EEP-Vol. 9

G. S. DULIKRAVICH

CAE/CAD APPLICATION TO ELECTRONIC PACKAGING

presented at
1994 International Mechanical Engineering Congress and Exposition
Chicago, Illinois
November 6–11, 1994

sponsored by
The Electrical and Electronic Packaging Division, ASME

edited by
Dereje Agonafer
IBM Corporation

Robert E. Fulton
Georgia Institute of Technology

THE AMERICAN SOCIETY OF MECHANICAL ENGINEERS
UNITED ENGINEERING CENTER / 345 EAST 47TH STREET / NEW YORK, NEW YORK 10017

EEP-Vol. 9, CAE/CAD Application to Electronic Packaging
ASME 1994

AN INTERACTIVE MULTIDISCIPLINARY ANALYSIS, DESIGN AND OPTIMIZATION (MDA&O) APPROACH TO ELECTRONIC PACKAGING

Norman F. Foster, George S. Dulikravich, and Thomas J. Martin
Department of Aerospace Engineering

Jonathan D. Halderman
Department of Engineering Science and Mechanics

Pennsylvania State University
University Park, Pennsylvania

ABSTRACT

We have developed a unique set of robust, highly computationally efficient, and interactive computer codes for analysis, inverse design, and optimization of two-dimensional fluid flow, heat conduction, electrostatics, magnetohydrodynamics, electrohydrodynamics, and elastostatics. These codes have been packaged together into a highly interactive menu-driven multidisciplinary analysis, design, and optimization (MDA&O) software product that simultaneously accounts for fluid flow, heat transfer, and elastic stresses and deformations. This package is graphically intensive, and provides the user an easy- to-operate, easy-to-understand platform from which to design and analyze many types of manufacturable components including electronic circuit board configurations. The MDA&O package allows the user to choose different algorithms and interact with the design process at any stage. All relevant design parameters and field quantities can be monitored and modified at any time. The entire package employs the C++ programming language which produces a fast, self-sufficient, highly modular product that is easy to upgrade. The programming methodology utilized has allowed the ability to take full command of the host hardware system. The MDA&O package self-configures to support a wide range of computer platforms. This range can be increased because the hardware shell links at run-time with the technical modules, thus new hardware shells can easily be added.

NOMENCLATURE

a	local speed of sound
b	body force
E	Young's modulus of elasticity
G	shear modulus of elasticity
k	coefficient of thermal conductivity
$\boldsymbol{n}$	outward unit normal vector to surface
$\boldsymbol{P}$	prescribed boundary traction vector
$\boldsymbol{p}$	boundary traction vector
q	fundamental solution of Green's function
T	temperature
$\boldsymbol{s}$	displacement field vector
U	prescribed displacement boundary conditions
u	heat function
Γ	surface
Δ	Kronecker delta
δ	Dirac's delta function
ε	strain tensor
ν	Poisson's ratio
Φ	velocity potential function
σ	stress tensor

INTRODUCTION

There are many commercially available analysis/visualization packages for specific engineering disciplines, each with different strengths and weaknesses. The vast majority of these products employ finite element or finite volume techniques. Our current package is unique in several aspects.

Firstly, the initial development rationale was to produce a MDA&O package that simultaneously accounts for the concerns related to different, sometimes conflicting, disciplines. For example, very often the best aerodynamic design is not manufacturable or will fail due to excessive loadings and stresses. On the other hand, the finest elastostatic design might correspond to an undesirable thermal-stress situation, depending on the utility. Our goal was to produce a product that will assist in determining the best overall design in light of these types of issues. The current package supports inverse design based on a specified criterion, and optimization based on multiple criteria.

In addition to the analysis and inverse determination of boundary conditions of steady 2-D heat conduction and elastostatic problems, our package performs fluid flow analysis, inverse shape design, and shape optimization.

Inverse shape design can loosely be described as the determination of the physical shape of a body such that some prescribed field (or boundary) conditions are satisfied. For example, if a particular temperature or heat flux distribution is desired on the surface of a configuration yet corresponding geometrical configuration is unknown, then the inverse shape design method is to find that configuration such that the prescribed boundary conditions are satisfied. Our package will perform such inverse shape design to satisfy thermal boundary conditions [Dulikravich and Martin, 1994], elastostatic boundary conditions, and aerodynamic boundary conditions [Malone and Vadyak, 1985]. Further, our package will perform constrained optimization of arbitrary configurations such that a prescribed mixed aerodynamic-thermal-elastostatic cost function is driven to a minimum/maximum.

Second, most of the algorithms used here utilize boundary element methods (BEM). These are non-iterative techniques that are very accurate and efficient, yet do not suffer from the common error growth and computing cost of typical numerical iterative techniques.

Finally, the current package employs a very sophisticated programming structure and style. All algorithms are written in C++ and make extensive use of object-oriented command/data structures. This has allowed the seamless integration of otherwise diverse sets of codes into a highly interactive graphical environment.

In this paper the general code structure is discussed, theoretical developments of the algorithms employed are presented, and computational results are shown and discussed.

GENERAL PACKAGE OVERVIEW

The current package consists of four separate windowed modules that can run separately or simultaneously with any/all of the others. Each module is briefly described below.

- Builder: This is a small CAD program in which the user can construct geometries that are used in the other modules. Popular formats are supported for import/export such as AutoCAD and IDEAS.
- Analyzer: This module allows fast and detailed aerodynamic, thermal, and elastostatic analysis of any geometry built (or imported) from the Builder.
- Inverse Design: This module generates a new design from some initial (guess) design such that a specified target (desired) distributed physical quantity (boundary condition) is satisfied. These target quantities can be such things as aerodynamic pressure coefficient or surface heat flux distributions, among others.
- Optimization: This module optimizes an initial design such that a specified cost function is minimized or maximized. This cost function is defined by the user and can be multi-objective. For example, the user might want to maximize buckling load while minimizing the maximum internal thermal gradient under the influence of external heating. Nonlinear constraints, as well as global bound functions can be imposed .

Each of the four modules are generally structured in the same form from a programming standpoint. Specific programming details will only be discussed superficially in this paper.

Figure 1 shows the general structure of the Analyzer module (the other three modules are similar). Each box in Figure 1 represents a C++ "class-object". A "class-object" will be familiar to programmers of C++, and in particular object-oriented programming (OOP). The arrows in Figure 1 denote class data/function access between each class-object. Of particular importance in Figure 1 are the italicized class-objects. These class-objects occupy most of the system resources and thus are of great concern. To avoid unnecessary system use, these class-objects link to the rest of the module at run-time and are dynamically allotted memory. Likewise, while not in use, these class objects can be dumped from memory when the system needs to do so. The result is a module that self-manages memory at very low overhead.

The window-object in Figure 1 is the graphical interface and is the most machine-dependent class-object. However, because OOP has been utilized, new window-objects can be derived from the base object and added to the module to support a wide range of systems. This is also true of the event-handler-object in which parallel virtual machine (PVM) networking, for instance, could be added. Further details of the software engineering will not be discussed here.

The theoretical development of the algorithms used and their results will now be presented. Discussion of the Builder (CAD) module will be omitted because its workings and utilities are similar to other popular CAD packages that are readily available.

HEAT CONDUCTION ANALYSIS AND INVERSE DETERMINATION OF BOUNDARY CONDITIONS

The Analyzer module supports the determination of unknown temperatures and heat fluxes on surfaces of arbitrarily shaped objects where the thermal boundary conditions cannot be measured or evaluated otherwise. A requirement for this technique is that both temperature and heat flux distributions must be available and specified together (over-specified) on at least a part of the object's surface. The algorithm also simultaneously computes the temperature field within the entire object. Our algorithm utilizes the boundary element method (BEM) which is a very accurate and computationally efficient technique [Brebbia, 1978; Brebbia and Dominguez, 1989; Martin, 1993]. It does not require any artificial smoothing technique and is very robust because it is non-iterative. It has been shown [Martin and Dulikravich, 1993] to compute meaningful and accurate thermal fields in a single analysis using a straight-forward modification to the classical BEM.

The Analyzer module can perform direct analysis when either temperatures, T, or heat fluxes, Q, are specified everywhere on the surface of the solid where one of these quantities is known while the other is unknown. When performing inverse evaluation, both T and Q must be specified on a part of the solid's surface, while both T and Q are unknown on another part of the surface. If temperatures or heat fluxes are known (from

measurements or otherwise) at isolated points within the solid, they can be directly added to the BEM equation set in order to enhance the accuracy of the inverse heat conduction procedure.

The governing equation for the steady-state heat conduction in a solid with a temperature-dependent coefficient of thermal conductivity is

$$\nabla \bullet \left(k(T)\nabla T\right) = 0 \tag{1}$$

where T is the temperature and k(T) is the temperature-dependent coefficient of thermal conductivity. Equation (1) can be linearized by the application of the classical Kirchoff transformation [Martin and Dulikravich, 1994a] which defines the heat function, u, as

$$u = \int_0^T \frac{k(T)}{k_0} dT \tag{2}$$

where k_0 is the reference coefficient of thermal conductivity. Equation (1) is subsequently transformed into Laplace's equation operating on the heat function, u. The boundary integral equation (BIE) for Laplace's Equation is obtained from the weighted residual statement or Green's Theorem

$$c(\xi)u(\xi) + \int_\Gamma (u^* q)d\Gamma = \int_\Gamma (q^* u)d\Gamma \tag{3}$$

where the integration is over the solid surface, Γ. Here, $q = \partial u/\partial n$, u^* is the fundamental solution [Brebbia and Dominguez, 1989], $q^* = \partial u^*/\partial n$, n is the direction of the outward normal to the surface, Γ, and $c(\xi)$ is a term arising from the integration over the singularity in the sense of the Cauchy principal value at the point ξ. The fundamental solution is a general Green's function solution for a point-source subject to the homogeneous boundary conditions. For the two-dimensional Laplace's equation it is

$$u^* = \frac{1}{2\pi} \ln\left(\frac{1}{|\xi - x|}\right) \tag{4}$$

respectively. The surface area of the object may be discretized into a number of surface panels connected by N nodes. The functions u and q can be constant or vary linearly or quadratically, etc., distributed over each panel. After adding the contributions from each surface integral, the whole set of boundary integral equations can be written in a matrix form as

$$[\mathbf{H}]\{\mathbf{U}\} = [\mathbf{G}]\{\mathbf{Q}\} \tag{5}$$

For example, for constant elements the entries to the [**H**] and [**G**] matrices are

$$h_{ij} = \int_{\Gamma_j} q^* d\Gamma \qquad g_{ij} = \int_{\Gamma_j} u^* d\Gamma \tag{6}$$

For a well-posed boundary value problem, at least one of the functions, u or q, will be known at each boundary point (either Dirichlet or von Neumann boundary condition). This will lead to N knowns and N unknowns. Equation (5) can then be rearranged so as to separate the knowns from the unknowns to obtain

$$[\mathbf{A}]\{\mathbf{X}\} = \{\mathbf{F}\} \tag{7}$$

where {**X**} contains the unknown functions (u's and q's). Equation (7) can be solved by standard procedures.

For an inverse analysis, or if the boundary conditions are applied as above, the problem becomes ill-posed and the number of unknowns may exceed the number of knowns. This type of problem can still be solved by the method above except Equation (7) can become highly singular. The Singular Value Decomposition (SVD) technique can be used to yield an acceptable solution. We have tested this entire methodology and proved it successful [Martin and Dulikravich, 1994a; 1994b].

Figure 2 shows results of a BEM analysis and an inverse analysis applied to a multiply connected domain. A plate was specified to be 1.0 cm square, with a 0.4 cm square hole off center. This test case was chosen to roughly approximate a portion of a circuit board with an attached component. First, a standard analysis was conducted with well-posed boundary conditions. The left side of the plate was specified with a temperature of 1°C, the right side was specified with a temperature of 0°C, the top and bottom sides were specified to be adiabatic, and the hole boundary was specified to be 0.5°C everywhere. As a by-product of the analysis, the interior temperature distribution was obtained and is plotted in Figure 2a. Then, the inverse problem was posed by specifying both temperature and heat fluxes on the walls of the plate, while nothing was specified about the hole. This inverse problem was solved and the interior temperature is plotted in Figure 2b. It can be seen that the inverse methodology (Figure 2b) yields a solution very close to the well-posed boundary value problem (Figure 2a). Figure 2c shows a window dump of the Analyzer (running on an IBM 486DX-33 personal computer with Microsoft Windows 3.1) operating on this test case. The test specimen is shown in the graphics area and is shaded in different hues corresponding to the varying internal temperature distribution. Figure 2d shows a window dump of every module in our package (overlapped). In the foreground is a pop-up window used for the specification of thermal boundary conditions. These figures (2c and 2d) demonstrate the flavor of our package. It is highly interactive, easy to use, and reliable.

Figure 3 shows the results of a BEM analysis and inverse problem applied to a more complicated, polygonal, circuit board configuration with twelve components. The thermal conductivity of the plate was adjusted to correspond to that of gold. For the analysis (shown in Figure 3a), all sides of the outer boundary were specified to have a convective heat transfer boundary condition to an ambient temperature of 300°K except for the extreme upper-most side on which the temperature was specified to be 200°K. Every internal component (chip) was specified to have a 500°K boundary. Figure 3a shows the temperature contours that were obtained by the BEM analysis

with the given boundary conditions. Heat fluxes on every boundary were also obtained as a by product of this analysis.

For the inverse problem, the temperatures and heat fluxes that were obtained in the standard analysis (above) were imposed as boundary conditions everywhere on the outer boundary, except on the left-most side of the plate from ordinate values of Y=0.0 to Y=0.4 where nothing was specified. On every boundary of every internal component, no information (boundary conditions) was specified. Figure 3b shows the temperature contours obtained from our inverse BEM heat conduction program. It can be seen that the inverse results differ mostly as a result of the lack of provided boundary conditions on the lower left-hand outer edge of the plate. Nevertheless, by looking at the actual numbers it is clear that the inverse results agree very well with the standard analysis results when the degree of unknown boundary conditions is considered.

Yet another inverse option is available with the heat conduction code. It is capable of determining the correct number of cooling holes, their locations, sizes and shapes subject to overspecified thermal boundary conditions [Dulikravich and Martin, 1994].

ELASTOSTATIC ANALYSIS AND INVERSE DETERMINATION OF BOUNDARY CONDITIONS

Our package will perform highly accurate 2-D steady elastostatic analyses for arbitrarily shaped multiply connected domains. The code utilizes the BEM with quadratic isoparametric surface contour elements. Stress and deformation components can vary quadratically along the panels. The code is non-iterative, thus it is extremely fast and reliable. Our algorithm will also determine elastostatic boundary conditions (stress and deformation components) where none are known, provided both deformation and force traction components are known on at least part of the boundary. A by-product of the analysis is that the entire stress and deformation fields within the solid are simultaneously computed. The algorithm is highly flexible in treating complex geometries and mixed elastostatic boundary conditions [Martin, Halderman and Dulikravich, 1994].

The formulation of this elastostatics code is very similar to that of the heat conduction code presented above. To perform simple steady-state elastostatic analysis using BEM, either displacement vectors, **s**, or surface traction vectors, **p**, are specified everywhere on the boundary of the solid where one of these quantities is known while the other is unknown. When performing an inverse evaluation of the steady-state elastostatics using the BEM, both **s** and **p** must be specified on a part of the solid's boundary, while both **s** and **p** are unknown on another part of the boundary. Elsewhere, either **s** or **p** should be applied. A surface section where both **s** and **p** are specified simultaneously is called an over-specified boundary.

The two-dimensional state of stress at a point is defined using a second order stress tensor, σ. These stress components must satisfy the following equilibrium equations [Brebbia, 1978]

$$\frac{\partial \sigma_{kj}}{\partial x_j} + b_k = 0 \qquad \text{where } j = 1,2 \text{ and } k = 1,2 \tag{8}$$

Here, b_k are the net body forces per unit volume. The surface force tractions on the object are denoted by the vector, p_k, and the prescribed boundary values of these tractions on the surface Γp are denoted by P_k. Equilibrium at the boundary requires the satisfaction of the following stress boundary conditions [Brebbia, 1978]

$$p_k = \sigma_{kj} n_j = P_k \tag{9}$$

where n_j is the unit outward normal vector to the surface. The state of strain at a point within the solid is denoted by the second order symmetric strain tensor, ε. The strain-displacement relations for linear theory can be written in indicial form as

$$\varepsilon_{kj} = \frac{1}{2}\left(\frac{\partial s_k}{\partial x_j} + \frac{\partial s_j}{\partial x_k} \right) \tag{10}$$

where s_k is the vector displacement field. Let Γ_u be the portion of the boundary where the displacement boundary conditions, U_k, are prescribed. The states of stress and strain for a solid body are related through Hooke's Law, which depend on the material behavior

$$\begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ 2\varepsilon_{yx} \end{Bmatrix} = \frac{1}{E} \begin{bmatrix} 1 & -\upsilon & 0 \\ -\upsilon & 1 & 0 \\ 0 & 0 & 2(1+\upsilon) \end{bmatrix} \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{yx} \end{Bmatrix} \tag{11}$$

where E is the Young's modulus of elasticity, ν is the Poisson's ratio. Lame's constants are related to E and ν in the follow way

$$\lambda = \frac{E\upsilon}{(1+\upsilon)(1-2\upsilon)} \tag{12}$$

$$G = \frac{E}{2(1+\upsilon)} \tag{13}$$

where G is the shear modulus. The principle of virtual displacements for linear elasticity can be written as [Brebbia, 1978]

$$\int_\Omega \left(\frac{\partial \sigma_{kj}}{\partial x_j} + b_k \right) s_k^* \, d\Omega = \int_{\Gamma_p} (p_k - P_k) s_k^* \, d\Gamma + \int_{\Gamma_u} (S_k - s_k) p_k^* \, d\Gamma \tag{14}$$

where $p_k^* = n_j \sigma_{jk}^*$ are the surface tractions corresponding to the virtual displacements s_k^*. We will assume that the virtual strain-displacement relationships and the material properties are linear. After integrating by parts twice and ignoring for now the body forces, we obtain

$$\begin{aligned} &\int_\Omega \frac{\partial \sigma_{kj}^*}{\partial x_j} s_k \, d\Omega + \int_{\Gamma_p} P_k s_k^* \, d\Gamma + \int_{\Gamma_u} p_k s_k^* \, d\Gamma \\ &= \int_{\Gamma_u} U_k p_k^* \, d\Gamma + \int_{\Gamma_p} u_k p_k^* \, d\Gamma \end{aligned} \tag{15}$$

The virtual displacement will be a fundamental solution satisfying the equilibrium equations

$$\frac{\partial \sigma_{ij}^{*}}{\partial \xi_j} + \delta_i(x_i - \xi_i) = 0 \qquad (16)$$

where δ is the Dirac's delta function and represents a unit load at the point "i" in the "l" direction. In general, we can write for any point "i" and for any direction "l" a boundary integral equation

$$c_i^l s_i^l + \int_\Gamma s_k p_{lk}^{*}\, d\Gamma = \int_\Gamma p_k s_{lk}^{*}\, d\Gamma \qquad (17)$$

where c_l is obtained with some special treatment of the surface integral on the left hand side [Brebbia, 1978]. Explicit calculation of this value can be obtained by augmenting the surface integral over the singularity that occurs when the integral includes the point "i". Fortunately, explicit calculation is not necessary as it can be obtained using rigid body motions. The fundamental solution for a two-dimensional isotropic plane strain case is

$$s_{lk}^{*} = \frac{1}{8\pi G(1-\upsilon)}\left[(3-4\upsilon)\ln\left(\frac{1}{r}\right)\Delta_{lk} + \frac{\partial r}{\partial x_l}\frac{\partial r}{\partial x_k}\right] \qquad (18)$$

where r is the distance from the "i" node to the point of integration on the boundary and Δ is the Kronecker delta.

The boundary Γ is discretized into surface panels connected by N nodes. A transformation from the global (x,y) coordinate system to a localized boundary fitted (ξ) coordinate system is required in order to numerically integrate each surface integral using Gaussian quadrature. The displacements and tractions are defined in terms of nodal values and interpolation functions if quadratic distributions are used. The whole set of boundary integral equations can be written in matrix form

$$[\mathbf{H}]\{\mathbf{S}\}=[\mathbf{G}]\{\mathbf{P}\} \qquad (19)$$

where vectors $\{\mathbf{S}\}$ and $\{\mathbf{P}\}$ contain the nodal values of the displacement and traction vectors. Each entry in the $[\mathbf{H}]$ and $[\mathbf{G}]$ matrices is developed by properly summing the contributions from each numerically integrated surface integral. The boundary tractions are allowed to be discontinuous between each neighboring boundary panel to allow for proper corner treatment. The set of boundary integral equations will contain a total of 2N equations and 6N nodal values of displacements and boundary tractions.

At this point this formulation is handled like the heat conduction system, Equation (5). Equation (19) can be reordered into a new system

$$[\mathbf{A}]\{\mathbf{X}\}=\{\mathbf{F}\} \qquad (20)$$

where $\{\mathbf{X}\}$ contains the unknown (desired) **s** and **p** values. For a well-posed boundary value problem, at least one of the functions, **s** or **p**, will be known at each node (either Dirichlet or von Neumann boundary conditions) so that the equation set will be composed of 2N knowns and 2N unknowns. This case would constitute a simple analysis, and (20) can be solved by standard methods. For ill-posed problems or where the number unknowns exceeds the number of knowns (such as in an inverse problem), Equation (20) must be solved by the SVD technique due to the probable high degree of singularity of the system. If the number of unknowns is greater than the number of knowns, the SVD method will find one solution, although its accuracy decreases as this differential increases.

The inverse elastostatic BEM algorithm was tested on an annular disk plate subject to an internal gauge pressure. The shear modulus for this problem was $G=8.0\times10^4$ N/mm^2 and Poisson's ratio was $\nu=0.25$. The radius of the inner surface of the disk was 10 mm and the outer radius was 25 mm. The inner and outer boundaries were discretized with 12 quadratic panels each. The internal gauge pressure was specified to be $P_\Gamma=100$ N/mm^2, while the outer boundary was specified with a zero traction. The two-dimensional elastostatic code computed the displacement and stress fields within the annular domain. Figure 4a illustrates the radial displacement vector field with lengths of the arrows corresponding to the scaled magnitudes of the local deformations. Figures 5a represents a contour plot of constant values of σ_{yy}, that was computed using the analysis version of our second-order accurate BEM elastostatic code. The numerical results of this well-posed boundary value problem were then used as boundary conditions applied to the following two ill-posed problems.

The displacement vectors computed on the outer circular boundary by the well-posed numerical analysis were used to over-specify the outer circular boundary. At the same time, nothing was specified on the inner circular boundary. Figure 4b shows the displacement field and Figure 5b is the contour plot of σ_{yy} as computed by the inverse BEM technique. The inner surface deformations were in error by less than 0.1%, while stress averaged less than 1.0% error from the analysis results.

It is clearly shown here that this new method is capable of directly determining unknown steady elastostatic boundary conditions on boundaries of solids where such quantities are unknown. Numerical results were in excellent agreement with the analytic values in regions relatively close to the over-specified data, and were meaningful in areas where no data could be supplied. Because it is non-iterative, our method is very fast, and robust, consuming only seconds on an advanced personal computer.

FLUID FLOW ANALYSIS

The MDA&O package also has the ability to perform flow field analyses to varying degrees of precision. This is possible because the package has several flow solver modules that can individually be utilized upon command. The fastest of these solvers assumes the flow field is incompressible, steady, irrotational, and inviscid. This potential flow governed by the Laplace's equation can be modeled by descretizing the boundary of the flow field into linear or parametric segments. Point 'vortices' can be distributed on these descretized boundary elements. The strengths of these point singularities are determined by enforcing flow tangency at a finite number of

boundary locations. Finally, the flow velocity and pressure can be determined once these strengths are known.

The potential flow solver is by no means the most accurate, or the most current method, for determining flow field properties. However, the speed at which it can be implemented makes it a valuable tool for producing a preliminary solution for input into the more accurate and sophisticated solvers.

The second flow solver available in the MDA&O package solves the Full Potential Equation (FPE) given by

$$\left(1-\frac{\Phi_x^2}{a^2}\right)\Phi_{xx}+\left(1-\frac{\Phi_y^2}{a^2}\right)\Phi_{yy}-\frac{2\Phi_x\Phi_y}{a^2}\Phi_{xy}=0 \qquad (21)$$

where Φ is a scalar function called the velocity potential and a is the local speed of sound [Anderson, 1990]. This equation assumes the flow is irrotational, inviscid, isentropic and compressible. The FPE is integrated using the finite volume technique and successive line over-relaxation.

We have also developed (Lee and Dulikravich 1991) a computer code capable of efficiently solving Navier-Stokes equations for two-dimensional steady laminar and turbulent flows (Choi and Dulikravich 1994) including strong heat transfer (Figure 6). The formulation uses Boussinesq approximation for thermally induced buoyancy effects and also accounts for the presence of arbitrarily configured externally applied magnetic (Dulikravich, Ahuja and Lee 1994) and electric fields (Dulikravich, Ahuja and Lee, 1993). The potential flow model can be used as a very efficient and robust module for the creation of a meaningful initial guess for the flow field to be consequently analyzed by the full Navier-Stokes code which uses finite volume formulation. The Navier-Stokes code utilizes characteristic and non-reflecting boundary conditions (Dulikravich, Ahuja and Lee, 1994) thus allowing for smaller computational domains and open flow separation at the exit boundary (Figure 7).

In the case of electronic board design, the ability of this package to perform flow field analyses allows the user to visualize immediately the transport of heat due to circulation.

COMPUTATIONAL GRID GENERATION

The computational grid generation package consists of two modules. First, a reasonable preliminary grid is generated using an exceptionally robust and efficient method (Kennon 1984) that enforces desired grid orthogonality at the solid boundaries. Since this grid generation code does not guarantee that the grid lines of the same family will not overlap, we use it as an input to our (Kennon and Dulikravich 1986) *a posteriori* grid generation code that utilizes optimization in order to unravel the possible overlapped grid cells and to minimize local grid non-orthogonality and non-smoothness.

INVERSE AERODYNAMIC SHAPE DESIGN

As a part of the Inverse Design module, two-dimensional geometries can be determined by specifying desired flow field properties. As an example, the user can specify the desired pressure distribution on the boundary of some solid. The code will then modify the geometry of the solid such that the desired (target) pressure distribution is obtained. The inverse design method used for this purpose is called the Modified Garabedian-McFadden (MGM) method. The MGM method models the motion of the solid boundary similar to a mass-spring-damper system with a forcing function defined as the difference of the current flow properties and the desired flow properties at the boundary [Malone and Vadyak, 1985].

The MGM method hinges on the governing equation in two dimensions given by,

$$\beta_0\frac{\partial y}{\partial t}+\beta_1\frac{\partial^2 y}{\partial x\partial t}+\beta_2\frac{\partial^2 y}{\partial x^2\partial t}=Q^2-q^2=\Delta Q^2 \qquad (22)$$

where (x,y) are the coordinates of a boundary node, β's are user-specified (and arbitrary) values to help speed convergence, Q is the desired flow property value at (x,y), q is the flow property value presently at (x,y). One of the nodal coordinates (x or y) is allowed to vary, while the other is held constant. For the present development, let us assume that the y coordinate is allowed to vary and x is not. The time derivative can be interpreted as the change in y between two design iterations, so equation (22) becomes,

$$\beta_0\frac{\Delta y}{\Delta t}+\frac{\beta_1}{\Delta t}\frac{\partial\Delta y}{\partial x}+\frac{\beta_2}{\Delta t}\frac{\partial^2\Delta y}{\partial x^2}=\Delta Q^2 \qquad (23)$$

where Δt is arbitrary and can be set equal to one. The goal is to solve equation (23) for Δy so that the present configuration can be updated by

$$y^{new}=y^{old}+\Delta y \qquad (24)$$

After further development [Malone and Vadyak, 1985] and paying close attention to the orientation of each boundary element with respect to the assumed flow direction, equation (22) takes its final form given by

$$\beta_0\Delta y+\beta_1\frac{\partial\Delta y}{\partial x}+\beta_2\frac{\partial^2\Delta y}{\partial x}=\beta_3\Delta Q^2 \qquad (25)$$

where, β_3 can equal 1.0 or -1.0 depending on the boundary element orientation [Malone and Vadyak, 1985]. Equation (25) can then be rewritten with finite difference approximations for the derivatives. By making these substitutions, an expression for the evaluation of equation (25) at the ith node becomes,

$$A_i\Delta y_{i-1}+B_i\Delta y_i+C_i\Delta y_{i+1}=\Delta Q_i^2 \qquad (26)$$

The MGM method, then, is essentially the implementation of equation (26) and requires solving the tridiagonal matrix formed by coefficients A, B, and C. Figure 8 shows the Inverse Design module operating on an airfoil. Here, the user selects a desired pressure distribution, previously produced in the Builder module, from hard storage.

CONCLUSION

We have developed a unique set of robust, highly computationally efficient and interactive computer codes for

analysis, inverse design and optimization of two-dimensional fluid flow, heat transfer and elastostatics. These codes have been recoded in C++ and made interchangeable modules in a master code for a highly interactive menu-driven multidisciplinary analysis, design and optimization of arbitrary two-dimensional configurations. The MDA&O package allows the user to actively interact with the design process at any stage and to continuously monitor multiple graphical displays of all relevant parameters and field quantities. The package is highly self-sufficient and easy to upgrade and transport. Several applications of the MDA&O package to the design and analysis of cooled electronic boards have been demonstrated.

REFERENCES

Anderson, J.D., 1990 "Modern Compressible Flow with Historical Perspective", McGraw-Hill Publishing Company, New York.

Brebbia, C.A., 1978 "The Boundary Element Method for Engineers", John Wiley & Sons, New York.

Brebbia, C.A., and Dominguez, J., 1989 "Boundary Elements, An Introductory Course", McGraw-Hill Book Company, New York.

Choi, K.-Y., and Dulikravich, G, S., 1994 "Global sensitivity-based minimal residual (SBMR) method for acceleration of iterative convergence rates", Proceedings of 14th International Conference on Numerical Fluid Dynamics, Bangalore, India, July 11-15.

Dulikravich, G.S., 1992 "Aerodynamic shape design and optimization: Status and trends", AIAA Journal of Aircraft, Vol. 29, No. 5, pp. 1020-1026.

Dulikravich, G. S., Ahuja, V., and Lee, S., 1993 "Simulation of electrohydrodynamic enhancement of laminar flow heat transfer", Journal of Enhanced Heat Transfer, Vol. 1, No. 1, pp. 255-262.

Dulikravich, G. S., Ahuja, V., and Lee, S., 1994 "Modeling three-dimensional solidification with magnetic fields and reduced gravity", International Journal of Heat and Mass Transfer, Vol. 37, No. 5, pp. 837-853.

Dulikravich, G.S., and Martin, T.J., 1994 "Inverse design of superelliptic cooling passages in coated turbine blade airfoils", AIAA Journal of Thermophysics and Heat Transfer, Vol. 2, pp. 288-294.

Kennon, S. R., and Dulikravich, G. S., 1986 "Optimization of computational grids", AIAA Journal of Aircraft, Vol. 23, No. 5, pp. 415-421.

Kennon, S. R., 1984 "Direct grid generation using complex spline fitting", Proceedings of the 5th International Symposium on Finite Elements and Flow Problems, University of Texas, Austin, TX, January 23-26, pp. 29-33.

Kuethe, A.M., and Chow, C.Y., 1986 " Foundations of Aerodynamics", 4th Edition, John Wiley & Sons, Inc. New York.

Lee, S., and Dulikravich, G. S., 1991 "Accelerated computation of viscous flows with heat transfer", Numerical Heat Transfer: Fundamentals, Part B, Vol. 19, pp. 223-241.

Martin, T.J., 1993 "Inverse design and optimization of two- and three-dimensional coolant flow passages", M.S. Thesis, Dept. of Aerospace Engineering, The Pennsylvania State University.

Martin, T.J., and Dulikravich, G.S., 1994a "Inverse determination of temperatures and heat fluxes on inaccessible surfaces", Proceedings of BETECH'94, Editors: C.A. Brebbia and A. Kassab, Orlando, FL, March 16-18, pp 69-76.

Martin, T.J., and Dulikravich, G.S., 1994b "Finding unknown surface temperatures and heat fluxes in steady heat conduction", Fourth Intersociety Conference on Thermal Phenomena in Electrical Systems, Washington, DC, May 4-7, pp. 214-221.

Martin, T.J., and Dulikravich, G.S., 1993 "A direct approach to finding unknown boundary conditions in steady heat conduction", Proc. of 5th Annual Thermal and Fluids Workshop, NASA CP-10122, Lewis Research Center, Ohio, Aug. 16-20, pp.137-149.

Malone, J., and Vadyak, J., 1985 "A Technique for the Inverse Aerodynamic Design of Nacelles and Wing Configurations", AAIA paper, AAIA-85-4096.

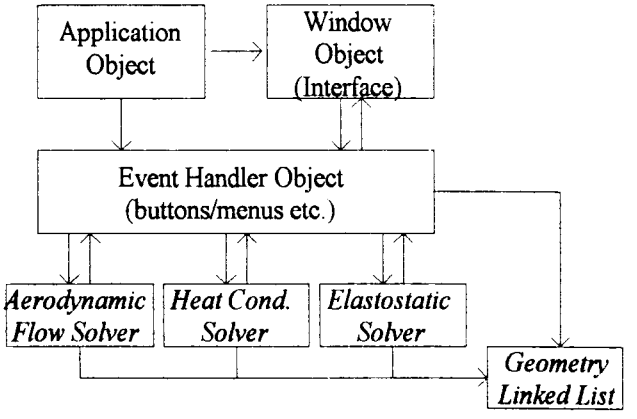

FIGURE 1. GENERAL (MACRO) STRUCTURE OF ANALYZER CODE. ALL *ITALICIZED* COMPONENTS ARE DYNAMICALLY ALLOCATED AND DEALLOCATED.

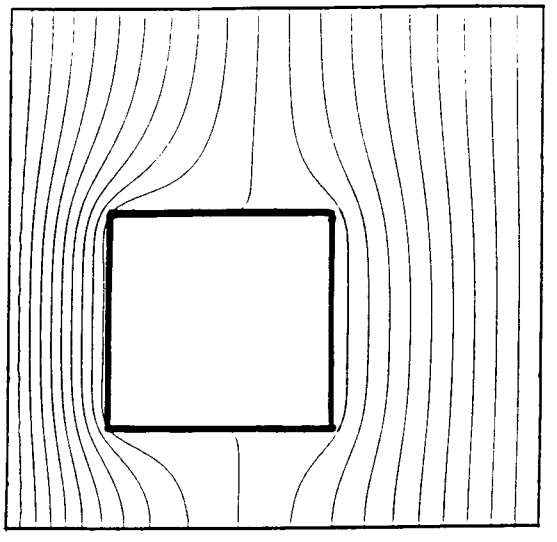

FIGURE 2A. SQUARE PLATE WITH SQUARE HOLE: ISOTHERMS OBTAINED WITH ANALYSIS BEM CODE.

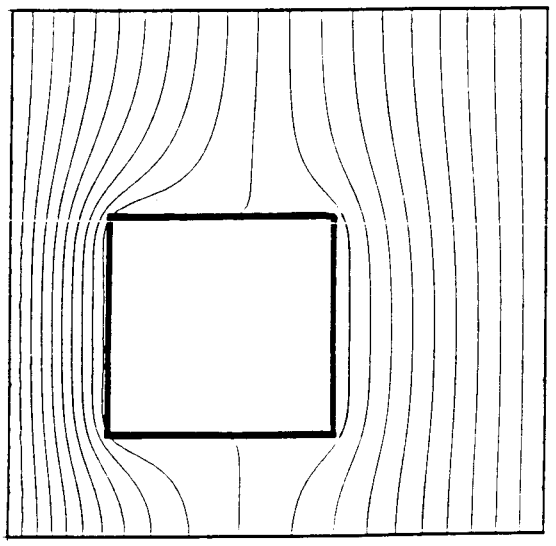

FIGURE 2B. SQUARE PLATE WITH SQUARE HOLE: ISOTHERMS OBTAINED WITH INVERSE BEM CODE ASSUMING THAT NOTHING IS KNOWN ON THE HOLE BOUNDARIES.

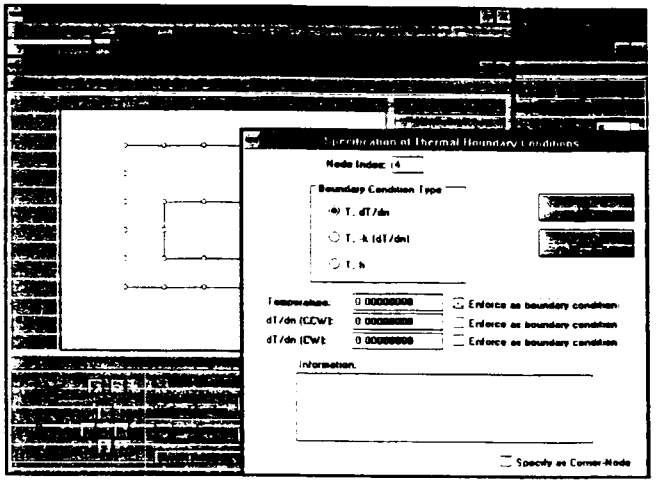

FIGURE 2D. WINDOW DUMP OF ANALYZER WITH BOUNDARY CONDITION SPECIFICATION WINDOW IN FOREGROUND. (IBM 486DX-33 MS WINDOWS VERSION).

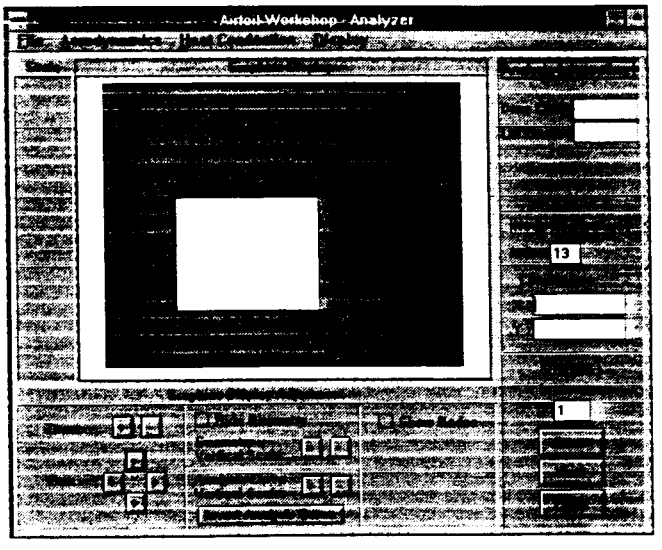

FIGURE 2C. WINDOW DUMP OF ANALYZER OPERATING ON SQUARE PLATE PROBLEM (FIGURE 2A AND 2C). THE PLATE IS SHADED IN HUES CORRESPONDING TO VARYING TEMPERATURES. (IBM 486DX-33 MS WINDOWS VERSION).

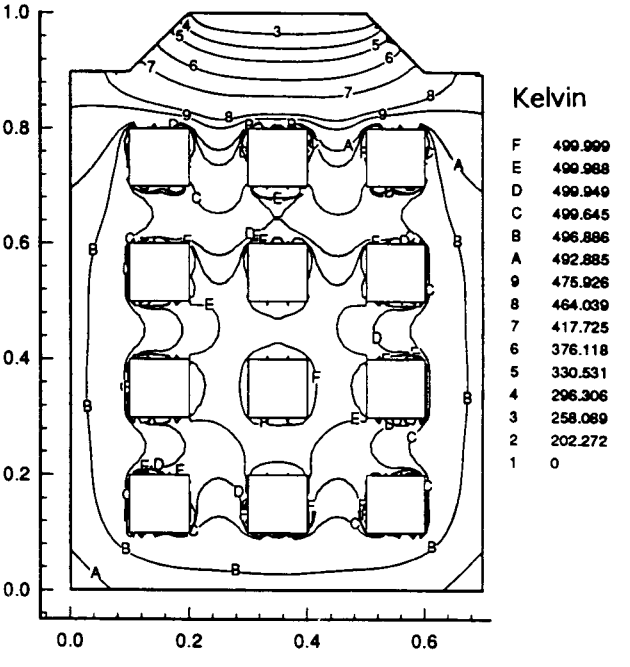

FIGURE 3A. CIRCUIT BOARD CONFIGURATION WITH TWELVE COMPONENTS: ISOTHERMS OBTAINED WITH BEM ANALYSIS CODE.

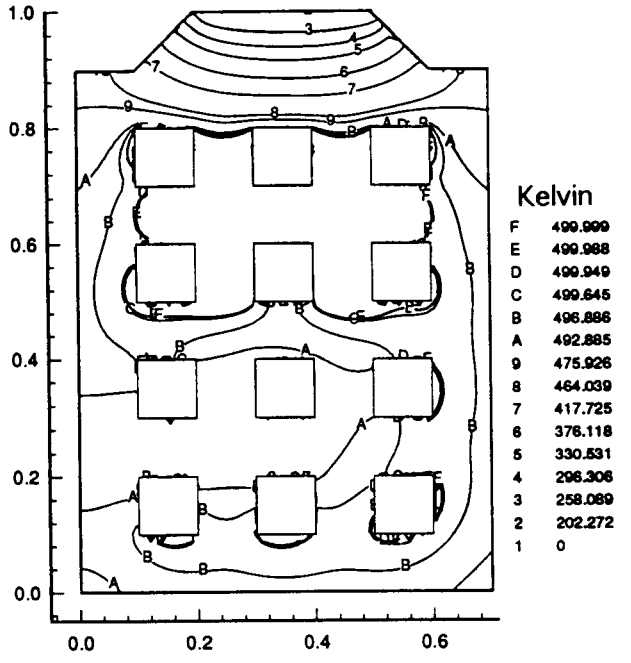

FIGURE 3B. CIRCUIT BOARD CONFIGURATION WITH TWELVE COMPONENTS: ISOTHERMS OBTAINED WITH INVERSE BEM CODE ASSUMING THAT NOTHING IS KNOWN ON INTERNAL BOUNDARIES, AND ON LEFT-MOST OUTER BOUNDARY FROM Y=0.0 TO Y=0.4.

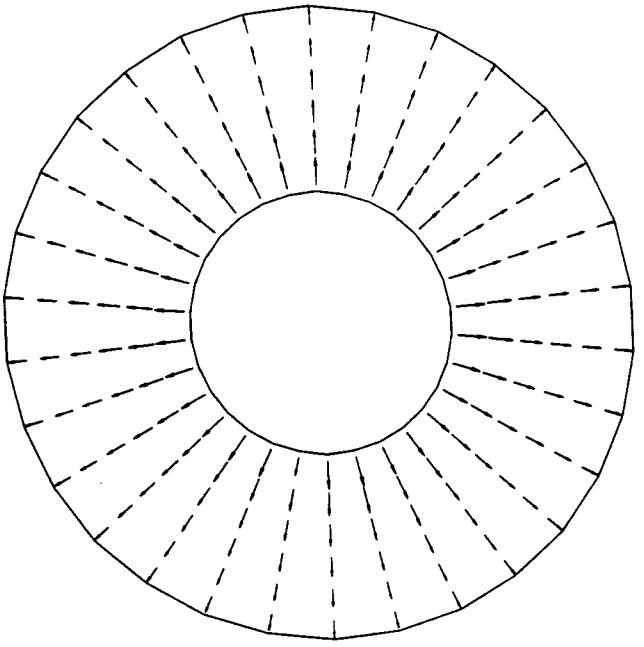

FIGURE 4B. VECTOR DISPLACEMENT FIELD FROM THE ILL-POSED ANALYSIS OF AN ANNULAR PRESSURIZED DISK: OUTER BOUNDARY OVER-SPECIFIED.

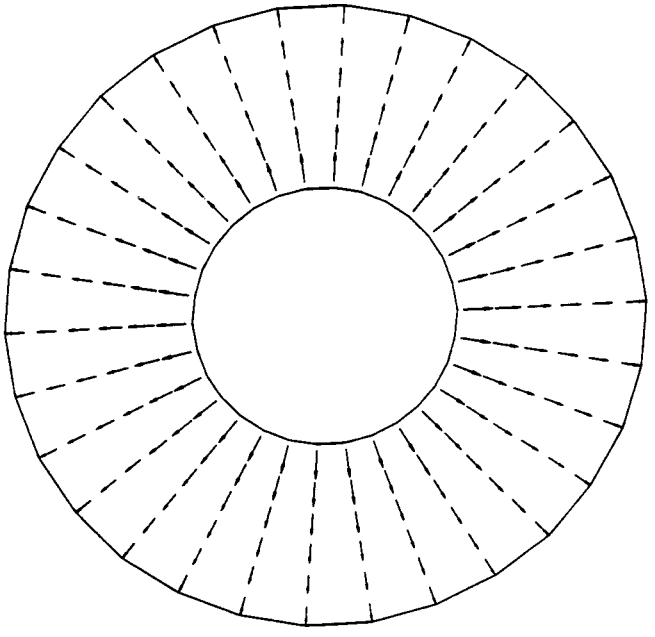

FIGURE 4A. VECTOR DISPLACEMENT FIELD FROM THE WELL-POSED ANALYSIS OF AN ANNULAR PRESSURIZED DISK.

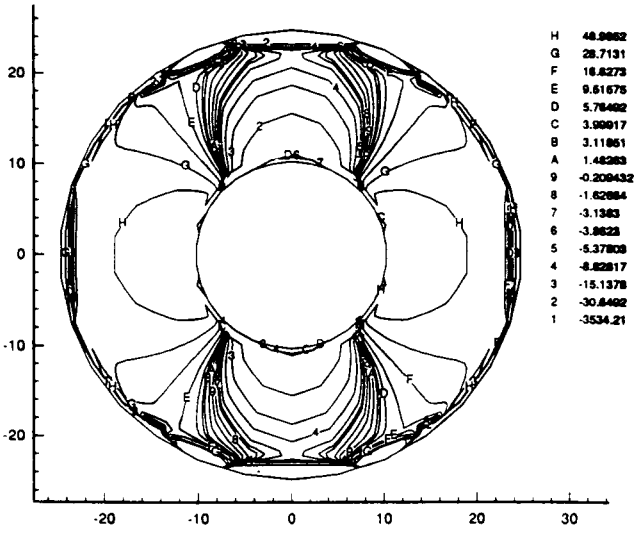

FIGURE 5A. CONTOURS OF CONSTANT STRESS, Σ_{YY}, FROM THE WELL-POSED ANALYSIS OF AN ANNULAR PRESSURIZED DISK.

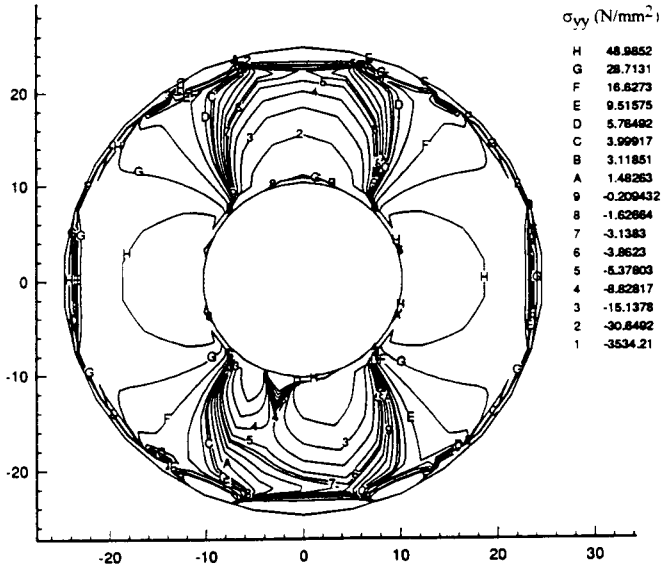

FIGURE 6B. CONTOURS OF CONSTANT STRESS, Σ_{YY}, FROM THE ILL-POSED ANALYSIS OF AN ANNULAR PRESSURIZED DISK: OUTER BOUNDARY OVER-SPECIFIED.

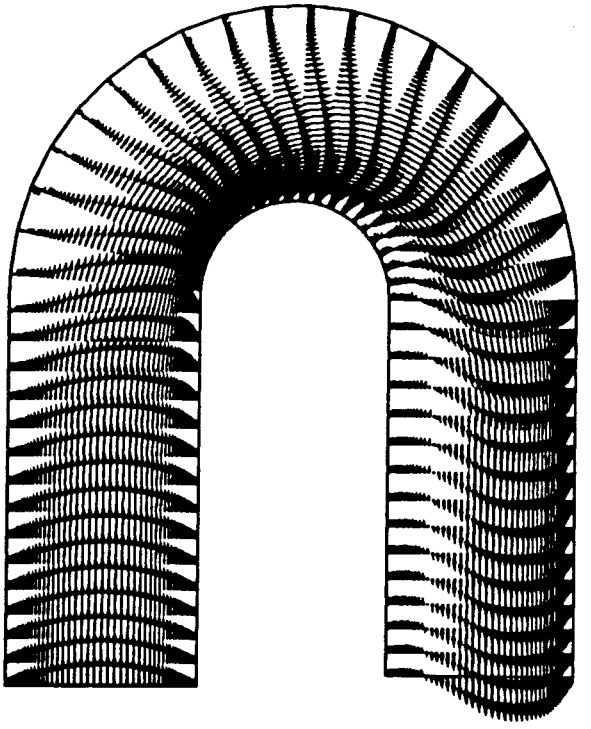

FIGURE 7. VELOCITY VECTOR PROFILES FOR AN ISOTHERMAL LAMINAR FLOW WITH AN OPEN SEPARATION CAPTURED WITH THE NON-REFLECTING EXIT BOUNDARY CONDITION FORMULATION.

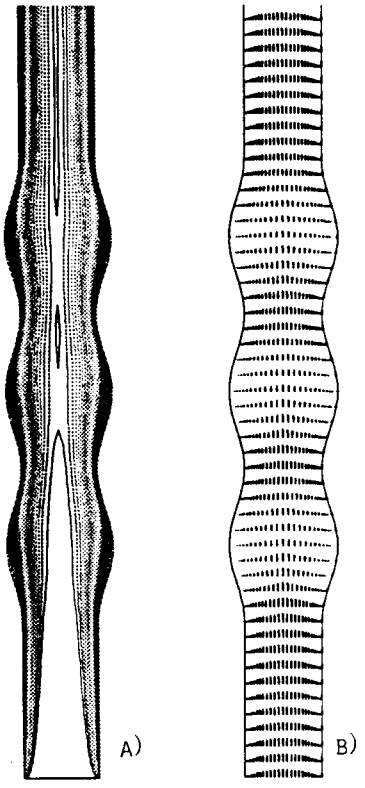

FIGURE 6. ISOTHERMS (a) AND VELOCITY VECTOR PROFILES (b) FOR A STEADY LAMINAR FLOW OF AN INITIALLY COLD FLUID PASSING THROUGH A CHANNEL WITH SINUSOIDAL WALLS KEPT AT A CONSTANT ELEVATED TEMPERATURE.

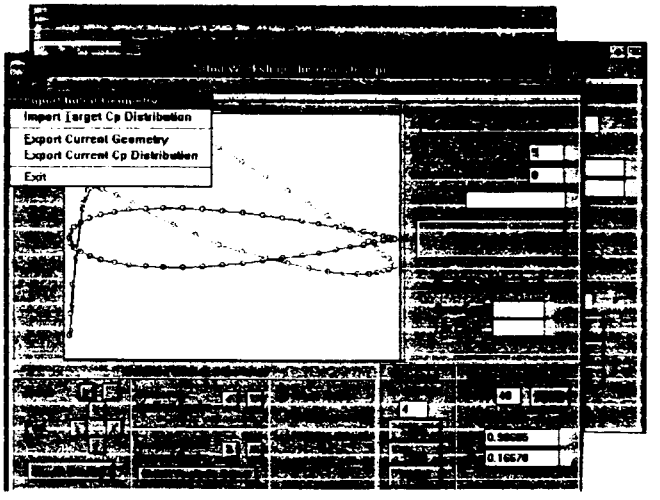

FIGURE 8. WINDOW DUMP OF INVERSE DESIGN MODULE. THE USER SELECTS A DESIRED PRESSURE DISTRIBUTION, AND THE CODE FINDS THE CORRESPONDING SOLID GEOMETRY BOUNDARY.