s) of vorticity/source distribution(s), wake surface (boundary elements) integral term(s) of free-vortex sheet(s) and a volume (field-elements) integral term of compressibility over a small limited domain around the source of disturbance. Solution procedure is an iterative procedure for non-linear flows. To consist with the mixed-nature of transonic flows, the Murman-Cole type-difference scheme is used to compute the derivatives of the density for non-linear flows. The present scheme is applied to flows around a rectangular wing with circular-arc section.

Key Words: integral equation method, full-potential equation, subsonic and transonic wing flows.

NOTATION

a	speed of sound
C_p	surface pressure coefficient
$\vec{d}$	distance vector pointed from sender to receiver
ds	infinitesimal surface area
$\vec{e}_d$	unit vector of d
g	wing surface
G	compressibility
l	wing surface panel length
M_∞	free-stream Mach number
$\vec{n}$	surface normal unit vector
q	surface source distribution
$\vec{V}$	field velocity vector
$\vec{V}_\infty$	free-stream velocity vector

w	wake surface
α	angle of attack
$\vec{\gamma}$	surface vorticity distribution
κ	gas specific heat ratio
ρ	density
Φ	velocity potential

INTRODUCTION

The finite-difference (FD) and finite-volume (FV) methods for solving transonic flows have been well developed during the past twenty years. Although the Navier-Stokes equation formulation for the transonic flow computations has been understood as the best model and the FD and FV methods are successful in dealing with transonic flows, the computation of the unsteady Navier-Stokes equations over complex three-dimensional configurations is very expensive, particularly for time-accurated unsteady flow computations. There are also major technical difficulties in FDM and FVM for generating suitable grids for complex three-dimensional aerodynamic configurations.

The experience has shown that rather accurate solutions can been obtained for many transonic flows using the inviscid modeling of the full-potential equation. For transonic flows without strong shocks and massive separations, the full-potential equation is an adequate approximation to the Navier-Stokes equations. The integral equation method (IEM) for the potential equation is an alternative to the FDM and FVM. Moreover, the IEM has several advantages over the FDM and FVM. The IEM involves evaluation of integrals, which is more accurate and simpler than the FDM and FVM, and hence a coarse grid (field-elements) can been used in IEM. The IEM automatically satisfies the far-field boundary conditions and therefore only a small limited computational domain is needed. The IEM does not suffer from the artificial viscosity effects as compared to FDM and FVM for shock capturing in transonic flow computations. The generation of the three-dimensional grid for complex configuration is not difficult in the IEM, since the mapping from physical plane to computational plane is not required.

Integral equation methods for transonic flows have been developed by several investigators[1-18] during the past few years for steady airfoil, wing and aircraft configurations and unsteady airfoils and wings. In the present paper, a method for computing general 3-D flows is presented along with a simple numerical example, as an initial stage of a research program to develop a general integral equation method for steady and unsteady subsonic and transonic flows around complex configurations.

FORMULATION

Governing Equations

The non-dimensional steady full-potential equation is given by:

$$\nabla^2\Phi = G \tag{1}$$

with

$$G_1 = -\frac{\nabla\rho}{\rho}\cdot\vec{V} \tag{2}$$

and

$$\rho = [1+\frac{\kappa-1}{2}(1-|\vec{V}|^2)]^{\frac{1}{\kappa-1}} \tag{3}$$

where the characteristic parameters, ρ_∞, a_∞ and l have been used; a is the speed of the sound, ρ the density, and l the wing surface panel length; and Φ is the velocity potential ($\vec{V}=\nabla\Phi$), G the compressibility, and κ the gas specific heat ratio.

Boundary Conditions

The boundary conditions are surface no-penetration condition, Kutta condition, infinity condition, and wake kinematic and dynamic conditions. They are described as follows:

$$\vec{V}\cdot\vec{n}_g = 0 \qquad \text{on} \quad g(\vec{r})=0 \tag{4}$$

$$\Delta C_p|_{sp} = 0 \tag{5}$$

$$\nabla\Phi \to 0 \quad \text{away from} \quad g(\vec{r})=0 \quad \text{and} \quad w(\vec{r})=0 \tag{6}$$

$$\vec{V}\cdot\vec{n}_w = 0 \qquad \text{on} \quad w(\vec{r})=0 \tag{7}$$

and

$$\Delta C_p = 0 \qquad \text{on} \quad w(\vec{r})=0 \tag{8}$$

where $\vec{n}_g$ is the unit normal vector of the wing (or a general body) surface, $g(\vec{r})=0$; C_p is the surface pressure coefficient; the subscripts sp refers to the edges of separation; and $w(\vec{r})=0$ is wake surface(s). It should be noticed that the infinity condition, Eq. (6), is automatically satisfied by the integral equation solution.

IE Solution

By using the Green's theorem, the integral equation solution of Eq. (1) in terms of the velocity field is given by

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty &+ \frac{1}{4\pi}\int\int_g \frac{q_g(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,ds(\xi,\eta,\zeta)\\ &+ \frac{1}{4\pi}\int\int_g \frac{\vec{\gamma}_g(\xi,\eta,\zeta)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta)\\ &+ \frac{1}{4\pi}\sum_{nw=1}^{NW}\int\int_w \frac{\vec{\gamma}_w(\xi,\eta,\zeta,t)\times\vec{d}}{d^3}ds(\xi,\eta,\zeta)\\ &+ \frac{1}{4\pi}\int\int\int_V \frac{G(\xi,\eta,\zeta)}{d^2}\vec{e}_d\,d\xi\,d\eta\,d\zeta\end{aligned} \tag{9}$$

where $\vec{V}_\infty$ is the free-stream velocity; q is the surface source distribution; $\vec{\gamma}$ is the surface vorticity distribution; NW is the total number of edges of separation; ds is the infinitesimal surface area; the vector $\vec{d}$ is given by $\vec{d} = (x-\xi)\vec{i} + (y-\eta)\vec{j} + (z-\zeta)\vec{k}$; and $\vec{e}_d$ is defined by $\vec{e}_d = \vec{d}/|\vec{d}|$.

COMPUTATIONAL SCHEME

Discretisation

The wing and its wake are represented by triangular vortex panels. A uniform rectangular parallelopiped type of volume elements are used throughout the flow field. A linear $\vec{\gamma}$-distribution is used over small surface panel, while a constant G-distribution is used over small field volume-element. The discretized integral equation solution becomes

$$\begin{aligned}\vec{V}(x,y,z) = \vec{V}_\infty &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{q_{g_{i,k}}(\xi,\eta,\zeta)}{d^2}\vec{e}_d ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LG}\sum_{k=1}^{NG}\int\int_{g_{i,k}} \frac{\vec{\gamma}_{g_{i,k}}(\xi,\eta,\zeta,t)\times\vec{d}}{d^3} ds(\xi,\eta,\zeta) \\ &+ \frac{1}{4\pi}\sum_{nw=1}^{NW}\sum_{i=1}^{LW_{nw}}\sum_{k=1}^{MW_{nw}}\int\int_{w_{i,k}} \frac{\vec{\gamma}_{w_{i,k}}(\xi,\eta,\zeta,t)\times\vec{d}}{d^3} ds(\xi,\eta,\zeta,t) \\ &+ \frac{1}{4\pi}\sum_{i=1}^{LV}\sum_{j=1}^{MV}\sum_{k=1}^{NV} G_{i,j,k}\int\int\int_{V_{i,j,k}} \frac{1}{d^2}\vec{e}_d d\xi d\eta d\zeta\end{aligned} \tag{10}$$

with

$$\begin{aligned}\vec{\gamma}\times\vec{d} = &[\gamma_y(z-\zeta) - \gamma_z(y-\eta)]\vec{i} \\ &+ [\gamma_z(x-\xi) - \gamma_x(z-\zeta)]\vec{j} \\ &+ [\gamma_x(y-\eta) - \gamma_y(x-\xi)]\vec{k}\end{aligned} \tag{11}$$

where the linear distributed $(\gamma_x, \gamma_y, \gamma_z)$ are three components of $\vec{\gamma}$; the indices, i, j and k refer to the surface panels and field elements; $LG \times NG$ is the total number of wing (or a general body) surface panels; $LW \times MW$ is the total number of wake surface panels; and $LV \times MV \times NV$ is the total number of field elements.

Iterative Scheme

Due to the nature of the nonlinearity of transonic flows, the solutions are obtained through a iterative procedure, where the compressibility, G, and the wake shape and its strength are updated within each iteration. The solution procedure follows the successful forms of Ref. 10 and Ref. 18, hence only a brief description is given:

Equations (10) along with the boundary conditions, Eqs. (4), (5), (7) and (8), are solved iteratively until the solution converges. Here, two loops are used. The inner loop is used to calculate and check the convergence

of the non-linear term, G. Here, G is set to be zero first to perform the incompressible flow calculations. The outer loop is used to update and check the convergence of the wake shape and wing surface pressure distribution. For transonic flows, the Murman-Cole type difference scheme is used. In subsonic flow region a central-differencing is used, while in supersonic flow region, a backward-differencing is used.

NUMERICAL EXAMPLES

The presesnt scheme has been applied to a rectangular wing of aspect ratio of 4 with a 6%-thick circular-arc section. The half-span of the wing (including upper and lower surfaces) is divided into 20×12 quadrilateral panels. Each quadrilateral panel consists of 2 triangular panels. The one-half of the computational domain is divided into $23 \times 9 \times 15$ field volume elements in x, y and z directions, respectively.

To simplify the problem in this initial stage of the research program, a symmetric flow with zero angle of attack is considered, and no separation has been considered. A transonic flow case (at $M_\infty = 0.908$) has been tested and the results (Figures 1-2) shew that the shock predicted is not as strong as one expected (see Ref. 19). This is possibly due to the large grid size ($\Delta x = 0.1$ chord length) used. A successive grid refinement technique is being applied in the region around the shock to sharpen the shock. The further results will be presented in the near furture.

Figures 3-4 shows the present computational results for the incompressible flow around the same wing at zero angle of attack. The results look quite correct.

CONCLUDING REMARKS

An integral equation scheme based on the full-potential equation formulation for transonic flows has been developed. The scheme is capable of handling flows around general three dimensional configurations, although only a simple case is tested in the present paper. It is necessary to emphasize that the result presented here is just an preliminarly result in this initial stage of a research program of developing a general steady and unsteady subsonic and transonic IEM for flows around complex 3-D configurations. Presently, the research has been focused on transonic flow computations, and in the near furture effects of unsteadiness and wake(s) (separation(s)) will be added.

ACKNOWLEDGEMENTS

This work has been supported by NASA Langley Research Center under the Grant No.: NAG-1-1170. Dr. Carson Yates, Jr. is the technical monitor. The valuable discussions with Dr. Carson Yates, Jr. at NASA-LaRC and Dr. L-C Chu at Lockheed are gratefully acknowledged.

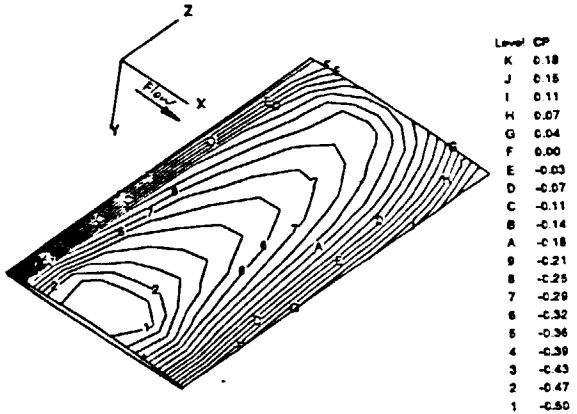


Figure 1. C_p contour, rectangular wing, $AR = 4$, 6%-thick circular-arc section, $\alpha = 0$, $M_\infty = 0.908$.

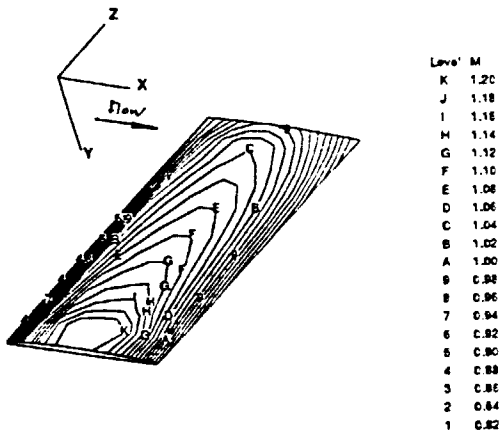


Figure 2. Mach contour, same case as Fig. 1.

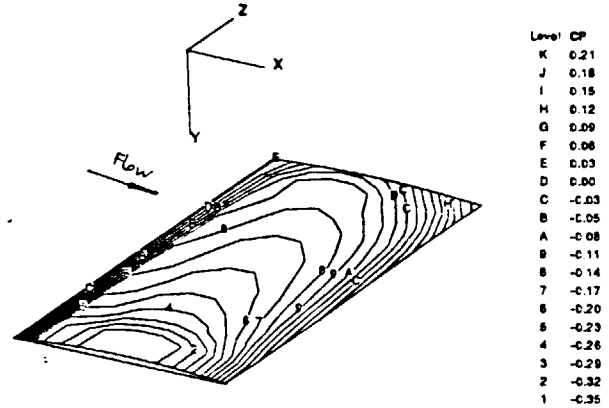


Figure 3. C_p contour, same case as Fig.1, but $M_\infty = 0$.

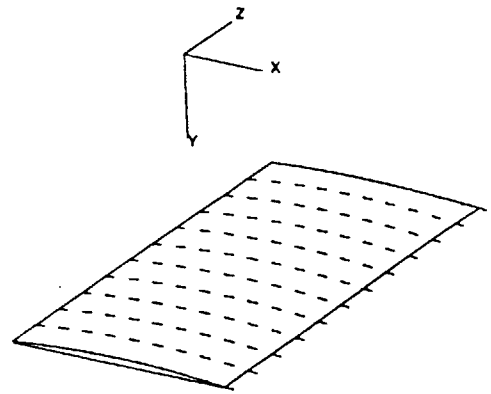


Figure 4. Velocity vector field, same case as Fig. 3.

REFERENCES

1. Piers, W. J. and Sloof, J. W. Calculation of transonic flow by means of a shock-capturing field panel method, AIAA Paper 79-1459, 1979.

2. Hounjet, M. H. L. Transonic panel method to determine loads on oscillating airfoils with shocks, *AIAA Journal*, 1981, Vol. 19, No. 5, pp.559-566.

3. Tseng, K. and Morino, L. Nonlinear Green's function methods for unsteady transonic flows, *Transonic Aerodynamics*, (Ed. Nixon, D), 1982, pp.565-603.

4. Oskam, B. Transonic panel method for the full potential equation applied to multicomponent airfoils, *AIAA Journal*, 1985, Vol. 23, No. 9, pp.1327-1334.

5. Erickson, L. L. and Strande, S. M. A theoretical basis for extending surface-paneling methods to transonic flow, *AIAA Journal*, 1985, Vol.23, No. 12, pp.1860-1867.

6. Sinclair, P. M. An exact integral (field panel) method for the calculation of two-dimensional transonic potential flow around complex configurations, *Aeronautical Journal*, June/July 1986, Vol. 90, No. 896, pp.227-236.

7. Kandil, O. A. and Yates, E. C., Jr. Computation of transonic vortex flows past delta wings-integral equation approach, *AIAA Journal*, 1986, Vol. 24, No. 11, pp.1729-1736.

8. Madson, M. D. Transonic analysis of the F-16A with under-wing fuel tanks: an application of the TranAir pull-potential code, AIAA Paper 87-1198, 1987.

9. Kandil, O. A. and Hu, H. Transonic airfoil computation using the integral equation with and without embedded Euler domains, *Boundary Elements IX*, Vol. 3, (Eds.: Brebbia, C. A., Wendland, W. L. and Kuhn, G.), Computational Mechanics Publications, Springer-Verlag, 1987, pp. 553-566.

10. Kandil, O. A. and Hu, H. Full-potential integral solution for transonic flows with and without embedded Euler domains, *AIAA Journal*, 1988, Vol. 26, No. 9, pp. 1074-1086.

11. Chu, L-C. Integral equation solution of the full potential equation for three-dimensional, steady, transonic wing flows, Ph.D. dissertation, Dept. of Mechanical Engineering and Mechanics, Old Dominion University, Norfolk, Virginia, March 1988.

12. Sinclair, P. M. A three-dimensional field-integral method for the calculation of transonic flow on complex configurations - theory and preliminary results, *Aeronautical Journal*, June/July 1988, Vol. 92, No. 916, pp. 235-241.

13. Kandil, O. A. and Hu, H. Unsteady transonic airfoil computation using the integral solution of full-potential equation, *Boundary Elements X*, Vol. 2, (Ed.: Brebbia, C. A.), Computational Mechanics Publications, Springer-Verlag, 1988, pp. 352-371.

14. Kandil, O. A. and Hu, H. Integral solution of unsteady full- potential equation for a transonic pitching airfoil, *Journal of Aircraft*, 1990, Vol. 27, No. 2, pp. 123-130.

15. Ogana, W. Transonic integro-differential and integral equations with artificial viscosity, *Engineering Analysis with Boundary Elements*, 1989, Vol. 6, No. 3, pp. 129-135.

16. Hu. H. Field-boundary element solution for a transonic pitching wing, *Topics in Engineering, Vol. 7: Advances in Boundary Elements Methods in Japan and USA*, (Eds.: Tanaka, M., Brebbia, C. A. and Shaw, R.), Computational Mechanics Publication, 1990, pp. 295-305.

17. Hu, H. Field-boundary element computation for transonic flows, *Computation Engineering with Boundary Elements*, Vol. 1, (Eds.: Grilli, S., Brebbia, C. A. and Cheng, A. H-D.), Computational Mechanics Publication, 1990, pp. 319-330.

18. Hu, H. and Chu, L-C. Unsteady Three-Dimensional Transonic Flow Computations Using Field Element Method, *Boundary Element Methods in Engineering*, (Eds.: Annigeri, B. S. and Tseng, K.), Springer-Verlag, 1990, pp.140-146.

19. Tseng, K. Application of Green's Function Method for 2- and 3- Dimensional Steady Transonic Flows, AIAA Paper 84-0425, 1984.