To the Graduate Council:

I am submitting herewith a thesis written by Satish P. Ketkar entitled "Determination of Optimal Hole Locations in a Two-Dimensional Heat Conductor Using Boundary Integral Method." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Mechanical Engineering.

M. Parang, Major Professor

We have read this thesis and recommend its acceptance:

Rao V. Arimilli

Accepted for the Council:

The Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature ____________________

Date March 1, 1985

DETERMINATION OF OPTIMAL HOLE LOCATIONS IN A TWO-DIMENSIONAL HEAT CONDUCTOR USING BOUNDARY INTEGRAL METHOD

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Satish P. Ketkar

March 1985

ACKNOWLEDGMENTS

The author wishes to acknowledge his major professor, Dr. Masood Parang, without whom this work would not have gone too far. His constant encouragement and willingness to go all out to help during the difficult periods as well as his friendly attitude made this work experience an enjoyable one. The invaluable knowledge and confidence he has given me cannot be acknowledged with a mere "Thank you."

My deep gratitude is also due to Dr. R. V. Arimilli who was always willing to help and guide during hard times. Thanks are also due to the members of the committee: Dr. R. J. Parsons and Dr. J. A. Euler, for their guidance and support. The author acknowledges the use of The University of Tennessee Computing Center, Knoxville, Tennessee, in obtaining the results shown here.

The author finally wishes to thank his parents, grandmother, sister and his girlfriend for their constant love and encouragement.

ABSTRACT

Steady state, two-dimensional conduction heat transfer problems in regions containing interior circular holes are considered. Such problems arise in the thermal design and analysis of molds and dies, where one of the objectives is to maintain a uniform temperature on the cavity surface.

In this study a boundary integral equation method especially developed for circular internal holes is used to develop a computer code. This code is used to accomplish two tasks. First, it is used to obtain the unknown values of temperature and heat transfer on the boundaries of an arbitrary two-dimensional heat conductor representing a mold or a die. Next, this computer code is used in conjunction with the subroutine CONMIN to determine the optimal locations of heating or cooling lines in order to minimize the variation of temperature on the die cavity surface.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE PAGE

FIGURE PAGE

LIST OF SYMBOLS

a_α	Radius of tube α
c_j	jth segment of discretized boundary
$g(X,Y)$	Harmonic function defined by Equation 3
m	Number of holes
q	Normal derivatives of temperature, $\partial\phi/\partial n$
q_0^α, q_1^α, q_2^α	Values of $\partial\phi/\partial n$ on hole as defined by Equation 6
X,Y	Position vectors of points on boundary
α,β	Tube identification number
ξ^β	Position vector of center of tube β
θ^α	Azimuthal angle of tube α
ϕ	Temperature
ϕ_0^α	Constant temperature on the surface of tubes (holes)
ψ	Angles defined in Figure 1
A	Radius of largest diameter tube
CD	Cavity surface
Bi	Biot number
K	Thermal conductivity

CHAPTER I

INTRODUCTION

Background Information

A number of important problems in engineering require the solution of Laplace's equation in regions enclosed by arbitrary boundaries. An example of these elliptic boundary value problems is steady conduction heat transfer where the temperature or its normal derivative will have to be found by numerical integration of the Laplace equation. Various numerical techniques have been proposed for the solution of elliptic boundary value problems. Among the most commonly used methods are finite difference and finite element Galerkin methods. These conventional methods can be used to obtain the unknown function (e.g., temperature or heat transfer in steady conduction heat transfer problems) over the entire region. A method of increasing popularity is the boundary integral equation method (BIE).

In many conduction heat transfer problems, the behavior of the temperature or its normal derivative is of interest only on the boundary of the domain under consideration. In this type of problem it is often advantageous to use boundary integral equation method in which the problem can be formulated in terms of the unknowns on the boundary of the domain. The major advantage of this formulation is that only the boundary of the region of interest needs to be discretized. Therefore BIE method offers a more efficient and less tedious numerical formulation of the problem relative to such methods as finite difference or finite element where the entire domain of interest is required to be discretized.

The use of boundary integral method for solving engineering problems is not a new phenomenon. Considerable amount of research using BIE method has been reported in a variety of areas such as conduction heat transfer, potential flow, elastic torsion, and fracture mechanics.

Problems governed by Laplace's equation have been successfully solved using BIE method [1,2]. Lachat and Watson [3] transformed the field equations of three-dimensional elastostatics to boundary integral equations and illustrated application of BIE in the analysis of a prestressed concrete reactor pressure vessel. Rizzo and Shippy [4,5] used BIE method for the analysis of three-dimensional thermoelasticity and two-dimensional potential problems. Khader and Hanna [6] presented an iterative boundary integral numerical solution for general steady heat conduction problems with a variety of different boundary conditions. The method also accounted for temperature dependent thermal conductivity with the use of Kirchoff transformation. Altenkirch, Rezayat, Eichhorn and Rizzo [7] used BIE method to calculate surface regression effects in flame spreading over pyrolyzing fuel.

Advantages of BIE Method

The type of problems solved by use of boundary integral equation method and some of the advantages of the method can be outlined as follows. Consider a function which satisfies Laplace's equation throughout region R.

$$\nabla^2\phi = 0 \text{ in } R$$

with ϕ satisfying one or more boundary conditions of the type

$$\phi = f \text{ or } \frac{\partial\phi}{\partial n} = g \text{ or } \frac{\partial\phi}{\partial n} + c\phi = 0 \text{ on } \partial R$$

where ϕ is the boundary of the region R and C, d, f and g are specified constants. The BIE method is based on Green's formula which enables one to reformulate the elliptic boundary value problem described above in terms of an integral equation. Such an integral equation involves the values of the function ϕ or its normal derivative only on the boundary of the region R. This unique feature of BIE method makes it possible to reduce the dimension of the problem by one. That is, since the boundary (surface) of the relevant region is dealt with exclusively, a three-dimensional problem can be reduced to a two-dimensional one, and a two-dimensional problem can be converted to a one-dimensional problem. The system of equations obtained from discretization of the boundaries of the region of interest is smaller than that obtained by conventional numerical methods which discretize the entire region of interest (for example, the finite element method). But, it should also be pointed out that the matrix of coefficients in BIE formulation are generally nonsymmetric and fully populated. In general, however, the number of nonzero elements in the coefficient matrix of the linear system arising from a finite element Galerkin method is comparable to the number of matrix elements which have to be calculated in BIE method if the number of boundary nodes chosen are the same in both problems [5].

The Problem and the Method of Solution

A number of technically important problems require the solution of Laplace's equation in regions with circular holes. Some examples are two-dimensional conduction heat transfer through dies or molds in casting processes where cooling or heating lines are positioned in the mold to decrease or increase the temperature of the cavity surface. Other examples may include heat conduction in cooled gas turbine blades, flow of an incompressible invisid fluid around circular cylinders, buried heat-exchanger tubes of ground coupled heat pump systems, elastic torsion in bars with circular holes and electrostatic field around gratings of charged wires.

In application of numerical methods to steady state conduction heat transfer problems (including boundary integral equation method) the discretization of the boundaries internal to the region of interest may pose special problems. Specifically, when these internal boundaries are small relative to the outer boundary (e.g., small holes) a fine discretization needs to be implemented on the entire inner boundary to avoid numerical conditioning problems. This in turn requires the solution of a large number of equations which often defeats the advantage of smaller number of computations inherent in the boundary integral equation method. To avoid this problem, it is desirable to express analytically the unknown function (temperature or its normal derivative) on the internal boundary. In that case the integration over the internal boundary can be performed analytically and the discretization of this boundary can be avoid entirely.

For circular internal boundaries Barone and Caulk [8,9] have developed a method where the need for the discretization of circular interior boundaries is completely eliminated. This is accomplished by choosing appropriate expansion schemes for the unknown function and their subsequent analytic integration over these circular interior boundaries, i.e., by expressing temperature and its normal derivative on the circular holes in the form of harmonic expansion schemes. The first term of these series is the zeroth-order solution, the second and third terms represent first-order solution and so on. Arimilli and Parang [10] used this method to determine zeroth- and first-order conduction heat transfer from circular tubes embedded in semi-infinite medium with convective boundary conditions. Caulk used a similar formulation to analyze steady heat conduction from infinite number of holes in half-space [11] and in a uniform slab [11] as well as to analyze elastic torsion in a bar with circular holes [12].

An area of practical interest involving steady conduction heat transfer is the molding process in tool design and manufacturing industry. In molding it is often desirable to determine and minimize the temperature variation on the mold cavity surface in order to improve quality and efficiency of the molding process. One way to change the temperature on the cavity surface is to vary the position of heating or cooling lines inside the mold. The locations of these lines can be optimized to produce minimum variation of temperature on the cavity surface of the die or the mold.

Objectives of the Study

The objectives of this study are several. First, the special method of Barone and Caulk is used to formulate the general problem of steady state conduction heat transfer from a set of circular tubes in a general two-dimensional region with arbitrary external boundary. The governing equations are developed from Green's second identity and are nondimensionalized appropriately. Upon discretization of the external boundary and evaluation of integrals on the discreet intervals and their analytical determination over the circular holes, the problem is reduced to the solution of a set of algebraic equations. The unknowns in these equations are temperatures at each of the nodes on the external boundary, and zeroth- and first-order normal derivative of temperature on the circular tubes (holes). A general computer program, BISOL, is developed to solve these equations for both temperature on an arbitrary external boundary and normal derivative of temperature on the circular holes.

To illustrate the numerical procedure, the computer program developed is applied to the specific problem of a mold surface with a cross section composed of three straight normal sides enclosed by an arbitrary contour is considered. The arbitrary contour models the cavity surface of the mold. Various shapes for the cavity surface are chosen for illustration purposes, e.g., semi-circle, combination of a curved and a straight boundary, etc. The temperature on the external boundary, including the cavity surface, and zeroth- and first-order solution of normal derivative of temperature on the circular holes (representing nondimensional heat transfer from circular holes) are numerically computed. These results are discussed in detail.

Another important problem addressed in this thesis is the determination of optimum location of cooling or heating lines for the purpose of designing high-quality molds and dies. To accomplish this the computer program BISOL is incorporated into a program called OPTLOC and is utilized in an optimization scheme where the deviation of cavity surface temperature from a desired reference temperature is minimized by appropriate choice of location of circular tubes. For this purpose the governing set of algebraic equations are solved for the coordinates of the circular holes (tubes), temperature on the external boundary, and normal derivative of temperature on the circular holes. These unknowns are found by using an iterative scheme with the objective of minimizing the cavity-surface temperature deviation. The optimization code CONMIN [14] is adapted and used in conjunction with the program OPTLOC developed in this thesis to give the desired optimum location of the circular holes in a two-dimensional mold with an arbitrary external boundary. For illustration purpose several specific mold geometries with given cavity surface shapes are considered and the developed optimization scheme is applied to them. The results are presented and discussed.

CHAPTER II

BOUNDARY INTEGRAL FORMULATION

Green's Second Identity

Consider a two-dimensional region R containing M number of circular holes centered at points ξ^α with $\alpha = 1,2,3 \ldots M$. Let a_α be the radius of the hole α, ∂c^α its boundary, and ∂R the outer boundary of the region is shown in Figure 1. This region represents a cross section of a two-dimensional heat conductor with circular heating or cooling lines. A steady state condition is assumed. The governing equation for this conduction heat transfer problem is the well-known Laplace equation

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

where ϕ is the temperature in R and satisfies regular boundary conditions on ∂R and ∂c^α. This equation is solved with either the temperature or its derivative specified on the boundaries.

This type of problem can be formulated in terms of boundary integral equation known as Green's second identity [8].

$$\lambda\phi(y) + \int_{\partial R} (\phi \frac{\partial g}{\partial n} - g \frac{\partial \phi}{\partial n})ds + \sum_{\alpha=1}^{M} \int_{\partial c^\alpha} (\phi^\alpha \frac{\partial g}{\partial n} - g \frac{\partial \phi^\alpha}{\partial n})ds = 0 \tag{2}$$

where $\phi(y)$ is the temperature at point y on the boundary and g(x,y) is a harmonic function over the region of interest and ϕ^α and q^α are

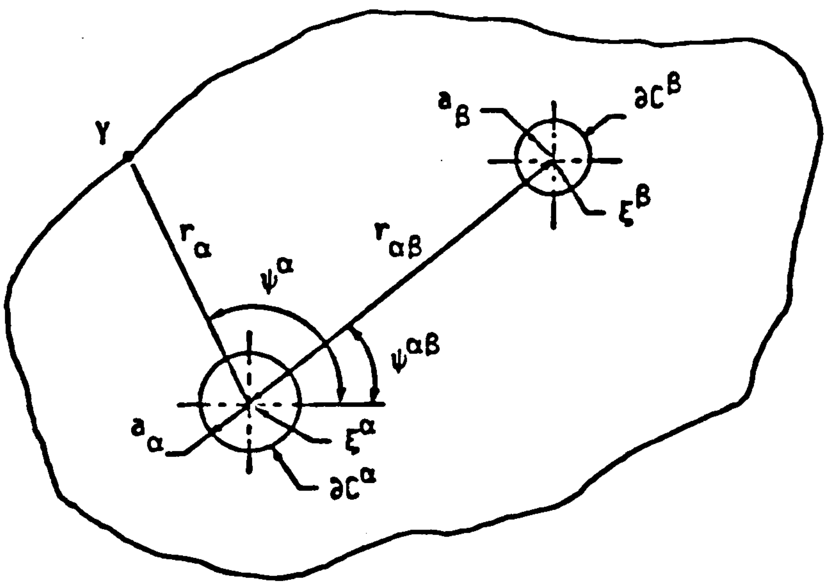


Figure 1. Two-dimensional region with circular tubes: geometry and notation.

Source: M. R. Barone and D. A. Caulk, "Special Boundary Integral Equations for Approximate Solution of Laplace's Equation in Two-Dimensional Region with Circular Holes," The Quarterly Journal of Mechanics and Applied Mathematics, Vol. XXXIV, 265-286 (1981).

the values of ϕ and its outward normal derivative on the hole α. For two-dimensional problem we have [13]

$$g(X,Y) = -\frac{1}{2\pi}\log|\vec{X}-\vec{Y}| \tag{3}$$

where $|\vec{X}-\vec{Y}|$ represent the distance between points X and Y defined in Figure 1. Therefore Eq. 2 relates the values of ϕ and $\frac{\partial\phi}{\partial n}$ at the boundaries of the region of interest. The constant λ in this equation is given by

$$\lambda = \left.\begin{array}{r} 1 \\ 1/2 \\ 0 \end{array}\right\} \begin{array}{l} Y \in R \\ Y \in \{\partial R,\ \partial c^{\alpha}\} \\ Y \notin \{\partial R,\ R,\ \partial c^{\alpha}\} \end{array} \tag{4}$$

Equation 2 forms the basis of the conventional boundary integral equation method. To apply this method, the boundary of the region of interest is divided into small segments. Integral Eq. 2 is then reduced to a set of linear algebraic equations. In applying Eq. 2 to regions with holes as interior boundaries, the boundary of each hole should be divided into a fairly large number of finite segments to gain an adequate representation of the local field variation. This leads to a fine discretization of inner boundary of the region and hence to numerical conditioning problems [8].

Kernel Functions

As mentioned earlier, Barone and Caulk [8] have developed a method that avoids such conditioning problems by carrying out the integration on the holes analytically and thus avoiding the need to discretize the hole boundary. Attention will now be confined to the problem where a constant potential is specified on the surface of each hole, i.e.,

$$\phi^{\alpha} = \phi_0^{\alpha} \quad \text{on} \quad \partial c^{\alpha} \tag{5}$$

and representing q^{α} by finite sum of circular harmonics on ∂c^{α} in the form

$$q^{\alpha} = \frac{\partial \phi^{\alpha}}{\partial n} = q_0^{\alpha} + \sum_{m=1} (q_{1m}^{\alpha} \sin(m\theta^{\alpha}) + q_{2m}^{\alpha} \cos(m\theta^{\alpha}) \tag{6}$$

where q_0 represents the zeroth-order term, and q_{1m}^{α} and q_{2m}^{α} mth-order expansion terms of the normal derivative of temperature. Also θ^{α} is the polar angle centered at ξ^{α} measured relative to ξ axis. To evaluate constants q_0^{α}, q_{1m}^{α}, q_{2m}^{α}, Baroné and Caulk [8] have considered the following kernel functions for the integral equations:

$$g_0^{\alpha}(x) = g(x,\xi^{\alpha}) = -\frac{1}{2\pi} \cdot \log|X-\xi^{\alpha}| \tag{7}$$

$$g_{1m}^{\alpha}(x) = \frac{\sin(m\theta^{\alpha})}{|X-\xi^{\alpha}|^m} \tag{8}$$

$$g_{2m}^{\alpha}(x) = \frac{\cos(m\theta^{\alpha})}{|X-\xi^{\alpha}|^m} \tag{9}$$

where $\beta = 1,2,3 \ldots M$ and m represents the order of approximation. In this study only the zeroth- and first-order terms of the expansion for the normal derivative are considered, i.e., m=1. The first-order solution is obtained by the substitution of g_0^α, g_{11}^α, g_{21}^α, q_0^α, q_{11}^α and q_{21}^α in Green's second identity. Since second or higher orders are not discussed the notation is simplified by omitting the second subscripts in the above. Thus g_0^α, g_1^α, g_2^α, q_0^α, q_1^α and q_2^α are substituted in Eq. 2 to give

$$\int_{\partial R} (\phi \frac{\partial {g_0}^\alpha}{\partial n} - g_0^\alpha \frac{\partial \phi}{\partial n})ds + \sum_{\alpha=1}^{M} (\phi^\alpha \frac{\partial {g_0}^\alpha}{\partial n} - g_0^\alpha(q_0^\alpha + q_1^\alpha \sin\theta + q_2^\alpha \cos\theta))ds = 0 \tag{10}$$

$$\int_{\partial R} (\phi \frac{\partial {g_1}^\alpha}{\partial n} - g_1^\alpha \frac{\partial \phi}{\partial n})ds + \sum_{\alpha=1}^{M} (\phi^\alpha \frac{\partial {g_1}^\alpha}{\partial n} - g_1^\alpha(q_0^\alpha + q_1^\alpha \sin\theta + q_2^\alpha \cos\theta))ds = 0 \tag{11}$$

$$\int_{\partial R} (\phi \frac{\partial {g_2}^\alpha}{\partial n} - g_2^\alpha \frac{\partial \phi}{\partial n})ds + \sum_{\alpha=1}^{M} (\phi^\alpha \frac{\partial {g_2}^\alpha}{\partial n} - g_2^\alpha(q_0^\alpha + q_1^\alpha \sin\theta + q_2^\alpha \cos\theta))ds = 0 \tag{12}$$

Special Boundary Integral Equations

The substitution of Eqs. 3, 7, 8, and 9 into Eqs. 2, 10, 11, and 12 results in the following integral equations (References [8] and [9]).

$$\frac{1}{2}\phi_i + \sum_{j=1}^{n}\left(\phi_j \int_{c_j} \frac{\partial g_i}{\partial n}\, ds - q_j \int_{c_j} g_i\, ds\right) + \sum_{\alpha=1}^{m} a_\alpha [q_0^\alpha \log r_{\alpha i}$$

$$- \frac{1}{2} b_{\alpha i} (q_1^\alpha \sin \psi^{\alpha i} + q_2^\alpha \cos \psi^{\alpha i})] = 0 \tag{13}$$

$$\sum_{j=1}^{n} \left(\phi_j \int_{c_j} \frac{\partial g_0^\beta}{\partial n} ds - q_j \int_{c_j} g_0^\beta \, ds \right) + \phi_0^\beta + q_0^\beta \, a_\beta \log a_\beta$$

$$\sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{m} a_\alpha [q_0^\alpha \log r_{\alpha\beta} - \frac{1}{2} b_{\alpha\beta}(q_1^\alpha \sin \psi^{\alpha\beta} + q_2^\alpha \cos \psi^{\alpha\beta})] = 0 \tag{14}$$

$$\sum_{j=1}^{n} \left(\phi_j \int_{c_j} \frac{\partial g_1^\beta}{\partial n} ds - q_j \int_{c_j} g_1^\beta \, ds \right) - \pi q_1^\beta + \pi \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{m} [2b_{\alpha\beta} \, q_0^\alpha \sin \psi^{\alpha\beta}$$

$$+ b_{\alpha\beta}^2 (q_2^\alpha \sin 2\psi^{\alpha\beta} - q_1^\alpha \cos 2\psi^{\alpha\beta})] = 0 \tag{15}$$

$$\sum_{j=1}^{n} \left(\phi_j \int_{c_j} \frac{\partial g_2^\beta}{\partial n} ds - q_j \int_{c_j} g_2^\beta \, ds \right) - \pi \, q_2^\beta + \pi \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{m} [2b_{\alpha\beta} \, q_0^\alpha \cos \psi^{\alpha\beta}$$

$$+ b_{\alpha\beta}^2 (q_1^\alpha \sin 2\psi^{\alpha\beta} + q_2^\alpha \cos 2\psi^{\alpha\beta})] = 0 \tag{16}$$

where

$$\begin{aligned} b_\alpha &= a_\alpha / r_\alpha, \qquad & r_\alpha &= |y-\xi^\alpha|, \qquad & \psi^\alpha &= \theta^\alpha(y) \\ b_{\alpha\beta} &= a_\alpha / r_{\alpha\beta}, \qquad & r_{\alpha\beta} &= |\xi^\alpha - \xi^\beta|, \qquad & \psi^{\alpha\beta} &= \theta^\alpha(\xi^\beta) \; . \end{aligned} \tag{17}$$

and c_j's are the discrete segments belonging to the outer boundary. Here it is assumed that boundary temperature and its normal derivative are constant over each finite segment of the outer boundary c_j. Y_i is a boundary point on c_i and

$$\phi_i = (Y_i)$$

$$q_i = \frac{\partial\phi(Y_i)}{\partial n} .$$

Equation 13 is the result of using g(x,y) as the kernel function. It represents N algebraic equations corresponding to the N points on the external boundary. Equations 14, 15, and 16 are the result of using g_0^α, g_1^α, and g_2^α as the kernel functions in the Green's identity, respectively. For M tubes in the region R Eqs. 14, 15, and 16 each represent M equations. The total number of resulting algebraic equations are (N+3M) and are to be solved for N number of nodal temperatures on the external boundary ∂R, ϕ_i^0, and three unknowns q_0^α, q_1^α, q_2^α on each of M holes.

CHAPTER III

ANALYSIS

The boundary integral equations (13, 14, 15, 16) presented in the previous section are solved to obtain the temperatures at the node points on the boundary and the zeroth- and first-order normal gradient of temperature at the hole boundaries. These equations will be applied to the following problem of interest.

Consider a two-dimensional region R with M number of circular holes where a constant temperature is specified on the boundary of each hole. The region R represents a heat conductor with a constant thermal conductivity. This heat conductor is surrounded by ambient air at a temperature ϕ_a.

Taking advantage of the linear nature of the governing differential equation and boundary conditions, a new temperature is defined by

$$\phi' = (\phi - \phi_a)$$

$$\phi_0^{\alpha'} = (\phi_0^\alpha - \phi_a) \ .$$

This makes it possible to recast the problem discussed into a new one with simpler boundary conditions. The Laplace equation for for region R is now of the form

$$\nabla^2\phi' = 0 \tag{18}$$

and the boundary conditions on R are

$$\phi' = \phi_0^{\alpha'} \quad \text{on} \quad \partial c^{\alpha} \tag{19}$$

$$-K \frac{\partial \phi'}{\partial n} = h \ \Phi' \quad \text{on } \partial R \tag{20}$$

where h is the convection heat transfer coefficient and n is the normal unit vector on ∂R.

We now nondimensionalize all lengths in these equations by the radius A of the largest diameter tube. The nondimensional temperature Φ^* is defined by

$$\Phi^* = \frac{\phi'}{\phi_c - \phi_a} \tag{21}$$

where ϕ_c is some specified reference temperature over a portion of the outer boundary and ϕ_a is the far field ambient temperature. Similarly

$$\Phi_0^{\alpha *} = \frac{\phi_0^{\alpha'}}{\phi_c - \phi_a} \ . \tag{22}$$

Upon nondimensionalization of Eqs. 13, 14, 15, 16 and dropping superscript & we obtain the dimensionless form of these equations as follows.

$$\frac{1}{2}\phi_i + \sum_{j=1}^{N}\left(\phi_J \int_{C_J} \frac{\partial g_i}{\partial n}\, ds - q_j \int_{c_j} [g_i + \log A]\right) ds$$

$$+ a_\alpha\left((\log r_{\alpha i} + \log A) q_0^\alpha - \frac{1}{2} b_{\alpha i}(q_1^\alpha \sin \psi^{\alpha i} + q_2^\alpha \cos \psi^{\alpha i})\right) = 0 \quad (23)$$

$$\sum_{j=1}^{N}\left(\phi_j \int_{c_j} \frac{\partial g_0^\beta}{\partial n}\, ds - q_j \int_{c_j} (g_0^\beta + \log A) ds\right) + q_0^\beta\, a_\beta(\log a_\beta + \log A)$$

$$+ \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} a_\alpha\left(q_0^\alpha(\log r_{\alpha\beta} + \log A) - \frac{1}{2} b_{\alpha\beta}(q_1^\alpha \sin \psi^{\alpha\beta} + q_2^\alpha \cos \psi^{\alpha\beta}\right) = -\Phi_0^\beta \quad (24)$$

$$\sum_{j=1}^{N}\left(\phi_j \frac{1}{A} \int_{c_j} \frac{\partial g_1^\beta}{\partial n}\, ds - q_j \frac{1}{A} \int_{c_j} g_1^\beta\, ds\right) - \frac{\pi}{A} q_1^\beta$$

$$+ \frac{\pi}{A} \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} [2b_{\alpha\beta}\, q_0^\alpha \sin \psi^{\alpha\beta} + b_{\alpha\beta}^2(q_2 \sin 2\psi^{\alpha\beta} - q_1^\alpha \cos 2\psi^{\alpha\beta})] = 0 \quad (25)$$

$$\sum_{j=1}^{N} \phi_j \frac{1}{A} \int_{c_j} \frac{\partial g_2^\beta}{\partial n}\, ds - \frac{q_j}{A} \int_{c_j} g_2^\beta\, ds - \frac{\pi}{A} q_2^\beta$$

$$+ \frac{\pi}{A} \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} 2b_{\alpha\beta} \cos \psi^{\alpha\beta}\, q_0^\alpha + b_{\alpha\beta}^2(q_1^\alpha \sin 2\psi^{\alpha\beta} + q_2^\alpha \cos 2\psi^{\alpha\beta}) = 0 \quad (26)$$

where

$$b_\alpha = a_\alpha / r_\alpha, \qquad r_\alpha = |y - \xi^\alpha|, \quad \psi^\alpha = \theta^\alpha(y)$$

$$b_{\alpha\beta} = a_\alpha / r_{\alpha\beta}, \qquad r_{\alpha\beta} = |\xi^\alpha - \xi^\beta|, \quad \psi^{\alpha\beta} = \theta^\alpha(\xi^\beta)$$

The nondimensional boundary conditions for this problem are:

$$\Phi = \Phi_0^{\alpha} \quad \text{on} \quad \partial c^{\alpha} \tag{27}$$

$$q_i = \frac{\partial \Phi_i}{\partial n} = - Bi \cdot \Phi_i \quad \text{on} \quad \partial R \tag{28}$$

where B_2^0 = ha/k is the Biot number. We can now make use of Eq. 28 to formulate $q_1, q_2, \ldots, q_j, \ldots, q_n$ in terms of temperatures $\Phi_1, \Phi_2, \ldots, \Phi_j, \ldots \Phi_n$ on the outer boundary of the region.

As discussed in Chapter II for each node point Y_i on the outer boundary there will be one algebraic equation resulting from Eq. 13 and for each hole Eqs. 14-16 will result in three algebraic equations. Thus there will be (N+3M) algebraic equations to be solved, N being the number of node points and M, the number of circular holes. The integrals in Eqs. 23-26 need to be evaluated on specified external boundaries of region R. In this thesis several boundaries of interest for region R are investigated. The evaluation of these integrals over these boundaries are presented in Appendix A for various geometries of interest. The algebraic equations obtained can be cast in the matrix form

$$AX = B \tag{29}$$

where $X = (\Phi_1, \Phi_2, \ldots, \Phi_n, q_0^1, q_1^1, q_2^1, \ldots, q_0^M, q_1^M, q_2^M)$ and A is the matrix of coefficients. The matrix A is computed from the integral equations and is a full matrix. Although A is a full matrix it tends to be well conditioned because the dominant entries lie on or near the diagonal [8]. The resulting matrix system is solved on a Dec-10 computer using the IMSL subroutine LEQT1F.

CHAPTER IV

OPTIMIZATION

A computer program is developed which by solving matrix Eq. 29 determines temperature on the external boundary and normal derivatives of temperature on the surface of the tubes in a two-dimensional heat conductor. It is assumed that the constant temperature on tube surface and the tube locations are specified. This program is presented in Appendix C.

One of the objectives in thermal design of molds or dies is to maintain a uniform temperature on the cavity surface. The external boundary of the heat conductor is fixed and from practical considerations the radius and surface temperature of the tubes are assumed constant. The design objective therefore should be achieved by varying the location of the tubes inside the heat conductor. Thus we consider two design variables for each hole (tube): ξ_1^α and ξ_2^α, the coordinates at the center of the hole.

The design objective is to minimize the distributed difference between the nodal temperature Φ_i and the desired temperature ϕ_c over some portion of the outer boundary (the cavity surface). This distributed difference is expressed as a least squared difference of the form

$$OBJ = \sum_{Y_i \in \partial R} (\Phi_i - 1) \tag{30}$$

where Φ is the nondimensional temperature given by

$$\Phi_i = \frac{\phi_i - \phi_a}{\phi_c - \phi_a}$$

ϕ_a is the ambient temperature outside the heat conductor. The normal derivative of temperature q_j can be formulated in terms of temperature Φ_j at the boundaries. That is, the general boundary condition on ∂R can be expressed as

$$q_j = \frac{\partial \Phi_j}{\partial n} = -Bi \ \Phi_j + p$$

Bi is the Biot number and p is the heat of reaction absorbed by the heat conductor expressed as a uniform heat flux across the cavity surface. Usually p is zero in most applications.

The constraints on this optimization problem are chosen to be minimum spacing acceptable between the holes and minimum spacing acceptable between the circular tube and the outer boundary. These constraints can be explicitly expressed as follows [8]:

$$|Y_i - \xi^\alpha| \geq a_\alpha + c_1 \qquad \alpha = 1, 2, \ldots M, \ i = 1,], \ldots N \tag{32}$$

$$|\xi^\alpha - \xi^\beta| \geq a_\alpha + a_\beta + c_2 \qquad \alpha, \beta = 1, 2, \ldots, M \tag{33}$$

where c_1 and c_2 are specified clearances. They are taken to be 0.05 in this study.

The nodal nondimensional temperature Φ_i which are represented in the objective function (30) depend on the location of the tubes through the solution of the linear system (29). Hence

$$\Phi_i = \Phi_i(\xi_1^\alpha, \xi_2^\alpha) \ . \tag{34}$$

The optimization problem formulated above must be solved numerically. This objective is achieved with the use of the code CONMIN [14]. CONMIN is a FORTRAN program in subroutine form for the minimization of a multivariable function subject to a set of inequality constraints. The main algorithm used in CONMIN is the Method of Feasible Directions. The gradient information of the objective function is calculated internally by finite difference unless analytic gradients are supplied by the user. Further details on this code and the optimization techniques may be found from References 14 and 15. The definition of necessary parameters in the code and their values used in this study is presented in Appendix B. The computer program of Appendix C is used in conjunction with CONMIN for the determination of the optimal locations of cooling or heating lines in two-dimensional molds. The resulting optimization program developed is presented in Appendix D. The results of the optimization problem will be discussed in Chapter VI.

CHAPTER V

HEAT TRANSFER RESULTS

Verification of the Numerical Procedure ($Bi \rightarrow \infty$)

The computer program developed and presented in Appendix C can be tested for accuracy by comparing its results with available analytical solution for a specific two-dimensional configuration and boundary conditions.

Consider a circular annulus with internal and external boundaries each at a different constant temperature. Using the same nondimensionalization used in Chapter III, assume the dimensionless inner radius of the annulus is 1.0 and the dimensionless outer radius is r. The inner and outer nondimensional temperatures are assumed 1.0 and 0.0, respectively. The analytical solution for radial heat transfer can easily be determined to be

$$q_0 = \frac{T_1 - T_2}{r_1 \log \frac{r}{r_1}} \tag{35}$$

where $T_1 = 1.0$, $T_2 = 0.0$, and $r_1 = 1.0$.

In order to verify the accuracy of the numerical procedure developed, the above problem was solved using the computer program presented in Appendix C. Here we set $\alpha = 1$, $a_\alpha = 1$, and $\Phi_0^\alpha = 1$. The constant temperature on the outer boundary is obtained by selecting a large values for Bio numbers at the external boundaries (approximately 10000). The numerical solution for the heat transfer showed excellent agreement with the analytical results and gave a maximum deviation of 0.3 percent. This adds confidence to the applicability and accuracy of numerical method employed and the computer program presented and discussed in this thesis.

Heat Transfer Results for the Selected Configurations

Several geometries of interest were assumed for region R. The numerical computation of zeroth- and first-order coefficients in the harmonic expansion (Eq. 6) were computed for various convecting boundary conditions imposed on region R. For all the configurations to follow, a_α and Φ_0^α are each assumed to be 1.0.

First the simplest case of a square region with one circular hole located on its vertical axis of symmetry was considered. Each side of the square was assumed to be L/A = 25 and the center of the hole was taken to be distance D above the lower horizontal side of the square (see inset of Figure 2). All sides of the square were subjected to the same convective boundary condition and hence are at equal Biot numbers. The computed values of q_0 and q_1 as a function of D/A for various Biot numbers are shown in Figures 2 and 3, respectively. As expected the minimum of q_0 and zero values of q_1 shown in these figures occur when the hole is at the center of the square. The values of q_2 are uniformly zero for all values of D/A due to the vertical symmetry of the problem considered.

Another case considered was similar to the previous problem except for the top boundary of the square subjected to a different

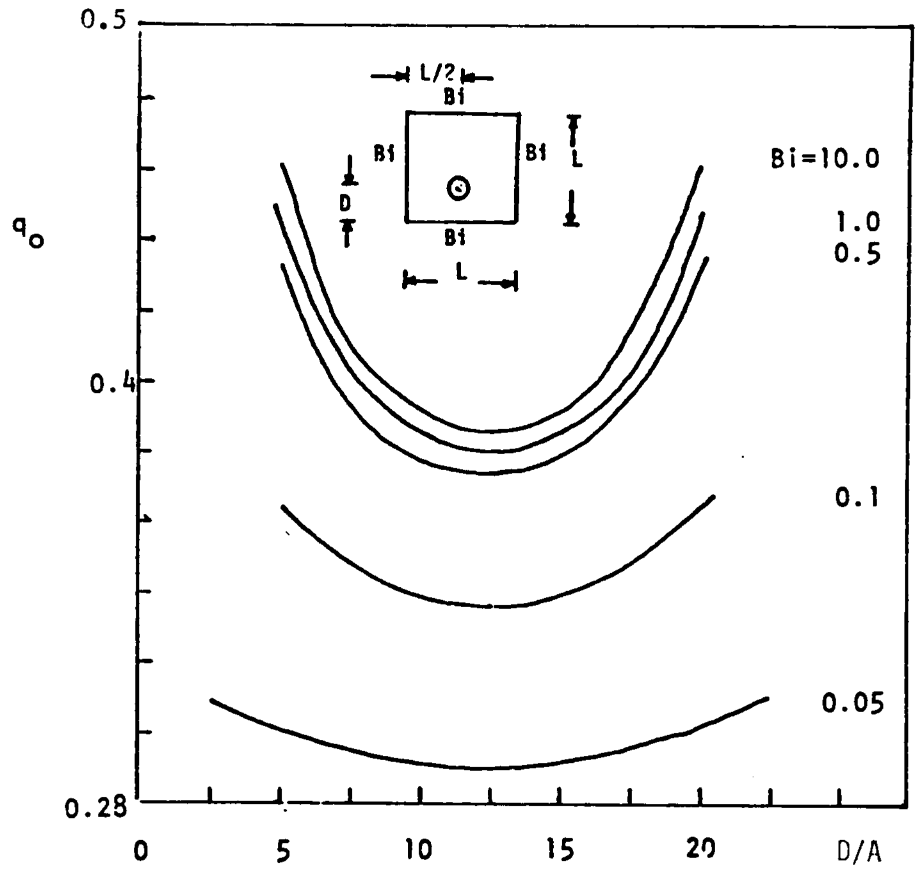


Figure 2. Dimensionless "zeroth order" normal temperature gradient, q_o, as a function of D/A for various Biot numbers (L/A = 25.0).

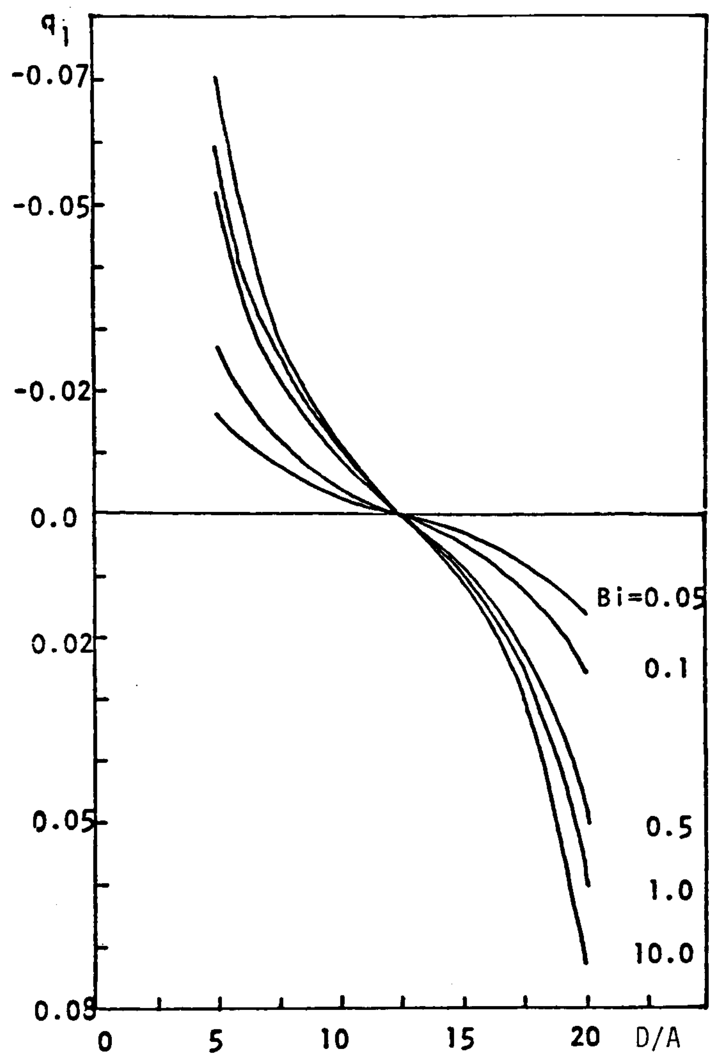


Figure 3. Dimensionless "first order" normal temperature gradient, q_1, as a function of dimensionless distance D/A for various Biot numbers (L/A = 25.0).

Biot number, Bi (see inset of Figure 4). For illustration purposes, we let Bi = 0.05 and results for q_0 as a function of D/A for various Biot numbers are presented in Figure 4. It is observed that the nonsymmetry of the convective conditions imposed on the boundaries result in a nonsymmetric profile for q_0 especially for large values of Biot numbers, $\overline{Bi}$.

The next example considered was the problem of two symmetrically located holes in the square region of the previous problem. The center of two holes was assumed to be a distance S apart (see inset of Figure 5). All sides of the square were again subjected to the same Biot number. The sum of values of q_0 for both holes was normalized using the results for the case of a single tube located at the center of the square (q_{0S}). The normalized results are shown in Figure 5 as a function of S/A for various Biot numbers. The computed values of q_1 are found to be zero due to the horizontal symmetry of the problem. The results for q_2 are normalized in respect to q_0 values and are shown in Figure 6.

The next example chosen was a region composed of three sides of L/A = 25 enclosed by a semi-circle and containing two circular holes with a fixed spacing of S/A distance apart (see inset of Figure 7). All boundaries are subject to the same Biot number. The computed values of q_0 as a function D/A and Biot number are shown in Figure 7. Figures 8 and 9 show the results of q_1 and q_2 (normalized using q_0) as a function of D/A for various Biot numbers. The symmetry of the problem makes q_2 values for the two tubes equal

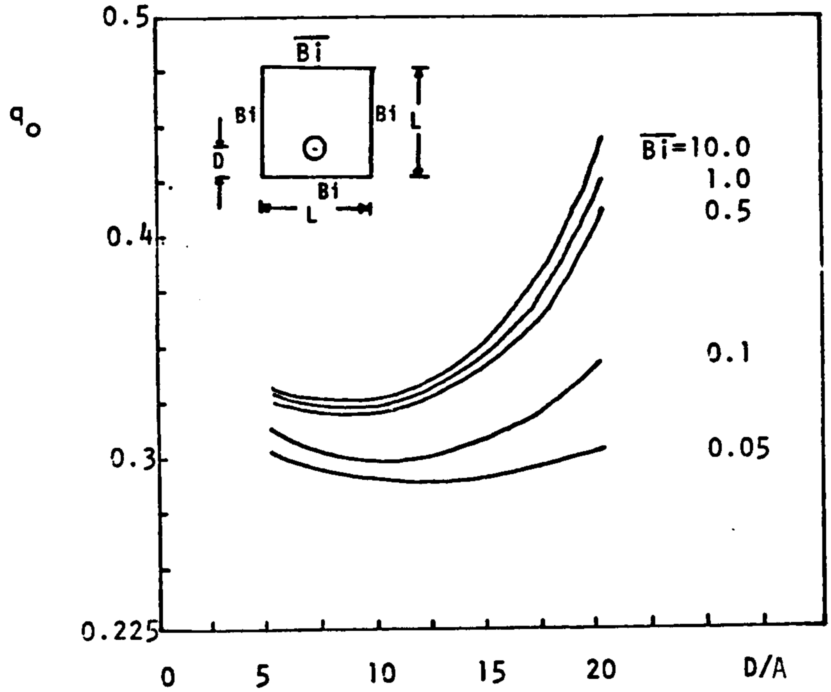


Figure 4. Dimensionless "zeroth order" normal temperature gradient, q_o, as a function of dimensionless distance D/A for various Biot numbers on top surface (Bi = 0.05, L/A = 25.0).

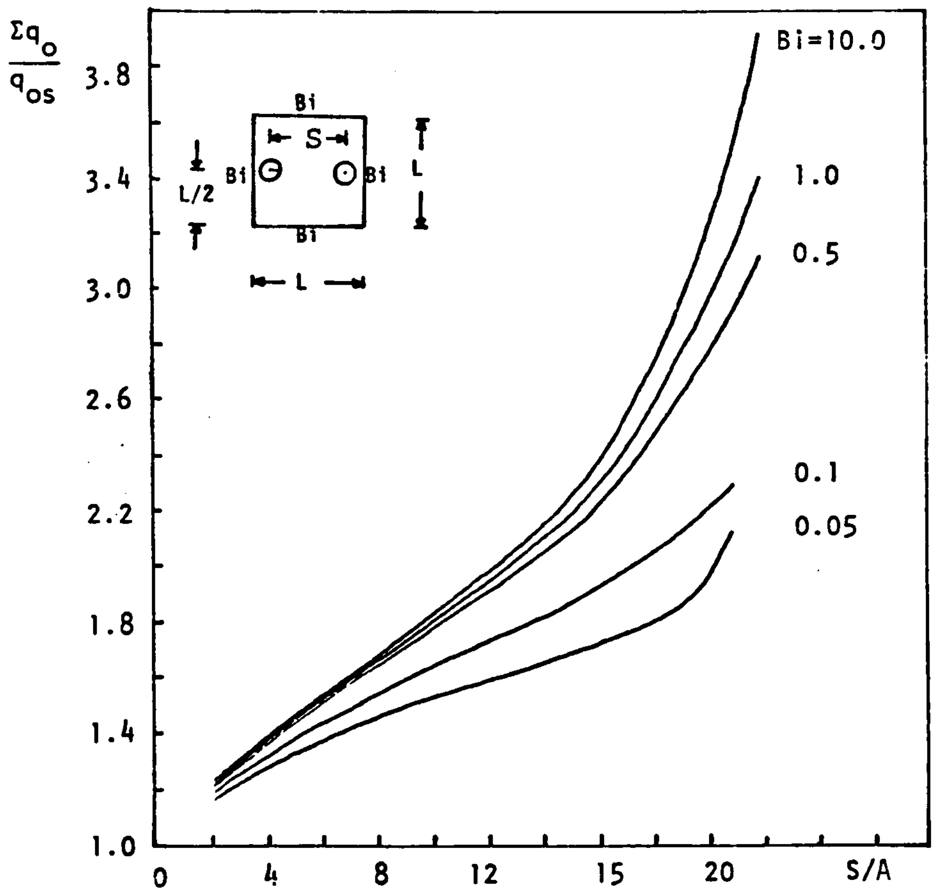


Figure 5. Dimensionless "zeroth order" normal temperature gradient, $\Sigma q_0/q_{0_S}$, as a function of dimensionless hole spacing S/A for various Biot numbers (L/A = 25.0).

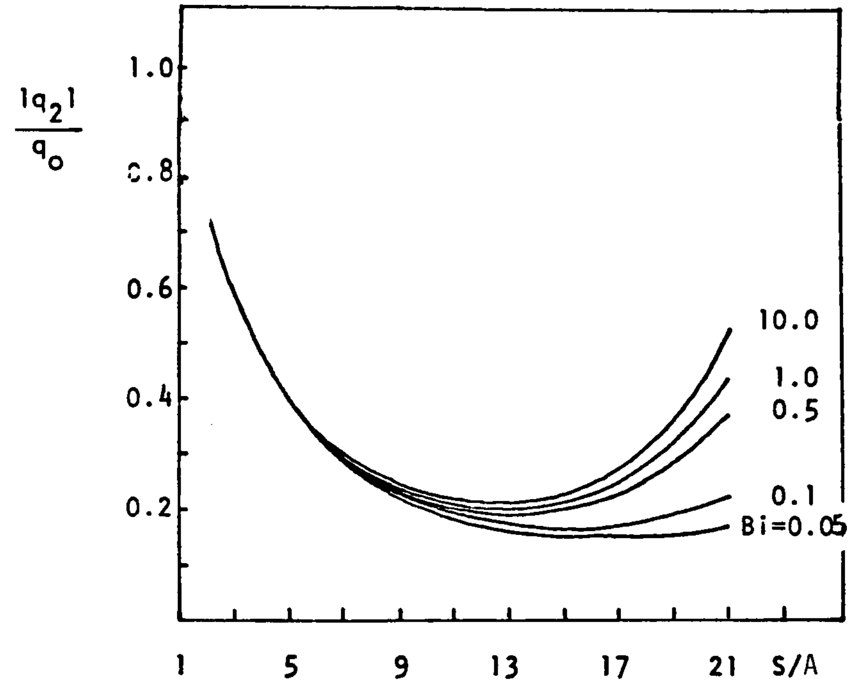


Figure 6. Ratio of normal temperature gradients, q_2/q_0, as a function of dimensionless hole spacing S/A for various Biot numbers (L/A = 25.0).

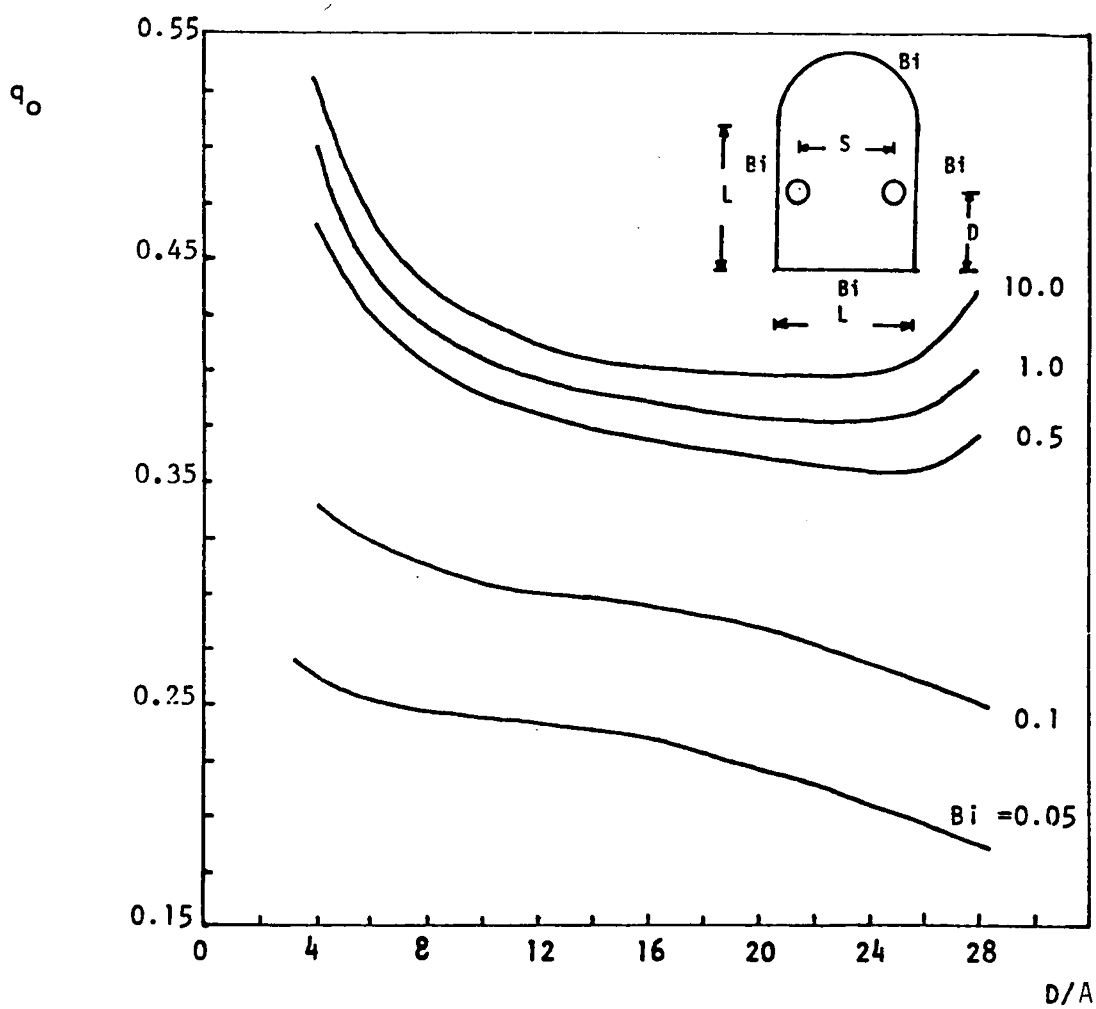


Figure 7. Dimensionless "zeroth order" normal temperature gradient, q_o, as a function of dimensionless distance D/A for various Biot numbers (S/A = 15.0, L/A = 25.0).

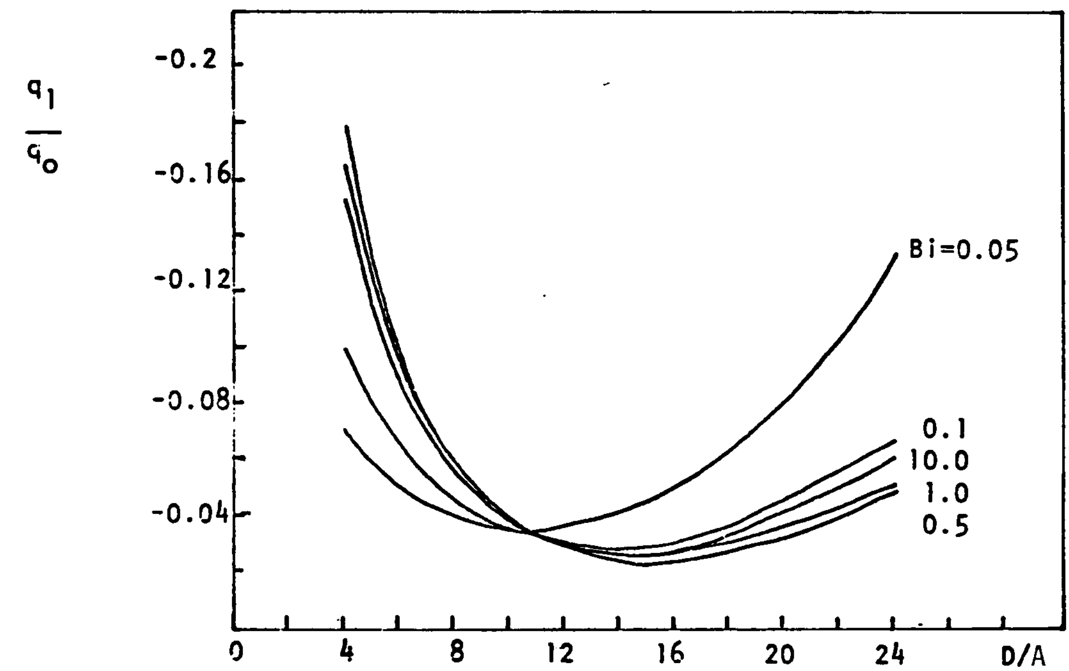


Figure 8. Ratio of dimensionless normal temperature gradients, q_1/q_0, as a function of dimensionless distance D/A for various Biot numbers (S/A = 15.0, L/A = 25.0).

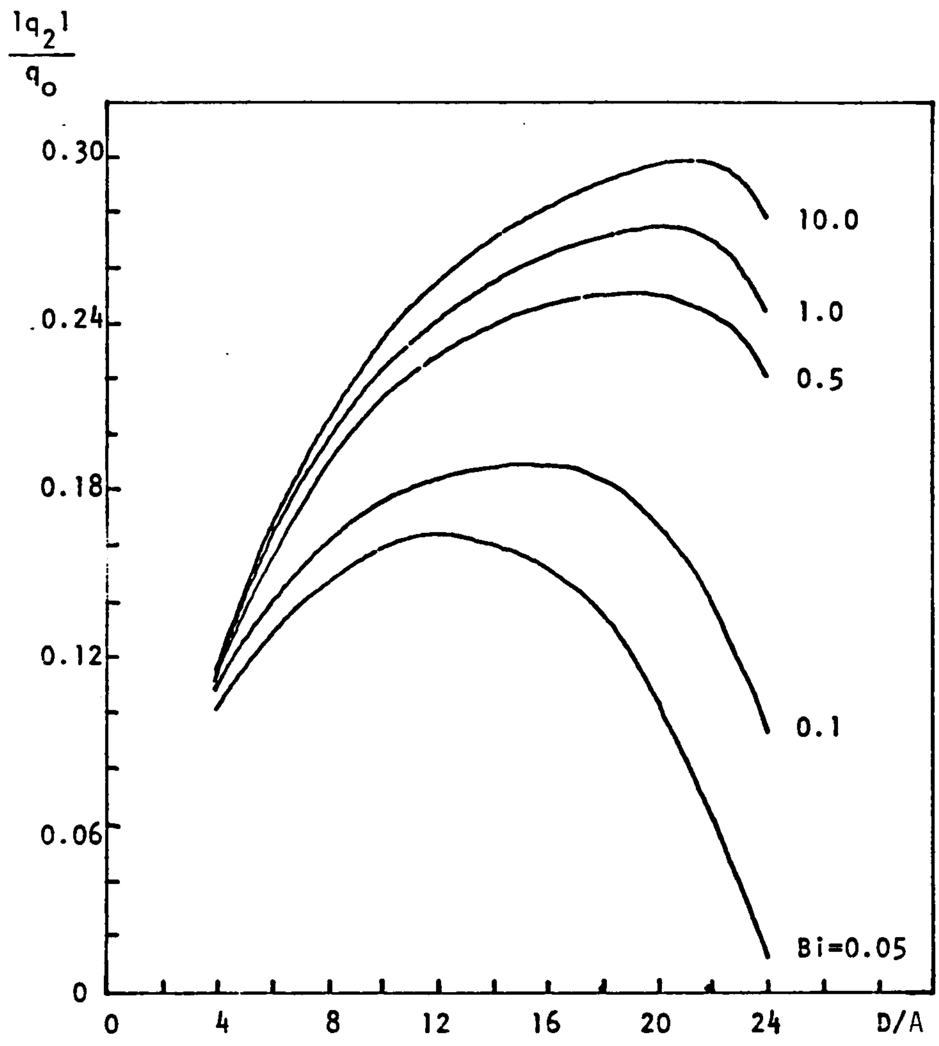


Figure 9. Ratio of dimensionless normal temperature gradient, q_2/q_o, as a function of dimensionless distance D/A for various Biot numbers (S/A = 15.0, L/A = 25.0).

in magnitude but opposite in sign. Therefore, $|q_2|$ value is used in Figure 9. These results indicate that the first-order results can be significant (up to 30 percent of q_0) for large Biot numbers.

Another example chosen was the replacement of the semicircular cavity surface of the previous problem with an S-shaped contour as shown in Figure 10. The results for q_0, q_1 and q_2 are tabulated in Table 1.

The CPU time required for the solution of temperature on the outer region and the solution of zeroth- and first-order normal derivative of temperature was less than 1 second. For example, the simplest case of a square region with one circular hole required a CPU time of 0.63 seconds with 20 discrete segments on each side of the square.

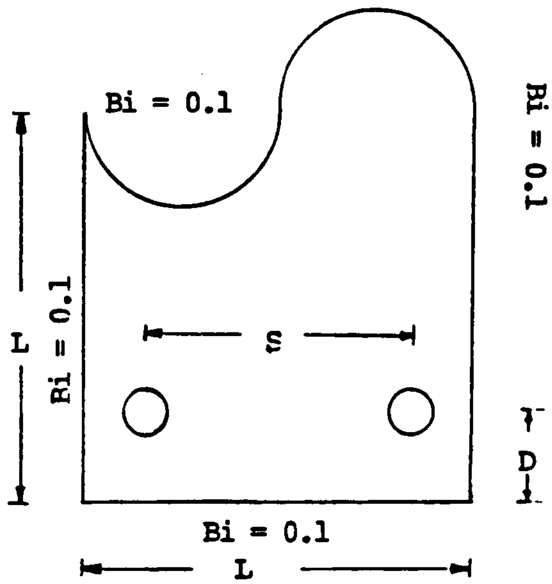


Figure 10. Sketch of the problem with an S-shaped contour analyzed with the results shown in Table I (L/A = 25.0).

TABLE I

VALUES OF q_0, q_1, AND q_2 FOR THE TWO HOLES IN THE REGION SHOWN AS A FUNCTION OF D/A (S/A = 15)

D/A	$\alpha = 1$			$\alpha = 2$		
	q_0	q_1	q_2	q_0	q_1	q_2
4	0.357	-0.033	-0.041	0.359	-0.033	0.039
6	0.341	-0.019	-0.048	0.341	-0.018	0.043
8	0.334	-0.010	-0.052	0.330	-0.010	0.046
10	0.332	-0.003	-0.054	0.325	-0.005	0.048
12	0.335	+0.003	-0.055	0.322	-0.001	0.047
14	0.345	0.013	-0.055	0.321	-0.001	0.045
16	0.368	0.022	-0.057	0.322	+0.003	0.041

CHAPTER VI

OPTIMIZATION RESULTS

In this section several geometrical configurations are selected for the mold surface in order to illustrate the use of the general computer program (Appendix D) in the optimization problem. Therefore, four different geometries for mold surfaces are selected and labeled as Cases I through IV. In all of the above cases all tubes are assumed to be of the same size, $a_\alpha = 1.0$, and at the same temperature which is selected arbitrarily at $\Phi_0^\alpha = 1.24$.

Case I corresponds to the rectangular region shown in Figure 11. Assuming two heating lines the hole locations were optimized for a set of boundary conditions on the external boundary. Here Biot numbers on all four external boundaries of the region are all taken to be equal to 0.2. The results of cavity surface temperature and coordinates of tube locations are shown in Figures 12 and 13, respectively.

Next the effect of nonuniform convection conditions on optimal hole locations are considered. Assuming different and nonuniform Biot numbers and $p = 0.5$, the results for the temperature distribution on the cavity surface and the optimal tube locations are presented in Figures 14 and 15.

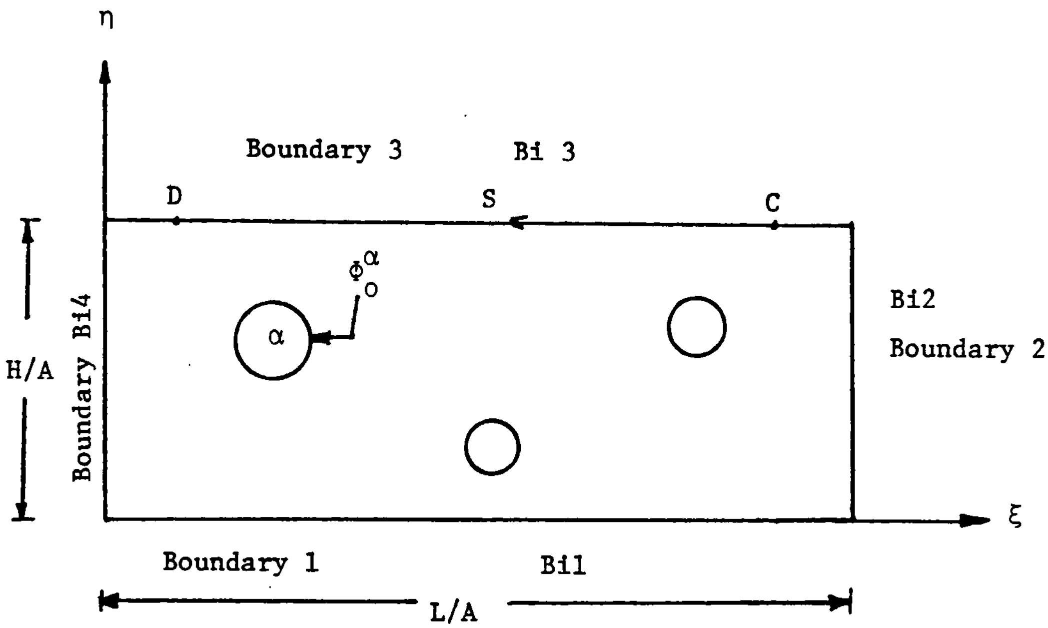


Figure 11. Sketch of the two-dimensional rectangular region with circular holes and convective boundary conditions used in Case I.

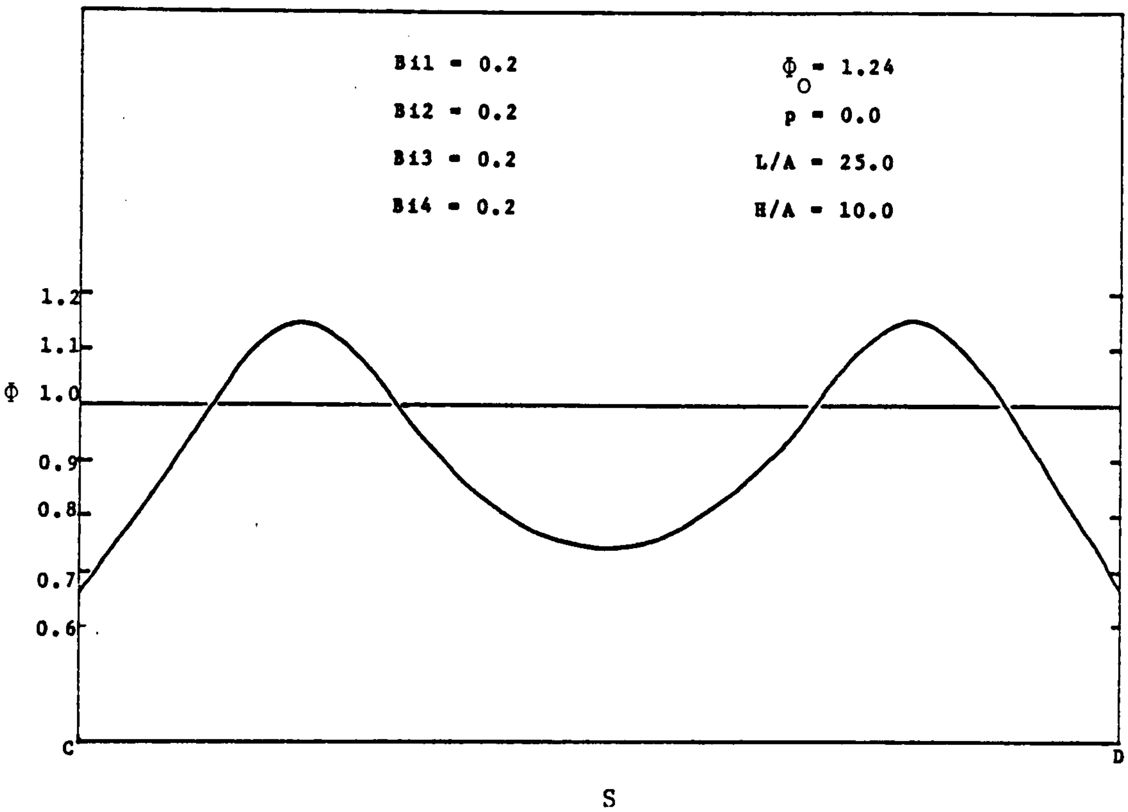


Figure 12. Variation of nondimensional temperature, Φ, along the cavity surface—Case I [optimal hole locations: (7.19, 7.51), (18.54, 7.49)].

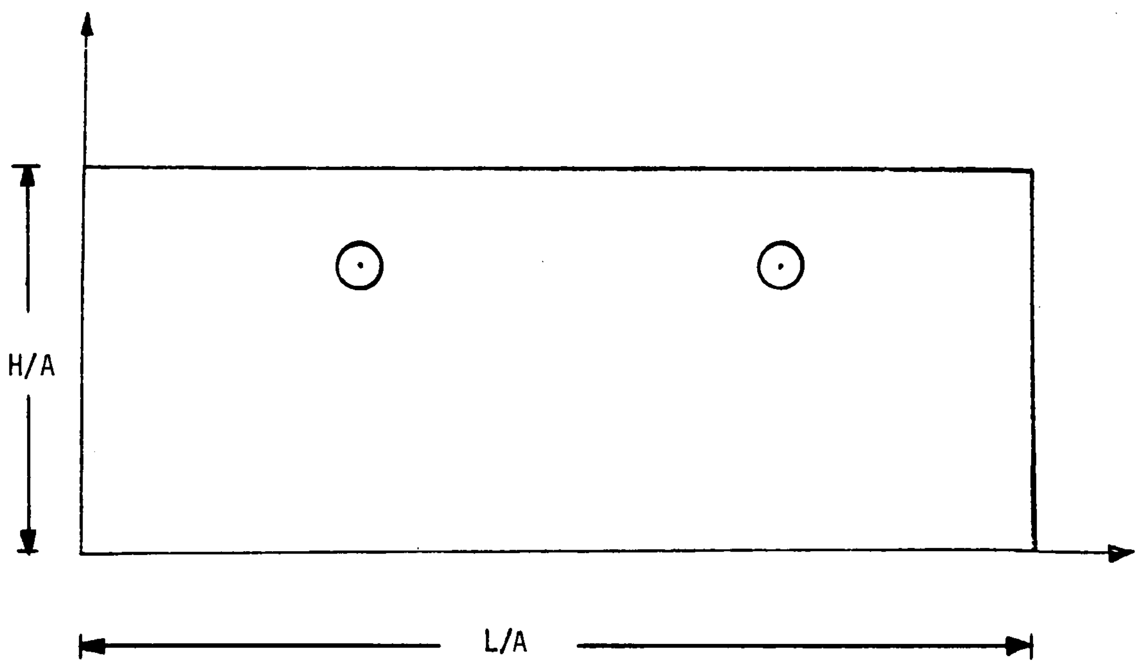


Figure 13. Optimal tube locations for Case I (all conditions defined in Figure 12).

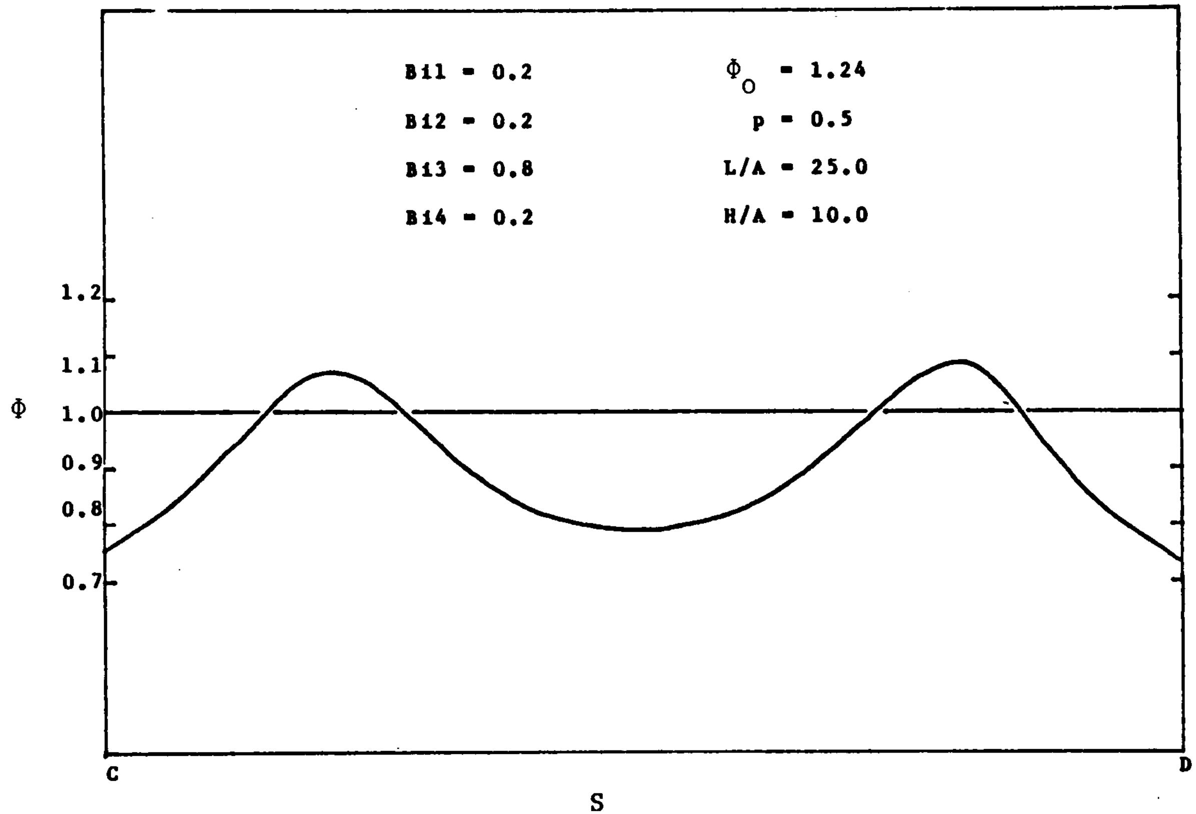


Figure 14. Variation of nondimensional temperature, Φ, along the cavity surface—Case I [optimal hole locations: 7.25, 7.71), (18.31, 7.69)].

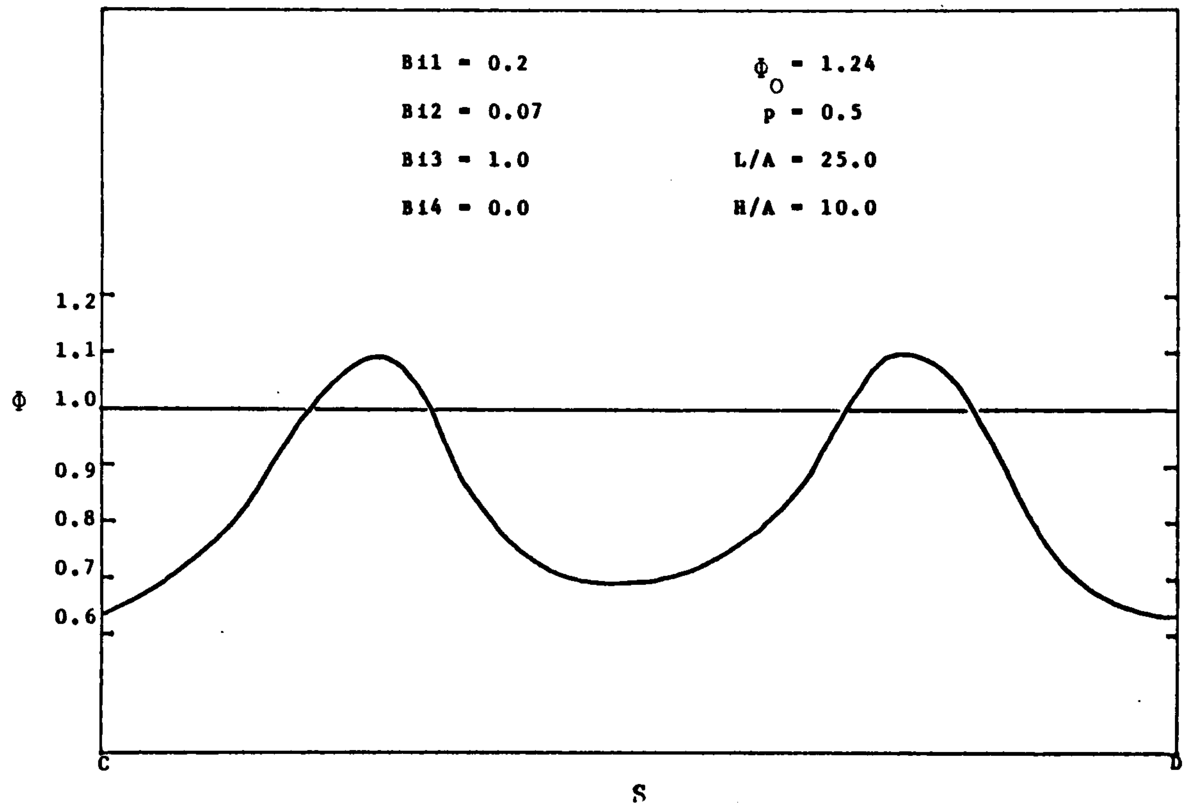


Figure 15. Variation of nondimensional temperature, Φ, along the cavity surface—Case I [optimal hole locations: (7.98, 8.22), (17.71, 8.17)].

In another numerical computation the rectangular region of Case I was chosen with three heating lines instead of two and their locations were optimized. The temperature distribution on the cavity surface and the optimal hole locations are shown in Figures 16 and 17, respectively. It can be observed that the temperature distribution shows less fluctuations than the results for the region with two holes.

Case II corresponds to the configuration shown in Figure 18. Here the cavity surface is assumed to be a semicircle. This geometry was used with both two and three holes in the optimization problem and the temperature on the external boundary as well as the optimal locations of holes in each case were determined. The temperature distribution on the cavity surface with two holes in the region is shown in Figure 19. Here Biot numbers on all four external boundaries of the region are assumed to be the same (0.2). In Figures 20 and 21 the temperature distribution for Case II configuration with three holes and coordinates of the optimal tube locations is shown, respectively. The numerical values of the Biot numbers on all boundaries is assumed to be the same as the previous case. As in the rectangular region, the temperature distribution on the cavity surface has smaller fluctuations and is more uniform for the three-hole region than the two-hole region. To illustrate the variation of Biot number on the cavity surface results for temperature distribution and optimal hole locations were obtained for Bi3 = 0.5 and Bi3 = 0.8. In these results nondimensional heat of reaction into the heat

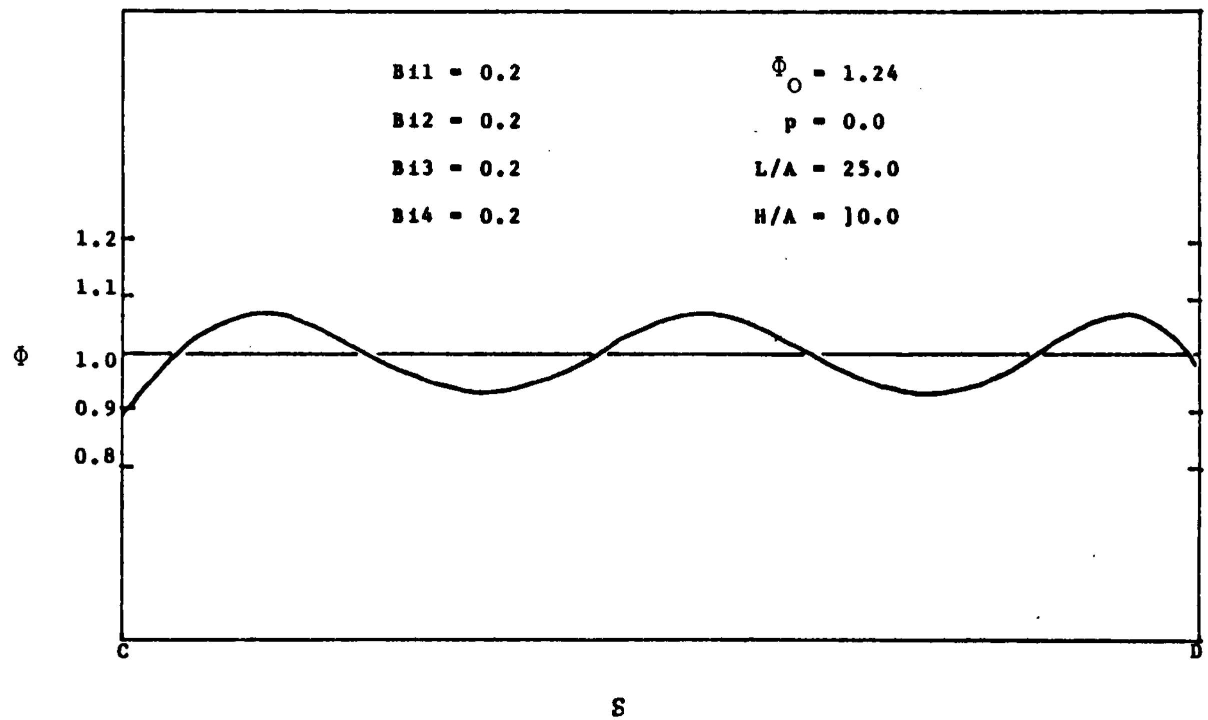


Figure 16. Variation of nondimensional temperature, Φ, along the cavity surface for three holes—[optimal hole locations: (12.22, 6.83), (20.50, 7.06), (4.71, 7.75)].

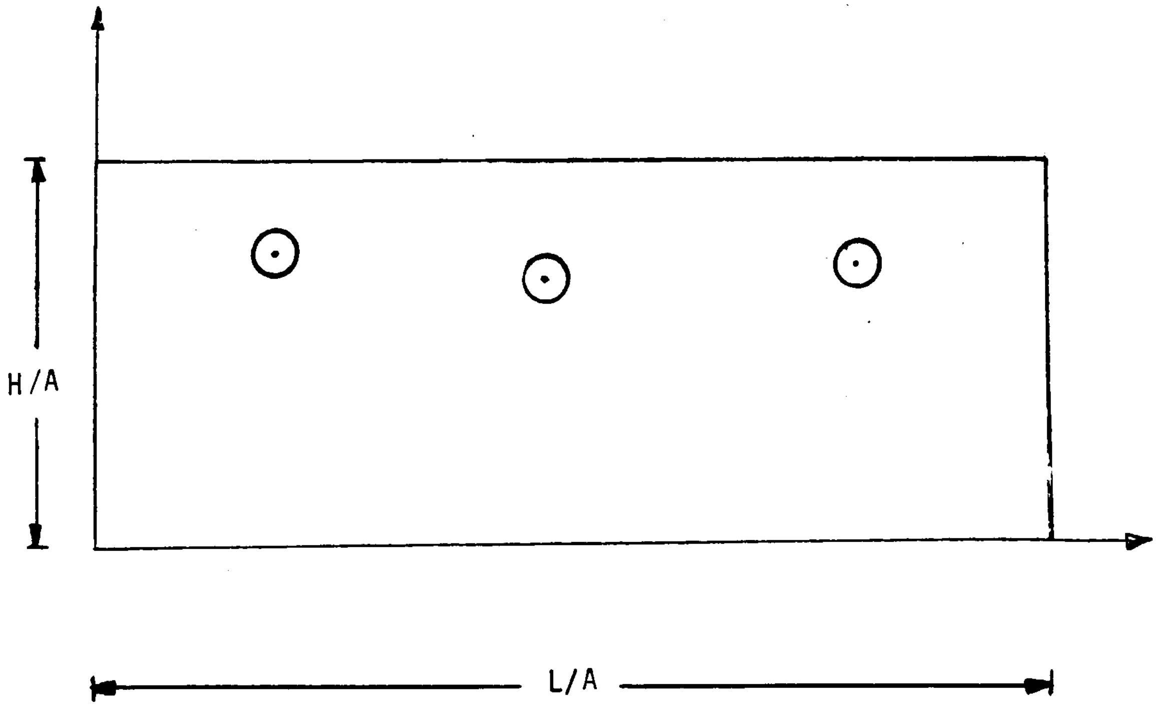


Figure 17. Optimal tube locations for Case I (all conditions defined in Figure 16).

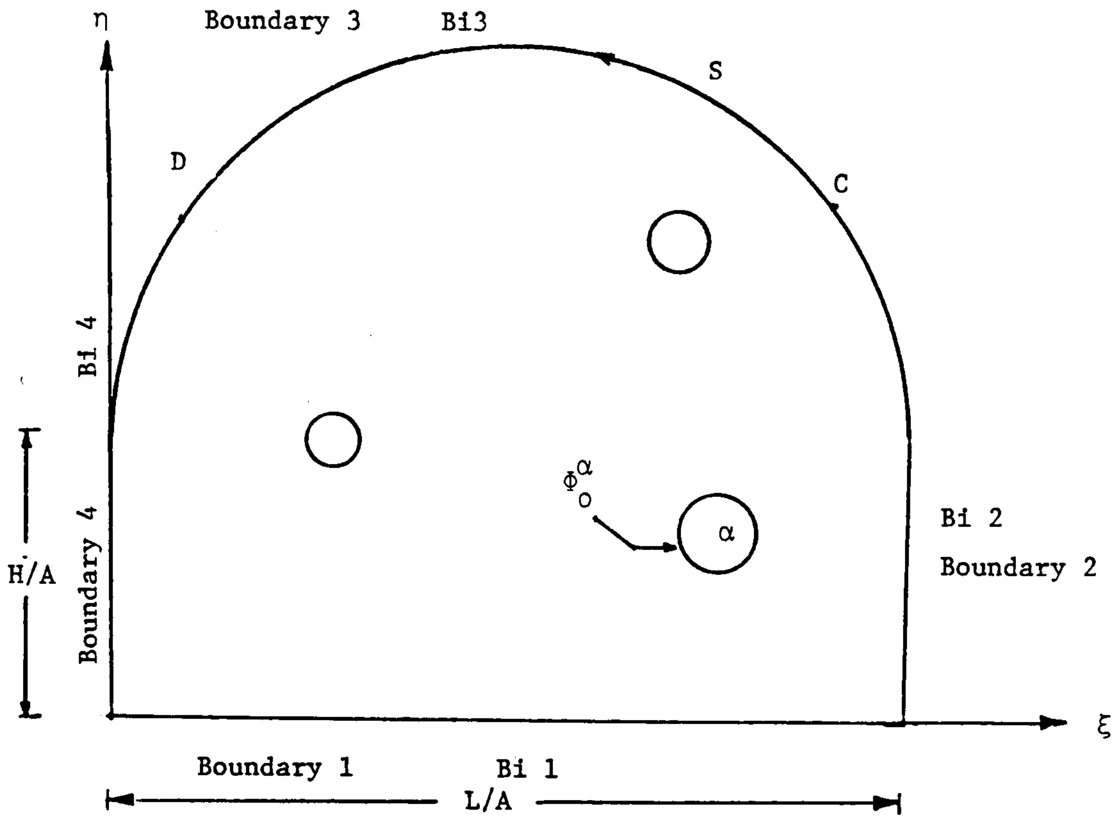


Figure 18. Sketch of the two-dimensional heat conductor with circular holes and convective boundary conditions used in Case II.

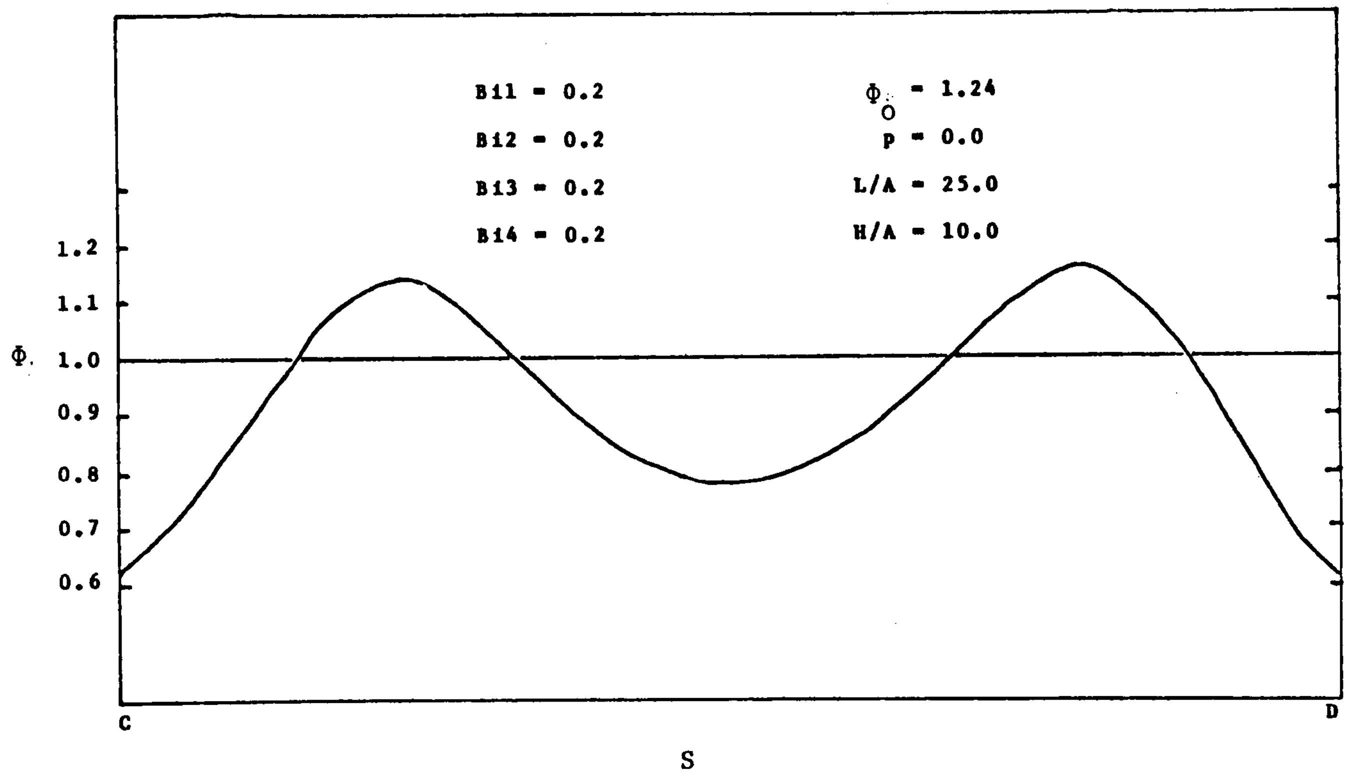


Figure 19. Variation of nondimensional temperature, Φ, along the cavity surface—Case II [optimal hole locations: (8.21, 19.45), (17.62, 18.90)].

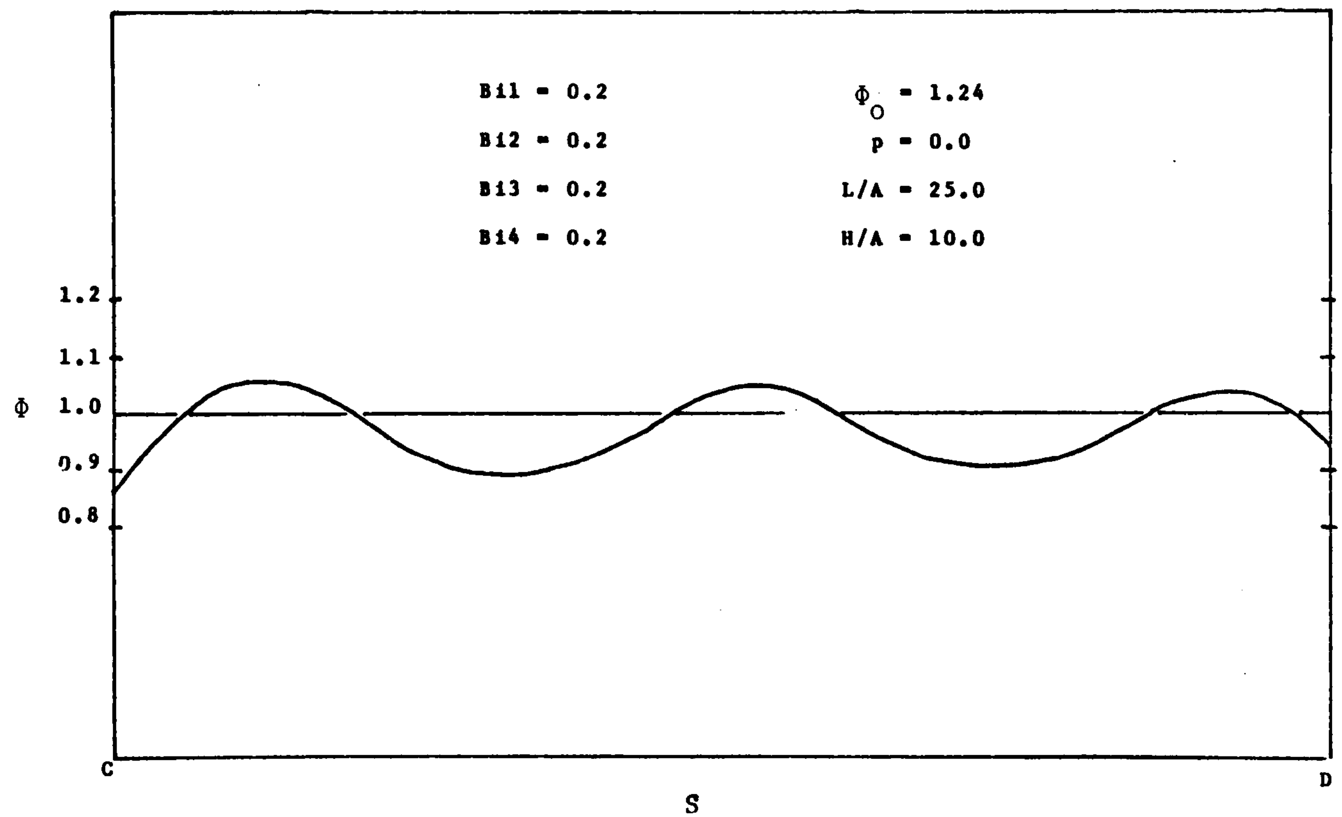


Figure 20. Variation of nondimensional temperature, Φ, along the cavity surface—Case II [optimal hole locations: (6.21, 18.02), (12.84, 20.35), (19.25, 17.65)].

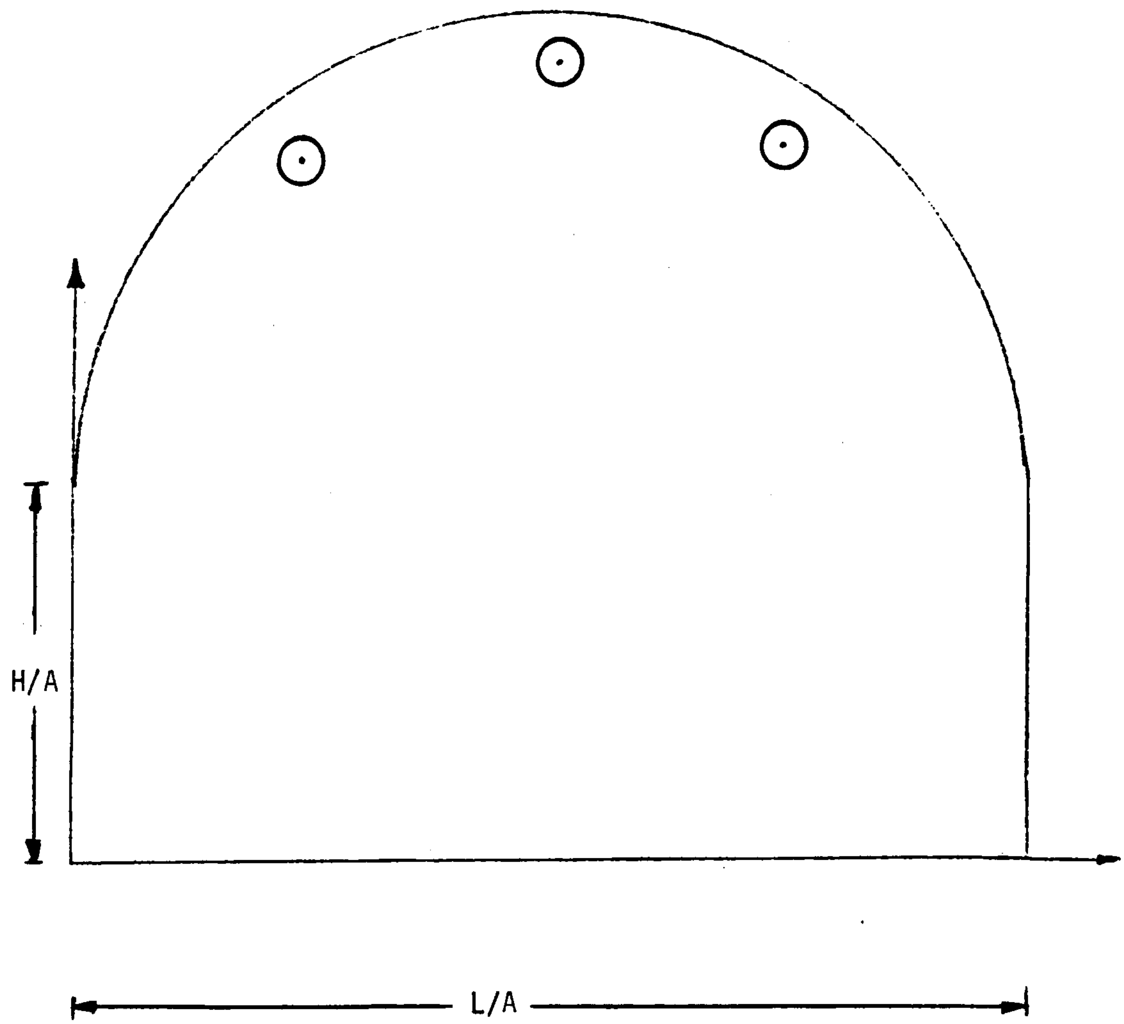


Figure 21. Optimal tube locations for Case II (all conditions defined in Figure 20).

conductor, p, is taken to be 0.3 and 0.5, respectively. These results are presented in Figures 22 and 23, respectively.

Case III corresponds to the configuration as shown in Figure Figure 24 where the semicircular cavity surface of the previous problem is replaced with an S-shaped contour. This region was optimized for the minimum variation of temperature on the cavity surface CD with three heating lines. The cavity temperature profile and the optimal line locations are shown in Figures 25 and 26, respectively.

The last geometrical configuration chosen for optimization (Case IV) is shown in Figure 27. This region was considered with three heating lines. The results for temperature distribution on the cavity surface and optimal hole locations are shown in Figure 28.

The examples discussed above are illustrations of application of general program developed for this study. In this program the hole size, convective boundary conditions, boundary of the region of interest, temperature imposed on circular holes and the number of holes can all be easily varied for other two-dimensional applications.

The CPU time used in the above numerical examples varies depending on the number of tubes and initial tube locations used in the optimization scheme. The typical range of the computation time in the above illustrations is in the range of to 1 to 4 minutes on a DECsystem-10 model 1080. For example, the optimization problem for a rectangular region with two heating lines presented in Figure 12 (p. 39) required a CPU time of 1 minute and 9 seconds. The CPU

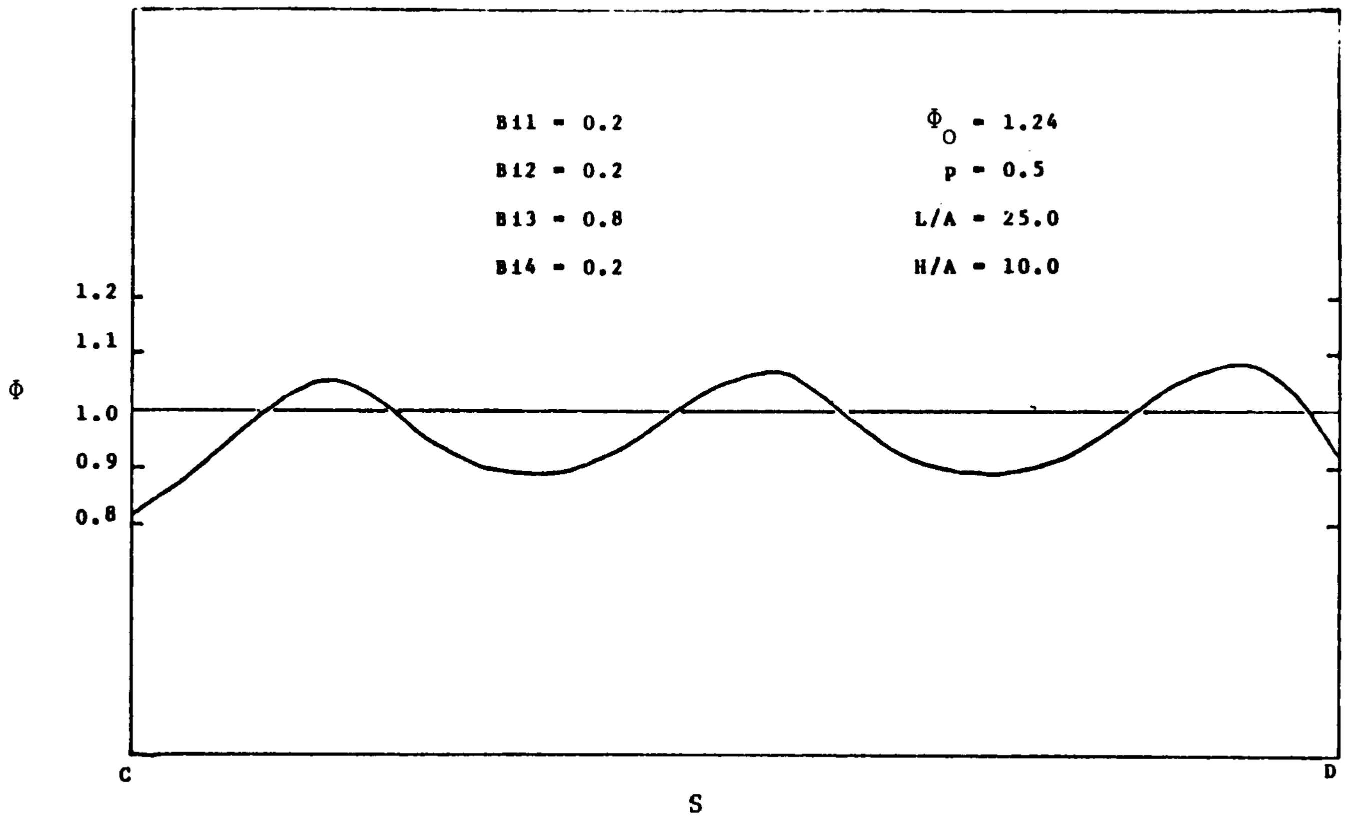


Figure 22. Variation of nondimensional temperature, Φ, along the cavity surface—Case II [optimal hole locations: (6.06, 18.60), (12.58, 20.91), (18.63, 18.72)].

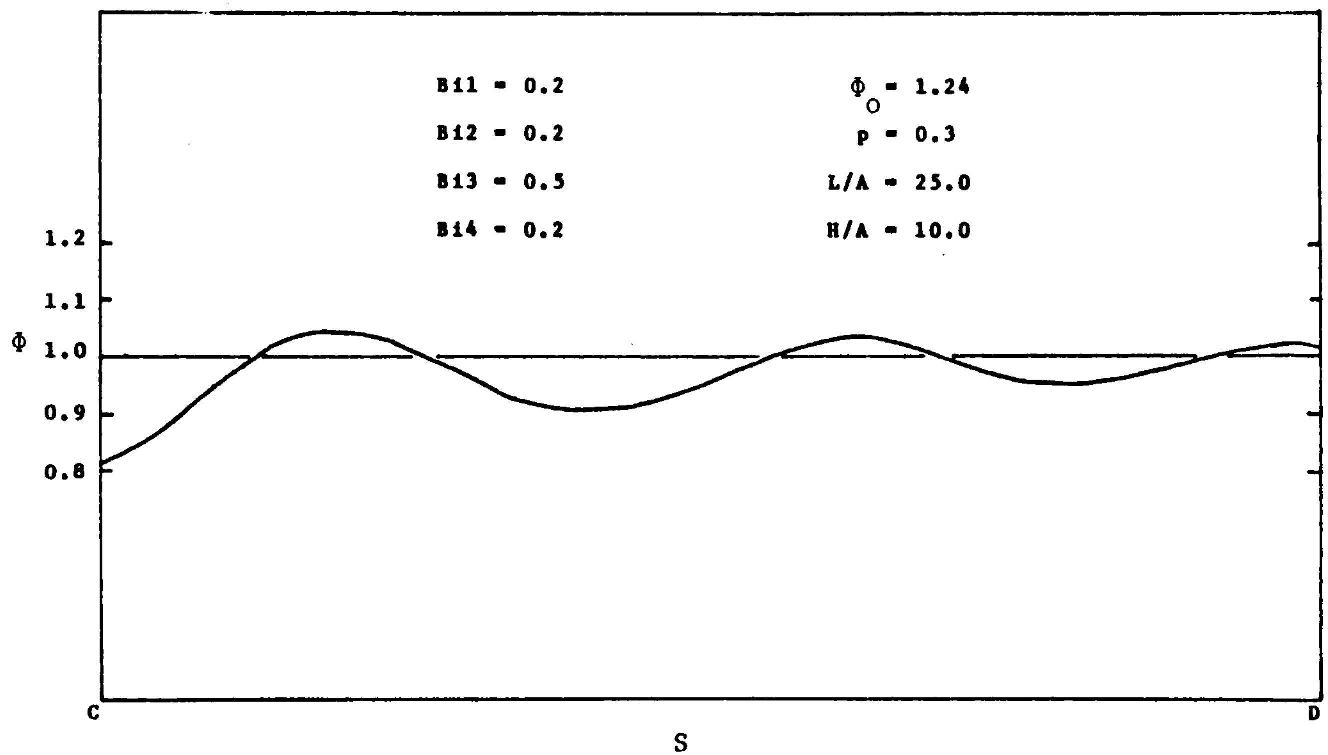


Figure 23. Variation of nondimensional temperature, Φ, along the cavity surface—Case II [optimal hole locations: (5.58, 17.19), (11.12, 20.07), (17.99, 18.55)].

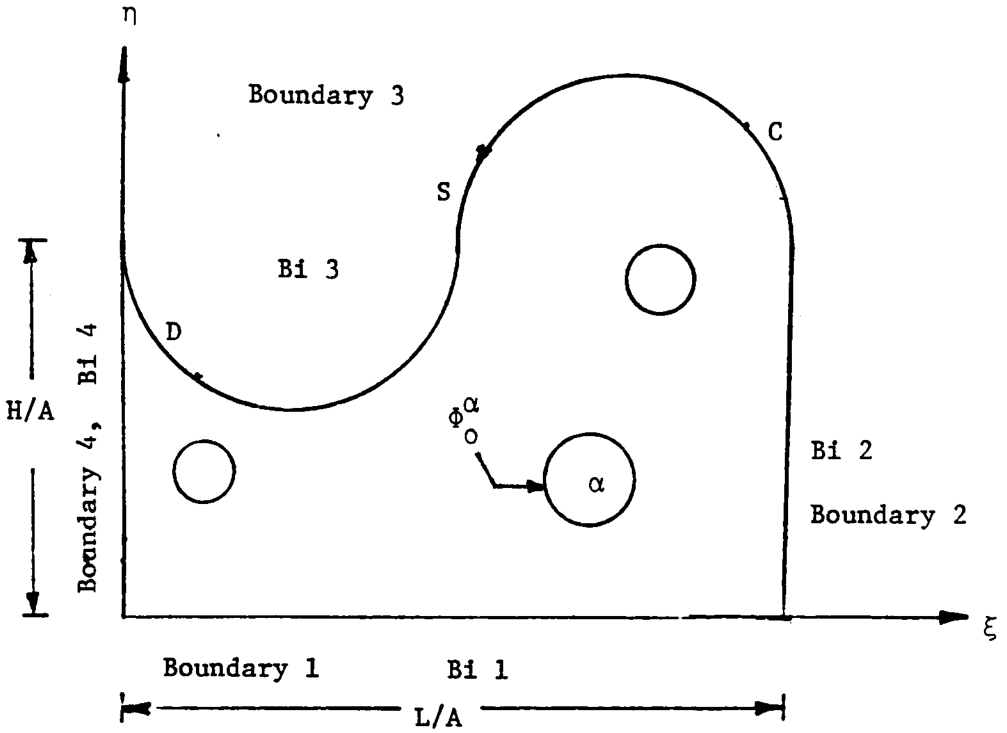


Figure 24. Sketch of the two-dimensional heat conductor with circular holes and convective boundary conditions used in Case III.

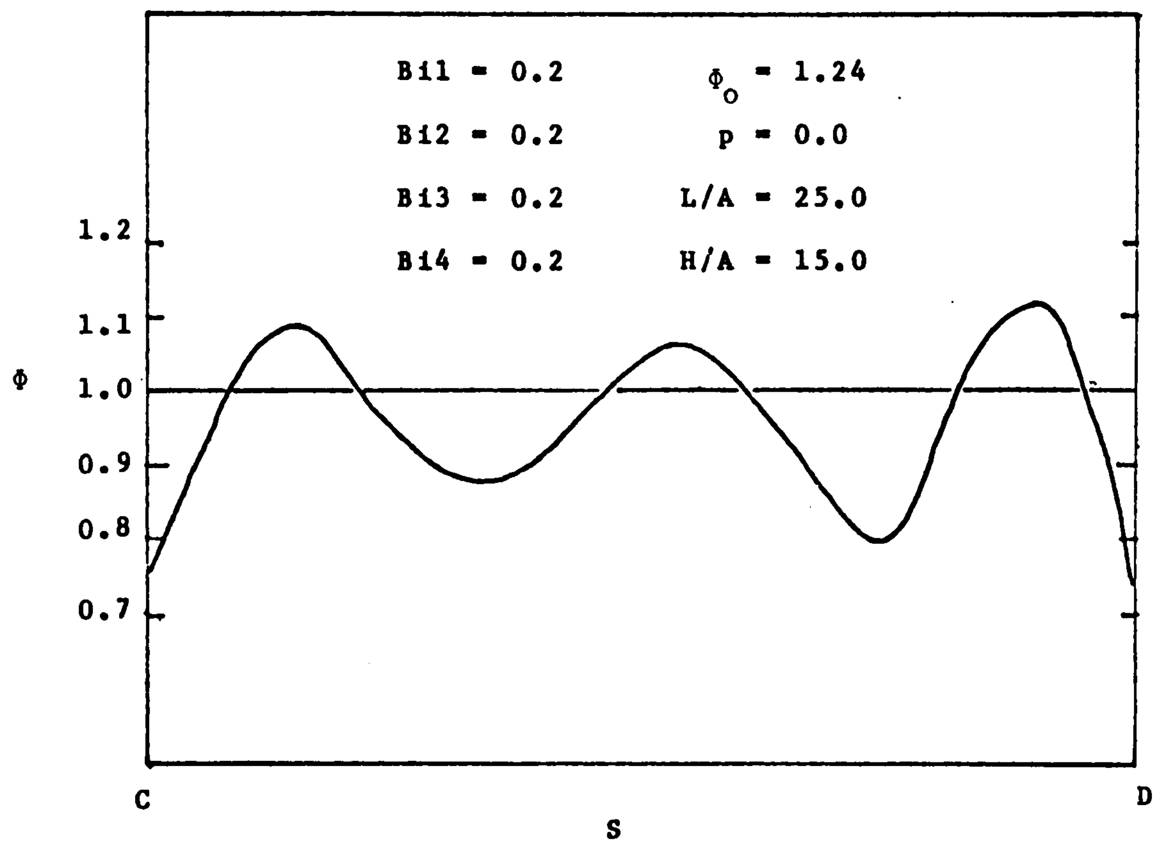


Figure 25. Variation of nondimensional temperature, Φ, along the cavity surface—Case III [optimal hole locations: (6.30, 5.10), (15.31, 14.89), (18.99, 18.96)].

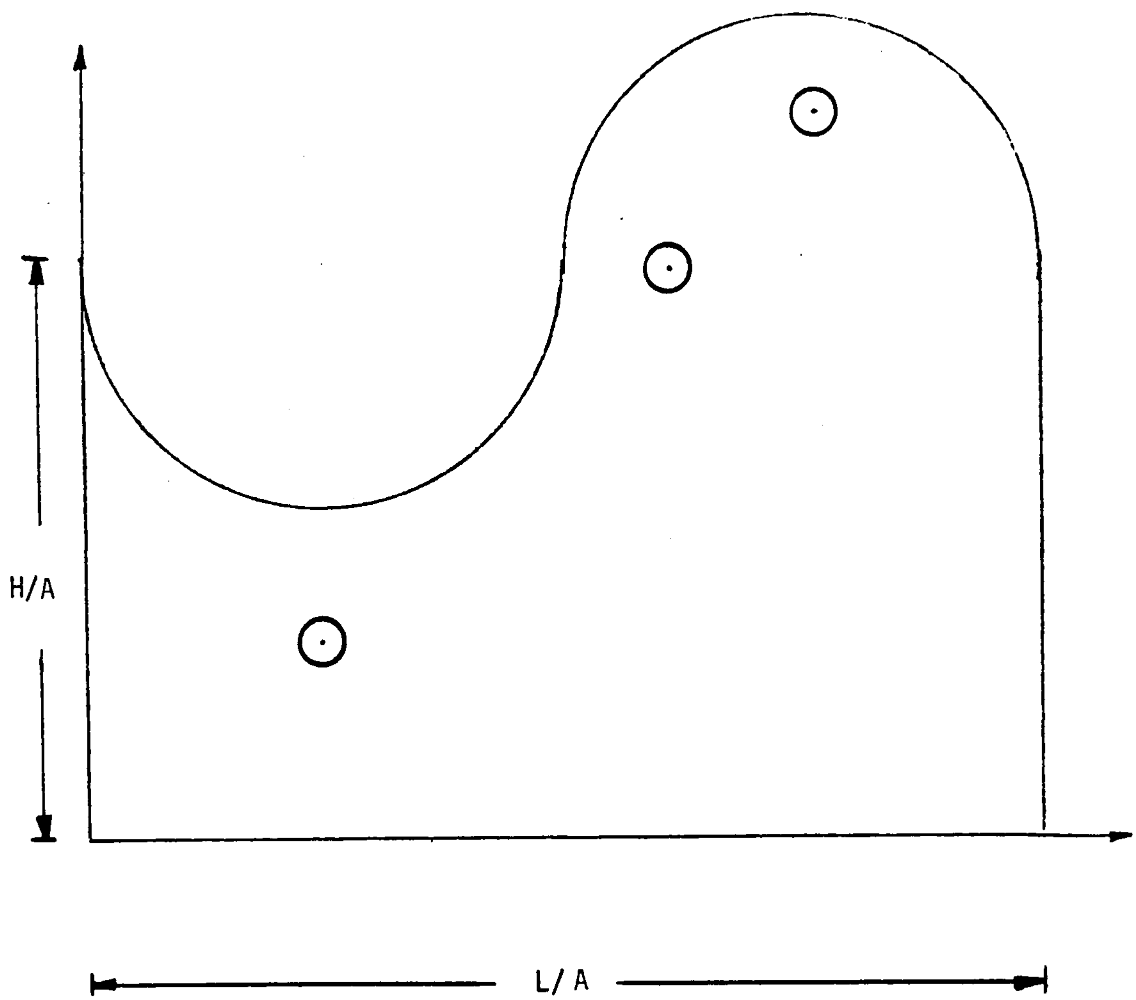


Figure 26. Optimal tube locations for Case III (all locations defined in Figure 25).

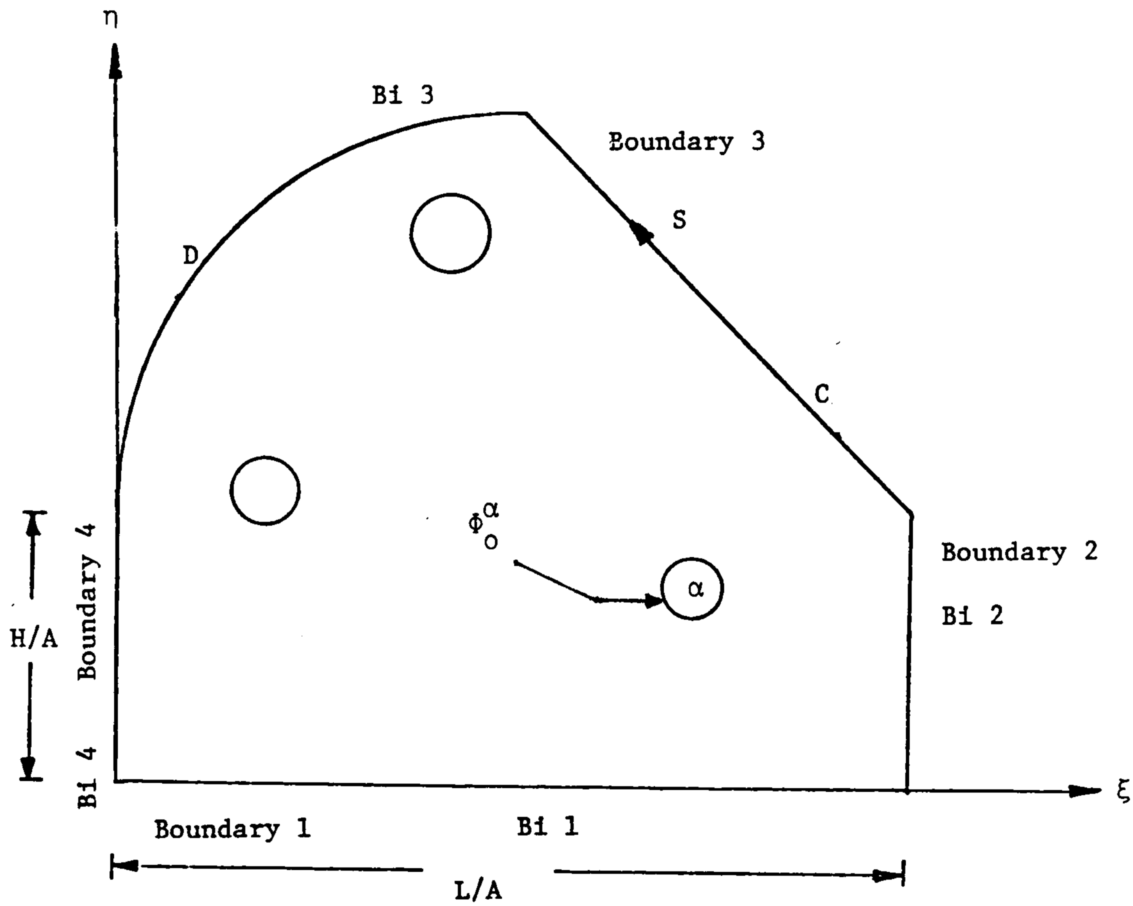


Figure 27. Sketch of the two-dimensional region of a heat conductor with circular holes and convective boundary conditions used in Case IV.

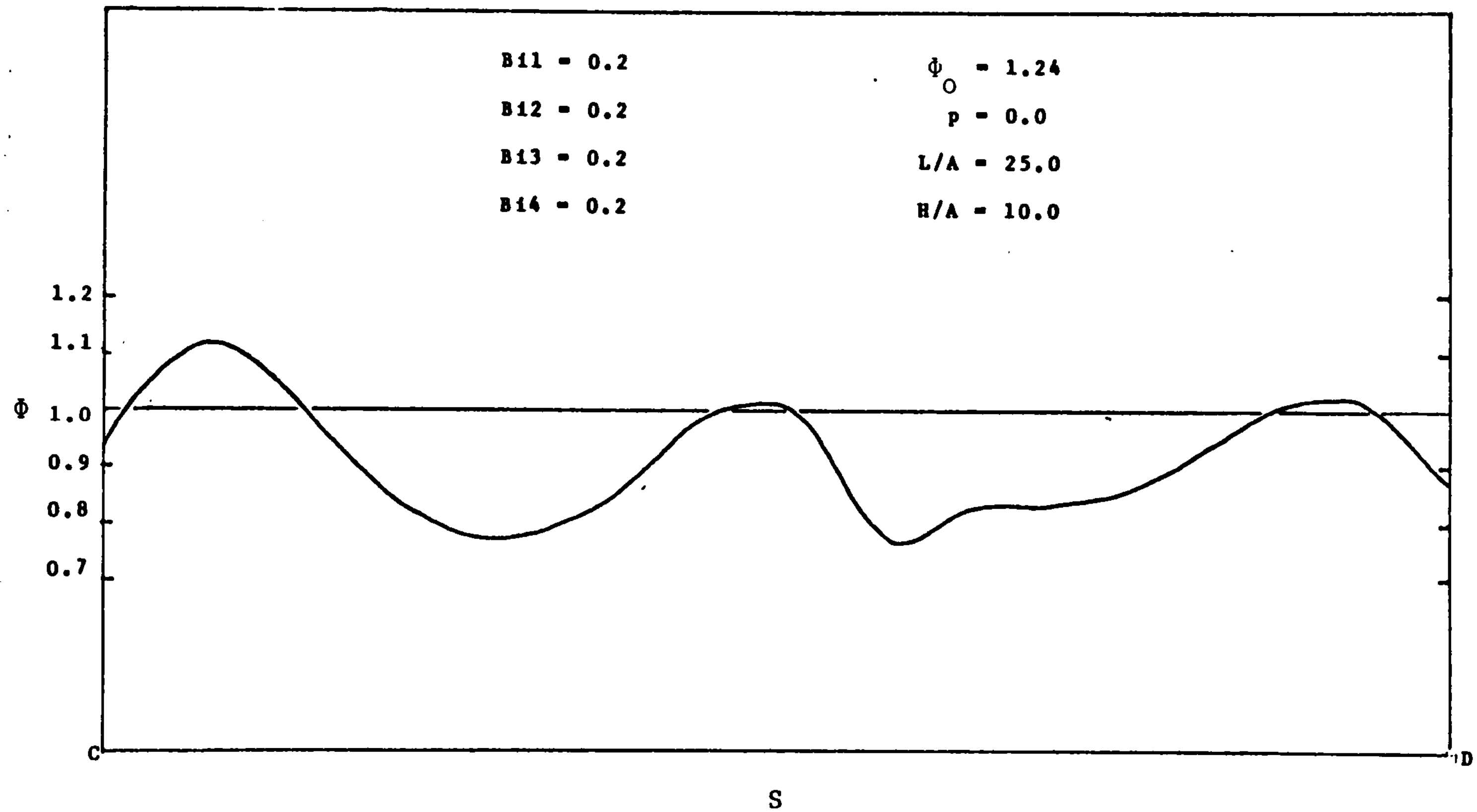


Figure 28. Variation of nondimensional temperature, Φ, along the cavity surface—Case IV [optimal hole locations: (6.51, 18.12), (12.49, 19.38), (19.24, 12.02)].

time increases as the number of tubes is increased. This is because the decision variables are increased in the optimization problem. However, computation time for any of the optimization cases presented never exceeded 4 minutes.

CHAPTER VII

CONCLUSIONS

A special boundary integral method for two-dimensional regions containing circular holes was applied to regions with arbitrary boundaries as cavity surfaces and arbitrary convective boundary conditions. This method was incorporated into a computer program BISOL. Several illustrative examples were solved and presented. The advantage of boundary integral method in conjunction with analytical representation of temperature or its normal derivative on the circular hole boundaries make the method especially powerful in solving design and optimization problems. The CPU time required for the temperature solution on the outer region and the solution of zeroth- and first-order normal derivative of temperature was less than 1 second.

The computer program BISOL developed based on the special boundary integral method is used in conjunction with an optimization subroutine CONMIN in an optimization scheme to determine the optimal locations of cooling or heating lines in a two-dimensional mold. The method used and the optimization program OPTLOC developed are very general, and can be used for any arbitrary cavity surface. Also on this program the hole size, convective boundary conditions, temperature imposed on the circular holes, and the number of holes can all be easily varied. Results for four sample geometries were presented and discussed. The computational time required for the examples discussed are found to be less than two minutes of CPU time.

REFERENCES

REFERENCES

1. M. A. Jaswon and G. T. Symm, Integral Equation Methods in Potential Theory and Elastostatics, Academic Press, New York 1977.

2. C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel, Boundary Element Techniques—Theory and Applications to Engineering, Springer-Verlag, New York (1984).

3. J. C. Lachat and J. O. Watson, "Effective Numerical Treatment of Boundary Integral Equations: A Formulation for Three-Dimensional Elastostatics," International Journal of Numerical Methods in Engineering, Vol. 10, 991-1005 (1976).

4. F. J. Rizzo and D. J. Shippy, "An Advanced Boundary Integral Equation Method for Three-Dimensional Thermoelasticity," International Journal of Numerical Methods in Engineering, Vol. 11, 1753-1768 (1977).

5. G. Fairweather, F. J. Rizzo, D. J. Shippy, and Y. S. Wu, "On the Numerical Solution of Two-Dimensional Potential Problems by an Improved Boundary Integral Equation Method," Journal of Computational Physics, Vol. 31, 96-112 (1979).

6. M. S. Khader and M. C. Hanna, "An Iterative Numerical Solution for General Steady Heat Conduction Problems," International Journal of Heat and Mass Transfer, Vol. 103, 26-31 (1981).

7. R. A. Altenkirch, M. Rezayat, R. Eichhorn, and F. J. Rizzo, "Boundary Integral Equation Method Calculations of Surface Regression Effects in Flame Spreading," Journal of Heat Transfer, Vol. 104, 734-740 (1982).

8. M. R. Barone and D. A. Caulk, "Special Boundary Integral Equations for Approximate Solution of Laplace's Equation in Two-Dimensional Region with Circular Holes," The Quarterly Journal of Mechanics and Applied Mathematics, Vol. XXXIV, 265-286 (1981).

9. M. R. Barone and D. A. Caulk, "Optimal Arrangement of Holes in a Two-Dimensional Heat Conductor by a Special Boundary Integral Method," International Journal of Numerical Methods in Engineering, Vol. 18, 675-685 (1982).

10. R. V. Armilli and M. Parang, "Numerical Analysis of Heat Transfer in Buried Horizontal Heat Exchanger Tubes," 21st AICHE/ASME National Heat Transfer Conference, July 24-27, 1983, Seattle, Washington.

11. D. A. Caulk, "Steady Heat Conduction from an Infinite Number of Holes in a Half Space or a Uniform Slab," International Journal of Heat and Mass Transfer, Vol. 26, No. 10, 1509-1513 (1983).

12. D. A. Caulk, "Analysis of Elastic Torsion in a Bar with Circular Holes by a Special Boundary Integral Method," Journal of Applied Mechanics, Vol. 50, 101-108 (1983).

13. W. D. McMillan, The Theory of Potential, McGraw-Hill, New York (1930).

14. CONMIN, "A FORTRAN Program for Constrained Function Minimization," NASA TM-X 62.282 (1972).

15. G. Zoutendijk, Methods of Feasible Directions, Elsevier Publishing Co., Amsterdam (1960).

APPENDICES

APPENDIX A

EVALUATION OF INTEGRALS

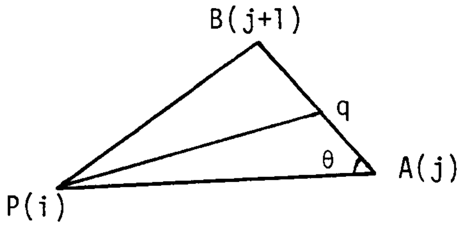
In this appendix several types of integrals that are encountered in Eqs. 23-26 are evaluated. These integrals can often be determined analytically on straight boundaries. In these cases the results are presented without showing all the algebraic details in their derivation. If the analytic evaluation of the integrals is not possible, the formulations shown here in detail are used in the computer program (Appendix C).

Integral of $\int_{c_j} g ds = \int_p^q \log|P-q| dq$

Consider the line integration of the function g over a straight line AB as shown below. The line AB subtends an angle ψ at point P. Let the length of this line be h, and let a and b be the distances of P from A and B, respectively. Let θ denote the angle PAB. If a, b, h are all nonzero, the angles ψ and θ are given by cosine rule as

$$\psi = \text{arc cos}\left(\frac{a^2 + b^2 - h^2}{2ab}\right) \qquad 0 \leq \psi \leq \pi \qquad \text{(A-1)}$$

$$= \text{arc cos}\left(\frac{a^2 + h^2 - b^2}{2ah}\right) \qquad 0 \leq \theta \leq \pi \qquad \text{(A-2)}$$



For a general point q on AB, let the distance from P, i.e., $|P-q|$, be denoted by r. The integral of kernel g over AB is

$$\int_A^B g\,dq = \int_0^h \log r\,ds = a\cos\theta(\log a - \log b) + h(\log b-1) + a\,\psi\sin\theta \qquad \text{(A-3)}$$

when the points P, A, B are collinear, i.e., $\theta = 0$ or π. Therefore, when $\theta = 0$

$$\int_A^B g\,dq = a(\log a-1) - b(\log b-1) \qquad \text{(A-4)}$$

and when $\theta = \pi$,

$$\int_A^b g\,dq = b(\log b-1) - a(\log a-1) \qquad \text{(A-5)}$$

Also when $a = 0$ or $b = 0$, $\int_A^B g\,dq = h(\log(h)-1)$.

<u>Integral of</u> $\int \frac{\partial g}{\partial n}\,ds = \int \underline{\log|P-q|'dq}$ (A-6)

This is the integral of kernel function differentiated along the normal to the point $q \in R$ (see illustration above).

$$\int_A^B \frac{\partial}{\partial n}\log|P-q|\,dq = [\theta]_A^B \qquad \text{(A-7)}$$

where $[\theta]_A^B$ is the change in angle θ which the vector $(\overline{q}-\overline{p})$ makes with some fixed direction, as q moves from A to B. It is assumed that the arc AB is described so as to keep the region on the left and the normal is directed outwards.

In a more convenient notation result (A-7) can be written as

$$\int_A^B \log|P-q|'dq = \theta_{AB,P} \tag{A-8}$$

where $\theta_{AB,P}$ is the angle subtended at point P by arc AB with appropriate sign attached. For example $\theta_{AB,P} = \psi$ for the arrangement shown in the illustration on p. 60.

If point P lies on the arc AB the integral becomes

$$\int_A^B \frac{\partial}{\partial n} \log|P-q|dq = \theta_{AP,P} + \theta_{PB,P} \tag{A-9}$$

where $\theta_{AP,P}$ is the signed angle between the straight line joining A and P and the tangent to the arc AP at P. Similarly, $\theta_{PB,P}$ is the signed angle between the tangent to the arc PB at P and the chord PB.

A special case occurs when p AB and AB is a circular arc of radius a_r and length h. In this case

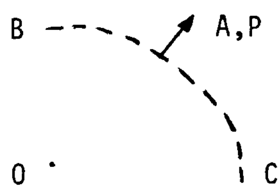
$$\int_A^B \frac{\partial}{\partial n} \log|P-q| = h/2\, a_r\ . \tag{A-10}$$

If the normal is directed inwards from the circle instead of outwards then

$$\int_A^B \frac{\partial}{\partial n} \log|P-q|dq = -h/2\ a_r\ . \tag{A-11}$$

The integral of normal derivative along a portion of boundary without any curvature is zero.

If AB is not a circular arc and P coincides with either A or B, the evaluation of the integral of $\frac{\partial}{\partial n} \log|P-q|$ needs to be performed numerically. This situation is presented in the illustration below. To evaluate the integral, a circle is passed through three points P, A, and C as shown below. The outward normal at point P may be either directed towards or away from the center O of the circle.



In the former case Eq. A-11 is used whereas in the latter case Eq. A-10 is utilized to evaluate this integral. The general computer program developed and presented in Appendix C computes this integral internally using the scheme described whenever point P (or i in Eqs. 23-26) coincides with A or B (j or J=1, respectively, in Eqs. 23-26).

<u>Integrals of g_1^α and g_2^α</u>

The integrals of the kernel functions g_1^α and g_2^α along a segment of outer boundary can be evaluated analytically only when the segment is parallel to either of the axes. For example, the integral of g_1^α along a line parallel to the ξ axis is $[-\theta]_j^{j+1}$ and along a line parallel to η axis is $[\log r_\alpha]_j^{j+1}$. Also the integral g_2^α along a line parallel to the ξ axis is $[\log q_\alpha]_j^{j+1}$ and along a line parallel to the η axis is $[\theta]_j^{j+1}$. In all other cases the integral is evaluated using a trapezoidal rule. The trapezoidal rule for the integration of a function f along a segment can be written as

$$\int_j^{j+1} f\, ds \approx \frac{1}{2} h_j \, [f_j + f_{j+1}] \qquad \text{(A-12)}$$

where h_j is the segment length from j to j+1 and f_j and f_{j+1} are the values of function f at points j and j+1, respectively.

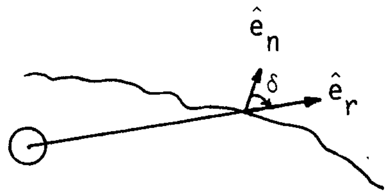
<u>Integrals of $\int \frac{\partial g_1^\alpha}{\partial n}$ ds, and $\int \frac{\partial g_2^\alpha}{\partial n}$ ds</u>

The integral of $\frac{\partial g_1^\alpha}{\partial n}$ and $\frac{\partial g_2^\alpha}{\partial n}$, where n is the outward normal along a portion of the boundary without any curvature is zero. In other cases the integral is evaluated as follows:

$$\int_j^{j+1} \frac{\partial g_{1,2}}{n} ds = \int_j^{j+1} \frac{\partial g_{1,2}^\alpha}{\partial n} \hat{e}_n \cdot \hat{e}_r + \frac{1}{} \frac{\partial g_{1,2}^\alpha}{\partial \theta} \hat{e}_n \cdot \hat{e} \,)ds \qquad \text{(A-13)}$$

where $\hat{e}_n \cdot \hat{e}_r = \cos\delta$ and $\hat{e}_n \cdot \hat{e} = \sin\delta$. δ is an angle between $\hat{e}_n$ and $\hat{e}_r$ as shown in the illustration below. The vector e_n is in the direction of the normal to the region at point P and vector e_r

is in the direction parallel to the direction of the line joining points P and the center of the hole pointing outwards from the region.



The integrals on the right hand side of Eq. A-13 are evaluated using a trapezoidal rule. Thus, for example,

$$\int_j^{j+1} \frac{\partial g_{1,2}^{\alpha}}{\partial n} \hat{e}_n \cdot \hat{e}_r \, ds = \frac{1}{2} h_j \left[\left(\frac{\partial g_{1,2}^{\alpha}}{\partial n} \cos \delta \right)_j + \left(\frac{\partial g_{1,2}^{\alpha}}{\partial n} \cos \delta \right)_{j+1} \right] \quad \text{(A-14)}$$

and similarly for the other integral in Eq. A-13.

As an example, the evaluation of these integrals and their substitution in Eqs. 23-26 for the simple case of rectangular region (Figure 11, p. 38) can be shown to yield the following algebraic equations:

$$\sum_{j=1}^{N} \left[-\frac{Bi}{2\pi} (a \cos \theta(\log a - \log b) + h(\log b - 1) + a\psi \sin \theta + h \log A) + \delta_{ij} \cdot \frac{1}{2} \phi_j \right] + \sum_{\alpha=1}^{M} a_{\alpha} \left[(\log r_{\alpha i} + \log A) q_0^{\alpha} - \frac{1}{2} b_{\alpha i} \sin \psi^{\alpha i} q_1^{\alpha} - \frac{1}{2} b_{\alpha i} \cos \psi^{\alpha i} q_2^{\alpha} \right] = 0 \quad \text{(A-15)}$$

$$\sum_{j=1}^{N} \left(- \frac{Bi}{2\pi} [a \cos \theta(\log a - \log b) + h(\log b - 1) + a\psi \sin\theta + h \log A]\right)$$

$$+ q_0^\beta \, \alpha\beta(\log a_\beta + \log A) + \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} a_\alpha \left((\log \gamma_{\alpha\beta} + \log A)q_0^\alpha\right.$$

$$\left. - \frac{1}{2} b_{\alpha\beta} \sin \psi^{\alpha\beta} q_1^\alpha - \frac{1}{2} b_{\alpha\beta} \cos \psi^{\alpha\beta} q_2^\alpha\right) = - \phi_0^\beta \qquad \text{(A-16)}$$

$$\sum_{j=1}^{N} \frac{Bi}{A} (u_{j+1} - u_j) - \frac{\pi}{A} \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} [2b_{\alpha\beta} \sin \psi^{\alpha\beta} q_0^\alpha$$

$$+ b_{\alpha\beta}^2 \sin 2\psi^{\alpha\beta} q_2^\alpha - b_{\alpha\beta}^2 \cos 2\psi^{\alpha\beta} q_1^\alpha] = 0 \qquad \text{(A-17)}$$

$$\sum_{j=1}^{N} \frac{Bi}{A} v_{j+1} - v_j) - \frac{\pi}{A} q_2^\beta + \frac{\pi}{A} \sum_{\substack{\alpha=1 \\ \alpha\neq\beta}}^{M} [2b_{\alpha\beta} \cos \psi^{\alpha\beta} q_0^\alpha$$

$$+ b_{\alpha\beta}^2 \sin 2\psi^{\alpha\beta} q_1^\alpha + b_{\alpha\beta}^2 \cos 2\psi^{\alpha\beta} q_2^\alpha] = 0 \qquad \text{(A-18)}$$

where $\delta_{ij} = 0$ and $i = j$ and $\delta_{ij} = 1$ when $i = j$

$b_\alpha = a_\alpha/\gamma_\alpha, \quad \gamma_\alpha = |y - \xi^\alpha|, \quad \psi^\alpha = \phi^\alpha(y)$

$b_{\alpha\beta} = a_\alpha/\gamma_{\alpha\beta}, \quad \gamma_{\alpha\beta} = |\xi^\alpha - \xi^\beta|, \quad \psi^{\alpha\beta} = \theta^\alpha(\xi^\beta)$

and

$u = \int g_1 \, ds$

$v = \int g_2 \, ds.$

where $u = -\theta^{\alpha}$ when j is on boundary 1 in Figure 11 (p. 38) and it is equal to $\log r_{\alpha}$ on boundary 2. The value of u on boundaries 3 and 4 is negative of its value on boundaries 1 and 2, respectively. The value of v when j is on boundary 1 is $\log r_{\alpha}$ and it is equal to θ^{α} on boundary 2. On boundaries 3 and 4 the value of v is negative of its value on boundaries 1 and 2, respectively.

APPENDIX B

CONMIN: DEFINITIONS OF PARAMETERS

In many mathematical problems, it is necessary to obtain the minimum or the maximum of a function of several variables subject to various linear and nonlinear inequality constraints. It is seldom possible to solve these problems analytically, and iterative methods are used to obtain the numerical solution.

CONMIN is a FORTRAN program, in subroutine form, for the minimization of a multivariable function subject to a set of inequality constraints. The general minimization problem can be described as follows:

Minimize OBJ

Subject to:

G(J).LE.0 J=1,NCON

where OBJ is a general function of the decision variables, X(I). OBJ need not be a simple analytic function, and may be any function which can be numerically evaluated. G(J) is the value of the Jth inequality constraint. NCON is the number of constraints [14].

The minimization algorithm is based on Zoutendijk's method of feasible directions [15]. Though the program is intended primarily for the efficient solution of constrainted functions, unconstrained functions may also be minimized, and the conjugate direction method of Fletcher and Reeves is used. In the following the definition of several basic parameters used in CONMIN are given and their selected values for the computer program used in this thesis are briefly presented.

DEFINITION OF PARAMETERS

PARAMETER	DEFINITION
NDV	Number of decision variables, X(I)
ITMAX	Maximum number of iterations (typically chosen to be 40)
NCON	Number of constraint functions, G(J)
Nside	Side constraint parameter (zero in this thesis)
NSCAL	Scaling control parameter (no scaling is used here)
NFDG	Gradient calculation control parameter (set to zero here)
DELFUN	Minimum relative change in OBJ function (set equal to 0.001)
DABFUN	Minimum absolute change in objective (set equal to 0.001)
LINOBJ	Linear objective function identifier
ITRM	Number of consecutive iterations to indicate convergence (set to 3 here)
X(N)	Vector of decision variables (coordinates of the hole positions)
ISC(N)	Linear constraint identification vector

For the Case I considered for optimization with two holes, the number of decision variables is four (the coordinates at the center of each hole). Therefore, NDV is taken to be four. The constraints are that the hole should not cross the boundary of the region and should not overlap each other. Therefore, the number of constraints becomes five, i.e., NCON = 5. Since the gradients are computed within the optimization code CONMIN, NFDG = 0. The values of DELFUN and DABFUN are taken to be 0.001. As the objective function is non-linear, LINOBJ = 0.

The reader should refer to Reference 14 for more detailed information about the control parameters used in CONMIN.

APPENDIX C

BISOL: FORTRAN PROGRAM FOR HEAT TRANSFER SOLUTION

```
C#############################################################################
C#                      APPENDIX C                                           #
C#                     PROGRAM BISOL                                         #
C#############################################################################
C#       THIS PROGRAM CALCULATES THE ZEROTH AND FIRST ORDER                  #
C#       NORMAL DERIVATIVE OF TEMPERATURE ON THE TUBE SURFACE                #
C#       USING BOUNDARY INTEGRAL ANALYSIS ...............                    #
C#############################################################################
C
C                    INPUTS USED ARE
C#############################################################################
C        PHIO(K)= THE NON-DIMENSIONAL TEMPERATURE SPECIFIED
C                 ON THE K'TH TUBE
C        BI1= BIOT NUMBER ON BOUNDARY ONE
C        BI2=BIOT NUMBER ON BOUNDARY TWO
C        BI3(J)=VARIABLE BIOT NUMBER ON BOUNDARY THREE
C        BI4=BIOT NUMBER ON BOUNDARY FOUR
C        XT(K)= XI CO-ORDINATE OF THE K'TH TUBE
C        YT(K)= ETA CO-ORDINATE OF THE K'TH TUBE
C        W1=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY ONE
C        W2=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY TWO
C        W3=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY THREE
C        W4=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY FOUR
C        HEAT=HEAT FLUX INTO THE HEAT CONDUCTOR
C        WW=WIDTH OF THE HEAT CONDUCTOR
C        LE=LENGTH OF THE HEAT CONDUCTOR
C        AA(K)=RADIUS OF THE K'TH TUBE
C-----------------------------------------------------------------------------
         DIMENSION D(100,100),XI(100),ETA(100),R1(100),R2(100)
         DIMENSION F(100),WKAREA(250),TEP(100),XIP(100),ETAP(100)
         DIMENSION XT(30),YT(30),PHIO(10),GGG1(10,10),GGG2(10,10)
         DIMENSION XP(60),YP(60),BETA(60) ,AA(10),RAL(60,60)
         DIMENSION FF(100),FFF(100),ZZ1(100),ZZ2(100),ZZZ1(100),ZZZ2(100)
         DIMENSION YTP(10),XTP(10),BI3(100)
         REAL LLL,LE
         INTEGER IA,IDGT
         DATA W1,W2,W3,W4/10,4,20,4/
         DATA WW,LE/10.,25./
         DATA AA(1),AA(2),AA(3),AA(4)/1.,1.,1.,1./
         DATA XT(1),XT(2),YT(1),YT(2)/5.,20.,5.,5./
         DATA XT(3),YT(3),XT(4),YT(4) /15.,5.,10.,5./
         DATA PHIO(1),PHIO(2)/150.,150./
         DATA PHIO(3),PHIO(4)/150.,150./
         DATA PHIC,PHIA/150.,25./
         DATA M/2/
         DATA HEAT/0.0/
         DATA BI1,BI2,BI4/.2,.2,.2/
         AD=0.0
         DO 11 K=1,M
         IF(AA(K).GT.AD) AD=AA(K)
```

```
 11     CONTINUE
C       NON-DIMENSIONALISE ALL PARAMETERS
        DO 13 K=1,M
        XT(K)=XT(K)/AD
        YT(K)=YT(K)/AD
        AA(K)=AA(K)/AD
        PHIO(K)=(PHIO(K)-PHIA)/(PHIC-PHIA)
 13     CONTINUE
        PI=2.*ASIN(1.)
        ALPHA1=PI/2.
        ALPHA2=PI/2.
        ALPHA3=PI/2.
        ALPHA4=PI/2.
        W=WW/AD
        LLL=LE/AD
        N=W1
        N1=W1+W2
        N2=W1+W2+W3
        N3=W1+W2+W3+W4
        NN=N+1
        NN1=N1+1
        NN2=N2+1
        NNN2=W1+W2+(W3/2)
        NNNN2=NNN2+1
        DO 12  J=NN1,N2
        BI3(J)=.2
 12     CONTINUE
        DX1=LLL/W1
        DY2=W/W2
        DX3=LLL/W3
        DY4=W/W4
C---------------------------------------------------------------------
C       GENERATE ALL CO-ORDINATES OF THE PROBLEM
C---------------------------------------------------------------------
        XI(1)=0.0
        DO 10 I=1,N
        ETA(I)=0.0
        XI(I+1)=XI(I)+DX1
  10    CONTINUE
        ETA(NN)=0.0
        DO 20 I=NN,N1
        XI(I)=LLL
        ETA(I+1)=ETA(I)+DY2
  20    CONTINUE
        XI(NN1)=LLL
        DO 30 I=NN1,N2
        XI(I+1)=XI(I)-DX3
  30    CONTINUE
        DO 31 I=NN1,NNN2
  31    ETA(I)=W
```

```
      ETA(NNNN2)=W
      DO 32 I=NNNN2,N2
 32   ETA(I)=W
      ETA(NN2)=W
      DO 40 I=NN2,N3
      XI(I)=0.0
      ETA(I+1)=ETA(I)-DY4
 40   CONTINUE
C----------------------------------------------------------------------
C     EQUATION 26
C----------------------------------------------------------------------
C     I ON BOUNDARY 1
C----------------------------------------------------------------------
      DO 100 I=1,N
      CON=0.0
      DO 110 J=1,N
      A1=ABS(XI(J+1)-XI(I))
      B1=ABS(XI(J)-XI(I))
      H=ABS(XI(J+1)-XI(J))
      IF(A1.EQ.0.0)D(I,J)=BI1*(  B1*(ALOG(B1)-1))+BI1*B1*ALOG(AD)
      IF(B1.EQ.0.0)D(I,J)=BI1*(A1*(ALOG(A1)-1))+BI1*A1*ALOG(AD)
      IF(A1.EQ.0.0.OR.B1.EQ.0.0)GO TO 110
      D(I,J)=BI1*(A1*(ALOG(A1)-1)-B1*(ALOG(B1)-1))+BI1*ALOG(AD)*
     * (A1-B1)
      IF(I.LT.J) GO TO 110
      IF(I.GT.J) D(I,J)=-D(I,J)
110   CONTINUE
      DO 120 J=NN,N1
      C1=LLL-XI(I)
      D1=ETA(J+1)
      E1=ETA(J)
      H=D1-E1
      A=SQRT(E1**2+C1**2)
      B=SQRT(D1**2+C1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      C11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI2*(C11+H*ALOG(AD))
120   CONTINUE
      DO 130 J=NN1,N2
      F1=ABS(XI(J+1)-XI(I))
      F2=ABS(XI(J)-XI(I))
      F3=ABS(ETA(J)-ETA(I))
      F4=ABS(ETA(J+1)-ETA(I))
      F5=ABS(ETA(J+1)-ETA(J))
      F6=ABS(XI(J+1)-XI(J))
      H=SQRT(F5**2+F6**2)
      A=SQRT(F2**2+F3**2)
      B=SQRT(F1**2+F4**2)
```

```
      YY=(A**2+H**2-B**2)/(2.*A*H)
      IF(YY.GT.1.) YY=1.
      IF(YY.LT.-1.) YY=-1.
      THETA1=ACOS(YY)
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      F11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+F11
      GIDS=BI3(J)*F11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 130
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS=PSI
      D(I,J)=(DGDNDS+GIDS)
 130  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      DO 140 J=NN2,N3
      A1=XI(I)
      BB=ETA(J)
      BB1=ETA(J+1)
      IF(A1.EQ.0.0.AND.BB1.EQ.0.0)D(I,J)=BI4*(BB*(ALOG(BB)-1))
     * +BB*ALOG(AD)
      IF(A1.EQ.0.0.AND.BB1.EQ.0.0) GO TO 140
      IF(A1.EQ.0.0)D(I,J)=-BI4*(BB1*(ALOG(BB1)-1)-BB*(ALOG(BB)-1))
     * +BI4*ALOG(AD)*(BB1-BB)
      IF(A1.EQ.0.0) GO TO 140
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(BB**2+A1**2)
      B=SQRT(BB1**2+A1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
```

```
      AA11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI4*(AA11+ALOG(AD))
 140  CONTINUE
 100  CONTINUE
C------------------------------------------------------------------
C     I ON BOUNDARY 2
C------------------------------------------------------------------
      DO 200 I=NN,N1
      CON=0.0
      DO 210 J=1,N
      GG1=LLL-XI(J+1)
      GG2=LLL-XI(J)
      G3=ETA(I)
      IF(GG1.EQ.0.0.AND.G3.EQ.0.0)D(I,J)= BI1*(GG2*ALOG(GG2)-GG2)
     * +BI1*GG2*ALOG(AD)
      IF(GG1.EQ.0.0.AND.G3.EQ.0.0) GO TO 210
      IF(G3.EQ.0.0)D(I,J)=-(BI1)*((GG1*ALOG(GG1)-GG1)-(GG2*
     1 ALOG(GG2)-GG2))+BI1*GG2*ALOG(AD)-BI1*GG1*ALOG(AD)
      IF(G3.EQ.0.0)GO TO 210
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(GG2**2+ETA(I)**2)
      B=SQRT(GG1**2+ETA(I)**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      G11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI1*(G11+H*ALOG(AD))
 210  CONTINUE
      DO 220 J=NN,N1
      H1=ABS(ETA(J+1)-ETA(I))
      H2=ABS(ETA(J)-ETA(I))
      H=ABS(ETA(J+1)-ETA(J))
      IF(H1.EQ.0.0)D(I,J)=BI2*( H2*(ALOG(H2)-1))+BI2*H2*ALOG(AD)
      IF(H2.EQ.0.0)D(I,J)=BI2*(H1*(ALOG(H1)-1))+BI2*H1*ALOG(AD)
      IF(H1.EQ.0.0.OR.H2.EQ.0.0)GO TO 220
      D(I,J)=BI2*(H1*(ALOG(H1)-1)-H2*(ALOG(H2)-1))
     * +BI2*(H1*ALOG(AD)-H2*ALOG(AD))
      IF(I.LT.J)GO TO 220
      IF(I.GT.J) D(I,J)=-D(I,J)
 220  CONTINUE
      DO 230 J=NN1,N2
      O1=ABS(ETA(J)-ETA(I))
      O2=ABS(XI(J)-XI(I))
      O3=ABS(ETA(J+1)-ETA(I))
      O4=ABS(XI(J+1)-XI(I))
      O5=ABS(ETA(J+1)-ETA(J))
      O6=ABS(XI(J+1)-XI(J))
      A=SQRT(O1**2+O2**2)
      B=SQRT(O3**2+O4**2)
```

```
      H=SQRT(O5**2+O6**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*H*A))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      O11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+O11
      GIDS=BI3(J)*O11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 230
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS= PSI
      D(I,J)=(DGDNDS+GIDS)
 230  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      DO 240 J=NN2,N3
      GG1=ABS(ETA(J+1)-ETA(I))
      GG2=ABS(ETA(J)-ETA(I))
      GG3=ETA(I)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(GG2**2+LLL**2)
      B=SQRT(GG1**2+LLL**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      GG11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI4*GG11+BI4*H*ALOG(AD)
 240  CONTINUE
 200  CONTINUE
C-----------------------------------------------------------------
C     I ON BOUNDARY 3
C-----------------------------------------------------------------
```

```
      DO 300 I=NN1,N2
      CON=0.0
      DO 310 J=1,N
      P1=ABS(XI(J+1)-XI(I))
      P2=ABS(XI(J)-XI(I))
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(P2**2+W**2)
      B=SQRT(P1**2+W**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      P11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI1*P11+BI1*H*ALOG(AD)
310   CONTINUE
      DO 320 J=NN,N1
      Q1=W-ETA(J+1)
      Q2=W-ETA(J)
      Q3=LLL-XI(I)
      IF(Q1.EQ.0.0.AND.Q3.EQ.0.0)D(I,J)=BI2*(Q2*ALOG(Q2)-Q2)
     * +BI2*Q2*ALOG(AD)
      IF(Q1.EQ.0.0.AND.Q3.EQ.0.0)GO TO 320
      IF(Q3.EQ.0.0)D(I,J)=-BI2*((Q1*ALOG(Q1)-Q1)-(Q2*ALOG
     1 (Q2)-Q2))+BI2*Q2*ALOG(AD)-BI2*Q1*ALOG(AD)
      IF(Q3.EQ.0.0)GO TO 320
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(Q2**2+Q3**2)
      B=SQRT(Q1**2+Q3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      Q11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI2*Q11+BI2*H*ALOG(AD)
320   CONTINUE
      DO 330 J=NN1,N2
      QQ1=ABS(XI(J+1)-XI(I))
      QQ2=ABS(XI(J)-XI(I))
      QQ3=ABS(ETA(J+1)-ETA(I))
      QQ4=ABS(ETA(J)-ETA(I))
      QQ5=ABS(ETA(J+1)-ETA(J))
      QQ6=ABS(XI(J+1)-XI(J))
      A=SQRT(QQ2**2+QQ4**2)
      B=SQRT(QQ1**2+QQ3**2)
      H=SQRT(QQ5**2+QQ6**2)
      IF(QQ1.EQ.0.0.OR.QQ2.EQ.0.0) GO TO 311
      YY=(A**2+B**2-H**2)/(2.*A*B)
      IF(YY.GT.1.) YY=1.0
      IF(YY.LT.-1.) YY=-1.
      YY1=(A**2+H**2-B**2)/(2.*A*H)
      IF(YY1.GT.1.) YY1=1.
      IF(YY1.LT.-1.) YY1=-1.
```

```
      THETA1=ACOS(YY1)
      PSI=ACOS(YY)
      Q=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      GIDS=BI3(J)*Q
      IF(QQ5.EQ.0.0.OR.QQ6.EQ.0.0)D(I,J)=GIDS
      IF(QQ5.EQ.0.0.OR.QQ6.EQ.0.0) GO TO 322
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS=PSI
      D(I,J)=(DGDNDS+GIDS)
      GO TO 322
 311  GIDS=BI3(J)*(H*(ALOG(H)-1))+H*ALOG(AD)*BI3(J)
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 322
      SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
      SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
      IF(SLOPE1.EQ.SLOPE2) D(I,J)=GIDS
      IF(SLOPE1.EQ.SLOPE2) GO TO 322
C     CALCULATE THE RADIUS OF CURVATURE
C     TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
      XP(I)=XI(I)-XI(I)
      XP(I+1)=XI(I+1)-XI(I)
      XP(I-1)=XI(I-1)-XI(I)
      YP(I)=ETA(I)-ETA(I)
      YP(I+1)=ETA(I+1)-ETA(I)
      YP(I-1)=ETA(I-1)-ETA(I)
      ALPHA=2*(XP(I-1)-XP(I+1))
      BETTA=2*(YP(I-1)-YP(I+1))
      GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
      ZETA=(XP(I+1)**2+YP(I+1)**2)
      MU=XP(I+1)*2
```

```
      DELTA=2*YP(I+1)
      ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
      BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
      AAA=SQRT(ABAR**2+BBAR**2)
C     CHECK FOR CONCAVITY & CONVEXITY
C     CALCULATE BETA'JS
      IF(XP(I+1).EQ.0.0)THEN
      BETA(I)=PI/2.
      ELSE
      BETA(I)=ATAN(YP(I+1)/XP(I+1))
      END IF
      IF(YP(I+1).GT.0.0.AND.XP(I+1).GT.0.0)BETA(I)=BETA(I)
      IF(YP(I+1).GT.0.0.AND.XP(I+1).LT.0.0)BETA(I)=BETA(I)+PI
      IF(YP(I+1).LT.0.0.AND.XP(I+1).LT.0.0)BETA(I)=BETA(I)+PI
      IF(YP(I+1).LT.0.0.AND.XP(I+1).GT.0.0)BETA(I)=BETA(I)+2*PI
C     CALCULATE BETA(I-1)
      IF(XP(I).EQ.0.0)THEN
      BETA(I-1)=PI/2.
      ELSE
      BETA(I-1)=ATAN(YP(I)/XP(I))
      END IF
      IF(YP(I).GT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)
      IF(YP(I).GT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)+2*PI
C     EVALUATE THE INTEGRAL DGDNDS
      IF(BETA(I).GT.BETA(I-1))THEN
      DGDNDS=H/(2*AAA)
      ELSE IF(BETA(I).LT.BETA(I-1))THEN
      DGDNDS=-H/(2*AAA)
      ELSE
      DGDNDS=0.0
      END IF
      D(I,J)=(DGDNDS+GIDS)
 322  CON=CON+GIDS/BI3(J)
 330  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      DO 340 J=NN2,N3
      PP1=XI(I)
      PP2=W-ETA(J)
      PP3=W-ETA(J+1)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(PP2**2+PP1**2)
      B=SQRT(PP3**2+PP1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      PP11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI4*PP11+BI4*H*ALOG(AD)
 340  CONTINUE
```

```
 300    CONTINUE
C-----------------------------------------------------------------------
C       I  ON BOUNDARY FOUR
C-----------------------------------------------------------------------
        DO 400 I=NN2,N3
        CON=0.0
        DO 410 J=1,N
        Z1=XI(J+1)
        Z2=XI(J)
        Z3=ETA(I)
        H=ABS(XI(J+1)-XI(J))
        A=SQRT(Z2**2+Z3**2)
        B=SQRT(Z1**2+Z3**2)
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        R11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)
        D(I,J)=BI1*R11+BI1*H*ALOG(AD)
 410    CONTINUE
        DO 420 J=NN,N1
        RR1=ABS(ETA(J+1)-ETA(I))
        RR2=ABS(ETA(J)-ETA(I))
        H=ABS(ETA(J+1)-ETA(J))
        A=SQRT(RR2**2+LLL**2)
        B=SQRT(RR1**2+LLL**2)
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        RR11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)
        D(I,J)=BI2*RR11+BI2*H*ALOG(AD)
 420    CONTINUE
        DO 430 J=NN1,N2
        S1=ABS(XI(J)-XI(I))
        S2=ABS(ETA(J)-ETA(I))
        S3=ABS(XI(J+1)-XI(I))
        S4=ABS(ETA(J+1)-ETA(I))
        S5=ABS(XI(J+1)-XI(J))
        S6=ABS(ETA(J+1)-ETA(J))
        A=SQRT(S1**2+S2**2)
        B=SQRT(S3**2+S4**2)
        H=SQRT(S5**2+S6**2)
        IF(S3.EQ.0.0.AND.S4.EQ.0.0) GO TO 411
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        SS2=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)+H*ALOG(AD)
        GIDS= BI3(J)*SS2
        IF(S5.EQ.0.0.OR.S6.EQ.0.0) D(I,J)=GIDS
        IF(S5.EQ.0.0.OR.S6.EQ.0.0) GO TO 422
C       TRANSFORM THE CO-ORDINATE SYSTEM
```

```
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS= PSI
      D(I,J)= (DGDNDS+GIDS)
      GO TO 422
 411  GIDS=BI3(J)*(H*(ALOG(H)-1))+BI3(J)*H*ALOG(AD)
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 422
      SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
      SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
      IF(SLOPE1.EQ.SLOPE2) D(I,J)=GIDS
      IF(SLOPE1.EQ.SLOPE2) GO TO 422
C     CALCULATE THE RADIUS OF CURVATURE
C     TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
      XP(I)=XI(I)-XI(I)
      XP(I+1)=XI(I+1)-XI(I)
      XP(I-1)=XI(I-1)-XI(I)
      YP(I)=ETA(I)-ETA(I)
      YP(I+1)=ETA(I+1)-ETA(I)
      YP(I-1)=ETA(I-1)-ETA(I)
      ALPHA=2*(XP(I-1)-XP(I+1))
      BETTA=2*(YP(I-1)-YP(I+1))
      GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
      ZETA=(XP(I+1)**2+YP(I+1)**2)
      MU=XP(I+1)*2
      DELTA=2*YP(I+1)
      ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
      BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
      AAA=SQRT(ABAR**2+BBAR**2)
C     CHECK FOR CONCAVITY & CONVEXITY
C     CALCULATE BETA(I)
      BETA(I)=(PI/2.)+PI
C     CALCULATE BETA(I-1)
```

```
      IF(XP(I).EQ.0.0)THEN
      BETA(I-1)=PI/2.
      ELSE
      BETA(I-1)=ATAN(YP(I)/XP(I))
      END IF
      IF(YP(I).GT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)
      IF(YP(I).GT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)+2*PI
C     EVALUATE THE INTEGRAL DGDNDS
      IF(BETA(I).GT.BETA(I-1))THEN
      DGDNDS=H/(2*AAA)
      ELSE IF(BETA(I).LT.BETA(I-1))THEN
      DGDNDS=-H/(2*AAA)
      ELSE
      DGDNDS=0.0
      END IF
      D(I,J)= (DGDNDS+GIDS)
 422  CON=CON+GIDS/BI3(J)
 430  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      DO 440 J=NN2,N3
      HH1=ABS(ETA(J+1)-ETA(I))
      HH2=ABS(ETA(J)-ETA(I))
      IF(HH1.EQ.0.0)D(I,J)=BI4*( HH2*(ALOG(HH2)-1))+HH2*BI4*ALOG(AD)
      IF(HH2.EQ.0.0)D(I,J)=BI4*(HH1*(ALOG(HH1)-1))+HH1*BI4*ALOG(AD)
      IF(HH1.EQ.0.0.OR.HH2.EQ.0.0) GO TO 440
      D(I,J)=BI4*(HH1*(ALOG(HH1)-1)-HH2*(ALOG(HH2)-1))
     *+BI4*(HH1*ALOG(AD)-HH2*ALOG(AD))
      IF(I.LT.J) GO TO 440
      IF(I.GT.J) D(I,J)=-D(I,J)
 440  CONTINUE
 400  CONTINUE
      DO 710 I=1,N3
      DO 710 J=1,N3
 710  D(I,J)=D(I,J)*(-0.159154)
      DO 800 I=1,N3
 800  D(I,I)=D(I,I)+0.5
      D(1,1)=D(1,1)+(ALPHA1/(2.*PI))-0.5
      D(NN,NN)=D(NN,NN)+(ALPHA2/(2.*PI))-0.5
      D(NN1,NN1)=D(NN1,NN1)+(ALPHA3/(2.*PI))-0.5
      D(NN2,NN2)=D(NN2,NN2)+(ALPHA4/(2.*PI))-0.5
C     TAKE CARE OF THE ANGLES FOR THE CURVED BOUNDARY
C     DO 99 J=NN1,N2
C     QQ5=ABS(ETA(J+1)-ETA(J))
C     QQ6=ABS(XI(J+1)-XI(J))
C     H=SQRT(QQ5**2+QQ6**2)
C     IF(XI(J).LT.(LLL/2.)) THEN
C     D(J,J)=D(J,J)-(H/(4*PI*6.25))
C     ELSE
C     D(J,J)=D(J,J)+(H/(4.*PI*6.25))
```

```
C       END IF
C99     CONTINUE
        F(NNN)=0.
C-----------------------------------------------------------------------
C       COMPUTE COEFF.  Q0,Q1,Q2......EQUATION 26.
C-----------------------------------------------------------------------
        NNN=N3+1
        DO 25  K=1,M
        DO 510 I=1,N
        R1(I)=SQRT((XI(I)-XT(K))**2+YT(K)**2)
        D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
        DAL1=(-0.5*AA(K)**2*(ETA(I)-YT(K)))/(R1(I)**2)
        DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
        D(I,NNN+1)=DAL1
        D(I,NNN+2)=DAL2
 510    CONTINUE
        DO 520 I=NN,N1
        R1(I)=SQRT((ETA(I)-YT(K))**2+(LLL-XT(K))**2)
        D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
        DAL1=(-0.5*AA(K)**2*( (ETA(I)-YT(K))))/(R1(I)**2)
        DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
        D(I,NNN+1)=DAL1
        D(I,NNN+2)=DAL2
 520    CONTINUE
        DO 530 I=NN1,N2
        R1(I)=SQRT((ETA(I)-YT(K))**2+(XI(I)-XT(K))**2)
        D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
        DAL1=(-0.5*AA(K)**2* (ETA(I)-YT(K)))/(R1(I)**2)
        DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
        D(I,NNN+1)=DAL1
        D(I,NNN+2) =DAL2
 530    CONTINUE
        DO 540 I=NN2,N3
        R1(I)=SQRT((ETA(I)-YT(K))**2+XT(K)**2)
        D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
        DAL1=(-0.5*AA(K)**2* (ETA(I)-YT(K)))/(R1(I)**2)
        DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
        D(I,NNN+1)=DAL1
        D(I,NNN+2)=DAL2
 540    CONTINUE
        NNN=NNN+3
 25     CONTINUE
C-----------------------------------------------------------------------
C       I AT THE CENTER OF THE TUBES , EQUATION 27...
C-----------------------------------------------------------------------
        DO 35 K= 1,M
        NNN=N3+K
        DO 610 J=1,N
        T1=ABS(XI(J+1)-XT(K))
        T2=ABS(XI(J)-XT(K))
```

```
      T3=YT(K)
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(T2**2+T3**2)
      B=SQRT(T1**2+T3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      T11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI1*T11+BI1*H*ALOG(AD)
 610  CONTINUE
      DO 620 J=NN,N1
      U1=ABS(ETA(J+1)-YT(K))
      U2=ABS(ETA(J)-YT(K))
      U3=LLL-XT(K)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(U2**2+U3**2)
      B=SQRT(U1**2+U3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      U11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI2*U11+BI2*H*ALOG(AD)
 620  CONTINUE
      CON=0.0
      DO 630 J=NN1,N2
      V1=ABS(XI(J+1)-XT(K))
      V2=ABS(XI(J)-XT(K))
      V3=ABS(ETA(J)-YT(K))
      V4=ABS(ETA(J+1)-YT(K))
      V5=ABS(XI(J+1)-XI(J))
      V6=ABS( ETA(J+1)-ETA(J))
      H=SQRT(V5**2+V6**2)
      A=SQRT(V2**2+V3**2)
      B=SQRT(V1**2+V4**2)
      YY=(A**2+H**2-B**2)/(2.*A*H)
      IF(YY.GT.1.) YY=1.
      IF(YY.LT.-1.) YY=-1.
      YY1=(A**2+B**2-H**2)/(2.*A*B)
      IF(YY1.GT.1.) YY1=1.
      IF(YY1.LT.-1.) YY1=-1.
      THETA1=ACOS(YY)
      PSI=ACOS(YY1)
      V11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+V11
      GIDS=BI3(J)*V11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(NNN,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 630
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XT(K)
```

```
      XIP(J+1)=XI(J+1)-XT(K)
      ETAP(J)=ETA(J)-YT(K)
      ETAP(J+1)=ETA(J+1)-YT(K)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS= PSI
      D(NNN,J)= (DGDNDS+GIDS)
 630  CONTINUE
      F(NNN)=(CON*HEAT)*(-0.159154)
      DO 640 J=NN2,N3
      TT1=ABS(ETA(J+1)-YT(K))
      TT2=ABS(ETA(J)-YT(K))
      TT3=XT(K)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(TT2**2+TT3**2)
      B=SQRT(TT1**2+TT3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      TT11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI4*TT11+BI4*H*ALOG(AD)
 640  CONTINUE
 35   CONTINUE
C---------------------------------------------------------------
C     EQUATION 27 - COEFFICIENT OF Q0,Q1,Q2
C---------------------------------------------------------------
      DO 77 KK=1,M
      M1=1
      DO 88 LL=1,M
      IF(KK.EQ.LL) GO TO 87
      GGG1(KK,LL)=YT(KK)-YT(LL)
      GGG2(KK,LL)=XT(KK)-XT(LL)
      RAL(KK,LL)=SQRT(GGG1(KK,LL)**2+GGG2(KK,LL)**2)
      D(N3+KK,N3+M1)=AA(LL)*(ALOG(RAL(KK,LL))+ALOG(AD))
      D(N3+KK,N3+M1+1)=-(0.5*AA(LL)**2*GGG1(KK,LL))/(RAL(KK,LL)**2)
      D(N3+KK,N3+M1+2)=-(0.5*AA(LL)**2*GGG2(KK,LL))/(RAL(KK,LL)**2)
      GO TO 86
```

```
 87     D(N3+KK,N3+M1)=AA(KK)*(ALOG(AA(KK))+ALOG(AD))
        D(N3+KK,N3+M1+1)=0.0
        D(N3+KK,N3+M1+2)=0.0
 86     M1=M1+3
 88     CONTINUE
 77     CONTINUE
C-----------------------------------------------------------------------
C       EQUATION 28 & EQUATION 29 , I AT THE CENTER OF THE TUBES
C-----------------------------------------------------------------------
        TH1=2*ASIN(1.)
        DO 810 K=1,M
        DO 820 I=1,N
        R1(I)=SQRT((XI(I)-XT(K))**2+YT(K)**2)
        R1(I+1)=SQRT((XI(I+1)-XT(K))**2+YT(K)**2)
        IF(XI(I).LT.XT(K).AND.XI(I+1).GT.XT(K)) GO TO 811
        D(N3+M+K,I)=(-BI1/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1  -ASIN((ABS(ETA(I)-YT(K)))/R1(I)))
        IF(XI(I).GE.XT(K)) D(N3+M+K,I)=-D(N3+M+K,I)
        GO TO 812
 811    D(N3+M+K,I)=(BI1/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))+
     1               ASIN((ABS(ETA(I)-YT(K)))/R1(I))-TH1)
 812    MM=2*M
        D(N3+MM+K,I)=(BI1/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
 820    CONTINUE
        DO 830 I=NN,N1
        R1(I)=SQRT((ETA(I)-YT(K))**2+(LLL-XT(K))**2)
        R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+(LLL-XT(K))**2)
        MM=2*M
        IF(ETA(I).LT.YT(K).AND.ETA(I+1).GT.YT(K)) GO TO 821
        D(N3+MM+K,I)=(BI2/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))
     1               - ASIN((ABS(XI(I)-XT(K)))/R1(I)))
        IF(ETA(I).GE.YT(K)) D(N3+MM+K,I)=-D(N3+MM+K,I)
        GO TO 822
 821    D(N3+MM+K,I)=(-BI2/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))+
     1    ASIN((ABS(XI(I)-XT(K)))/R1(I))-TH1)
 822    D(N3+M+K,I)=(BI2/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
 830    CONTINUE
        CON1=0.0;CON2=0.0
        DO 840 I=NN1,N2
        R1(I)=SQRT((ETA(I)-YT(K))**2+(XI(I)-XT(K))**2)
        R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+(XI(I+1)-XT(K))**2)
        IF(ETA(I).EQ.ETA(I+1).OR.XI(I).EQ.XI(I+1)) THEN
        IF(XI(I).GT.XT(K).AND.XI(I+1).LT.XT(K)) GO TO 831
        G1DS=(BI3(I)/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1         -ASIN((ABS(ETA(I)-YT(K)))/R1(I)))
        IF(XI(I).LE.XT(K))G1DS=-G1DS
        GO TO 832
 831    G1DS=(-BI3(I)/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1  + ASIN((ABS(ETA(I)-YT(K)))/R1(I))-TH1)
 832    D(N3+M+K,I)=G1DS
```

```
      CON1=CON1+G1DS/BI3(I)
      G2DS=(-BI3(I)/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
      MM=2*M
      D(N3+MM+K,I)=G2DS
      CON2=C0N2+G2DS/BI3(I)
      ELSE
C     EVALUATION OF G1DS,DG1DS,G2DS,DG2DS
      FF(I)=(ETA(I)-YT(K))/(R1(I)**2)
      FF(I+1)=(ETA(I+1)-YT(K))/(R1(I+1)**2)
      FFF(I)=(XI(I)-XT(K))/(R1(I)**2)
      FFF(I+1)=(XI(I+1)-XT(K))/(R1(I+1)**2)
      VV5=ABS(XI(I+1)-XI(I))
      VV6=ABS(ETA(I+1)-ETA(I))
      H=SQRT(VV5**2+VV6**2)
      G1DS= 0.5*H*BI3(I)*(FF(I)+FF(I+1))/AD
      G2DS= 0.5*H*BI3(I)*(FFF(I)+FFF(I+1))/AD
      CON1=CON1+G1DS/BI3(I)
      CON2=CON2+G2DS/BI3(I)
      MM=2*M
      SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
      SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
      IF(SLOPE1.EQ.SLOPE2) D(N3+M+K,I)=G1DS;D(N3+MM+K,I)=G2DS
      IF(SLOPE1.EQ.SLOPE2) GO TO 840
C     TRANSFORM THE CO-ORDINATE SYSTEM TO THE CETRE OF THE TUBE
      XIP(I)=XI(I)-XT(K)
      ETAP(I)=ETA(I)-YT(K)
      YTP(K)=YT(K)-YT(K)
      XTP(K)=XT(K)-XT(K)
      ETAP(I+1)=ETA(I+1)-YT(K)
      XIP(I+1)=XI(I+1)-XT(K)
      DET2= ASIN((ETAP(I)-YTP(K))/R1(I))
      IF(ETAP(I).GT.0.0.AND.XIP(I).LT.0.0) DET2=PI-DET2
C     CALCULATE THE RADIUS OF CURVATURE
C     TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
      XP(I)=XI(I)-XI(I)
      XP(I+1)=XI(I+1)-XI(I)
      XP(I-1)=XI(I-1)-XI(I)
      YP(I)=ETA(I)-ETA(I)
      YP(I+1)=ETA(I+1)-ETA(I)
      YP(I-1)=ETA(I-1)-ETA(I)
      YTP(K)=YT(K)-ETA(I)
      XTP(K)=XT(K)-XI(I)
      ALPHA=2*(XP(I-1)-XP(I+1))
      BETTA=2*(YP(I-1)-YP(I+1))
      GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
      ZETA=(XP(I+1)**2+YP(I+1)**2)
      MU=XP(I+1)*2
      DELTA=2*YP(I+1)
      ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
      BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
```

```
      AAA=SQRT(ABAR**2+BBAR**2)
      DET1=ASIN(ABS(YP(I)-BBAR)/AAA)
      IF(ABAR.GT.0.0.AND.BBAR.LT.0.0)DET1=PI-DET1
      IF(ABAR.LT.0.0.AND.BBAR.GT.0.0)DET1=PI-DET1
      DET= DET2 - DET1
      ZZ1(I)=-((YP(I)-YTP(K))/(R1(I)**3))*COS(DET)
      ZZ2(I)=((XP(I)-XTP(K))/(R1(I)**3))*SIN(DET)
      ZZ1(I+1)=-((YP(I+1)-YTP(K))/(R1(I+1)**3))*COS(DET)
      ZZ2(I+1)=((XP(I+1)-XTP(K))/(R1(I+1)**3))*SIN(DET)
      ZZZ1(I)=-((XP(I)-XTP(K))/(R1(I)**3))*COS(DET)
      ZZZ2(I)=-((YP(I)-YTP(K))/(R1(I)**3))*SIN(DET)
      ZZZ1(I+1)=-((XP(I+1)-XTP(K))/(R1(I+1)**3))*COS(DET)
      ZZZ2(I+1)=-((YP(I+1)-YTP(K))/(R1(I+1)**3))*SIN(DET)
      DG1DS= 0.5*H*(ZZ1(I)+ZZ1(I+1)+ZZ2(I)+ZZ2(I+1))/AD
      DG2DS= 0.5*H*(ZZZ1(I)+ZZZ2(I)+ZZZ1(I+1)+ZZZ2(I+1))/AD
      D(N3+M+K,I)=(G1DS+DG1DS)
      D(N3+MM+K,I)=(G2DS+DG2DS)
      END IF
 840  CONTINUE
      F(N3+M+K)=CON1*HEAT
      F(N3+MM+K)=CON2*HEAT
      DO 850 I=NN2,N3
      R1(I)=SQRT((ETA(I)-YT(K))**2+XT(K)**2)
      R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+XT(K)**2)
      MM=2*M
      IF(ETA(I).GT.YT(K).AND.ETA(I+1).LT.YT(K)) GO TO 851
      D(N3+MM+K,I)=(-BI4/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))-
     1             ASIN((ABS(XI(I)-XT(K)))/  R1(I)))
      IF(ETA(I).LE.YT(K)) D(N3+MM+K,I)=-D(N3+MM+K,I)
      GO TO 852
 851  D(N3+MM+K,I)=(BI4/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))+
     1             ASIN((ABS(XI(I)-XT(K)))/  R1(I))-TH1)
 852  D(N3+M+K,I)=(-BI4/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
 850  CONTINUE
 810  CONTINUE
C------------------------------------------------------------------
C     EQUATION 28 & 29 COEFFICIENT OF Q0,Q1,Q2
C------------------------------------------------------------------
      DO 360 K=1,M
      M1=1
      DO 370 L=1,M
      IF(K.EQ.L) GO TO 380
      D(N3+M+K,N3+M1)=(2*PI*AA(L)*GGG1(K,L))/(AD*RAL(K,L)**2)
      D(N3+M+K,N3+M1+1)=-((AA(L)**2*PI)/(AD*RAL(K,L)**2))
     1                  *(COS(2*ACOS(GGG2(K,L)/RAL(K,L))))
      D(N3+M+K,N3+M1+2)=-((AA(L)**2*PI)/(AD*RAL(K,L)**2))
     1                  *(SIN(2*ASIN(GGG1(K,L)/RAL(K,L))))
      GO TO 390
 380  D(N3+M+K,N3+M1)=0.0
      D(N3+M+K,N3+M1+1)= -PI/AD
```

```
      D(N3+M+K,N3+M1+2)=0.0
 390  M1=M1+3
 370  CONTINUE
 360  CONTINUE
C-----------------------------------------------------------------
      DO 1001 K=1,M
      M1=1
      DO 1010 L=1,M
      IF(K.EQ.L) GO TO 1030
      MM=2*M
      D(N3+MM+K,N3+M1)=(2*PI*AA(L)*GGG2(K,L))/(AD*RAL(K,L)**2)
      D(N3+MM+K,N3+M1+1)=((PI*AA(L)**2)/(AD*RAL(K,L)**2))
     1                   *(SIN(2*ASIN(GGG1(K,L)/RAL(K,L))))
      D(N3+MM+K,N3+M1+2)=((PI*AA(L)**2)/(AD*RAL(K,L)**2))
     1                  *(COS(2*ACOS(GGG2(K,L)/RAL(K,L))))
      GO TO 1040
 1030 D(N3+MM+K,N3+M1)=0.0
      D(N3+MM+K,N3+M1+1)=0.0
      D(N3+MM+K,N3+M1+2)=-PI/AD
 1040 M1=M1+3
 1010 CONTINUE
 1001 CONTINUE
      NNN=N3+M
      NN33=N3+1
      DO 700 I=NN33,NNN
      DO 700 J=1,N3
      D(I,J)=D(I,J)*(- 0.159154)
 700  CONTINUE
C-----------------------------------------------------------------
C     GENERATE THE R.H.S.
C-----------------------------------------------------------------
      DO 45  K= 1,M
      NNN=N3+K
      F(NNN)=F(NNN)-PHIO(K)
 45   CONTINUE
      DO 28 L=1,M
      F(N3+M+L)=0.
 28   CONTINUE
      DO 29 L=1,M
      MM=2*M
      F(N3+MM+L)=0.
 29   CONTINUE
C-----------------------------------------------------------------
C     SOLVE THE ASSEMBLED MATRIX
C-----------------------------------------------------------------
      NNNN=N3+3*M
C     IDGT IS AN INPUT OPTION.IF IDGT IS GREATER THAN ZERO,THE
C     ELEMENTS OF THE MATRIX AND THE R.H.S. VECTOR ARE ASSUMED
C     TO BE CORRECT TO THE IDGT DECIMAL DIGITS AND THE ROUTINE
C     PERFORMS AN ACCURACY TEST.
C     IF IDGT IS EQUAL TO ZERO, THE ACCURACY TEST IS BYPASSED
```

```
      MM1=1;N9=NNNN;IA=100;IDGT=3
      CALL LEQT1F(D,MM1,N9,IA,F,IDGT,WKAREA,IER)
C-----------------------------------------------------------------------
C     PRINT ALL THE INPUTS USED
C-----------------------------------------------------------------------
      WRITE(23,90)
 90   FORMAT(1H1//1X,75(1H-),//15X,'INPUT QUANTITIES'//1X,75(1H-),//)
      WRITE(23,101) M
 101  FORMAT(1H ,'THE NUMBER OF TUBES IN THE HEAT CONDUCTOR = ',I3)
      WRITE(23,102) LLL,W
 102  FORMAT(1H ,'NON-DIM LENGTH= ',F8.3,1X,'NON-DIM WIDTH=',F8.3)
      DO 103 K=1,M
      WRITE(23,104) K,XT(K),K,YT(K),K,PHIO(K),K,AA(K)
 104  FORMAT(1H ,'XT(',I2,') = 'F8.3,3X,'YT(',I2,') = 'F8.3,3X,
     1 'PHIO(',I2,') = 'F8.3,3X,'AA(',I2,') = 'F8.3)
 103  CONTINUE
      WRITE(23,204) BI1
 204  FORMAT(1H ,'BIOT NO. ON BOUNDARY ONE = ',F8.3)
      WRITE(23,205) BI2
 205  FORMAT(1H ,'BIOT NO. ON BOUNDARY TWO = ',F8.3)
      WRITE(23,206) BI4
 206  FORMAT(1H ,'BIOT NO. ON BOUNDARY FOUR = ',F8.3)
      DO 207 J= NN1,N2
      WRITE(23,208) J,BI3(J)
 208  FORMAT(1H ,'BI(',I3,') = ',F8.3)
 207  CONTINUE
      WRITE(23,211) HEAT
 211  FORMAT(1H ,'THE NON-DIM HEAT FLUX IN  TO THE
     1 HEAT CONDUCTOR = ',F8.3)
      WRITE(23,209)
 209  FORMAT(1H1//1X,75(1H-),/25X,'OUTPUT QUANTITIES',//75(1H-),//)
      WRITE(5,*)IER
C     PRINT ERROR MESSAGES FROM THE MATRIX SOLVER
      IF(IER.EQ.129) WRITE(23,301)
 301  FORMAT(1H ,20X,'ERROR MESSAGES',//75(1H-),//
     1 'MATRIX IS ALGORITHMICALLY SINGULAR')
      IF(IER.EQ.34) WRITE(23,302)
 302  FORMAT(1H ,20X,'ERROR MESSAGES',//75(1H-),//
     1 2X,'ACCURACY TEST FAILED.')
C     PRINT THE TEMPERATURE DISTRIBUTION
      WRITE(23,41)
 41   FORMAT(/,1X,'THE NON-DIM TEMPERATURES ON THE BOUNDARY',/)
      DO 900 J=1,N3
      WRITE(23,42) J,F(J)
 42   FORMAT(1H ,'PHI(',I3,') = 'F8.3)
 900  CONTINUE
C     PRINT THE HEAT TRANSFER SOLUTION
      WRITE(23,43)
 43  FORMAT(/,1X,'THE QO S ARE',/)
```

```
      WRITE(23,51)
51    FORMAT(/2X'TUBE',7X,'QO',10X,'Q1',7X,'Q2')
      M1=N3+1
      M2=M1+1
      M3=M1+2
      DO 52 L=1,M
      WRITE(23,54)L,F(M1),F(M2),F(M3)
54    FORMAT(/1X,I3,5X,3F10.6)
      M1=M1+3
      M2=M1+1
      M3=M1+2
52    CONTINUE
      STOP
      END
C#######################################################################
```

APPENDIX D

OPTLOC: FORTRAN PROGRAM FOR OPTIMIZATION SOLUTION

```
C###########################################################################
C#                     APPENDIX D                                          #
C#                    PROGRAM OPTLOC                                       #
C############################################################################
C#        THIS PROGRAM CALCULATES THE TEMPERATURE DISTRIBUTION              #
C#        ON THE BOUNDARY OF A TWO DIMENSIONAL HEAT CONDUCTOR               #
C#        USING THE BOUNDARY INTEGRAL ANALYSIS ...............              #
C#        THE PROGRAM USES SUBROUTINE CONMIN TO DETERMINE THE               #
C#        OPTIMAL HOLE LOCATIONS FOR MINIMUM VARIATION IN SURFACE           #
C#        TEMPERATURE ON THE CAVITY SURFACE .................               #
C############################################################################
C
C                    INPUTS USED ARE
C############################################################################
C        PHIO(K)= THE NON-DIMENSIONAL TEMPERATURE SPECIFIED
C                 ON THE K'TH TUBE
C        BI1= BIOT NUMBER ON BOUNDARY ONE
C        BI2=BIOT NUMBER ON BOUNDARY TWO
C        BI3(J)=VARIABLE BIOT NUMBER ON BOUNDARY THREE
C        BI4=BIOT NUMBER ON BOUNDARY FOUR
C        XT(K)= XI CO-ORDINATE OF THE K'TH TUBE
C        YT(K)= ETA CO-ORDINATE OF THE K'TH TUBE
C        W1=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY ONE
C        W2=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY TWO
C        W3=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY THREE
C        W4=NUMBER OF DISCRETE SEGMENTS ON BOUNDARY FOUR
C        HEAT=HEAT FLUX INTO THE HEAT CONDUCTOR
C        WW=WIDTH OF THE HEAT CONDUCTOR
C        LE=LENGTH OF THE HEAT CONDUCTOR
C        AA(K)=RADIUS OF THE K'TH TUBE
C---------------------------------------------------------------------------
        COMMON/CNMN1/DELFUN,DABFUN,FDCH,FDCHM,CT,CTMIN,CTL,CTLMIN,
     *  ALPHAX,ABOBJ1,THETA1,OBJ,NDV,NCON,NSIDE,IPRINT,NFDG,NSCAL,
     *  LINOBJ,ITMAX,ITRM,ICNDIR,IGOTO,NAC,INFO,INFOG,ITER
        DIMENSION X(7),VLB(7),VUB(7),G(50),SCAL(7),DF(7),AAAA(7,30),
     *  SS1(7),G1(50),G2(50),BBB(30,30),C(10),ISC(50),IC(60),MS1(20)
C
        DIMENSION D(100,100),XI(100),ETA(100),R1(100),R2(100),SS(80,80)
        DIMENSION F(100),WKAREA(250),TEP(100),XIP(100),ETAP(100),S(80)
        DIMENSION XT(30),YT(30),PHIO(10),GGG1(10,10),GGG2(10,10)
        DIMENSION XP(60),YP(60),BETA(60) ,AA(10),RAL(60,60)
        DIMENSION FF(100),FFF(100),ZZ1(100),ZZ2(100),ZZZ1(100),ZZZ2(100)
        DIMENSION YTP(10),XTP(10),BI3(100)
        REAL LLL,LE
        INTEGER IA,IDGT
        DATA W1,W2,W3,W4/10,4,20,4/
        DATA WW,LE/10.,25./
        DATA AA(1),AA(2),AA(3),AA(4)/1.,1.,1.,1./
        DATA XT(1),XT(2),YT(1),YT(2)/5.,20.,5.,5./
        DATA XT(3),YT(3),XT(4),YT(4) /15.,5.,10.,5./
```

```
      DATA PHIO(1),PHIO(2)/180.,180./
      DATA PHIO(3),PHIO(4)/180.,180./
      DATA PHIC,PHIA/150.,25./
      DATA M/2/
      DATA HEAT/0.0/
      DATA BI1,BI2,BI4/.2,.2,.2/
      AD=0.0
      DO 11 K=1,M
      IF(AA(K).GT.AD) AD=AA(K)
 11   CONTINUE
C     NON-DIMENSIONALISE ALL PARAMETERS
      DO 13 K=1,M
      XT(K)=XT(K)/AD
      YT(K)=YT(K)/AD
      AA(K)=AA(K)/AD
      PHIO(K)=(PHIO(K)-PHIA)/(PHIC-PHIA)
 13   CONTINUE
      PI=2.*ASIN(1.)
      ALPHA1=PI/2.
      ALPHA2=PI/2.
      ALPHA3=PI/2.
      ALPHA4=PI/2.
      W=WW/AD
      LLL=LE/AD
      N=W1
      N1=W1+W2
      N2=W1+W2+W3
      N3=W1+W2+W3+W4
      NN=N+1
      NN1=N1+1
      NN2=N2+1
      NNN2=W1+W2+(W3/2)
      NNNN2=NNN2+1
      DO 12  J=NN1,N2
      BI3(J)=.2
 12   CONTINUE
      K1=7;K2=50;K3=30;K4=10;K5=20
C     INITIALISE CONTROL PARAMETERS
      IPRINT=2;NDV=4;ITMAX=40;NCON=5;NFDG=0;NSIDE=0;ICNDIR=0;NSCAL=0
      LINOBJ=0;ITRM=0;FDCH=0.;FDCHM=0.;CT=0.;CTMIN=0.;THETA=0.
      ABOBJ1=0.5;DELFUN=1.0E-3;DABFUN=1.0E-3;ALPHAX=.01;PHI=0.
C     INITIALIZE X & ISC VECTORS
      DO 39 I=1,NCON
 39   ISC(I)=0
      X(1)=5.;X(2)=5.;X(3)=20.;X(4)=5.
      IGOTO=0
      DX1=LLL/W1
      DY2=W/W2
      DX3=LLL/W3
      DY4=W/W4
C-----------------------------------------------------------------
```

```
C       GENERATE ALL CO-ORDINATES OF THE PROBLEM
C-----------------------------------------------------------------------
        XI(1)=0.0
        DO 10 I=1,N
        ETA(I)=0.0
        XI(I+1)=XI(I)+DX1
  10    CONTINUE
        ETA(NN)=0.0
        DO 20 I=NN,N1
        XI(I)=LLL
        ETA(I+1)=ETA(I)+DY2
  20    CONTINUE
        XI(NN1)=LLL
        DO 30 I=NN1,N2
        XI(I+1)=XI(I)-DX3
  30    CONTINUE
        DO 31 I=NN1,NNN2
  31    ETA(I)=W
        ETA(NNNN2)=W
        DO 32 I=NNNN2,N2
  32    ETA(I)=W
        ETA(NN2)=W
        DO 40 I=NN2,N3
        XI(I)=0.0
        ETA(I+1)=ETA(I)-DY4
  40    CONTINUE
C-----------------------------------------------------------------------
C       EQUATION 26
C-----------------------------------------------------------------------
C       I ON BOUNDARY 1
C-----------------------------------------------------------------------
        DO 100 I=1,N
        CON=0.0
        DO 110 J=1,N
        A1=ABS(XI(J+1)-XI(I))
        B1=ABS(XI(J)-XI(I))
        H=ABS(XI(J+1)-XI(J))
        IF(A1.EQ.0.0)D(I,J)=BI1*(  B1*(ALOG(B1)-1))+BI1*B1*ALOG(AD)
        IF(B1.EQ.0.0)D(I,J)=BI1*(A1*(ALOG(A1)-1))+BI1*A1*ALOG(AD)
        IF(A1.EQ.0.0.OR.B1.EQ.0.0)GO TO 110
        D(I,J)=BI1*(A1*(ALOG(A1)-1)-B1*(ALOG(B1)-1))+BI1*ALOG(AD)*
     *  (A1-B1)
        IF(I.LT.J) GO TO 110
        IF(I.GT.J) D(I,J)=-D(I,J)
 110    CONTINUE
        DO 120 J=NN,N1
        C1=LLL-XI(I)
        D1=ETA(J+1)
        E1=ETA(J)
        H=D1-E1
```

```
      A=SQRT(E1**2+C1**2)
      B=SQRT(D1**2+C1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      C11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI2*(C11+H*ALOG(AD))
 120  CONTINUE
      DO 130 J=NN1,N2
      F1=ABS(XI(J+1)-XI(I))
      F2=ABS(XI(J)-XI(I))
      F3=ABS(ETA(J)-ETA(I))
      F4=ABS(ETA(J+1)-ETA(I))
      F5=ABS(ETA(J+1)-ETA(J))
      F6=ABS(XI(J+1)-XI(J))
      H=SQRT(F5**2+F6**2)
      A=SQRT(F2**2+F3**2)
      B=SQRT(F1**2+F4**2)
      YY=(A**2+H**2-B**2)/(2.*A*H)
      IF(YY.GT.1.) YY=1.
      IF(YY.LT.-1.) YY=-1.
      THETA1=ACOS(YY)
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      F11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+F11
      GIDS=BI3(J)*F11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 130
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS=PSI
```

```
      D(I,J)=(DGDNDS+GIDS)
130   CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      S(I)=(CON*HEAT)*(-0.159154)
      DO 140 J=NN2,N3
      A1=XI(I)
      BB=ETA(J)
      BB1=ETA(J+1)
      IF(A1.EQ.0.0.AND.BB1.EQ.0.0)D(I,J)=BI4*(BB*(ALOG(BB)-1))
     * +BB*ALOG(AD)
      IF(A1.EQ.0.0.AND.BB1.EQ.0.0) GO TO 140
      IF(A1.EQ.0.0)D(I,J)=-BI4*(BB1*(ALOG(BB1)-1)-BB*(ALOG(BB)-1))
     * +BI4*ALOG(AD)*(BB1-BB)
      IF(A1.EQ.0.0) GO TO 140
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(BB**2+A1**2)
      B=SQRT(BB1**2+A1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      AA11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI4*(AA11+ALOG(AD))
140   CONTINUE
100   CONTINUE
C-----------------------------------------------------------------------
C     I ON BOUNDARY 2
C-----------------------------------------------------------------------
      DO 200 I=NN,N1
      CON=0.0
      DO 210 J=1,N
      GG1=LLL-XI(J+1)
      GG2=LLL-XI(J)
      G3=ETA(I)
      IF(GG1.EQ.0.0.AND.G3.EQ.0.0)D(I,J)= BI1*(GG2*ALOG(GG2)-GG2)
     * +BI1*GG2*ALOG(AD)
      IF(GG1.EQ.0.0.AND.G3.EQ.0.0) GO TO 210
      IF(G3.EQ.0.0)D(I,J)=-(BI1)*((GG1*ALOG(GG1)-GG1)-(GG2*
     1 ALOG(GG2)-GG2))+BI1*GG2*ALOG(AD)-BI1*GG1*ALOG(AD)
      IF(G3.EQ.0.0)GO TO 210
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(GG2**2+ETA(I)**2)
      B=SQRT(GG1**2+ETA(I)**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      G11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI1*(G11+H*ALOG(AD))
210   CONTINUE
      DO 220 J=NN,N1
      H1=ABS(ETA(J+1)-ETA(I))
```

```
      H2=ABS(ETA(J)-ETA(I))
      H=ABS(ETA(J+1)-ETA(J))
      IF(H1.EQ.0.0)D(I,J)=BI2*( H2*(ALOG(H2)-1))+BI2*H2*ALOG(AD)
      IF(H2.EQ.0.0)D(I,J)=BI2*(H1*(ALOG(H1)-1))+BI2*H1*ALOG(AD)
      IF(H1.EQ.0.0.OR.H2.EQ.0.0)GO TO 220
      D(I,J)=BI2*(H1*(ALOG(H1)-1)-H2*(ALOG(H2)-1))
     * +BI2*(H1*ALOG(AD)-H2*ALOG(AD))
      IF(I.LT.J)GO TO 220
      IF(I.GT.J) D(I,J)=-D(I,J)
 220  CONTINUE
      DO 230 J=NN1,N2
      O1=ABS(ETA(J)-ETA(I))
      O2=ABS(XI(J)-XI(I))
      O3=ABS(ETA(J+1)-ETA(I))
      O4=ABS(XI(J+1)-XI(I))
      O5=ABS(ETA(J+1)-ETA(J))
      O6=ABS(XI(J+1)-XI(J))
      A=SQRT(O1**2+O2**2)
      B=SQRT(O3**2+O4**2)
      H=SQRT(O5**2+O6**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*H*A))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      O11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+O11
      GIDS=BI3(J)*O11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 230
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS= PSI
      D(I,J)=(DGDNDS+GIDS)
```

```
 230    CONTINUE
        F(I)=(CON*HEAT)*(-0.159154)
        S(I)=(CON*HEAT)*(-0.159154)
        DO 240 J=NN2,N3
        GG1=ABS(ETA(J+1)-ETA(I))
        GG2=ABS(ETA(J)-ETA(I))
        GG3=ETA(I)
        H=ABS(ETA(J+1)-ETA(J))
        A=SQRT(GG2**2+LLL**2)
        B=SQRT(GG1**2+LLL**2)
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        GG11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)
        D(I,J)=BI4*GG11+BI4*H*ALOG(AD)
 240    CONTINUE
 200    CONTINUE
C-----------------------------------------------------------------------
C       I ON BOUNDARY 3
C-----------------------------------------------------------------------
        DO 300 I=NN1,N2
        CON=0.0
        DO 310 J=1,N
        P1=ABS(XI(J+1)-XI(I))
        P2=ABS(XI(J)-XI(I))
        H=ABS(XI(J+1)-XI(J))
        A=SQRT(P2**2+W**2)
        B=SQRT(P1**2+W**2)
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        P11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)
        D(I,J)=BI1*P11+BI1*H*ALOG(AD)
 310    CONTINUE
        DO 320 J=NN,N1
        Q1=W-ETA(J+1)
        Q2=W-ETA(J)
        Q3=LLL-XI(I)
        IF(Q1.EQ.0.0.AND.Q3.EQ.0.0)D(I,J)=BI2*(Q2*ALOG(Q2)-Q2)
     *  +BI2*Q2*ALOG(AD)
        IF(Q1.EQ.0.0.AND.Q3.EQ.0.0)GO TO 320
        IF(Q3.EQ.0.0)D(I,J)=-BI2*((Q1*ALOG(Q1)-Q1)-(Q2*ALOG
     1  (Q2)-Q2))+BI2*Q2*ALOG(AD)-BI2*Q1*ALOG(AD)
        IF(Q3.EQ.0.0)GO TO 320
        H=ABS(ETA(J+1)-ETA(J))
        A=SQRT(Q2**2+Q3**2)
        B=SQRT(Q1**2+Q3**2)
        THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
        PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
        Q11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
```

```
     1  SIN(THETA1)
        D(I,J)=BI2*Q11+BI2*H*ALOG(AD)
 320    CONTINUE
        DO 330 J=NN1,N2
        QQ1=ABS(XI(J+1)-XI(I))
        QQ2=ABS(XI(J)-XI(I))
        QQ3=ABS(ETA(J+1)-ETA(I))
        QQ4=ABS(ETA(J)-ETA(I))
        QQ5=ABS(ETA(J+1)-ETA(J))
        QQ6=ABS(XI(J+1)-XI(J))
        A=SQRT(QQ2**2+QQ4**2)
        B=SQRT(QQ1**2+QQ3**2)
        H=SQRT(QQ5**2+QQ6**2)
        IF(QQ1.EQ.0.0.OR.QQ2.EQ.0.0) GO TO 311
        YY=(A**2+B**2-H**2)/(2.*A*B)
        IF(YY.GT.1.) YY=1.0
        IF(YY.LT.-1.) YY=-1.
        YY1=(A**2+H**2-B**2)/(2.*A*H)
        IF(YY1.GT.1.) YY1=1.
        IF(YY1.LT.-1.) YY1=-1.
        THETA1=ACOS(YY1)
        PSI=ACOS(YY)
        Q=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1  SIN(THETA1)+H*ALOG(AD)
        GIDS=BI3(J)*Q
        IF(QQ5.EQ.0.0.OR.QQ6.EQ.0.0)D(I,J)=GIDS
        IF(QQ5.EQ.0.0.OR.QQ6.EQ.0.0) GO TO 322
C       TRANSFORM THE CO-ORDINATE SYSTEM
        XIP(J)=XI(J)-XI(I)
        XIP(J+1)=XI(J+1)-XI(I)
        ETAP(J)=ETA(J)-ETA(I)
        ETAP(J+1)=ETA(J+1)-ETA(I)
        IF(XIP(J).EQ.0.0)THEN
        TEP(J)=PI/2.
        ELSE
        TEP(J)=ATAN(ETAP(J)/XIP(J))
        END IF
        IF(XIP(J+1).EQ.0.0)THEN
        TEP(J+1)=PI/2.
        ELSE
        TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
        END IF
        IF(TEP(J).EQ.(PI/2.)) THEN
        PSI=TEP(J+1)+PI-TEP(J)
        ELSE
        PSI=TEP(J+1)-TEP(J)
        END IF
        DGDNDS=PSI
        D(I,J)=(DGDNDS+GIDS)
        GO TO 322
```

```
 311      GIDS=BI3(J)*(H*(ALOG(H)-1))+H*ALOG(AD)*BI3(J)
          IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
          IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 322
          SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
          SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
          IF(SLOPE1.EQ.SLOPE2) D(I,J)=GIDS
          IF(SLOPE1.EQ.SLOPE2) GO TO 322
C         CALCULATE THE RADIUS OF CURVATURE
C         TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
          XP(I)=XI(I)-XI(I)
          XP(I+1)=XI(I+1)-XI(I)
          XP(I-1)=XI(I-1)-XI(I)
          YP(I)=ETA(I)-ETA(I)
          YP(I+1)=ETA(I+1)-ETA(I)
          YP(I-1)=ETA(I-1)-ETA(I)
          ALPHA=2*(XP(I-1)-XP(I+1))
          BETTA=2*(YP(I-1)-YP(I+1))
          GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
          ZETA=(XP(I+1)**2+YP(I+1)**2)
          MU=XP(I+1)*2
          DELTA=2*YP(I+1)
          ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
          BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
          AAA=SQRT(ABAR**2+BBAR**2)
C         CHECK FOR CONCAVITY & CONVEXITY
C         CALCULATE BETA'JS
          IF(XP(I+1).EQ.0.0)THEN
          BETA(I)=PI/2.
          ELSE
          BETA(I)=ATAN(YP(I+1)/XP(I+1))
          END IF
          IF(YP(I+1).GT.0.0.AND.XP(I+1).GT.0.0)BETA(I)=BETA(I)
          IF(YP(I+1).GT.0.0.AND.XP(I+1).LT.0.0)BETA(I)=BETA(I)+PI
          IF(YP(I+1).LT.0.0.AND.XP(I+1).LT.0.0)BETA(I)=BETA(I)+PI
          IF(YP(I+1).LT.0.0.AND.XP(I+1).GT.0.0)BETA(I)=BETA(I)+2*PI
C         CALCULATE BETA(I-1)
          IF(XP(I).EQ.0.0)THEN
          BETA(I-1)=PI/2.
          ELSE
          BETA(I-1)=ATAN(YP(I)/XP(I))
          END IF
          IF(YP(I).GT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)
          IF(YP(I).GT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
          IF(YP(I).LT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
          IF(YP(I).LT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)+2*PI
C         EVALUATE THE INTEGRAL DGDNDS
          IF(BETA(I).GT.BETA(I-1))THEN
          DGDNDS=H/(2*AAA)
          ELSE IF(BETA(I).LT.BETA(I-1))THEN
          DGDNDS=-H/(2*AAA)
```

```
      ELSE
      DGDNDS=0.0
      END IF
      D(I,J)=(DGDNDS+GIDS)
 322  CON=CON+GIDS/BI3(J)
 330  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      S(I)=(CON*HEAT)*(-0.159154)
      DO 340 J=NN2,N3
      PP1=XI(I)
      PP2=W-ETA(J)
      PP3=W-ETA(J+1)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(PP2**2+PP1**2)
      B=SQRT(PP3**2+PP1**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      PP11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI4*PP11+BI4*H*ALOG(AD)
 340  CONTINUE
 300  CONTINUE
C-----------------------------------------------------------------------
C       I  ON BOUNDARY FOUR
C-----------------------------------------------------------------------
      DO 400 I=NN2,N3
      CON=0.0
      DO 410 J=1,N
      Z1=XI(J+1)
      Z2=XI(J)
      Z3=ETA(I)
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(Z2**2+Z3**2)
      B=SQRT(Z1**2+Z3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      R11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(I,J)=BI1*R11+BI1*H*ALOG(AD)
 410  CONTINUE
      DO 420 J=NN,N1
      RR1=ABS(ETA(J+1)-ETA(I))
      RR2=ABS(ETA(J)-ETA(I))
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(RR2**2+LLL**2)
      B=SQRT(RR1**2+LLL**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      RR11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
```

```
      D(I,J)=BI2*RR11+BI2*H*ALOG(AD)
 420  CONTINUE
      DO 430 J=NN1,N2
      S1=ABS(XI(J)-XI(I))
      S2=ABS(ETA(J)-ETA(I))
      S3=ABS(XI(J+1)-XI(I))
      S4=ABS(ETA(J+1)-ETA(I))
      S5=ABS(XI(J+1)-XI(J))
      S6=ABS(ETA(J+1)-ETA(J))
      A=SQRT(S1**2+S2**2)
      B=SQRT(S3**2+S4**2)
      H=SQRT(S5**2+S6**2)
      IF(S3.EQ.0.0.AND.S4.EQ.0.0) GO TO 411
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      SS2=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      GIDS= BI3(J)*SS2
      IF(S5.EQ.0.0.OR.S6.EQ.0.0) D(I,J)=GIDS
      IF(S5.EQ.0.0.OR.S6.EQ.0.0) GO TO 422
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XI(I)
      XIP(J+1)=XI(J+1)-XI(I)
      ETAP(J)=ETA(J)-ETA(I)
      ETAP(J+1)=ETA(J+1)-ETA(I)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
      DGDNDS= PSI
      D(I,J)= (DGDNDS+GIDS)
      GO TO 422
 411  GIDS=BI3(J)*(H*(ALOG(H)-1))+BI3(J)*H*ALOG(AD)
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(I,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 422
      SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
      SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
      IF(SLOPE1.EQ.SLOPE2) D(I,J)=GIDS
      IF(SLOPE1.EQ.SLOPE2) GO TO 422
C     CALCULATE THE RADIUS OF CURVATURE
C     TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
```

```
      XP(I)=XI(I)-XI(I)
      XP(I+1)=XI(I+1)-XI(I)
      XP(I-1)=XI(I-1)-XI(I)
      YP(I)=ETA(I)-ETA(I)
      YP(I+1)=ETA(I+1)-ETA(I)
      YP(I-1)=ETA(I-1)-ETA(I)
      ALPHA=2*(XP(I-1)-XP(I+1))
      BETTA=2*(YP(I-1)-YP(I+1))
      GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
      ZETA=(XP(I+1)**2+YP(I+1)**2)
      MU=XP(I+1)*2
      DELTA=2*YP(I+1)
      ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
      BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
      AAA=SQRT(ABAR**2+BBAR**2)
C     CHECK FOR CONCAVITY & CONVEXITY
C     CALCULATE BETA(I)
      BETA(I)=(PI/2.)+PI
C     CALCULATE BETA(I-1)
      IF(XP(I).EQ.0.0)THEN
      BETA(I-1)=PI/2.
      ELSE
      BETA(I-1)=ATAN(YP(I)/XP(I))
      END IF
      IF(YP(I).GT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)
      IF(YP(I).GT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).LT.0.0)BETA(I-1)=BETA(I-1)+PI
      IF(YP(I).LT.0.0.AND.XP(I).GT.0.0)BETA(I-1)=BETA(I-1)+2*PI
C     EVALUATE THE INTEGRAL DGDNDS
      IF(BETA(I).GT.BETA(I-1))THEN
      DGDNDS=H/(2*AAA)
      ELSE IF(BETA(I).LT.BETA(I-1))THEN
      DGDNDS=-H/(2*AAA)
      ELSE
      DGDNDS=0.0
      END IF
      D(I,J)= (DGDNDS+GIDS)
 422  CON=CON+GIDS/BI3(J)
 430  CONTINUE
      F(I)=(CON*HEAT)*(-0.159154)
      S(I)=(CON*HEAT)*(-0.159154)
      DO 440 J=NN2,N3
      HH1=ABS(ETA(J+1)-ETA(I))
      HH2=ABS(ETA(J)-ETA(I))
      IF(HH1.EQ.0.0)D(I,J)=BI4*( HH2*(ALOG(HH2)-1))+HH2*BI4*ALOG(AD)
      IF(HH2.EQ.0.0)D(I,J)=BI4*(HH1*(ALOG(HH1)-1))+HH1*BI4*ALOG(AD)
      IF(HH1.EQ.0.0.OR.HH2.EQ.0.0) GO TO 440
      D(I,J)=BI4*(HH1*(ALOG(HH1)-1)-HH2*(ALOG(HH2)-1))
     * +BI4*(HH1*ALOG(AD)-HH2*ALOG(AD))
      IF(I.LT.J) GO TO 440
```

```
      IF(I.GT.J) D(I,J)=-D(I,J)
 440  CONTINUE
 400  CONTINUE
      DO 710 I=1,N3
      DO 710 J=1,N3
 710  D(I,J)=D(I,J)*(-0.159154)
      DO 800 I=1,N3
 800  D(I,I)=D(I,I)+0.5
      DO 1000 I=1,N3
      DO 1000 J=1,N3
 1000 SS(I,J)=D(I,J)
      SS(1,1)=SS(1,1)+(ALPHA1/(2.*PI))-0.5
      SS(NN,NN)=SS(NN,NN)+(ALPHA2/(2.*PI))-0.5
      SS(NN1,NN1)=SS(NN1,NN1)+(ALPHA3/(2.*PI))-0.5
      SS(NN2,NN2)=SS(NN2,NN2)+(ALPHA4/(2.*PI))-0.5
C     TAKE CARE OF THE ANGLES FOR THE CURVED BOUNDARY
C     DO 99 J=NN1,N2
C     QQ5=ABS(ETA(J+1)-ETA(J))
C     QQ6=ABS(XI(J+1)-XI(J))
C     H=SQRT(QQ5**2+QQ6**2)
C     IF(XI(J).LT.(LLL/2.)) THEN
C     SS(J,J)=SS(J,J)-(H/(4*PI*6.25))
C     ELSE
C     SS(J,J)=SS(J,J)+(H/(4.*PI*6.25))
C     END IF
C99   CONTINUE
 1111 CONTINUE
      F(NNN)=0.
      XT(1)=X(1)
      YT(1)=X(2)
      XT(2)=X(3)
      YT(2)=X(4)
C     XT(3)=X(5)
C     YT(3)=X(6)
      WRITE(5,*) X(1),X(2),X(3),X(4)
C------------------------------------------------------------------------
C     COMPUTE COEFF.  Q0,Q1,Q2........EQUATION 26
C------------------------------------------------------------------------
      NNN=N3+1
      DO 25  K=1,M
      DO 510 I=1,N
      R1(I)=SQRT((XI(I)-XT(K))**2+YT(K)**2)
      D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
      DAL1=(-0.5*AA(K)**2*(ETA(I)-YT(K)))/(R1(I)**2)
      DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
      D(I,NNN+1)=DAL1
      D(I,NNN+2)=DAL2
 510  CONTINUE
      DO 520 I=NN,N1
      R1(I)=SQRT((ETA(I)-YT(K))**2+(LLL-XT(K))**2)
```

```
      D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
      DAL1=(-0.5*AA(K)**2*( (ETA(I)-YT(K))))/(R1(I)**2)
      DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
      D(I,NNN+1)=DAL1
      D(I,NNN+2)=DAL2
520   CONTINUE
      DO 530 I=NN1,N2
      R1(I)=SQRT((ETA(I)-YT(K))**2+(XI(I)-XT(K))**2)
      D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
      DAL1=(-0.5*AA(K)**2* (ETA(I)-YT(K)))/(R1(I)**2)
      DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
      D(I,NNN+1)=DAL1
      D(I,NNN+2) =DAL2
530   CONTINUE
      DO 540 I=NN2,N3
      R1(I)=SQRT((ETA(I)-YT(K))**2+XT(K)**2)
      D(I,NNN)=AA(K)*(ALOG(R1(I))+ALOG(AD))
      DAL1=(-0.5*AA(K)**2* (ETA(I)-YT(K)))/(R1(I)**2)
      DAL2=(-0.5*AA(K)**2*(XI(I)-XT(K)))/(R1(I)**2)
      D(I,NNN+1)=DAL1
      D(I,NNN+2)=DAL2
540   CONTINUE
      NNN=NNN+3
25    CONTINUE
C-----------------------------------------------------------------
C     I AT THE CENTER OF THE TUBES , EQUATION 27...
C-----------------------------------------------------------------
      DO 35 K= 1,M
      NNN=N3+K
      DO 610 J=1,N
      T1=ABS(XI(J+1)-XT(K))
      T2=ABS(XI(J)-XT(K))
      T3=YT(K)
      H=ABS(XI(J+1)-XI(J))
      A=SQRT(T2**2+T3**2)
      B=SQRT(T1**2+T3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      T11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI1*T11+BI1*H*ALOG(AD)
610   CONTINUE
      DO 620 J=NN,N1
      U1=ABS(ETA(J+1)-YT(K))
      U2=ABS(ETA(J)-YT(K))
      U3=LLL-XT(K)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(U2**2+U3**2)
      B=SQRT(U1**2+U3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
```

```
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      U11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI2*U11+BI2*H*ALOG(AD)
 620  CONTINUE
      CON=0.0
      DO 630 J=NN1,N2
      V1=ABS(XI(J+1)-XT(K))
      V2=ABS(XI(J)-XT(K))
      V3=ABS(ETA(J)-YT(K))
      V4=ABS(ETA(J+1)-YT(K))
      V5=ABS(XI(J+1)-XI(J))
      V6=ABS( ETA(J+1)-ETA(J))
      H=SQRT(V5**2+V6**2)
      A=SQRT(V2**2+V3**2)
      B=SQRT(V1**2+V4**2)
      YY=(A**2+H**2-B**2)/(2.*A*H)
      IF(YY.GT.1.) YY=1.
      IF(YY.LT.-1.) YY=-1.
      YY1=(A**2+B**2-H**2)/(2.*A*B)
      IF(YY1.GT.1.) YY1=1.
      IF(YY1.LT.-1.) YY1=-1.
      THETA1=ACOS(YY)
      PSI=ACOS(YY1)
      V11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)+H*ALOG(AD)
      CON=CON+V11
      GIDS=BI3(J)*V11
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) D(NNN,J)=GIDS
      IF(XI(J).EQ.XI(J+1).OR.ETA(J).EQ.ETA(J+1)) GO TO 630
C     TRANSFORM THE CO-ORDINATE SYSTEM
      XIP(J)=XI(J)-XT(K)
      XIP(J+1)=XI(J+1)-XT(K)
      ETAP(J)=ETA(J)-YT(K)
      ETAP(J+1)=ETA(J+1)-YT(K)
      IF(XIP(J).EQ.0.0)THEN
      TEP(J)=PI/2.
      ELSE
      TEP(J)=ATAN(ETAP(J)/XIP(J))
      END IF
      IF(XIP(J+1).EQ.0.0)THEN
      TEP(J+1)=PI/2.
      ELSE
      TEP(J+1)=ATAN(ETAP(J+1)/XIP(J+1))
      END IF
      IF(TEP(J).EQ.(PI/2.)) THEN
      PSI=TEP(J+1)+PI-TEP(J)
      ELSE
      PSI=TEP(J+1)-TEP(J)
      END IF
```

```
      DGDNDS= PSI
      D(NNN,J)= (DGDNDS+GIDS)
630   CONTINUE
      F(NNN)=(CON*HEAT)*(-0.159154)
      DO 640 J=NN2,N3
      TT1=ABS(ETA(J+1)-YT(K))
      TT2=ABS(ETA(J)-YT(K))
      TT3=XT(K)
      H=ABS(ETA(J+1)-ETA(J))
      A=SQRT(TT2**2+TT3**2)
      B=SQRT(TT1**2+TT3**2)
      THETA1=ACOS((A**2+H**2-B**2)/(2.*A*H))
      PSI=ACOS((A**2+B**2-H**2)/(2.*A*B))
      TT11=A*COS(THETA1)*(ALOG(A)-ALOG(B))+H*(ALOG(B)-1)+A*PSI*
     1 SIN(THETA1)
      D(NNN,J)=BI4*TT11+BI4*H*ALOG(AD)
640   CONTINUE
35    CONTINUE
C-----------------------------------------------------------------------
C     EQUATION 27 - COEFFICIENT OF Q0,Q1,Q2
C-----------------------------------------------------------------------
      DO 77 KK=1,M
      M1=1
      DO 88 LL=1,M
      IF(KK.EQ.LL) GO TO 87
      GGG1(KK,LL)=YT(KK)-YT(LL)
      GGG2(KK,LL)=XT(KK)-XT(LL)
      RAL(KK,LL)=SQRT(GGG1(KK,LL)**2+GGG2(KK,LL)**2)
      D(N3+KK,N3+M1)=AA(LL)*(ALOG(RAL(KK,LL))+ALOG(AD))
      D(N3+KK,N3+M1+1)=-(0.5*AA(LL)**2*GGG1(KK,LL))/(RAL(KK,LL)**2)
      D(N3+KK,N3+M1+2)=-(0.5*AA(LL)**2*GGG2(KK,LL))/(RAL(KK,LL)**2)
      GO TO 86
87    D(N3+KK,N3+M1)=AA(KK)*(ALOG(AA(KK))+ALOG(AD))
      D(N3+KK,N3+M1+1)=0.0
      D(N3+KK,N3+M1+2)=0.0
86    M1=M1+3
88    CONTINUE
77    CONTINUE
C-----------------------------------------------------------------------
C     EQUATION 28 & EQUATION 29 , I AT THE CENTER OF THE TUBES
C-----------------------------------------------------------------------
      TH1=2*ASIN(1.)
      DO 810 K=1,M
      DO 820 I=1,N
      R1(I)=SQRT((XI(I)-XT(K))**2+YT(K)**2)
      R1(I+1)=SQRT((XI(I+1)-XT(K))**2+YT(K)**2)
      IF(XI(I).LT.XT(K).AND.XI(I+1).GT.XT(K)) GO TO 811
      D(N3+M+K,I)=(-BI1/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1 -ASIN((ABS(ETA(I)-YT(K)))/R1(I)))
      IF(XI(I).GE.XT(K)) D(N3+M+K,I)=-D(N3+M+K,I)
```

```
      GO TO 812
811   D(N3+M+K,I)=(BI1/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))+
     1             ASIN((ABS(ETA(I)-YT(K)))/R1(I))-TH1)
812   MM=2*M
      D(N3+MM+K,I)=(BI1/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
820   CONTINUE
      DO 830 I=NN,N1
      R1(I)=SQRT((ETA(I)-YT(K))**2+(LLL-XT(K))**2)
      R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+(LLL-XT(K))**2)
      MM=2*M
      IF(ETA(I).LT.YT(K).AND.ETA(I+1).GT.YT(K)) GO TO 821
      D(N3+MM+K,I)=(BI2/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))
     1             - ASIN((ABS(XI(I)-XT(K)))/R1(I)))
      IF(ETA(I).GE.YT(K)) D(N3+MM+K,I)=-D(N3+MM+K,I)
      GO TO 822
821   D(N3+MM+K,I)=(-BI2/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))+
     1  ASIN((ABS(XI(I)-XT(K)))/R1(I))-TH1)
822   D(N3+M+K,I)=(BI2/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
830   CONTINUE
      CON1=0.0;CON2=0.0
      DO 840 I=NN1,N2
      R1(I)=SQRT((ETA(I)-YT(K))**2+(XI(I)-XT(K))**2)
      R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+(XI(I+1)-XT(K))**2)
      IF(ETA(I).EQ.ETA(I+1).OR.XI(I).EQ.XI(I+1)) THEN
      IF(XI(I).GT.XT(K).AND.XI(I+1).LT.XT(K)) GO TO 831
      G1DS=(BI3(I)/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1      -ASIN((ABS(ETA(I)-YT(K)))/R1(I)))
      IF(XI(I).LE.XT(K))G1DS=-G1DS
      GO TO 832
831   G1DS=(-BI3(I)/AD)*(ASIN((ABS(ETA(I+1)-YT(K)))/R1(I+1))
     1 + ASIN((ABS(ETA(I)-YT(K)))/R1(I))-TH1)
832   D(N3+M+K,I)=G1DS
      CON1=CON1+G1DS/BI3(I)
      G2DS=(-BI3(I)/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
      MM=2*M
      D(N3+MM+K,I)=G2DS
      CON2=CON2+G2DS/BI3(I)
      ELSE
C     EVALUATION OF G1DS,DG1DS,G2DS,DG2DS
      FF(I)=(ETA(I)-YT(K))/(R1(I)**2)
      FF(I+1)=(ETA(I+1)-YT(K))/(R1(I+1)**2)
      FFF(I)=(XI(I)-XT(K))/(R1(I)**2)
      FFF(I+1)=(XI(I+1)-XT(K))/(R1(I+1)**2)
      VV5=ABS(XI(I+1)-XI(I))
      VV6=ABS(ETA(I+1)-ETA(I))
      H=SQRT(VV5**2+VV6**2)
      G1DS= 0.5*H*BI3(I)*(FF(I)+FF(I+1))/AD
      G2DS= 0.5*H*BI3(I)*(FFF(I)+FFF(I+1))/AD
      CON1=CON1+G1DS/BI3(I)
      CON2=CON2+G2DS/BI3(I)
```

```
      MM=2*M
      SLOPE1=(ETA(I)-ETA(I-1))/(XI(I)-XI(I-1))
      SLOPE2=(ETA(I+1)-ETA(I))/(XI(I+1)-XI(I))
      IF(SLOPE1.EQ.SLOPE2) D(N3+M+K,I)=G1DS;D(N3+MM+K,I)=G2DS
      IF(SLOPE1.EQ.SLOPE2) GO TO 840
C     TRANSFORM THE CO-ORDINATE SYSTEM TO THE CETRE OF THE TUBE
      XIP(I)=XI(I)-XT(K)
      ETAP(I)=ETA(I)-YT(K)
      YTP(K)=YT(K)-YT(K)
      XTP(K)=XT(K)-XT(K)
      ETAP(I+1)=ETA(I+1)-YT(K)
      XIP(I+1)=XI(I+1)-XT(K)
      DET2= ASIN((ETAP(I)-YTP(K))/R1(I))
      IF(ETAP(I).GT.0.0.AND.XIP(I).LT.0.0) DET2=PI-DET2
C     CALCULATE THE RADIUS OF CURVATURE
C     TRANSFORM THE CO-ORDINATE SYSTEM TO I'TH POINT
      XP(I)=XI(I)-XI(I)
      XP(I+1)=XI(I+1)-XI(I)
      XP(I-1)=XI(I-1)-XI(I)
      YP(I)=ETA(I)-ETA(I)
      YP(I+1)=ETA(I+1)-ETA(I)
      YP(I-1)=ETA(I-1)-ETA(I)
      YTP(K)=YT(K)-ETA(I)
      XTP(K)=XT(K)-XI(I)
      ALPHA=2*(XP(I-1)-XP(I+1))
      BETTA=2*(YP(I-1)-YP(I+1))
      GAMMA=(XP(I-1)**2-XP(I+1)**2)+(YP(I-1)**2-YP(I+1)**2)
      ZETA=(XP(I+1)**2+YP(I+1)**2)
      MU=XP(I+1)*2
      DELTA=2*YP(I+1)
      ABAR=((GAMMA*DELTA-BETTA*ZETA)/(ALPHA*DELTA-BETTA*MU))
      BBAR=((MU*GAMMA-ALPHA*ZETA)/(MU*BETTA-ALPHA*DELTA))
      AAA=SQRT(ABAR**2+BBAR**2)
      DET1=ASIN(ABS(YP(I)-BBAR)/AAA)
      IF(ABAR.GT.0.0.AND.BBAR.LT.0.0)DET1=PI-DET1
      IF(ABAR.LT.0.0.AND.BBAR.GT.0.0)DET1=PI-DET1
      DET= DET2 - DET1
      ZZ1(I)=-((YP(I)-YTP(K))/(R1(I)**3))*COS(DET)
      ZZ2(I)=((XP(I)-XTP(K))/(R1(I)**3))*SIN(DET)
      ZZ1(I+1)=-((YP(I+1)-YTP(K))/(R1(I+1)**3))*COS(DET)
      ZZ2(I+1)=((XP(I+1)-XTP(K))/(R1(I+1)**3))*SIN(DET)
      ZZZ1(I)=-((XP(I)-XTP(K))/(R1(I)**3))*COS(DET)
      ZZZ2(I)=-((YP(I)-YTP(K))/(R1(I)**3))*SIN(DET)
      ZZZ1(I+1)=-((XP(I+1)-XTP(K))/(R1(I+1)**3))*COS(DET)
      ZZZ2(I+1)=-((YP(I+1)-YTP(K))/(R1(I+1)**3))*SIN(DET)
      DG1DS= 0.5*H*(ZZ1(I)+ZZ1(I+1)+ZZ2(I)+ZZ2(I+1))/AD
      DG2DS= 0.5*H*(ZZZ1(I)+ZZZ2(I)+ZZZ1(I+1)+ZZZ2(I+1))/AD
      D(N3+M+K,I)=(G1DS+DG1DS)
      D(N3+MM+K,I)=(G2DS+DG2DS)
      END IF
```

```
 840    CONTINUE
        F(N3+M+K)=CON1*HEAT
        F(N3+MM+K)=CON2*HEAT
        DO 850 I=NN2,N3
        R1(I)=SQRT((ETA(I)-YT(K))**2+XT(K)**2)
        R1(I+1)=SQRT((ETA(I+1)-YT(K))**2+XT(K)**2)
        MM=2*M
        IF(ETA(I).GT.YT(K).AND.ETA(I+1).LT.YT(K)) GO TO 851
        D(N3+MM+K,I)=(-BI4/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))-
     1              ASIN((ABS(XI(I)-XT(K)))/  R1(I)))
        IF(ETA(I).LE.YT(K)) D(N3+MM+K,I)=-D(N3+MM+K,I)
        GO TO 852
 851    D(N3+MM+K,I)=(BI4/AD)*(ASIN((ABS(XI(I+1)-XT(K)))/R1(I+1))+
     1              ASIN((ABS(XI(I)-XT(K)))/  R1(I))-TH1)
 852    D(N3+M+K,I)=(-BI4/(AD))*(ALOG(R1(I+1))-ALOG(R1(I)))
 850    CONTINUE
 810    CONTINUE
C-----------------------------------------------------------------------
C       EQUATION 28 & 29 COEFFICIENT OF Q0,Q1,Q2
C-----------------------------------------------------------------------
        DO 360 K=1,M
        M1=1
        DO 370 L=1,M
        IF(K.EQ.L) GO TO 380
        D(N3+M+K,N3+M1)=(2*PI*AA(L)*GGG1(K,L))/(AD*RAL(K,L)**2)
        D(N3+M+K,N3+M1+1)=-((AA(L)**2*PI)/(AD*RAL(K,L)**2))
     1                  *(COS(2*ACOS(GGG2(K,L)/RAL(K,L))))
        D(N3+M+K,N3+M1+2)=-((AA(L)**2*PI)/(AD*RAL(K,L)**2))
     1                  *(SIN(2*ASIN(GGG1(K,L)/RAL(K,L))))
        GO TO 390
 380    D(N3+M+K,N3+M1)=0.0
        D(N3+M+K,N3+M1+1)= -PI/AD
        D(N3+M+K,N3+M1+2)=0.0
 390    M1=M1+3
 370    CONTINUE
 360    CONTINUE
C-----------------------------------------------------------------------
        DO 1001 K=1,M
        M1=1
        DO 1010 L=1,M
        IF(K.EQ.L) GO TO 1030
        MM=2*M
        D(N3+MM+K,N3+M1)=(2*PI*AA(L)*GGG2(K,L))/(AD*RAL(K,L)**2)
        D(N3+MM+K,N3+M1+1)=((PI*AA(L)**2)/(AD*RAL(K,L)**2))
     1                    *(SIN(2*ASIN(GGG1(K,L)/RAL(K,L))))
        D(N3+MM+K,N3+M1+2)=((PI*AA(L)**2)/(AD*RAL(K,L)**2))
     1                   *(COS(2*ACOS(GGG2(K,L)/RAL(K,L))))
        GO TO 1040
 1030   D(N3+MM+K,N3+M1)=0.0
        D(N3+MM+K,N3+M1+1)=0.0
```

```
      D(N3+MM+K,N3+M1+2)=-PI/AD
 1040 M1=M1+3
 1010 CONTINUE
 1001 CONTINUE
      NNN=N3+M
      NN33=N3+1
      DO 700 I=NN33,NNN
      DO 700 J=1,N3
      D(I,J)=D(I,J)*(- 0.159154)
 700  CONTINUE
C-----------------------------------------------------------------------
C     GENERATE THE R.H.S.
C-----------------------------------------------------------------------
      DO 45  K= 1,M
      NNN=N3+K
      F(NNN)=F(NNN)-PHIO(K)
 45   CONTINUE
      DO 28 L=1,M
      F(N3+M+L)=0.
 28   CONTINUE
      DO 29 L=1,M
      MM=2*M
      F(N3+MM+L)=0.
 29   CONTINUE
      DO 1100 I=1,N3
      DO 1100 J=1,N3
 1100 D(I,J)=SS(I,J)
      DO 95 I=1,N3
  95  F(I)=S(I)
C-----------------------------------------------------------------------
C     SOLVE THE ASSEMBLED MATRIX
C-----------------------------------------------------------------------
      NNNN=N3+3*M
C     IDGT IS AN INPUT OPTION.IF IDGT IS GREATER THAN ZERO,THE
C     ELEMENTS OF THE MATRIX AND THE R.H.S. VECTOR ARE ASSUMED
C     TO BE CORRECT TO THE IDGT DECIMAL DIGITS AND THE ROUTINE
C     PERFORMS AN ACCURACY TEST.
C     IF IDGT IS EQUAL TO ZERO, THE ACCURACY TEST IS BYPASSED
      MM1=1;N9=NNNN;IA=100;IDGT=3
      CALL LEQT1F(D,MM1,N9,IA,F,IDGT,WKAREA,IER)
C     EVALUATE OBJECTIVE FUNCTION AND CONSTRAINTS
      SIGMA=0.
      DO 127 J=17,32
      SIGMA1=(F(J)-1.)**2
      SIGMA=SIGMA1+SIGMA
 127  CONTINUE
      OBJ=SIGMA
      G(1)=AA(1)+0.05+X(1)-LLL
      G(2)=AA(2)+0.05+X(3)-LLL
```

```
      G(3)=AA(1)+0.05+X(2)-W
      G(4)=AA(2)+0.05+X(4)-W
      DO 94 KK=1,M
      DO 96 LL=1,M
      IF(KK.EQ.LL) GO TO 96
      GGG1(KK,LL)=YT(KK)-YT(LL)
      GGG2(KK,LL)=XT(KK)-XT(LL)
      RAL(KK,LL)=SQRT(GGG1(KK,LL)**2+GGG2(KK,LL)**2)
      G(5)=-RAL(KK,LL)+AA(KK)+AA(LL)+0.05
 96   CONTINUE
 94   CONTINUE
C     CALL THE OPTIMIZING ROUTINE CONMIN.....
      CALL CONMIN(X,VLB,VUB,G,SCAL,DF,AAAA,SS1,G1,G2,BBB,C,ISC,IC,MS1,
     * K1,K2,K3,K4,K5)
      IF(IGOTO.GT.0) GO TO 1111
C-----------------------------------------------------------------------
C     PRINT ALL THE INPUTS USED
C-----------------------------------------------------------------------
      WRITE(23,90)
 90   FORMAT(1H1//1X,75(1H-),//15X,'INPUT QUANTITIES'//1X,75(1H-),//)
      WRITE(23,101) M
 101  FORMAT(1H ,'THE NUMBER OF TUBES IN THE HEAT CONDUCTOR = ',I3)
      WRITE(23,102) LLL,W
 102  FORMAT(1H ,'NON-DIM LENGTH= ',F8.3,1X,'NON-DIM WIDTH=',F8.3)
      DO 103 K=1,M
      WRITE(23,104) K,XT(K),K,YT(K),K,PHIO(K),K,AA(K)
 104  FORMAT(1H ,'XT(',I2,') = 'F8.3,3X,'YT(',I2,') = 'F8.3,3X,
     1 'PHIO(',I2,') = 'F8.3,3X,'AA(',I2,') = 'F8.3)
 103  CONTINUE
      WRITE(23,204) BI1
 204  FORMAT(1H ,'BIOT NO. ON BOUNDARY ONE = ',F8.3)
      WRITE(23,205) BI2
 205  FORMAT(1H ,'BIOT NO. ON BOUNDARY TWO = ',F8.3)
      WRITE(23,206) BI4
 206  FORMAT(1H ,'BIOT NO. ON BOUNDARY FOUR = ',F8.3)
      DO 207 J= NN1,N2
      WRITE(23,208) J,BI3(J)
 208  FORMAT(1H ,'BI(',I3,') = ',F8.3)
 207  CONTINUE
      WRITE(23,211) HEAT
 211  FORMAT(1H ,'THE NON-DIM HEAT FLUX IN  TO THE
     1 HEAT CONDUCTOR = ',F8.3)
      WRITE(23,209)
 209  FORMAT(1H1//1X,75(1H-),/25X,'OUTPUT QUANTITIES',//75(1H-),//)
      WRITE(5,*)IER
C     PRINT ERROR MESSAGES FROM THE MATRIX SOLVER
      IF(IER.EQ.129) WRITE(23,301)
 301  FORMAT(1H ,20X,'ERROR MESSAGES',//75(1H-),//
     1 'MATRIX IS ALGORITHMICALLY SINGULAR')
      IF(IER.EQ.34) WRITE(23,302)
```

```
 302     FORMAT(1H ,20X,'ERROR MESSAGES',//75(1H-),//
     1   2X,'ACCURACY TEST FAILED.')
C        PRINT THE TEMPERATURE DISTRIBUTION
         WRITE(23,41)
 41      FORMAT(/,1X,'THE NON-DIM TEMPERATURES ON THE BOUNDARY',/)
         DO 900 J=1,N3
         WRITE(23,42) J,F(J)
 42      FORMAT(1H ,'PHI(',I3,') = 'F8.3)
 900     CONTINUE
C        PRINT THE HEAT TRANSFER SOLUTION
         WRITE(23,43)
 43     FORMAT(/,1X,'THE QO S ARE',/)
         WRITE(23,51)
 51      FORMAT(/2X'TUBE',7X,'QO',10X,'Q1',7X,'Q2')
         M1=N3+1
         M2=M1+1
         M3=M1+2
         DO 52 L=1,M
         WRITE(23,54)L,F(M1),F(M2),F(M3)
 54      FORMAT(/1X,I3,5X,3F10.6)
         M1=M1+3
         M2=M1+1
         M3=M1+2
 52      CONTINUE
         STOP
         END
C########################################################################
```

VITA

Satish Prabhakar Ketkar was born in Poona, India, on November 25, 1960. He attended elementary and secondary school at N.M.V., Poona, India, through tenth grade, and got his Secondary School Certificate in July 1975. He got his Higher Secondary School Certificate in July 1977.

In August 1977, he joined the College of Engineering, Poona, India. In August 1981 he received a Bachelor of Engineering degree with a major in Mechanical Engineering.

In Fall of 1982 he enrolled in the Master's program in Mechanical Engineering at The University of Tennessee, Knoxville. This degree was awarded in March 1985.

The author is a student member of The American Institute of Mechanical Engineers and The American Institute of Aeronautics and Astronautics.