AD NUMBER

ADB017022

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; NOV 1976. Other requests shall be referred to Rome Air Development Center, Griffiss AFB, NY.

AUTHORITY

RADC ltr 18 Apr 1977

RADC-TR-76-269(II)
IN-HOUSE REPORT
NOVEMBER 1976

FG

(2)

Computer Solutions to Heat and Diffraction Equations in High Energy Laser Windows

Volume II

PETER D. GIANINO
BERNARD BENDOW
N. GRIER PARKE, III
THEODORE A. BARRETT

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
GRIFFISS AIR FORCE BASE, NEW YORK 13441

APPROVAL SHEET

This technical report has been reviewed and approved by RADC (OI) for publication.

APPROVED:

NICHOLAS F. YANNONI, Chief
Optical Techniques Branch
Solid State Sciences Division

APPROVED:

ROBERT M. BARRETT
Director
Solid State Sciences Division

FOR THE COMMANDER:

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: RADC-TR-76-269 (Vol-2)	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): COMPUTER SOLUTIONS TO HEAT AND DIFFRACTION EQUATIONS IN HIGH ENERGY LASER WINDOWS, VOLUME II	5. TYPE OF REPORT & PERIOD COVERED: Scientific. Interim.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Peter D. Gianino, N. Grier Parke, III, Bernard Bendow, Theodore B. Barrett*	8. CONTRACT OR GRANT NUMBER(s): 182 P.
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Deputy for Electronic Technology (RADC/ETSS) Hanscom AFB Massachusetts 01731	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62601F 33260802
11. CONTROLLING OFFICE NAME AND ADDRESS: Deputy for Electronic Technology (RADC/ETSS) Hanscom AFB Massachusetts 01731	12. REPORT DATE: November 1976 / 13. NUMBER OF PAGES: 193
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Interim rept.,	15. SECURITY CLASS. (of this report): Unclassified / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U. S. Government Agencies only; (Test & Evaluation); (November 1976). Other requests for this document must be referred to RADC/ETE, Hanscom AFB, Massachusetts 01731

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 16 3326 17 08

18. SUPPLEMENTARY NOTES

*Parke Mathematical Laboratories, Inc.
Carlisle, Mass. 01741

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Laser windows
Laser window materials
Thermal lensing
Numerical solution of vector Kirchhoff diffraction equation
Numerical solution of heat equation

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

The LQ-10 Infrared Laser Window Program was initiated at AFCRL (now RADC/DET) in 1971 to develop a mechanically superior, highly transmitting optical window for CO_2 and chemical lasers to be used by the Air Force for military applications. From the earliest stages of that program it became evident that the principal failure mechanism of the window would be thermal lensing, that is, the distorting and defocusing of the laser beam as it propagated through the solid. For this reason analytical tools were

next page →

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

309 050 dn

20. Abstract (Continued)

developed to predict the extent of this lensing in various candidate materials under a variety of conditions. This work contributed to selection of appropriate materials, as well as to design of geometrical configurations, in which the lensing could be reduced. To quantify the effects of thermal lensing, an efficient computer program package was developed and programmed to run on a CDC6600 computer. The package was written to handle Gaussian-shaped beams incident on either a thin disc- or annular-shaped cylindrical window. Three coupled programs make up the package: TEMP5, which solves the full heat transport equation within the window for any given set of initial and boundary conditions on each surface; TIKIRK, which solves the vector Kirchhoff diffraction integrals for the beam transmitted to the far field; and DISPLAY, which plots these temperatures and/or intensities in a variety of ways, including three-dimensional perspective views. Volume I of this report lays the theoretical foundations underlying these programs and presents graphical results for two model problems using disc- and annular-shaped windows. Volume II is a "user's manual." It describes how each program functions, enumerates the constituent subroutines and subprograms, gives complete Fortran listings, and even provides typical detailed commands to initiate and run the programs in both the Intercom and Batch modes of operation. Results of this work should substantially aid engineers in planning configurations and specifications for current and conceptual systems.

Contents

VOLUME I

Contents

VOLUME II

Contents

Illustrations

VOLUME I

Illustrations

VOLUME II

Tables

VOLUME I

VOLUME II

Computer Solutions to Heat and Diffraction Equations in High Energy Laser Windows Volume II

7. INTRODUCTION

In Volume II we will give a detailed documentation of the Fortran programs TEMP5, TIKIRK and DISPLAY, explaining how to implement them. This will include: listings of main and ancillary programs and subroutines, plus an explanation of their functions; derivations of how the heat, boundary condition (BC) and diffraction intensity equations are transformed into algorithms solvable by the computer; flow charts; and, glossaries of variables for some of the more important subroutines. Since programs TEMP5 and TIKIRK have been coded to permit systems operation under both an Intercom and Batch mode, we will list typical interactive commands and card deck setups which control these two types of operation. Because program DISPLAY can function only in the Batch mode, we will list typical card deck setups for its operation.

Much of the details presented in Volume II appeared originally in the following unpublished reports from Parke Mathematical Laboratories, Inc., Carlisle, Mass.:

(i) N.G. Parke, III, "Program TEMP5," Sci. Rpt. No. 1 (April 1973); also documented as AFCRL Rpt TR-73-0039 by the same author.

(Received for publication 26 November 1976)

(ii) T.B. Barrett, "An Interactive Set of Programs Using Program TEMP5 for the Determination of Calorimetric Material Parameters from Experimental Data on Cylindrical IR Laser Window Materials," Tech. Memo. No. 16 (Oct. 1973).

(iii) T.B. Barrett, "TIKIRK Program," PML Rpt. 110, with revision (April 1974).

(iv) T.B. Barrett, "GETDATA Subroutine," PML Rpt. 111 (May 1974).

(v) T.B. Barrett, "DISPLAY Program," PML Rpt. 116 (May 1974).

8. TEMP5 PROGRAM

8.1 Introductory Remarks

Initial attempts to code the numerical solution to the heat and BC equations used the Crank-Nicolson method.[11] This procedure leads to a pentagonal system of linear difference equations, which are usually solved by an appropriate iteration technique.[12] However, if the edges of the "net" of points – at which the temperature is to be evaluated – is situated at the boundaries, three problems arise:

(1) Iteration techniques must be used.

(2) Symmetry dictates that along the window axis ($\rho = 0$) there be no heat flow across the window center, that is, $\partial u/\partial \rho = 0$. Under this condition, however, the term $(1/\rho)(\partial u/\partial \rho)$ which occurs in the partial differential equation would be indeterminate.

(3) A satisfactory finite-difference analog must be found for the general BC's, which have the form:

$$\partial u/\partial \nu + hu = g \quad . \tag{38}$$

[cf, Eq. (32) of Volume I].

These difficulties were resolved as follows:

(1) The Crank-Nicolson method was replaced by the Implicit Alternating Difference (IAD) method.[11] This procedure reduces the algebraic problem at each stage to the inversion of a tridiagonal matrix. The Thomas algorithm is employed and iteration is avoided. The cost of this approach for a problem involving two space variables is a two-time-level pair of difference equations.

---

11. Carnahan, B., Luther, H.A., and Wilkes, J.O. (1969) Applied Numerical Methods, Wiley and Sons, Inc., New York.

12. Parke, N.G., III (1971) Technical Memorandum No. 4, Parke Mathematical Laboratories, Inc., Carlisle, Massachusetts, unpublished.

(2) By applying L'Hospital's rule along the cylinder axis there results: $\lim_{\rho \to 0} (\rho^{-1})(\partial u/\partial \rho) = \partial^2 u/\partial \rho^2$.

(3) A suitable finite difference analog for Eq. (38) is established by shifting the "net" half an increment off the boundaries.

To see how the finite difference method is applied, consider a transverse cross-sectional cut through the window's center (that is, the plane of the cross-section is perpendicular to the window's faces). The borders of the resulting rectangular cross-section are parallel to the ρ and ζ axes (see Figure 1, Volume I). Because of the rotational symmetry, only one half of the section need be shown. The geometry of the choice of net points superimposed on this cut is indicated in Figure 19. The window faces occur at the lines marked $\zeta = \zeta_1$, and $\zeta = \zeta_2$; the inner and outer cylindrical surfaces are denoted by the lines marked $\rho = \rho_1$ and 1, respectively. The ρ, ζ coordinates of each net point are represented by the indices i, j, respectively, with i running from 0 to M+1, and, j running from 0 to N+1. That is,

$$\begin{aligned} \rho_i &= \rho_1 + \left(i - \frac{1}{2}\right) \cdot \Delta\rho \quad , \quad i = 0, 1, \ldots, M+1 \\ \zeta_j &= \zeta_1 + \left(j - \frac{1}{2}\right) \cdot \Delta\zeta \quad , \quad j = 0, 1, \ldots, N+1 \quad . \end{aligned} \tag{39}$$

These coordinates are measured relative to the surfaces ρ_1 and ζ_1, respectively. All of the net points bearing one or both of the indices 0, M+1 and N+1 fall outside of the window itself and are considered to be "fictitious" or "corner" points.

8.2 Finite Difference Analogs for the General BC's

It is now possible to write the finite difference analog of the general BC's for the shifted net. First, we note that the derivative term $\partial u/\partial \nu$ in Eq. (38) differs for each surface due to the sign conventions chosen for ρ and ζ. For example, at the surfaces $\rho = \rho_1$ and $\rho = 1$, the term $\partial u/\partial \nu$ becomes - and + $\partial u/\partial \rho$, respectively; while at the surfaces $\zeta = \zeta_1$, and $\zeta = \zeta_2$ it becomes - and + $\partial u/\partial \zeta$, respectively. Thus, the finite difference analogs of the BC's become:

$$\frac{u_{o,j} - u_{1,j}}{\Delta\rho} + h_1 \frac{u_{o,j} + u_{1,j}}{2} = g_1 \text{ at } \rho = \rho_1 \tag{40}$$

$$\frac{u_{M+1,j} - u_{M,j}}{\Delta\rho} + h_2 \frac{u_{M+1,j} + u_{M,j}}{2} = g_2 \text{ at } \rho = 1 \tag{41}$$

for $j = 1, 2, \ldots, N-1$

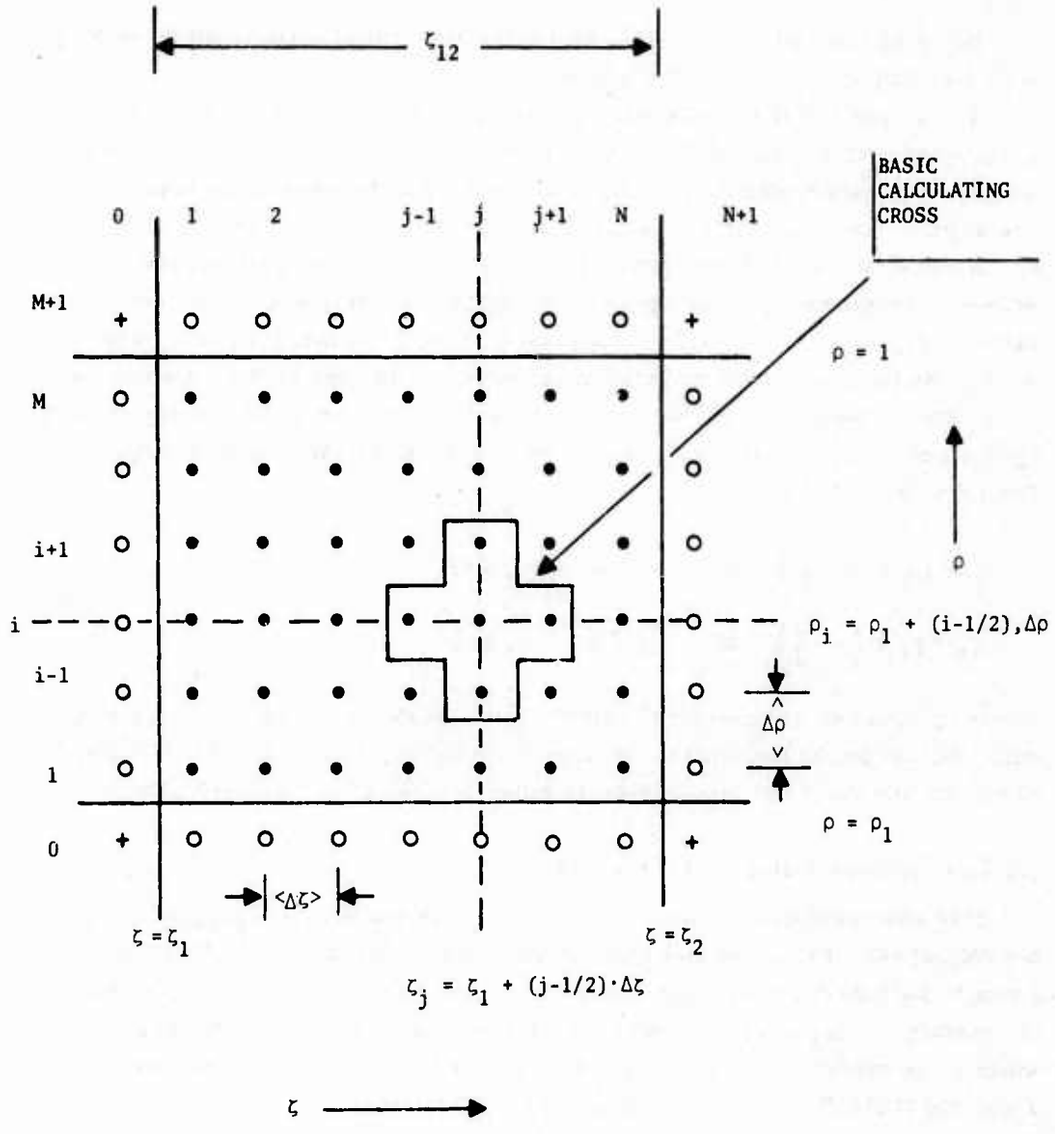

Figure 19. Geometry of Finite Difference Net. The boundaries are straddled by a net point and a fictitious point

$$\frac{u_{i,o} - u_{i,1}}{\Delta\zeta} + h_3 \frac{u_{i,o} + u_{i,1}}{2} = g_3 \text{ at } \zeta = \zeta_1 \tag{42}$$

$$\frac{u_{i,N+1} - u_{i,N}}{\Delta\zeta} + h_4 \frac{u_{i,N+1} + u_{i,N}}{2} = g_4 \text{ at } \zeta = \zeta_2 \tag{43}$$

When these are solved for the "fictitious" points, one obtains

$$u_{o,j} = \left[\frac{2 - h_1 \cdot \Delta\rho}{2 + h_1 \cdot \Delta\rho}\right] u_{1,j} + \left[\frac{2 \cdot \Delta\rho \cdot g_1}{2 + h_1 \cdot \Delta\rho}\right] \tag{44}$$

$$u_{M+1,j} = \left[\frac{2 - h_2 \cdot \Delta\rho}{2 + h_2 \cdot \Delta\rho}\right] u_{M,j} + \left[\frac{2 \cdot \Delta\rho \cdot g_2}{2 + h_2 \cdot \Delta\rho}\right] \tag{45}$$

$$u_{i,o} = \left[\frac{2 - h_3 \cdot \Delta\zeta}{2 + h_3 \cdot \Delta\zeta}\right] u_{i,1} + \left[\frac{2 \cdot \Delta\zeta \cdot g_3}{2 + h_3 \cdot \Delta\zeta}\right] \tag{46}$$

$$u_{i,N+1} = \left[\frac{2 - h_4 \cdot \Delta\zeta}{2 + h_4 \cdot \Delta\zeta}\right] u_{i,N} + \left[\frac{2 \cdot \Delta\zeta \cdot g_4}{2 + h_4 \cdot \Delta\zeta}\right] . \tag{47}$$

We saw in Section 3.3, Volume I, that all BC's of practical interest can be represented by appropriate choices of the g_i and h_i. With this capability in the above analogs, the resulting computer program becomes very flexible.

8.3 Finite Difference Equations for I.A.D. Method

Having set up the "net," we shall now use the I.A.D. method on the parabolic heat equation having the general form [cf Eq. (29), Volume I]:

$$\partial u/\partial\tau = \partial^2 u/\partial\rho^2 + \rho^{-1}\partial u/\partial\rho + \partial^2 u/\partial\zeta^2 + q \tag{48}$$

where

$$q = A \exp(-\rho^2/2\sigma_e^2).$$

In the first finite difference equation set, the analog of the partial derivatives with respect to ρ will be written at the new time level n + 1, and the analog of the ζ-derivative written at the old level n. Here, n is even starting with n = 0. To complete the cycle, the second finite difference equation set is written at the new time level n + 2 for derivatives in the ζ direction. In other words, the equations are now implicit in ζ-direction and explicit in the ρ-direction. Partial derivatives with respect to ρ are written in terms of values of u at the, now old, time level n + 1. These "intermediate" values of u are sometimes designated u^* (meaning a correction). They are not accurate representations of the u. This point is discussed in detail by von Rosenberg.[13]

Our analogs for the various partial derivatives take the forms:

$$(u_{\rho\rho})_{i,j,n+1} = \frac{u_{i+1,j,n+1} - 2u_{i,j,n+1} + u_{i-1,j,n+1}}{(\Delta\rho)^2} , \tag{49}$$

$$\left(\frac{1}{\rho} u_{\rho}\right)_{i,j,n+1} = \frac{u_{i+1,j,n+1} - u_{i-1,j,n+1}}{2\rho_i(\Delta\rho)} , \tag{50}$$

$$(u_{\zeta\zeta})_{i,j,n} = \frac{u_{i,j+1,n} - 2u_{i,j,n} + u_{i,j-1,n}}{(\Delta\zeta)^2} , \tag{51}$$

$$(u_{\tau})_{i,j,n+1/2} = \frac{u_{i,j,n+1} - u_{i,j,n}}{(\Delta\tau)} , \tag{52}$$

and

$$q(\rho_i, \zeta_j) = q_{i,j} . \tag{53}$$

The subscripts i, j merely represent generalized indices and extend over the generalized ranges: i = 1, 2, . . . , M and j = 1, 2, . . . , N. After substitution, the first set of I. A. D. equations are:

13. von Rosenberg, D. U. (1969) Methods for the Numerical Solution of Partial Differential Equations, American-Elsevier Publishing Co., Inc., New York.

$$\frac{u_{i+1,j,n+1} - 2u_{i,j,n+1} + u_{i-1,j,n+1}}{(\Delta\rho)^2} + \frac{u_{i+1,j,n+1} - u_{i-1,j,n+1}}{2\rho_i(\Delta\rho)}$$

$$+ \frac{u_{i,j+1,n} - 2u_{i,j,n} + u_{i,j-1,n}}{(\Delta\zeta)^2} + q_{i,j} = \frac{u_{i,j,n+1} - u_{i,j,n}}{\Delta\tau} \quad . \tag{54}$$

The second set of I.A.D. equations is:

$$\frac{u_{i+1,j,n+1} - 2u_{i,j,n+1} + u_{i-1,j,n+1}}{(\Delta\rho)^2} + \frac{u_{i+1,j,n+1} - u_{i-1,j,n+1}}{2\rho_i(\Delta\rho)}$$

$$+ \frac{u_{i,j+1,n+2} - 2u_{i,j,n+2} + u_{i,j-1,n+2}}{(\Delta\zeta)^2} + q_{i,j} = \frac{u_{i,j,n+2} - u_{i,j,n+1}}{\Delta\tau} \, . \tag{55}$$

It is convenient to introduce the parameters

$$\lambda = \frac{\Delta\tau}{(\Delta\rho)^2} \quad , \quad \mu = \frac{\Delta\tau}{(\Delta\zeta)^2} \quad . \tag{56}$$

Observe that Eqs. (54) are tridiagonal, containing the unknowns

$$u_{i+1,j,n+1} \quad , \quad u_{i,j,n+1} \quad , \quad u_{i-1,j,n+1}$$

and can be solved by the Thomas algorithm. Likewise, Eqs. (55) are tridiagonal, containing the unknowns

$$u_{i,j+1,n+2} \quad , \quad u_{i,j,n+2} \quad , \quad u_{i,j-1,n+2}$$

and can likewise be solved by the Thomas algorithm.

Before continuing, let us take up the mathematics of the Thomas algorithm (which will be incorporated in the subroutine TRIDAG, to be explained later).

8.4 The Solution of a Tridiagonal System of Equations

The whole purpose of the implicit-alternating direction method is to reduce the number of unknown variables at the "next" time level to three in any one equation. Such a set of equations is called a tridiagonal system that has a relatively

simple solution. This strategy avoids "iteration" techniques of the Crank-Nicolson approach, described in Parke.[12]

The general form of a tridiagonal system of equations is

$$
\begin{aligned}
& b_1 v_1 + c_1 v_2 = d_1 \\
& a_2 v_1 + b_2 v_2 + c_2 v_3 = d_1 \\
& a_3 v_2 + b_3 v_3 + c_3 v_4 = d_2 \\
& \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
& a_i v_{i-1} + b_i v_i + c_i v_{i+1} = d_i \\
& \cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
& a_{N-1} v_{N-2} + b_{N-1} v_{N-1} + c_{N-1} v_N = d_{N-1} \\
& a_N v_{N-1} + b_N v_N = d_N
\end{aligned}
\tag{57}
$$

where*

d_i .=. known quantities

a_i, b_i, c_i .=. known coefficients

v_i .=. unknown quantities .

The tridiagonal matrix is defined as the matrix of coefficients a, b, c alone.

We follow the treatment in Carnahan.[11] To continue, the validity of the form

$$
v_i = \gamma_i - \frac{c_i}{\beta_i} v_{i+1} \tag{58}
$$

can be demonstrated. The constants γ_i and β_i are to be determined. Indeed, substitution into the i-th equation of (57) gives

$$
a_i \left(\gamma_{i-1} - \frac{c_{i-1}}{\beta_{i-1}} v_i \right) + b_i v_i + c_i v_{i+1} = d_i \quad .
$$

---

*The symbol .=. means "is defined as."

As a result

$$v_i = \frac{d_i - a_i\gamma_{i-1}}{b_i - \frac{a_i c_{i-1}}{\beta_{i-1}}} - \frac{c_i v_{i+1}}{b_i - \frac{a_i c_{i-1}}{\beta_{i-1}}}$$

where we have the recursion relations

$$\beta_i = b_i - \frac{a_i c_{i-1}}{\beta_{i-1}} \quad ; \quad \gamma_i = \frac{d_i - a_i\gamma_{i-1}}{\beta_i} . \tag{59}$$

From the first of Eqs. (57), we have

$$v_1 = \frac{d_1}{b_1} - \frac{c_1 v_2}{b_1}$$

where

$$\beta_1 = b_1 \quad , \quad \gamma_1 = d_1/\beta_1 \quad . \tag{60}$$

Finally, from the last of Eqs. (57), we have

$$v_N = \frac{d_N - a_N v_{N-1}}{b_N} = \frac{d_N - a_N\left(\gamma_{N-1} - \frac{c_{N-1}}{\beta_{N-1}} v_N\right)}{b_N} \tag{61}$$

where

$$v_N = \frac{d_N - a_N(\gamma_{N-1})}{b_N - \frac{a_N c_{N-1}}{\beta_{N-1}}} = \frac{d_N - a_N\gamma_{N-1}}{\beta_N} = \gamma_N \quad . \tag{61a}$$

To summarize the complete algorithm for the solution of the tridiagonal system, we have

$$v_N = \gamma_N$$

$$v_i = \gamma_i - \frac{c_i\, v_{i+1}}{\beta_i} \quad ; \quad i = N-1, N-2, \ldots, 1 \tag{62}$$

where the β's and γ's are determined by the recursion formulae

$$\beta_1 = b_1 \quad , \quad \gamma_1 = d_1/\beta_1$$

$$\beta_i = b_i - \frac{a_i\, c_{i-1}}{\beta_{i-1}} \qquad i = 2, 3, \ldots, N \tag{63}$$

$$\gamma_i = \frac{d_i - a_i\gamma_{i-1}}{\beta_i} \qquad i = 2, 3, \ldots, N \quad .$$

8.5 Putting the I.A.D. Equations into Tridiagonal Form

Using Eqs. (56), the first I.A.D. set becomes, from Eq. (54)

$$\lambda[u_{i+1,j,n+1} - 2u_{i,j,n+1} + u_{i-1,j,n+1}] + \frac{\lambda \cdot \Delta\rho}{2(\rho_1 + (i - 1/2) \cdot \Delta\rho)}^{*} \times$$

$$\times [u_{i+1,j,n+1} - u_{i-1,j,n+1}] = d_i + u_{i,j,n+1} \quad . \tag{64}$$

Writing in the standard form

$$\begin{aligned}
&b_1 u_{1,j,n+1} + c_1 u_{2,j,n+1} = d_1 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_i u_{i-1,j,n+1} + b_i u_{i,j,n+1} + c_i u_{i+1,j,n+1} = d_i \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_M u_{M-1,j,n+1} + b_M u_{M,j,n+1} = d_M
\end{aligned} \tag{65}$$

we find that, in general, that is, for $i \neq 1$, $i \neq M$

$$a_i = \lambda \left(1 - \frac{\Delta\rho}{2\rho_1 + (2i - 1) \cdot \Delta\rho}\right) = \lambda \left(1 - \frac{\Delta\rho}{2\rho_i}\right) \tag{66}$$

* $\frac{\lambda \cdot \Delta\rho}{2\rho_i}$

$$b_i = -(2\lambda + 1) \tag{67}$$

$$c_i = \lambda \left(1 + \frac{\Delta\rho}{2\rho_1 + (2i - 1) \cdot \Delta\rho}\right) = \lambda \left(1 + \frac{\Delta\rho}{2\rho_i}\right) \tag{68}$$

$$d_i = -\mu(u_{i,j+1,n} - 2u_{i,j,n} + u_{i,j-1,n}) - \Delta\tau \cdot q_{ij} - u_{i,j,n} \quad . \tag{69}$$

We notice that from Eq. (44)

$$a_1 u_{o,j,n+1} = a_1 \left\{ \left[\frac{2 - h_1 \cdot \Delta\rho}{2 + h_1 \cdot \Delta\rho}\right] u_{1,j} + \left[\frac{2 \cdot \Delta\rho \cdot g_1}{2 + h_1 \cdot \Delta\rho}\right] \right\} \quad . \tag{70}$$

Hence, we have to change b_1 according to

$$b_1 \leftarrow b_1 + a_1 \left[\frac{2 - h_1 \cdot \Delta\rho}{2 + h_1 \cdot \Delta\rho}\right] \quad . \tag{71}$$

We also have to change d_1

$$d_1 \leftarrow d_1 - a_1 \left[\frac{2 \cdot \Delta\rho \cdot g_1}{2 + h_1 \cdot \Delta\rho}\right] \quad . \tag{72}$$

Similarly we have to change b_M

$$b_M \leftarrow b_M + c_M \left[\frac{2 - h_2 \cdot \Delta\rho}{2 + h_2 \cdot \Delta\rho}\right] \tag{73}$$

using Eq. (45). Likewise

$$d_M \leftarrow d_M - c_M \left[\frac{2 \cdot \Delta\rho \cdot g_2}{2 + h_2 \cdot \Delta\rho}\right] \quad . \tag{74}$$

It should be observed that as long as $j \neq 1$ or $j = N$, the d's require no further modification because their computation involves only "net" points at time level n. However, when $j = 1$, the fictitious points $u_{i,o,n}$ are involved. Also, when $j = N$,

the fictitious points $u_{i,N+1,n}$ are involved. The simplest way to handle this is to "border" the $u_{i,j,n}$ array by computing $u_{i,o,n}$ with Eq. (46) and $u_{i,N+1,n}$ with Eq. (47) just before computing the first sets of a, b, c, d for the ζ-explicit, ρ-implicit I.A.D. set.

To summarize, in step I we modify b_1, d_1, b_M, d_M having extended $u_{i,j,n}$ with Eqs. (46) and (47).

Now let us turn to step II of the I.A.D. method. We begin with Eqs. (55) and write

$$\mu[u_{i,j+1,n+2} - 2u_{i,j,n+2} + u_{i,j-1,n+2}] = d_j + u_{i,j,n+2} - \Delta\tau \cdot q_{ij} \quad . \tag{75}$$

These equations are to be written in the standard form:

$$\begin{aligned}
&b_1 u_{i,1,n+2} + c_1 u_{i,2,n+2} = d_1 \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_j u_{i,j-1,n+2} + b_j u_{i,j,n+2} + c_j u_{i,j+1,n+2} = d_j \\
&\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
&a_N\, u_{i,N-1,n+2} + b_{N-1}\, u_{i,N,n+2} = d_N \quad .
\end{aligned} \tag{76}$$

We find again that, in general, that is, for $j \neq 1$ or $j \neq N$

$$\begin{aligned}
&a_j = \mu \quad , \quad b_j = -(2\mu + 1) \quad , \quad c_j = \mu \quad , \\
&d_j = -\lambda[u_{i+1,j,n+1} - 2u_{i,j,n+1} + u_{i-1,j,n+1}] - \frac{\lambda \cdot \Delta\rho^*}{2\rho_1 + (2i-1)\cdot\Delta\rho} \\
&\qquad [u_{i+1,j,n+1} - u_{i-1,j,n+1}] - \Delta\tau \cdot q_{ij} - u_{i,j,n+1} \quad .
\end{aligned} \tag{77}$$

Using Eq. (46) for $u_{i,o,n+2}$, we change b_1 to

$$b_1 \leftarrow b_1 + a_1 \left[\frac{2 - h_3 \cdot \Delta\zeta}{2 + h_3 \cdot \Delta\zeta}\right] \quad . \tag{78}$$

---

*$-\lambda\Delta\rho / 2\rho_i$

We also change d_1 to

$$d_1 \leftarrow d_1 - a_1 \left[\frac{2 \cdot \Delta\zeta \cdot g_3}{2 + h_3 \cdot \Delta\zeta} \right] . \tag{79}$$

Similarly we use Eq. (47) for $u_{i, N, n+2}$ to change b_{N-1} to

$$b_N \leftarrow b_N + c_N \left[\frac{2 - h_4 \cdot \Delta\zeta}{2 + h_4 \cdot \Delta\zeta} \right] . \tag{80}$$

and

$$d_N \leftarrow d_N - c_N \left[\frac{2 \cdot \Delta\zeta \cdot g_4}{2 + h_4 \cdot \Delta\zeta} \right] . \tag{81}$$

Again, as long as $i \neq 1$, or $i \neq M$, the d's require no further modification because their computation involves only "net" points at time level n + 1. However, when $i = 1$, the fictitious points $u_{o, j, n+1}$ are involved. Also, when $i = M$, the fictitious points $u_{M+1, j, n+1}$ are involved. The simplest way to handle this is to "border" the $u^* = u_{i, j, n+1}$ array by computing $u_{o, j, n+1}$ with Eq. (44) and $u_{M+1, j, n+1}$ with Eq. (45) before starting on the second half of the I. A. D. set. We shall keep the $0, 2, \ldots, n$ even level u's in an array U(I, J). We shall keep the $1, 3, \ldots n$ odd level u's in an array USTAR(I, J).

8.6 The Time Coordinate

The time coordinate τ_n is constructed so as to be controlled by an integer n and an increment $\Delta\tau$ in a special way. Because we are using the I. A. D. method, the "net" values of temperature are only valid when u is an even integer. In addition, it should be noted that $\Delta\tau$ may be changed before entering a new cycle involving an alternating difference pair of finite difference equations.

Initially, temperature varies relatively rapidly with time. This means that rather closely spaced time units should be selected at which to calculate the temperature. Later, as the temperature approaches its steady state value, its change is less rapid so that it seems reasonable, especially from the viewpoint of conserving computer time, to calculate temperatures at much larger intervals of time. This can be accomplished by allowing the time interval $\Delta\tau$ at a particular choice of n to increase according to the scheme:

$$\Delta\tau(n) = 2^{n/n_o} \Delta\tau_o . \tag{82}$$

$\Delta\tau_o$ is some arbitrarily-selected initial value of $\Delta\tau$, n is a positive even integer and n_o is some arbitrary positive integer, called the "doubling count number," because when n reaches n_o, $\Delta\tau$ will double to $2\Delta\tau_o$. Making n_o very large is equivalent to holding $\Delta\tau(n) = \Delta\tau_o$.

Meanwhile, the time coordinate τ_n is formed according to the prescription:

$$\tau_n = 2\sum_{k=o}^{n} \Delta\tau(k) = 2\Delta\tau_o \sum_{k=o}^{n} 2^{k/n_o} \tag{83}$$

in which k is incremented in steps of 2 up to n. This prescription will hold up to the limit $n = n_L$. The actual value of n_L will be determined by the parameter n_{max}, a positive integer which is inputted at the start of the program. The integer n_L will be equal to $n_{max} - 2$ if n_{max} is even, or, to $n_{max} - 1$ if n_{max} is odd. At $n = n_L$, the increment is designated by $\Delta\tau(n_L)$ and the time by τ_{n_L}. That is:

$$\Delta\tau(n_L) = 2^{n_L/n_o}\,\Delta\tau_o \tag{84}$$

$$\tau_{n_L} = 2\,\Delta\tau_o \sum_{k=o}^{n_L} 2^{k/n_o} \quad . \tag{85}$$

For times greater than τ_{n_L}, corresponding to $n > n_L$, the increment will remain fixed at $\Delta\tau(n_L)$, whereas the time will be given by:

$$\tau_n = \tau_{n_L} + \Delta\tau(n_L) \cdot (n - n_L)/2 \quad . \tag{86}$$

The time τ_n will keep increasing by these fixed increments until it reaches some arbitrarily-fixed upper limit τ_{max}, at which point the program initiates a termination procedure.

By virtue of another time-control parameter, the program also makes provision for turning the source off and then determining the temperature changes as the window cools off. This occurs at $\tau = \tau_{off}$, where, of course, τ_{off} must be $\leq \tau_{max}$.

Should τ_n, as determined by Eq. (86), become greater than τ_{off} at the start of a time loop, and if $\tau_{off} < \tau_{max}$, then the time-incrementing procedure is reinitiated. On the other hand, should τ_n exceed τ_{off} at the start of a time loop, and if $\tau_{off} = \tau_{max}$, then the subroutine CYLTMP (to be described later) does not

continue with the calculation, but returns control to the main program. Thus, the actual maximum time value will usually be slightly less than τ_{off} if the window is being irradiated, and will be slightly less than τ_{max} if the window is experiencing a cooling phase.

From the analysis above, we see that, apart from the running index n, usually 4, or, at most 5, parameters are required to control the time coordinate, viz, n_o, n_{max}, $\Delta\tau_o$, τ_{max} and τ_{off}. The time interval $\Delta\tau$ can be enlarged by increasing $\Delta\tau_o$ or n_{max}, or by decreasing n_o.

8.7 The Main Program and Principal Subroutines

The coding necessary to input all of the data, to carry out all of the required calculations, including the I.A.D. prescriptions, and finally, to print out the results constitutes a major programmed package, named the TEMP5 program. This package consists of eight principal subroutines called into execution by one very short main program. This latter program has also been designated as TEMP5. However, whenever we use the term "TEMP5 program" throughout this report we will always mean the collective "package" rather than this one main program, unless stated otherwise.

The TEMP5 program is composed of the following:

(1) TEMP5 - a very short program whose principal purpose is to call the 2 principal subroutines DATINIT and CYLTMP.

(2) DATINIT - a subroutine which inputs all required parameters necessary for program operation by a call to subroutine GETDATA. It initializes the principal arrays used by subroutine CYLTMP. It also calls subroutine GAUSS.

(3) CYLTMP - the "core" subroutine of the TEMP5 program. It calculates the temperature u according to the I.A.D. method using both subroutines TRIDAG and SPLNI and then the related integrals F1 and F2 again using SPLNI. It stores u, F1 and F2 in unformatted form on a file named TAPE3. These temperatures (u) are calculated at the RHO, ZED lattice points and are designated by the variable name U(I, J).

(4) TRIDAG - the subroutine which implements the Thomas algorithms for solving a system of simultaneous linear equations having a tridiagonal coefficient matrix.

(5) GAUSS - a subroutine for loading the volume heating source term Q with a truncated Gaussian distribution into the program.

(6) SPLNI - The subroutine which finds the third order spline function for a function y(x) given at the points (X(I), Y(I)). It is used both for integrating the F1 and F2 functions as well as for interpolating values of u at the RFIN, ZFIN lattice points, which occur halfway between the RHO, ZED lattice points. These interpolated temperatures are designated by the variable name UFIN(I, J). They enable

us to calculate temperatures out to the window's edges. SPLNI, which follows Chapter 8 of Ralston and Wilf, [14] is a modification of the IBM standard Scientific Subroutine Package subroutine SPLIE.

(7) GETDATA - obtains data from the operator. It can be used as a universal inputting subroutine for either the Batch or Intercom modes of operation of any program requiring input data. (More will be said about these two modes later.) However, it was written mainly for Intercom operation. It also calls on subroutines SSWTCH and RJUST.

(8) SSWTCH - reads in the first three data values (and prints out appropriate messages) for GETDATA control. (It should be noted that SSWTCH is not the same as the CDC Fortran subroutine bearing the same name.)

(9) RJUST - right adjusts all numerical values.

The TEMP5 program has been coded to permit operation under either Batch processing or Intercom. The latter mode permits relatively easy interactive use, as implemented under CDC Scope 3.4 with the CDC6600.

The complete Fortran listings for each of the above are given in Appendix A.

8.8 Implementation of Some of the Subroutines

8.8.1 DATINIT

The implementation of the TEMP5 program begins with the inputting of all required program parameters and the initializing of the working arrays which will eventually be used by CYLTMP. DATINIT accomplishes all of this by a call to GETDATA. Furthermore, DATINIT assigns default values to the VALUE portion of DATAIN, which is an array of TEMP5 parameters, and also assigns names and format codes to the NAME and FORMAT portions of DATAIN. This will be described in more detail in Section 8.8.4 on GETDATA.

A complete tabulation of all the required input data is given in Table 2. The table lists both the data and the variable names, their corresponding default values, formats, the particular major programming package in which each quantity is ultimately used, and a succinct description. The default values listed in Table 2 for the material properties such as refractive index, absorption coefficient, etc., pertain to KCl.

It should be noted that although all of the variables itemized in Table 2 may be inputted at this stage of the program, not all of them will actually be used in TEMP5. Many of them will be called up later in the TIKIRK and DISPLAY programs.

---

14. Ralston, A., and Wilf, H. S. (1967) Mathematical Methods for Digital Computers, Vol. II, Wiley and Sons, Inc., New York.

Another factor to be noted in Table 2 is that the values of M and N have been chosen to be 80 and 20, respectively. Since the indices on the RHO and ZED coordinates extend from 0 to M+1, and, 0 to N+1, respectively, this means that the net depicted in Figure 3 consists of 82 points along the radial direction and 22 points along the axis. Meanwhile, since the indices on the RFIN, ZFIN coordinates extend from 1 to M+1, and, 1 to N+1, respectively, then the array of points at which interpolation occurs consists of 81 points along the radial direction and 21 points along the axis.

8.8.2 CYLTMP

The temperature-related terms U, F1 and F2 really constitute the principal output of the entire TEMP5 program. Computation of these quantities, as prescribed by Eqs. (29), (30), (16) and (17), are actually carried out by the subroutine CYLTMP, with the aid of TRIDAG and SPLNI. Figure 20 shows a flow chart for the CYLTMP algorithm. Table 3 gives a glossary of the variable names.

Source turn-off is accomplished in subroutine CYLTMP by setting the volume source term (array q) and "boundary" source term (array g) to 0 at the appropriate time. At the end of each τ-cycle through CYLTMP, the temperature distribution at $\tau + \Delta\tau$ has been computed where τ is the time at the start of the cycle. Thus, a check is made at the start of each cycle to see if $\tau + \Delta\tau$ is less than τ_{off}. If it is, then the cycle continues normally with the source terms "on." When $\tau + \Delta\tau$ first becomes equal to or greater than τ_{off}, a "flag" is set and a new $\Delta\tau$ is computed such that $\tau + \Delta\tau = \tau_{off}$ and the cycle continues. When the subroutine returns to the start of the next cycle, the source term is set to 0. In addition the variable NN is reset to 0 and $\Delta\tau$ is computed as was done for source turn-on.

8.8.3 TRIDAG

This is a subroutine for solving a system of linear simultaneous equations having a tridiagonal coefficient matrix. The equations are numbered from IF through L, and their subdiagonal, diagonal, and superdiagonal coefficients are stored in arrays A, B, C. The computed solution vector (V(IF), ..., V(L)) is stored in array V.

The mathematical details of all of the steps involved in solving the tridiagonal equations have been given in Section 8.4.

The coding for subroutine TRIDAG is taken from Carnahan et al[11] on page 446.

8.8.4 GETDATA

8.8.4.1 Description

This subroutine is designed for inputting problem data when a program is run under CDC6600 INTERCOM control. It may also be used, however, for inputting

Table 2. Input Data for the Implementation of the TEMP5 Program

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
1*	I1	I1	2	0	T	Print/punch F1, F2 and parameters.
2*	I2	I2	2	0	T, K	Print TAU, LMDA, MU, MN, NQ, INIT, ICNTR. Also, use 1 if IKIRKP option is desired; use 2 for IKIRK option. (see Sec. 9.2.)
3*	I3	I3	2	0	T	Print arrays: KK, A, B, C, D, UPRIM in CYLTMP; also initial values of U, USTAR, etc.
4*	I4	I4	2	0	T	Print U and Q after initial data read-in or computation.
5*	I5	I5	2	0	T	Print array UFIN at every fifth value of both RFIN and ZFIN.
6*	I6	I6	2	0	T	Punch array UFIN and parameters.
7*	I7	I7	2	0	T	Print array U at the following RHO(I) and ZED(J) points: I=2, 2+MI, 2+2MI, 2+3MI, .., ≤ 81 J=2, 2+NI, 2+2NI, 2+3NI, .., ≤ 21.
8	M	M	80	0	T	M+1 is the number of radial points at which temperature data is outputted.
9	N	N	20	0	T	N+1 is the number of axial points at which temperature data is outputted.

*For I1 through I7: If the value is set equal to 1, then appropriate output will be printed; if the value is set equal to 2, then output will be suppressed.

Table 2. Input Data for the Implementation of the TEMP5 Program (Cont.)

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
10	MI	MI	1	0	T	Every MI-th point in the radial direction is printed (see I7).
11	NI	NI	1	0	T	Every NI-th point in the axial direction is printed (see I7).
12	ICNT	ICNT	1	0	T	Array U is printed out for every ICNT-th time cycle (see I7).
13	IU	IU	0	0	T	If 0, temperature distribution U initialized to U0. If 1, initial temperature distribution read-in on file tape ICARD.
14	IQ	IQ	1	0	T	If IQ = 0, initialize source Q to zero. If IQ = 1, initialize source Q to Gaussian. If IQ ≠ 0, 1 read source Q from file tape ICARD.
15	N0	N0	2	0	T	The arbitrary positive integer n_0 in Eq. (82) of text.
16	NMX	NMX	11	0	T	n_{max} (see Sec. 8.6).
17	IRUN	IRUN	100	0		Not used.
18	ICARD	ICARD	5	0	T	Input file for some of the input controlled by IU, IQ.

Table 2. Input Data for the Implementation of the TEMP5 Program (Cont.)

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
19	IPRINT	IPRINT	6	0	T	Output file for all TEMP5 output except for some of the "interactive" output and unformatted temperature output.
20	IPNCH	IPNCH	-	0	N	In original TEMP5, identifies "punch" output file.
21	ITAP3	ITAP3	3	0	T	Unformatted output file for time-temperature.
22	ITAP4	ITAP4	4	0	T	"Interactive" input file.
23	RHO1	RHO1	0	1	T	ρ_1
24	RHO12	RHO12	1	1	T	ρ_{12}
25	ZED1	ZED1	-.5546	1	T	ζ_1
26	ZED12	ZED12	1.1092	1	T	ζ_{12}
27	DTAU0	DTAU0	.0035	1	T	$\Delta\tau_o$
28	TAUMX	TAUMX	5.0	1	T, K	τ_{max} (see Sec.8.6).
29	TAUOFF	TAUOFF	5.0	1	T	τ_{off} (see Sec. 8.6).
30	SIG	SIG	.1292	1	T, K	Either σ or σ_e (see Eq. (33) or (37)).
31	Q0	Q0	0	1	T	If you want $A = \Delta\tau_c/2\sigma^2$ (see Eq. 37), then set Q0 = A. (Be sure $A \geq .001$; highly unlikely to be otherwise). If you want $A = 1/2\ \sigma_e^2$ (see Eq. 33), then set Q0 < .001 (say, 0).
32	U0	U0	0	1	T	Initial (uniform) temperature distribution.

Table 2. Input Data for the Implementation of the TEMP5 Program (Cont.)

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
33	EPS	EPS	.001	1	T	Error tolerance in spline interpolation.
34 to 37	G1(1), G1(2), G1(3), G1(4)	G1	0, 0, 0, 0	1	T	Surface heat flux (see Eqs. (33) or (37)).
38 to 41	H1(1), H1(2), H1(3), H1(4)	H1	0, .0113 .0113 .0113	1	T	Surface heat transfer coefficient (see Eqs. (33) or (37)).
42	MATERIAL	MATER	KCL	-1		Cylinder material identifier. Used for identification purposes only. It is not "used" by any program, but can, of course, be printed in TIKIRK listings and on DISPLAY plots.
43	REF. IND.	NX	1.47	1	K	Cylinder refractive index.
44	BETA	BETA	.00048	1	K	Bulk absorption coefficient (cm^{-1}).
45	THER. COND	K	.0653	1	K	Thermal conductivity ($W/cm\text{-}^{o}C$).
46	LAMBDA	LAMBDA	10.6	1	K	Wavelength (microns).
47	S1R	S1R	-.34E-5	1	K	Stress optic coefficient $S_1^{\rho}(^{o}C)^{-1}$.
48	S1T	S1T	.05E-5	1	K	Stress optic coefficient $S_1^{\theta}(^{o}C)^{-1}$.
49	S2R	S2R	.1E-5	1	K	Stress optic coefficient $S_2^{\rho}(^{o}C)^{-1}$.
50	S2T	S2T	-.1E-5	1	K	Stress optic coefficient $S_2^{\theta}(^{o}C)^{-1}$.
51	DENSITY	DEN	1.98	1	K	Density (gm/cm^3).
52	SPEC. HEAT	CP	.691	1	K	Specific heat ($J/gm\text{-}^{o}C$).
53	RADIUS	R	1.258	1	K	Radius (cm).

Table 2. Input Data for the Implementation of the TEMP5 Program (Cont.)

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
54	EXPER	EXPER	2	-1		Not used.
55	PWR	PW	24.7	1	K	Transmitted power P_t, in watts (see Eq. 9).
56	R1	R1	81	0		Not used.
57	Z1	Z1	11	0		Not used.
58	R2	R2	1	0		Not used.
59	PLT?IY, 2N	IPLOT	1	0		Not used.
60	PROBNO	PROBNO	2347	-1		Problem number-for plot identification.
61	TICU	TICU	.5	1		Not used.
62	XLEN	XLEN	20.	1		Not used.
63	YLEN	YLEN	9.	1		Not used.
64	X-SCALE	SCALEX	12.	1		Not used.
65	Y1-SCALE	SCALEY1	.2	1		Not used.
66	Y2-SCALE	SCALEY2	.2	1		Not used.
67 to 71	XTITLE1 2, 3, 4, 5	XTITLE	time	-1	D	The x-axis is given a title of the form "XTITLE scale is n units/tic" where n may be scalex. Also parameter title.
72 to 76	YTITLE1 2, 3, 4, 5	YTITLE1	temp-deg. C above amb	-1	D	Similar to XTITLE, "surface" title in DISPLAY.
77 to 81	YTITLE2 2, 3, 4, 5	YTITLE2	mean temp above amb	-1		Not used.
82	OPERATOR	NAME	GIANINO	-1	D	Plot identification (required for picking up plots at central site).

Table 2. Input Data for the Implementation of the TEMP5 Program (Cont.)

(1) DATAIN Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Useage Code	(7) Description
83 to 87	XTI 2, 3, 4, 5		radial distance rho-(cm)	-1	D	Title for "x-axis" in DISPLAY.
88 to 92	YT1 2, 3, 4, 5		axial distance, z-(cm)	-1	D	Title for "y-axis" in DISPLAY.

N.B. O means "zero"; 0 means "oh".

Explanation of Columns

(1) The sequence number of the datum stored in array DATAIN.

(2) Datum name used by operator when he inputs the data. This name is a character string. E.g., I1 means 2HI1 in Hollerith notation.

(3) The symbolic name used in the TEMP5 program. The same quantity may be given a different variable name in other programs that used the quantity.

(4) The value that will be assigned to each item listed, unless a different value is inputted.

(5) 0 ≡ integer (I10) format.
1 ≡ floating point (E10.0) format.
-1 ≡ character string (6A10) format, that is, up to 60 characters are permitted.

(6) N ≡ not used.
Y ≡ used in TEMP5 program.
K ≡ used in TIKIRK program.
D ≡ used in DISPLAY program.

Table 3. Glossary of Variable Names*†

A(I)	.=. Tridiagonal system .=. coefficient vectors.
B(I)	.=. Tridiagonal system .=. coefficient vectors.
BETA(I)	.=. Auxiliary variable: Thomas Algorithm for TRIDAG matrix inversion.
C(I)	.=. Tridiagonal system .=. coefficient vectors.
D(I)	.=. Tridiagonal system .=. coefficient vectors.
DRHO	.=. Program control parameter .=. $\Delta\rho = (\rho_2 - \rho_1)/M$.
DTAU	.=. Time increment parameter .=. $\Delta\tau$.
DTAU0	.=. Time increment parameter .=. $\Delta\tau_o$.
DZED	.=. Program control parameter .=. $\Delta\zeta = (\zeta_2 - \zeta_1)/N$.
E(I)	.=. Space increment array .=. DRHO, DZED.
EPS	.=. Error tolerance in iterative steps.
F1(I)	.=. An array of integrals.
F2(I)	.=. An array of integrals.
G(I)	.=. Coefficient in general boundary condition.
GAMMA	.=. Auxiliary working variables-Thomas Algorithm (see Eq. (58)).
GAUSS	.=. Subroutine for loading Q with a truncated Gaussian distribution.
GF(X, Y, Z)	.=. Statement function = (2.*Y*Z)/(2.+X*Y).
G/H	.=. Given nondimensional surface temperature.
G(I)	.=. g_1, g_2, g_3, g_4, .=. boundary condition parameter.
G1(I)	.=. Buffer to retain input G(I). G(I) modified during program execution.
H(I)	.=. h_1, h_2, h_3, h_4 .=. boundary condition parameter. Film coefficient.
HF(X, Y)	.=. Statement function .=. boundary condition .=. (2.-X*Y)/(2.+X*Y) (see Eqs. (44)-(47)).

*The symbol .=. means "is defined as."

† 0 means "zero"; O means "oh."

Table 3. Glossary of Variable Names (Cont.)

H=0, G=0		.=.	No heat crosses boundary .=. physical significance of H and G.
H1(I)		.=.	Buffer to retain input H(I). H(I) is modified during program execution.
I		.=.	Indexing variable.
ICARD		.=.	Logical device number for card reader.
ICNT		.=.	Number of I.A.D. cycles between printouts.
ICNTR		.=.	Number of I.A.D. cycles since last printouts.
IF		.=.	Indexing variable.
II		.=.	I-1.
IKEY		.=.	Logical device number for keyboard (or card) input.
IPM(I)		.=.	Basic program parameters integer: see equivalence statements.
IPNCH		.=.	Logical device number for the card punch.
IPRINT		.=.	Logical device number for line printer.
IQ	= 0	.=.	Initialize Q to zero .=. no absorption.
	= 1	.=.	Calculated Q for Gaussian distribution.
	= 2	.=.	Read in value of Q.
IRUN		.=.	Run number.
ITAP3		.=.	Logical device number for TAPE3.
ITAP4		.=.	Logical device number for TAPE4.
ITYPE		.=.	Printing out on operators terminal (if possible). Logical device number.
IU	= 0	.=.	Initialize U (temp) to zero.
	= 1	.=.	Read in initial value of U.
I1		.=.	Punch and print F1, F2, and parameters. Control for output.
I2		.=.	Print TAU, LMDA, MU, NN, NO, ICNT, ICNTR. Control for output.
I3		.=.	Print KK, A, B, C, D, UPRIM .=. Initialize values of U, USTAR, etc. TRIDAG debug.
I4		.=.	Print U and Q after initial data read-in or computed.

Table 3. Glossary of Variable Names (Cont.)

I5	.=.	Print I, J, UFIN(I, J, K).
I6	.=.	Punch UFIN and parameters.
I7	.=.	Print I, J, U(I, J) on half increment shifted lattice.
J	.=.	Indexing variable.
JJ	.=.	Varies with J for indexing.
K	.=.	Indexing variable .=. see cross reference.
KK, KS, L	.=.	Indexing variables.
LMDA	.=.	$\lambda = \Delta\zeta(\Delta\rho)$.=. Special parameter.
M	.=.	ρ-net length .=. $M \cdot \Delta\rho = 1 - \rho_1$.
MI	.=.	Step size for output do-loop .=. ρ-direction.
MS	.=.	Number of given data points.
MU	.=.	$\mu = \Delta\tau/(\Delta\zeta)^2$.
M1	.=.	M+1. Loop indexing variable.
M2	.=.	M1+1. Loop indexing variable.
N	.=.	Special parameter .=. Count of Tau increments.
NF	.=.	Number of time intervals.
NFF	.=.	Duplicate storage for NF.
NI	.=.	Step size for output do-loop .=. zed direction.
NMX	.=.	N_{max}.
NN	.=.	n.
NS	.=.	Number of Spline Interpolated arguments.
NSEQ	.=.	Sequencing index for punched card output.
N0	.=.	Delta Tau doubling count.
N1	.=.	N+1.
N2	.=.	N1+1.
PARAM(I)	.=.	Basic program parameters, REAL, see equivalence statements.
Q(I, J)	.=.	Source distribution.

Table 3. Glossary of Variable Names (Cont.)

QQ	.=.	Working variable used in do-loop for Q(I, J).
QUA(I)	.=.	Values of integral SS from X(I) to X(N).
Q0	.=.	Control parameter for calculation of Q(I, J) in GAUSS.
REX	.=.	RNN/RNO.
RFIN(I)	.=.	Even R-Lattice point coordinates.
RHO(I)	.=.	Half-interval shifted.
RHO1	.=.	ρ_1.
RHO12	.=.	$1 - \rho_1$.
RI	.=.	II.
RJ	.=.	JJ .=. RJ/2.
RM	.=.	M.
RN	.=.	N.
RNN	.=.	Real representation of NN to avoid mixed mode in Delta Tau calculation.
RN0	.=.	Real representation of N0 to avoid mixed mode in Delta Tau calculation.
RRR(I)	.=.	RFIN(I).
SIG	.=.	Variance of Gaussian beam intensity dist.
SIG2	.=.	SIG squared.
SS1(I)	.=.	First derivatives of U.
SS2(I)	.=.	Second derivatives of U.
TAU	.=.	Nondimensional time .=. τ.
TAUMX	.=.	Maximum tau to be computed .=. τ_{max}.
TFIN	.=.	Array of τ-values for which we take printed or punched output.
U(I, J)	.=.	Array of nondimensional temperatures on RHO-ZED lattice.
UCARD	.=.	UFIN buffer for card and line printer output.
UFIN(I, J, K)	.=.	U on RFIN-ZFIN lattice at Kth time.

Table 3. Glossary of Variable Names (Cont.)

UPRIM(I)	.=.	Storage of results of solutions to tridiagonal equations.
USPLN(I)	.=.	Temporary work space used between Rho-splining and Zed-splining.
USTAR(I, J)	.=.	Intermediate temperature distribution in I. A. D. method.
U0	.=.	U(I, J). For uniform initial temperature option.
V	.=.	Computed solution vectors in TRIDAG.
X(I)	.=.	Array of strictly increasing abscissa.
XR(I)	.=.	RHO(I).
XX	.=.	Work space for desired abscissas.
XZ(I)	.=.	RHO protection.
YU(I)	.=.	UFIN(I, J).
ZED(I)	.=.	Nondimensional axial coordinate [cm] .=. Z/A.
ZED1	.=.	Lower Zed boundary .=. ζ_1.
ZED12	.=.	$\zeta_2 - \zeta_1$.
ZFIN(I)	.=.	Lattice coordinates that land on boundary instead of half shifted position. Used for CYLTMP Algorithm.
ZJ	.=.	JJ.
ZZZ(I)	.=.	ZFIN(I).

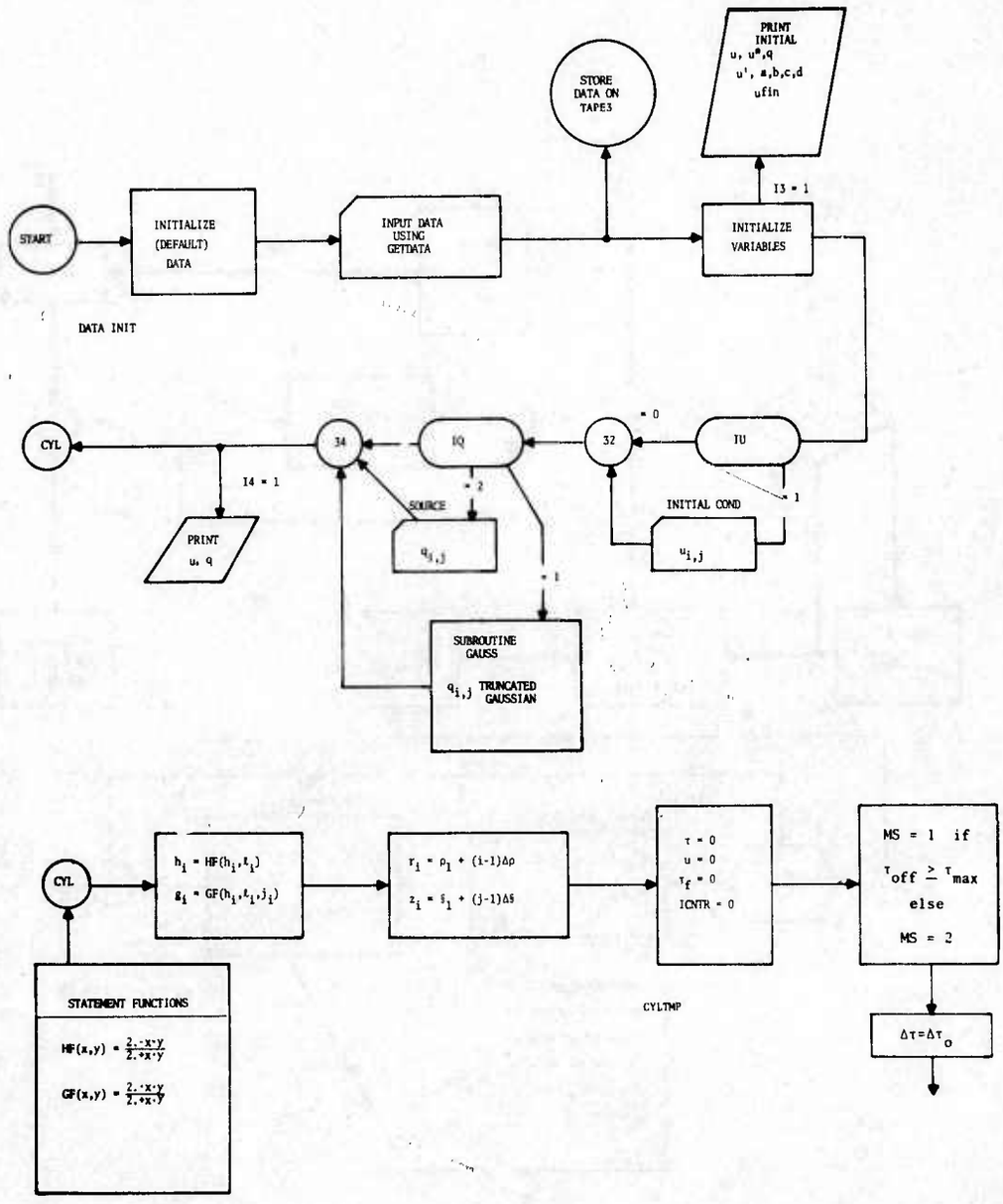

Figure 20. Flow Chart for the CYLTMP Algorithm (Sheet 1 of 4)

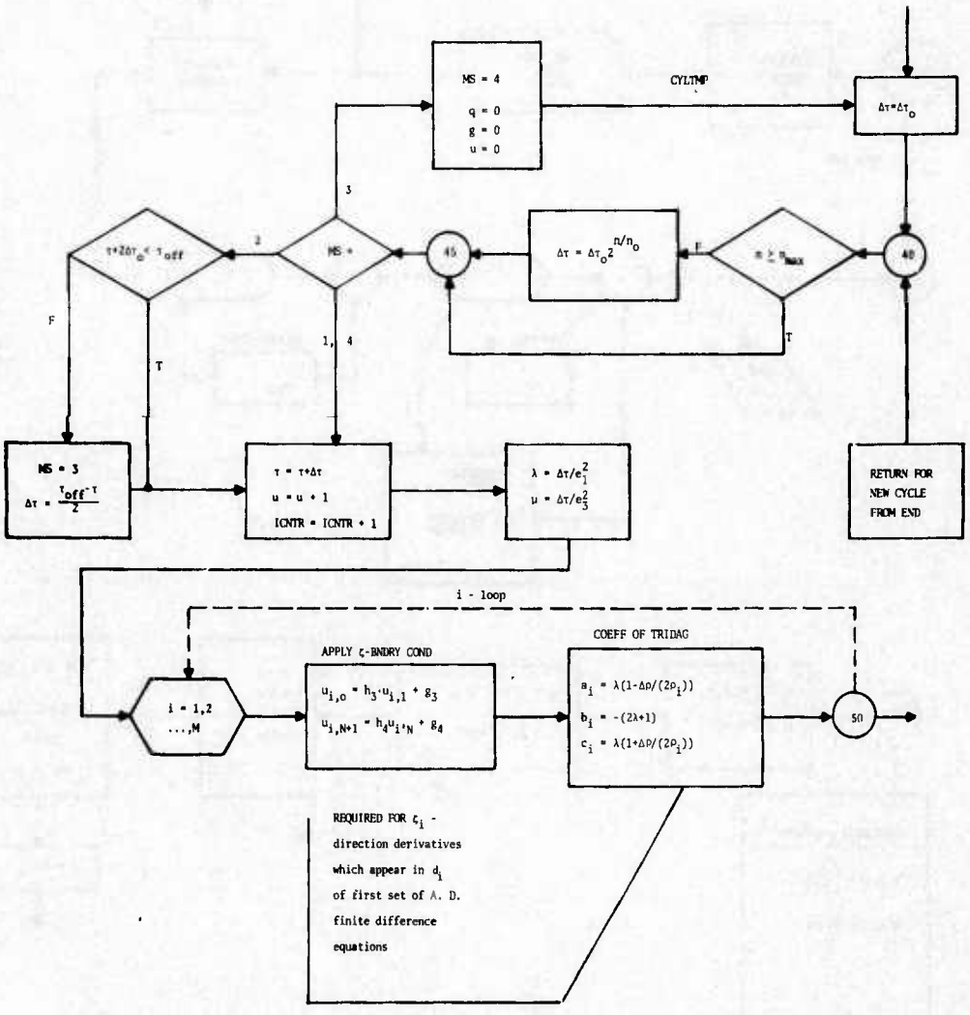

Figure 20. Flow Chart for the CYLTMP Algorithm (Sheet 2 of 4)

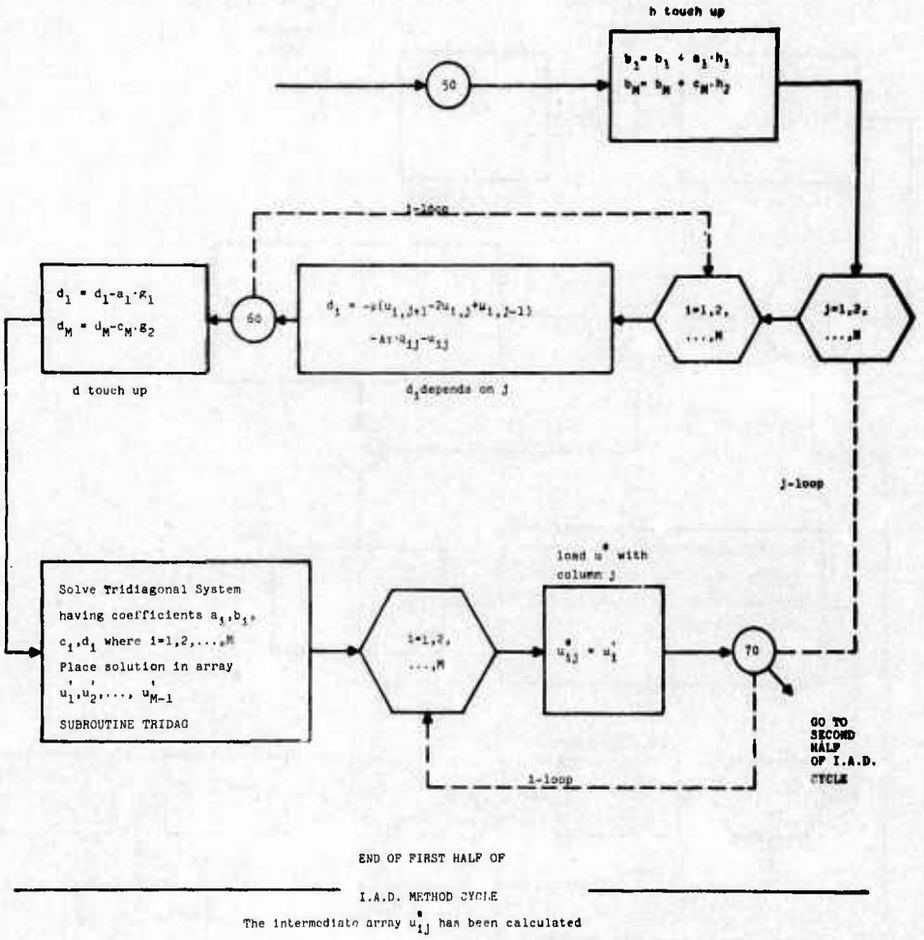

Figure 20. Flow Chart for the CYLTMP Algorithm (Sheet 3 of 4)

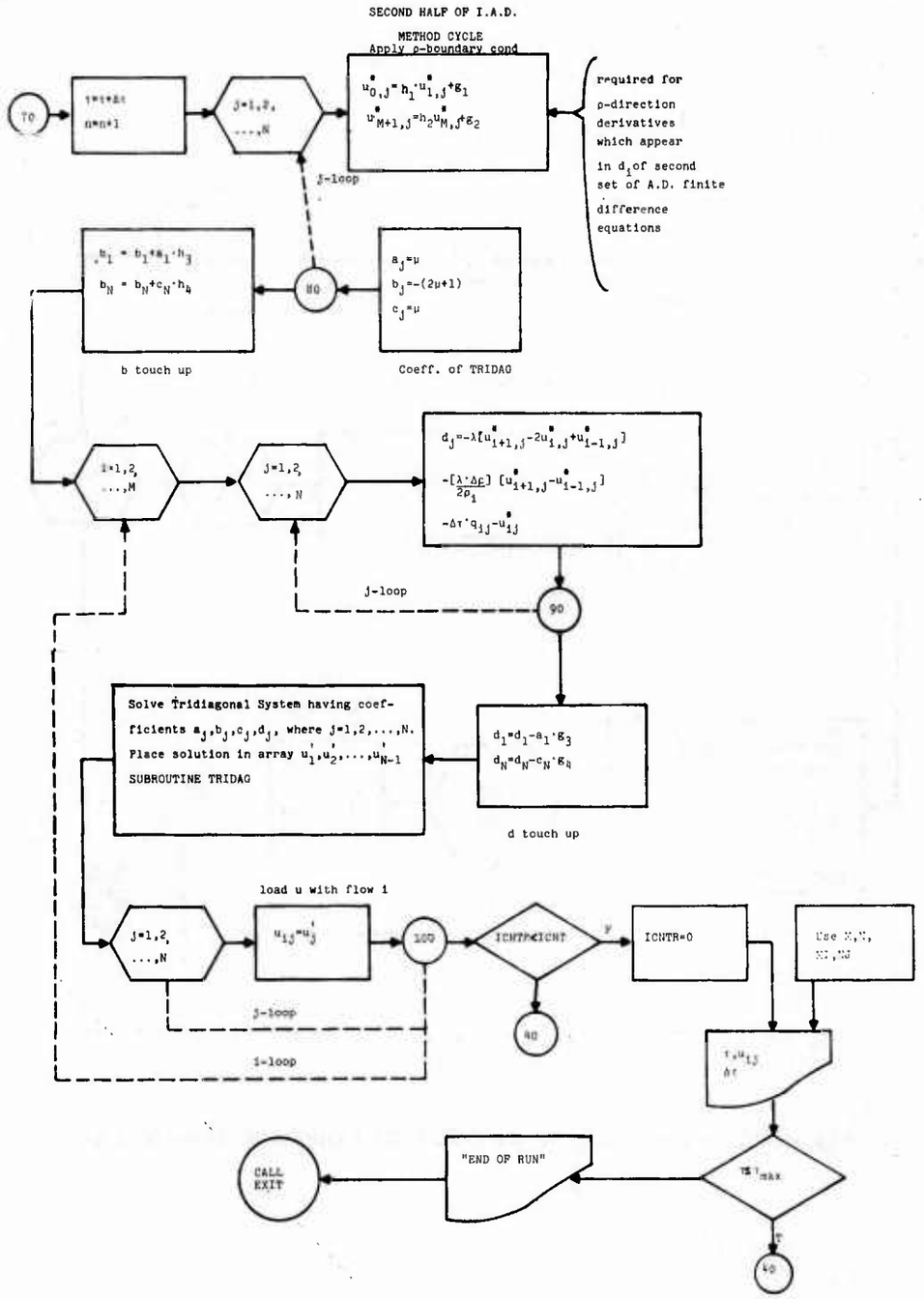

Figure 20. Flow Chart for the CYLTMP Algorithm (Sheet 4 of 4)

data when the program is "batched." The data which is obtained through GETDATA is stored in array DATAIN as an n by 3 array, where n is the maximum number of data to be inputted. Each datum consists of three parts (hence n x 3), which is herein referred to as VALUE, NAME, and FORMAT. VALUE is the numerical or character string value which will be stored for the datum; NAME is a character string (up to ten characters) by which the datum may be identified. FORMAT is a code number (-1, 0, or 1) which is used to indicate that the datum is to be interpreted respectively as: character string, integer, or real (floating point) number.

The first three data required by GETDATA are not part of DATAIN, but are used to control the operation of GETDATA. In INTERCOM mode each of these three data are preceded by questions as follows:

1) READ DATA FILE -n?- (n is the file number)
2) DEFAULTS LISTED?-
3) NAME-VALUE MODE?-

The input data is either YES or NO (the default is NO). If the answer to 1) is YES, then array DATAIN is filled from the first record on TAPE-n. Normally, DATAIN should be filled with defaults in the calling program. These default values will then be replaced by new "default" values obtained from TAPE-n.

If the answer to 2) is YES, then the default DATAIN is printed out in the form:

NAME=VALUE.

FORMAT is indicated by the form in which VALUE is printed. Integers are numerical values with no decimal point; floating point numbers are printed with a decimal point and possibly an exponent. Character strings are indicated by single quotes. The default values are listed in the order in which they are stored in DATAIN.

If the answer to question 3) is YES, then the NAME-VALUE input-mode is used to input data; otherwise, the LIST mode is used. These two input modes are described in detail below.

After these "control" data have been inputted, the DATAIN data are inputted.

8.8.4.2 List Mode

In the LIST mode, GETDATA starts out by printing the first NAME in DATAIN and then waits for the operator to type the VALUE to be assigned to NAME. Similarly, it sequences up to the last NAME in DATAIN, and then prints out a message "data input complete," after the last VALUE in the sequence has been given by the operator. The operator defaults a value by punching the "space," "carriage return" keys (denoted below as SP, CR, respectively). If at any time the operator realizes that an error has been made in typing any preceding VALUE (not just the

current one), it may be corrected by typing $, CR. This causes a shift to the NAME-VALUE mode, under which both the VALUE and NAME must be given by the operator. By this means, VALUES which were assigned earlier in the sequence can be changed. After the correct name and value have been given by the operator, the LIST mode continues where it left off. For example, if the NAME, OPERATOR was listed and the operator had typed $ next to this NAME, the same NAME would be typed again upon return from the NAME-VALUE mode. It might be noted that if a VALUE has been typed (and not signaled with $) which is incompatible with the FORMAT to be associated with that value, then the subroutine automatically goes into the NAME-VALUE mode after typing the message: "wrong data type-try again."

The NAME-VALUE mode has the feature of terminating input, whenever SP, CR are typed, when the program is waiting for a new NAME, VALUE pair. Thus, it may be convenient to terminate inputting when in the LIST mode, by typing $, then SP, CR after the NAME-VALUE mode has been initiated.

8.8.4.3 Name-Value Mode

In this mode, the program first types out the words "name" (6 blanks) "value.....", and then waits expectantly with the type head directly under the "n" of "name." The operator must then type the NAME and corresponding VALUE, with the VALUE starting under or after the "v" of "value." The "....." after "value" indicate the maximum field for inputting numerical data. If the operator types a NAME which is unrecognizable, the message "try again" is typed, followed by another "name" (6 blanks) value.....". A mistake is corrected by merely typing the NAME of the datum to be corrected followed by its correct VALUE. (If a format mistake has been made--see LIST MODE--then the message "wrong data type-try again" is printed indicating that the name-value should be retyped. The "name value....." header is printed only once (or every time an unrecognizable NAME has been inputted) since thereafter it is easy to start NAME and VALUES in the correct positions. (The typehead is always placed under "n" when new data is expected.)

Data input is terminated by typing SP, CR whenever the typehead is under the "n" column of the header.

The field width for all data is ten. Specifically, integers (format=0) are read under I10 format; reals (format=1) are read under E10.0 format. Character strings (format=-1) are read under A10 format, but up to 60 characters may be inputted "at once" provided contiguous space in DATAIN has been provided for them. For example, suppose that DATAIN(10, 2), DATAIN(11, 2), DATAIN(12, 2) are given the NAMES XTITLE(1), XTITLE(2), XTITLE(3), respectively (implying that space for up to 30 "XTITLE" characters has been provided for), then the entire "XTITLE" could be inputted as follows:

NAME VALUE.....

XTITLE(1) DISTANCE ALONG X-AXIS (INCHES)

In the LIST MODE it is also possible to input six numerical data values at "one time." This should only be done when the job is "batched," in which case six data values may be placed on one card, each one occupying a 10-column field. Data which is not to be changed should be replaced by blanks. All or some of the data may be defaulted by using an end-of-record card after the last data to be inputted, causing GETDATA to return to the program or subroutine. If no data is included in the input file a call to GETDATA will have no effect (default DATAIN as provided by the calling program will be used), except the end-of-file indicator INDIC will be set to 1.

GETDATA has a special "gimmick" in that if it is called with a value of INDIC other than 0, it may be used to input a single datum in the LIST mode. To be used in this mode, INDIC should be set equal to the location in DATAIN of the value desired, for example, 50 for DATAIN (50, 1).

According to standard Fortran practice, trailing blanks (of numerical values) are treated as zeros. For example, 10E1 would be interpreted as 10E1␣␣␣␣ (that is, 10^{10000}) and 1 would be interpreted as 1␣␣␣␣␣␣␣␣␣ (that is, 1×10^{9}). GETDATA calls a subroutine (RJUST) which removes all trailing blanks from numerical (but not character string) data so that trailing blanks are not treated as zeros.

8.8.4.4 Entry-Parameter List:

SUBROUTINE GETDATA (DATAIN, NV, IIN, IOUT1, IOUT2, IIN1, ISIZE, ISIZET, INDIC) Parameters:

DATAIN - 3 dimensional array of values, names and formats.

NV - amount of data to be inputted.

IIN - input file number for GETDATA.

IOUT1 - "interactive" (primary) output file for GETDATA.

IOUT2 - secondary output file (stores formatted names and values which are returned by GETDATA).

IIN1 - "scratch" input/output file for reading in "default" values of DATAIN and outputting DATAIN as modified by GETDATA. (Unformatted i/o).

ISIZE - DATAIN is assumed to be dimensioned (ISIZE, 3) in the calling program. It is the size of the first dimension of DATAIN.

ISIZET - 3* ISIZE.

INDIC - an end of file indicator. A value of 1 is returned if an end-of-file on input occurs. If INDIC is given a value other than 0 in the calling program, GETDATA will go into the LIST mode to obtain DATAIN (INDIC-).

When operating under INTERCOM, files IIN and IOUT1 should be "connected."

The storage required for GETDATA plus its two required subroutines RJUST and SSWTCH is 266 words.

8.8.4.5 Algorithm

All data "cards" are read using 6A10 format. If the first word is blank, then the subroutine goes to the next "card" (LIST-mode) or terminates (NAME-VALUE mode). In the LIST mode, fields are searched for the first blank field, upon which the next "card" is read. In NAME-VALUE mode, the search is made only if the FORMAT code for the datum is -1, indicating a possible character string greater than 10 characters. In LIST mode, a search is made of each field to see if $ occurs using AFCRL subroutine MXGETX. If it does, a jump is made to the NAME-VALUE mode of input. All values which are to be interpreted as numbers (integer or real) have their character string representations right adjusted using PML subroutine RJUST. The conversion from character string to coded number is done with the DECODE statement using the appropriate format (I10, E10.0, or A10).

The CDC subroutine ERRSET is called in case of a bad format. A bad format causes a jump to NAME VALUE mode of input.

8.8.4.6 Special Caution and Features

There are three different formats recognized by GETDATA: floating point (E10.0), integer (I10), and character string (6A10).

These formats are given the codes 1, 0, -1, respectively. The E10.0 format converts any decimal number which can be "sensibly" written as a string of ten or less characters into CDC6600 floating point number representation. Examples of permissible character strings are:

328.5678E4
328.5678+4 (the E may be omitted)
3285678.
3285678 (a decimal point is not necessary)
-5.77E-10

Specifically, a floating point number may be written with or without an exponent (which may be indicated by the letter E followed by a signed or unsigned integer OR a signed integer). It may or may not have a decimal point. Blanks are ignored.

The I10 format converts any decimal integer which can be written as a sequence of 10 or less characters from the set [+, -, 0, . . . , 9] into a CDC6600 integer. Trailing, as well as leading blanks within the field are effectively ignored. This is convenient since it is easier to input 1 for instance, as opposed to ␣␣␣␣␣␣␣␣␣1 or 0000000001. Note, however, that 3␣␣2 would be interpreted as 3002, that is, intermediate blanks are considered to be zeros. (Note that an all blank field is NOT equivalent to 0 for GETDATA.)

The 6A10 format allows character strings of up to 60 characters to be inputted "at once." However, one must make sure that sufficient space has been provided to receive character strings of length greater than 10, since each computer word holds a maximum of 10 characters. Any of the 64 characters listed in Appendix A of the CDC Fortran Extended Manual (more or less equivalent to the set on an INTERCOM teletype terminal) are permissible characters in the string. However, the character $ has special significance. When operating in the LIST mode, its appearance signifies that a mistake has been made in inputting some value, and the subroutine temporarily reverts to the NAME-VALUE mode. In this mode, $ has no special significance and is accepted as a legitimate character.

GETDATA assumes that DATAIN has been filled with default VALUES, as well as with the desired NAMES and FORMATS. This initialization of DATAIN can be done by the calling program OR by GETDATA itself, by reading in a DATAIN record from file IIN1.

It is usually convenient to equivalence DATAIN to a block of variables in the calling program. This simplifies subsequent handling of the values returned by GETDATA to the calling program.

GETDATA also prints out the following error messages to aid the programmer:

i) "try again"	Occurs when in the NAME-VALUE mode and an unrecognizable NAME is given.
ii) "file n is empty"	Occurs when an attempt is made to fill DATAIN from an empty file.
iii) "wrong data type - try again"	Occurs when a bad format is given for the datum.

9. TIKIRK PROGRAM

9.1 Introductory Remarks

The principal objective of the TIKIRK program is to compute the Kirchhoff intensity function, as given in Eqs. (23) or (28), Volume I. Before this can be accomplished, however, the file TAPE3 containing the nondimensional temperature w versus nondimensional time τ, as outputted by program TEMP5, must be available. Besides providing the nondimensional mean temperature distributions F1 and F2 mentioned previously, this file also provides as the first record an array of constants which are required to dimensionalize the data into real temperature versus real time (cf, Sections 3, 4, and 5 of Volume I and Table 2 of Volume II). For a more complete description of this file, see Section 9.5.

9.2 Program Options

The TIKIRK program has been set up to operate under two different options; each one being brought into play by an appropriate choice of the parameter I2 in TEMP5 (see Table 2). The first option uses subroutine IKIRK, which calculates the intensity I as a function of space and time for a Gaussian source term. It does so using a 24-point Gaussian integration routine. It is the option that would be used under most circumstances. The second option constitutes a special test case. A subroutine called IKIRKP is used to evaluate the intensity functions on either one of the two mutually orthogonal axes going through the Gaussian focal point for the special case of a window having a uniform mean temperature.

Allowing the control parameter I2 in the TEMP5 program to remain equal to its default value of 2, causes the first option to be utilized, while setting I2=1 brings the second option into play.

Since the form of numerical quadrature employed in option #1 may not be accurate for all parameter values which occur in practice, a third option, which uses a subroutine called IKIRK1, is also available. This program is highly accurate (because its integration methods are more exact), but it is extremely slow. It should be used for relatively small ranges of the space-time variables, for example, as a "spot check" for IKIRK.

9.3 Principal Functions and Subroutines

The TIKIRK program requires that various operations, such as integration, interpolation, Bessel function computation, etc., be carried out during the process of its execution. These operations are performed by various function subprograms and subroutines. The names of these functions and subroutines are listed below, together with their principal tasks:

(1) TIKIRK - the main program which calls the other subroutines and subprograms into execution. It also acts as the input/output interface for the main real functions IKIRK, IKIRKP and IKIRK1 (see below).

(2) IKIRK - the main real function for option #1.

(3) PHI - the function which computes $\Phi^{\rho,\theta}$.

(4), (5) J0 and J1 - real functions; compute $J_0(\rho v)$ and $J_1(\rho v)$, respectively.

(6) RTAPE3 - subroutine; reads and linearly interpolates (in time) temperature values from TEMP5. It also linearly interpolates in time and then outputs the dimensionalized window temperature function in a form suitable for plotting via program DISPLAY.

(7), (8) ALI and ATSE - interpolation subroutines for PHI. ALI uses the Aitken-Lagrange method.

(9) DQG24A - subroutine; computes x-values for Gaussian integration.

(10) DQG24B - subroutine; does Gaussian integration.

(11) IKIRKP - the main real function for option #2.

(12) COMPUTE - subroutine; computes the approximations to the integrals which are used in option IKIRKP.

(13) JI - a real function which computes moments of Bessel functions.

(14), (15) BESJF and BESJ - a function and a subroutine, respectively, which compute the Bessel function for a given argument and order.

(16) GETDATA - a subroutine for interactively inputting data.

(17), (18) SSWTCH and RJUST - subroutine used by GETDATA. (See Sections 8.7 and 8.8 for more detailed explanations of GETDATA, SSWTCH and RJUST.) The listings for these two subroutines have already been given in Appendices A.8 and A.9.

(19) PRT - printed output subroutine.

All three options mentioned above use the same "core package" of the following function subprograms and subroutines in their execution: TIKIRK, PHI, J0, J1, RTAPE3, ALI, ATSE, GETDATA, SSWTCH, RJUST and PRT. In addition to these, option #1 uses IKIRK plus the subroutines DQG24A and DQG24B, while option #2 utilizes IKIRKP plus COMPUTE, JI, BESJF and BESJ.

We have given this entire package of 19 main programs, subroutines, etc., constituting the TIKIRK program for options #1 and #2 only, the permanent file name (PFN) of TIBX.

The complete Fortran listings for each of the above function subprograms and subroutines are given in Appendix B.

9.4 Inputting the Data

After program TEMP5 has been executed, output file TAPE3 has been produced and the TIKIRK program attached, various program control data and constants pertaining to the calculation of the intensity function must be inputted, regardless of the option desired. A list of these data is given in Table 4. (The array containing this information is called DATAIN1 in the TIKIRK program.) The column headings are identical with those of Table 2, except that there is no Useage Code column included here. Their meanings are cited in a footnote in the table.

The total "load" storage required for the TIKIRK program is 56133 B words. (The program will operate with a core memory of 60K.) Single precision is used for most calculations. In fact, a comparison of IKIRK and IKIRKP indicates agreement to 4 significant figures.

Since the maximum number of sample "v-values" is 100, then MP should be ≤ 100. If more sample values are required, then array BUF should be dimensioned accordingly. Producing the full array of 100×100 function values takes 327 cpu secs (with OPT=1).

9.5 Program Files

The TIKIRK program depends on unformatted output from the TEMP5 program and requires up to 6 files. The names of the files used are (in the order they appear in the program statement):

TAPE4 - "Interactive" input file for inputting data via GETDATA. It is used for formatted read and must be set to INPUT for batch operation.

TAPE5 - "Interactive" output file for outputting messages from GETDATA and and for outputting a small amount of program flow information.

TAPE3 - Unformatted file outputted by TEMP5. This file, among other things, contains the mean temperature distribution functions F1 and F2 required by IKIRK and IKIRKP. In addition, the first record is the 100 by 3 array referred to as DATAIN containing various constants required by TIKIRK and contained in Table 2. The following temperature records are assumed to be of the form:

NF, TFIN, RFIN(82), ZFIN(22), UFIN(82, 22), F1(82), F2(82)

TAPE7 - Unformatted file containing the intensity distribution function in a form suitable for DISPLAY. The first two records are the 100 by 3 data arrays DATAIN and DATAIN1, the contents of which contain all pertinent program parameters as well as labeling information for DISPLAY. All subsequent records are of the form:

Table 4. Program Control Data and Constants Pertaining to the Calculation of the Intensity Function. The column headings are identical with those of Table 2, except that there is no Useage Code column included here.

(1) Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Description
1	X0	X0	1500.	1	Gaussian focus X_o (meters).
2	X1	X1	1000.	1	Minimum X-value (meters).
3	X2	X2	2000.	1	Maximum X-value (meters).
4	RHOP1	RHOP1	0.	1	Minimum ρ'-value (cm).
5	RHOP2	RHOP2	2.	1	Maximum ρ'-value (cm).
6	MP	MP	100	0	To calculate J values of intensity along the ρ' (or v) axis, where J = any integer, and have each calculation spaced by ρ'_{max}/J (or v_{max}/J) units, set MP = J + 1.
7	NP	NP	100	0	To calculate K values of intensity along the X (or u) axis, where K = any integer, and have each calculation spaced by $(X_{max} - X_{min})/K$ [or, $(u_{max} - u_{min})/K$] units, set NP = K + 1.
8	T1	TIM(1)	10.	1	Array of time values for function evaluation (seconds). Note that if $t_{i+1} < t_i$ then the program stops. Never set T1 = 0; use some small number instead, e.g., 1E-6.
9	T2	TIM(2)	-1	1	
-	-	-	-	-	
17	T10	TIM(10)	-1	1	
18	EPSI	EPSI	.001	1	Used by IBM Sci. Sub. ALI in interpolation of Φ^ρ, Φ^θ.
19	MINT	MINT	6	0	Used by IBM Sci. Sub. ALI and ATSE in interpolation of Φ^ρ and Φ^θ, MINT is the number of points used in the interpolation.
20	IPRNT	IPRINT	1	0	Use 1 for controlling debug output. Use 2 when producing a TAPE8 file for plotting purposes.
21	NGAUS	NGAUS	24	0	Number of points for Gaussian integration. Note that this number should be changed if and only if the Gaussian integration subroutine is changed.

Table 4. Program Control Data and Constants Pertaining to the Calculation of the Intensity Function. The column headings are identical with those of Table 2, except that there is no Useage Code column included here. (Cont.)

(1) Seq. No.	(2) Datum Name	(3) Variable Name	(4) Default Value	(5) Format Code	(6) Description
22	MODE	MODE	2	0	Use 1 if you want I(X, ρ', t). Use 2 if you want I'(u, v, t). (See Eqs. (28) and (23)).
23	UMIN	UMIN	-40.	1	Minimum u-value.
24	UMAX	UMAX	40.	1	Maximum u-value.
25	VMIN	VMIN	0.	1	Minimum v-value.
26	VMAX	VMAX	10.	1	Maximum v-value.
27 to 31	ST1 2, 3, 4, 5	-	Kirchhoff intensity function	-1	"Surface" title (see DISPLAY).
32 to 36	PTI 2, 3, 4, 5	-	Time (seconds)	-1	"Parameter" title (see DISPLAY).
37 to 41	XTI 2, 3, 4, 5	-	Nondimensional radial distance, V	-1	"X-axis" title (see DISPLAY and note below).
42 to 46	YTI 2, 3, 4, 5	-	Nondimensional axial distance, U	-1	"Y-axis" title (see DISPLAY and note below).
47	MSKIP	-	5	0	Of the J values of intensity calculated along the ρ' (or v) axis (see MP), every MSKIP-th value will be printed out.
48	NSKIP	-	5	0	Of the K values of intensity calculated along the X (or u) axis (see NP), every NSKIP-th value will be printed out.

NOTE: If mode = 1, then the above x, y-titles are replaced by "radial distance, rho-prime (cm)" and "axial distance, X (relative to gauss focus) (cm)," respectively.

N.B. 0 means "zero; o means "oh".

Explanation of Columns

(1) The sequence number of the datum stored in array DATAIN1.
(2) Same as in Table 2.
(3) The symbolic name used in the TIKIRK program. A blank variable name means that that part of DATAIN1 has not been equivalenced to another variable.
(4) Same as Table 2.
(5) Same as Table 2.

I, NP, U, T, MP, XMIN, XMAX, (MP intensity values)

where

I = 1, ... NP for each value of time, T

NP = number of axial points

U = axial distance (either u or x depending on mode)

T = dimensionalized time

MP = number of radial points

XMIN = minimum radial distance (either VMIN or RHOP1 depending on mode)

XMAX = maximum radial distance (either VMAX or RHOP2 depending on mode)

TAPE8 - Unformatted file containing the temperature distribution function in a form suitable for display. The first record contains the 100 by 3 data array DATAIN. All subsequent records have the same form as that listed above for TAPE7 except:

U = distance along window axis (cm)

XMIN = inner window radius (usually 0)

XMAX = outer window radius
(MP temperature values)

TAPE6 - Formatted "output" file. This file contains output suitable for printing in the following sequence:

1) Contents of DATAIN
2) Contents of DATAIN1
3) Array of x-values used for Gaussian integration (if IPRNT=1 and IKIRK is called).
4) I, U, XMIN, XMAX, T — every r^{th} I value
   Intensity (I, J), J=1, MP, 5 — if IPRNT=1

9.6 Implementing the Program

9.6.1 GENERAL INSTRUCTIONS

All of the TIKIRK data listed on Table 4 are inputted by two calls to subroutine GETDATA, regardless of the option desired. Normally, in the first call to GETDATA the required data (viz, the first DATAIN array) are obtained from file

TAPE3 by answering "yes" to the query READ DATA FILE-3? Then no changes are made to these data by returning a "space" in the "name value" mode of inputting data. On the second call to GETDATA, the operator always answers "no" to the query READ DATA FILE-7? and "yes" to the query NAME-VALUE MODE? At this point he enters the names and values of all of the input data whose numerical values differ from the default values as shown in Table 4.

The above information is sufficient for inputting the data when using the first option (that is, IKIRK). However, for those circumstances in which options #2 or #3 are desired, additional instructions are required and will be discussed in the next two sections.

Typical detailed commands which can be used to run all of the programs in both the Intercom and Batch modes are listed in various Attachments after Section 11. The "A" attachments pertain to full system operation in the Intercom mode; the "B" attachments exhibit typical control deck setups used to operate the system in the Batch mode.

Attachment 1 shows how to initiate and run a typical TEMP5- and TIKIRK-type calculation, catalog the results on permanent file and then print out these results. If a TEMP5 calculation has already been made and cataloged, Attachment 2 reveals how to change a few of the input variables so that a new temperature distribution may be obtained. If the TEMP5 calculation has been completed and cataloged, Attachment 3 indicates how to make changes in the TIKIRK parameters so that either a different portion of the former diffraction pattern or a completely new diffraction pattern will be produced with each change. Attachment 4 lists the commands necessary to produce the file, called TAPE8, which contains the temperature distribution in the window. This file is required in order to plot the temperatures.

Throughout all of the attachments, PFN stands for "permanent file name" and LFN for "logical file name."

9.6.2 SPECIAL INSTRUCTIONS FOR IKIRKP OPTION

As mentioned previously, the IKIRKP option will be employed when I2=1 in TEMP5. It then calculates the intensities along either the X- or the ρ'-axes if MODE is set equal to 1, or, along either the u- or the v-axes if MODE=2. The choice of the value for MODE is made, of course, at the second call to GETDATA when inputting the data for the TIKIRK program. Specifically, IKIRKP assumes that the window temperature is constant throughout and that it may or may not be a function of time. This assumption considerably simplifies the integrations delineated in Eq. (23), Volume I, which lead to the intensity function. The details of the mathematics leading to the evaluation of $I'(u, 0, t)$ and $I'(0, v, t)$ are enumerated in Appendix C.

In addition to this choice of I2, a few other input parameters in TEMP5 must be fixed to assure that all of the conditions imposed on the window are properly accounted for. If the window temperature is to remain fixed with time (implying that there is no source), the IQ must be set to zero and U0 to the appropriate temperature. Also, all H1(I) must equal zero, otherwise, if the window temperature is allowed to vary with time, then IQ must be set equal to 1, U0 to the appropriate initial temperature and all H1(I) to zero. Furthermore, the numerical value of σ must be ≥ 0.601 (corresponding to $\alpha \leq 0.8325$), otherwise, the message "α* * 2 is out of range" will be outputted. The reason for this restriction on σ is given in Appendix C.

In Table 5 we list the values that should be used for certain TIKIRK input parameters whenever IKIRKP is employed.

Table 5. Values Used for Certain TIKIRK Input Parameters Whenever IKIRKP is Employed.

MODE	Intensity Function Wanted	MP	NP	X1	X2	RHOP2
1	I(X, 0, t)	1	*	*	*	-
	I(0, ρ', t)	*	1	|X0|	-	*
MODE	Intensity Function Wanted	MP	NP	UMIN	UMAX	VMAX
2	I'(u, 0, t)	1	*	*	*	-
	I'(0, v, t)	*	1	0	-	*

*Means that operator should insert whatever value he desires.
-Means that that particular parameter is of no consequence.

The commands listed in the attachments apply equally as well to the IKIRKP option. The only responses which the operator will change will be those special values of the input parameters discussed above.

9.7 The IKIRK1 Option and an Alternate TIKIRK Package

The IKIRK1 option #3 is not contained in the TIKIRK program as described in Sections 9.1 through 9.6. However, a second TIKIRK program package has been assembled which not only offers option #3 but incorporates #1 and #2 as well. Thus, it can be substituted for the original program, if desired. This package consists essentially of four parts, each one to be stored in the computer under its own PFN. The first part contains a main program, also called TIKIRK, plus those

function subprograms and subroutines making up the "core package" listed in Section 9.3 (viz, PHI, J0, J1, RTAPE3, ALI, ATSE, GETDATA, SSWTCH, RJUST and PRT). This "core package" will be utilized by the last three parts. The main program, called TIKIRK, is a slightly modified form of the main program TIKIRK first introduced in Section 8.3. As before, its principal role is to call the other subroutines and subprograms into execution. Since this first part would play a major role in computing the diffraction pattern of the transmitted beam and call upon the various options, it too has been given the PFN of TIBX by us, whenever it has been used.

The next three parts pertain to options #3, #1, and #2 in that order and we have assigned them their own particular PFN. Each one of these parts includes a major function subroutine which is also given the name IKIRK. The modified TIKIRK main program mentioned above, summons a given option by putting in a call to IKIRK. Which IKIRK (and its concomitant subroutines) gets executed depends upon which one was attached just previous to the call. (It should be noted that in this alternate TIKIRK package, the control parameter I2 no longer plays any role in determining which option will be utilized.) This procedure is demonstrated in Attachment 5, which lists typical control commands for running this new TIKIRK package under either Intercom or Batch mode of operation:

The second part, as noted above, pertains to option #3 (IKIRK1). Besides its major function subroutine IKIRK, it contains the following three subroutines:

(1) FREAL1 - computes the products of the functions $f_w f_x$ and $f_w f_y$ (see Eqs. 23-27, Volume I). It has 4 entry points since actually 4 functions have to be integrated.

(2) DCADRE - an integration subroutine from the IMSL set of routines. It requires the external function subroutine FREAL1 as one of the arguments.

(3) UERTST - required by DCADRE to output error messages.

We have stored this second part under the PFN of IK1BX.

The third part pertains to option #1, which was also referred to as IKIRK in Sections 9.2-9.6. Besides its major function subroutine, called IKIRK, it contains the two subroutines DQG24A and DQG24B, mentioned in Section 9.3. To this part, we have given the PFN of IKBX.

The fourth part contains option #2 (IKIRKP). Besides its major function subroutine, called IKIRK, it utilizes the four subroutines COMPUTE, JI, BESJF and BESJ, previously mentioned in Section 9.3. We have given the PFN of IKPBX to this part.

The Fortran listings for the modified TIKIRK program – the three major function subroutines which are each called IKIRK and which lead into each of the three options, as well as the subroutines FREAL1, DCADRE and UERTST – are given in Appendix D. All of the other subprograms and subroutines mentioned above are the same as those listed in Appendix B.

It has already been noted that option #3 (IKIRK1) is exceedingly slow. However, it may be possible to speed it up by increasing the absolute and relative errors which are presently set at 10^{-3} and 10^{-6}, respectively. In addition, the maximum value of the error parameter IER in DCADRE, as well as the maximum estimated bound on the absolute error in integration, is always printed out.

Whenever the results of IKIRK1 disagree with the results of IKIRK (option #1), those of the former should be preferred because of its greater inherent accuracy.

10. DISPLAY PROGRAM

10.1 Introductory Remarks

Program DISPLAY is a general purpose program for displaying two-dimensional arrays of numbers which, intuitively at least, can be thought of as a surface which has been sampled over an evenly spaced grid. The array size is essentially limited (by computer storage capabilities) to a maximum of 100 by 100; there is no minimum size other than the practical one that it does not make good sense to use a program such as this to display a single point. However, it may make sense to use the program to display one-dimensional arrays; for example, an array of size 1 by n. As will be explained in detail later, three types of display are offered by the program: contour map, perspective view, and multiple cross sections (parallel to two rectangular coordinate axes only). Moreover, the program is designed to display several such arrays at one RUN with the idea in mind that these arrays represent the evolution in time (or with some other parameter) of a function of two variables.

In addition to merely plotting the arrays in one or more of the above mentioned forms, this program labels all plots (provided the labeling information is furnished) and lists pertinent experimental parameters (if desired). Most of this labeling information is supplied in one or more data arrays, the structure of which has been described in DATAIN (see Section 8.8). The rest of the labeling information is provided in the records which contain the array rows.

Program control is afforded by a set of input "commands" which are supplied by input cards. The program interprets each command, obeys each command, and terminates when all commands have been followed (or when time runs out).

Before the program is used, it must be somewhat "tailored to fit." This is accomplished through the use of a fixed number of input data which must be supplied in entirety. Examples of such data are: plot id, maximum plot size, array of indices for obtaining labeling information, etc. A complete list of such data is provided in Section 10.2.

The program allows for selection of the starting "time" and its increment assuming that the surface to be displayed is a function of "time." ("Time" may be any suitable parameter.) A surface cross-section may be displayed as a function of time on a single coordinate frame.

10.2 General Instructions

The surfaces to be displayed are assumed to be stored on file TAPE3 in unformatted records where each logical record consists of one "row" of the mp x np array of floating point numbers representing the surface. In addition, each record contains the information: row number (I), total number of rows (NP), y-value to be associated with that row (Y), parameter value (if any) to be associated with (T), number of samples in the row (MP), x-value to be associated with the first element of the row (XMIN), x-value to be associated with the last element of the row (XMAX). Specifically, each record must have the form:

I, NP, Y, T, MP, XMIN, XMAX, (F(I, J;T), J=1, MP)

where F(·, · ;T) is the function to be displayed. Thus, for each value of the parameter T, NP logical records represent one "surface." Several such "surfaces" corresponding to several values of T may be stored on file TAPE3 and exhibited by DISPLAY.

In addition to the above mentioned "surfaces," file TAPE3 may contain any number (including 0) of "information" records, the contents of which include information (such as experimental parameters) which should be printed on each plot and plot titles. Each such information record must be an array (which will be called here DATAIN) of the form DATAIN(100, 3). Thus, each datum is represented by three parts called (in the order in which they appear) VALUE, NAME, and FORMAT. DATAIN is stored in unformatted form. VALUE is the numerical or character string value of the datum, NAME is a character string of up to ten characters which may be used to identify the datum, while FORMAT is a format code number (-1, 0 or 1) which specifies whether the datum value is to be interpreted as a character string (-1), integer (0), or floating point number (1). (See GETDATA, Section 8.8.4.)

These DATAIN records (if any) must be the first records on file TAPE3. Titles (if any) must appear only in the last DATAIN record.

10.3 Data Cards

The following data cards must be inputted to tailor the program for the user's particular application. The number in parenthesis in front of each datum name is the card column at which to start the datum. The number in parenthesis following the datum is the default value of the datum. If the default value is to be used, leave the corresponding card field blank.

Data Card #1

(1)	XMAX	(100.)	Maximum plot length in inches.
(11)	YMAX	(12.)	Maximum plot width in inches.
(21)	PPI	(10.)	Number of points/inch for contours.
(31)	TICU	(.5)	Number of inches between tic-marks for user defined x-y plots.
(41)	XLEN	(10.)	x-y plot coordinate frame x-size.
(51)	YLEN	(8.)	x-y plot coordinate frame y-size.
(61)	SCALEX	(1.)	x-y plot x-scale, that is, no. of x-units/ tic-mark.
(71)	SCALEY	(1.)	x-y plot y-scale, that is, no. of y-units/ tic-mark.

Data Card #2

(1)	XMIN	(0.)	The minimum x-value and y-value to be plotted for user-defined x-y coordinate frame x-y plots.
(11)	YMIN	(0.)	
(21)	NAME	(GIANINO)	User's name to appear on plot.
(31)	PROB. NO.	(2347)	4-digit user's problem number.

Data Card #3

This card provides information for the two-dimensional array INDEX, which is the index of locations for labeling information assumed to be contained in the last DATAIN on the file containing the surfaces to be plotted. INDEX consists of pairs of numbers wherein the first number is the starting location of the label and the second number is the length of the label. If there is no such data, then this card may be left blank and the labeling will not be done. If the letter D is placed in column 1 of this card, the default values shown below are used. In addition to the values of INDEX, the field starting in column 41 should contain the number of DATAIN arrays on TAPE3.

(1)	INDEX(1, 1)	(1)	Surface title.
(6)	INDEX(1, 2)	(30)	Surface title length (characters).
(11)	INDEX(2, 1)	(4)	Parameter title.
(16)	INDEX(2, 2)	(30)	Parameter title length.
(21)	INDEX(3, 1)	(7)	x-title.
(26)	INDEX(3, 2)	(30)	x-title length.
(31)	INDEX(4, 1)	(10)	y-title.
(36)	INDEX(4, 2)	(30)	y-title length.
(41)	NDA	(2)	Number of DATAIN arrays on TAPE3. The INDEX information is taken from the last DATAIN array.

We employ the following numerical values on data card #3 for displaying the intensity (using TAPE7) and the temperature (using TAPE8):

for TAPE7: 27 29 32 14 37 33 42 32

for TAPE8: 72 20 67 13 83 24 88 31 1

Data Cards #4A, 4B, etc.

These cards contain the sequence numbers of data in DATAIN which are to be listed at the beginning of each run of DISPLAY. The sequence numbers pertaining to the TEMP5 parameters have been listed in Table 2, while those pertaining to the TIKIRK parameters have been listed in Table 4. There must be one card for each DATAIN array (see the last entry number in data card #3 above). For example, in our particular data card #3 above for the TAPE7 case, the default value of 2 is implied as the NDA entry, signifying that 2 cards must be used. The first card contains the sequence numbers of the TEMP5 parameters, while the second contains the sequence numbers of the TIKIRK parameters. The objective is to list both sets of parameters on the plots. In data card #3 above for the TAPE8 case, the NDA value of 1 was employed, indicating that only one card is to be used, viz, that containing the sequence numbers of the TEMP5 parameters which are to be listed on the plots.

The sequence numbers start in columns 1, 3, 5, 7, ... for a total of up to 40 indices per card. The default is a blank card. When the default is used, no DATAIN data is to be listed.

The above cards comprise the mandatory data cards. The remaining cards are the "command" cards which indicate what kinds of plots are wanted.

10.4 Command Cards

As stated previously, the DISPLAY program has the capability of 3 different types of plots, viz, multiple x-y, perspective view and contour map. Consequently, there is a command to control each type and they are indicated by the keywords PLOT, PERSPECTIVE and CONTOUR, respectively. The abbreviations PL, P and C, respectively, may also be used. These keywords are modified by certain parameters p_i. We now enumerate all of the modifying parameters of these command keywords and their meanings:

(1) Command: PLOT (p_1, p_2, p_3, p_4, p_5, p_6)

The parameters p_1, p_2, p_4 and p_5 are integers. In the process of creating a TAPE7 or TAPE8 file, up to ten times had to be chosen at each of which a temperature or an intensity distribution was calculated. These ten times were designated by the datum names T1, T2, ...,T10 (see Table 4). The first two parameters in the above command allow for control in selecting which times are to be chosen in the DISPLAY program. For example, the above command directs that the appropriate data corresponding to every p_1-th time value is to be displayed, starting with the p_2-th value (that is, Tp_2).

Recall that both the temperature and the intensity can be plotted either as a function of radial distance or of axial distance. The parameter p_3 can account for either of these two types of plots by taking on the code symbol X when it is a radial distance plot that is desired, or, the code symbol Y, when the axial distance plot is wanted.

For a temperature calculation, the TEMP5 program sets up a net of 82 temperature points in the radial direction (r), extending from the inner to the outer window radius, and 22 temperature points in the axial direction (z), extending from the entrance to the exit faces. Consequently, there are 22 cross-sectional surfaces of T versus r (that is, 22 type-X plots) and 82 cross-sectional surfaces of T versus z (that is, 82 type-Y plots). Thus, depending on the symbol given by p_3, the above PLOT command directs that on one coordinate frame every p_4-th T-versus-distance surface is to be plotted starting with the p_5-th surface.

On the other hand, for intensity calculations, the TIKIRK program establishes a net in the far field consisting of NP intensity points along the axial line (X), extending from some minimum to some maximum axial distance, and MP intensity points in the radial direction (ρ'), starting from the axial line and going perpendicular to it. Consequently, there are NP cross-sectional surfaces of I versus r (that is, NP type-X plots) and MP cross-sectional surfaces of I versus z (that is, MP type-Y plots). Again, depending on the symbol given by p_3, the PLOT

command directs that every p_4-th I-versus-distance surface is to be plotted on one coordinate frame starting with the p_5-th surface.*

The parameter p_6 can take on any one of the following code symbols: NA, NAS, NE, NES, DN or DNS. These symbols will be explained in Section 10.4, paragraph (4).

The default values for the above 6 parameters are: $p_1 = p_2 = p_4 = p_5 = 1$, p_3 = X and p_6 = NA.

(2) Command: PERSPECTIVE (p_1, p_2, p_3, p_4)

The parameters p_1 and p_2 are the same as in the PLOT command above. The numerical value p_3 is the magnitude of the view angle in degrees, measured from the plane of the window's exit face. Parameter p_4 can take on the code symbols NA or NE only.

The default values are: $p_1 = p_2 = 1$, $p_3 = 45$ and p_4 = NA.

(3) Command: CONTOUR (p_1, p_2, p_3, p_4)

The parameters p_1 and p_2 are the same as in the PLOT command above. The integer p_3 refers to the number of contour levels of constant temperature, or intensity, that are to be plotted, up to a maximum of 50. Parameter p_4 can take on any one of the code symbols NA, NE or DN.

The default values are: $p_1 = p_2 = 1$, $p_3 = 10$ and p_4 = NA.

Note that in the above three commands, parameters are separated by commas. Missing parameters are indicated by commas (with no blank spaces between commas), or by a right parenthesis. If the keyword only appears, then default parameters are assumed.

(4) Meaning of Code Symbols NA, NAS, NE, NES, DN and DNS

The letter N means that the functions to be displayed are first normalized before being plotted, that is, the transformation

$$z \rightarrow \frac{c}{(zmax - zmin)} (z - zmin)$$

(where z refers to the value of the ordinate) is made, where the value of c = 100 for CONTOUR plots; for PERSPECTIVE plots, it depends on the size of the array to be displayed. The latter N has a slightly different connotation for X-Y plots. Here, it means that the array is scaled such that it will fit in a coordinate frame with nice scale values. The second letter A or E indicates whether the normalization is over all (A) surfaces or whether each (E) surface is normalized separately, that is, the zmax, zmin are searched for over all arrays or over each array individually. The letters DN mean don't normalize, that is, do not do the above

*There is a circumstance in which the above meaning for p_4 does not apply. See Section 10.4, paragraph (4).

transformation. Note that for PERSPECTIVE displays normalization always occurs. Therefore, DN should never be used. This is done mainly to force the plot to remain within the plot paper boundaries. Again, for X-Y plots the connotation is slightly different, in that in this case the user must provide plot scale values.

The letter S indicates that there will be superimposed on a single coordinate frame many time curves (as chosen by p_1 and p_2) for a given cross-section, rather than having several cross-sections appear on a single frame for one particular time. Since the time variable can have as many as 10 values, then there can be as many as 10 time surfaces, that is, curves, superimposed on one frame. If it is desired to have one coordinate frame for each value of time that has been utilized, then the S should be omitted. This situation definitely pertains to CONTOUR and PERSPECTIVE plots.

When NA or NAS is used, the values assigned to SCALEX, SCALEY on data card #1 and XMIN, YMIN on data card #2 are ignored, so then the computer selects values which are more appropriate for the ranges of ordinate and abscissa involved. When NE or NES is used, the operator must select his own values for these data.

For X-Y plots, it is recommended that either NA or NAS be used. If NE or NES is employed instead, the operator should beware of erroneous coordinate scaling by ensuring that the values for any subsequent maxima and minima do not exceed those of the initial maximum and minimum. If the S is used in these plots, then parameter p_4 has no effect since several surfaces corresponding to several time values at one cross-section are to be plotted, rather than several cross-sections for one time value.

If no command card is included, then the default command of PERSPECTIVE is assumed. Several commands, one per card, may be given for any single run of the program. For example, a perspective display might be followed by a contour display or several perspective displays from several view angles might be called for by a sequence of PERSPECTIVE commands.

Typical detailed commands used for running the DISPLAY program in the Batch mode only are listed in Attachment 6. We do not run this program in the Intercom mode because usually there is not sufficient space allocated on Intercom to allow the program to run to completion.

10.5 Examples of the Use of the Three Different Plotting Commands

In Section 6 of Volume I, we presented many examples of the three different kinds of plots that program DISPLAY was capable of generating, listing in the figure captions the plot commands which controlled the actual plotting of these

curves. We are now in a position to understand and to analyze how these commands control the graphing.

For example, in Figure 3 the command is PLOT (1, 1, Y, 100, 1, NAS). The keyword PLOT indicates that a multiple X-Y plot is involved. The first two parameters (1, 1) mean that every time value available is to be utilized, starting with time value #1 (that is, T1). In this example, there were 9 time curves used (T1 through T9), which are shown on the right hand side of the figure. Whether the curves drawn will represent temperature – or intensity – versus distance depends on whether the permanent file attached previously to the PLOT command was a TAPE8 – or TAPE7 – type file, respectively. For this particular case, it was a TAPE8 file. The third parameter (Y) signifies that the abscissa is the axial distance through the window. The fourth and fifth parameters (100, 1) signify that every 100th T - versus - z surface is to be plotted, starting with the first (that is, starting with the surface existing at zero radial distance, which is through the center of the window). Since there are only 82 T - versus - z surfaces, then setting $p_4 = 100$ can be seen as a ploy for selecting only the first surface ($p_5 = 1$) to the exclusion of all others. In other words, the selection of any value of p_4 greater than 82 would have ensured the same result. The sixth parameter (NAS) indicates that the normalization is to occur over all surfaces and that all of the (9) time curves are to be superimposed on one coordinate frame.

In Figures 4 and 5 the same kind of information was desired except that the T - versus - z profiles were to occur at constant radial distances of 15 and 30 cm, respectively. Since these distances represent ~50 percent and ~100 percent of the radial distance up the window, the parameter p_5 was chosen to be equal to 41 and 81, respectively. (Actually, because of the way the 82 radial positions were chosen, p_5-values of 1, 41 and 81 represent distances of 0.6 percent, 49.4 percent and 99.4 percent up the radial axis, respectively.)

In Figure 6 we wanted to superimpose plots of T - versus - z at the above 3 radial positions at the fixed time of 5 sec, which is the 8th time value. The p_1, p_2 pair of 4, 8 signifies that every 4th time value is to be plotted, starting with T8. As above, this is a ploy to select only T8 and to exclude the others, since there are only 9 times available. Any $p_1 > 2$ would have produced the same result. The p_4, p_5 pair of 40, 1 means that every 40th surface is to be plotted, starting with surface #1. Thus, surfaces #1, 41 and 81 are plotted. Because we want these 3 surfaces, properly normalized, to be superimposed on one coordinate frame at the one fixed time, we leave off the letter S in the parameter p_6.

Temperature is plotted against radial distance for the same model problem in Figure 7 (hence, p_3 = X). Here, we wanted to plot all 21 T - versus - r surfaces (therefore, $p_4 = p_5 = 1$) at 1 and 5 sec. (The reason for there being only 21 rather

than 22 surfaces is due to the way we chose the points; the 22nd point falls outside of the window.) Thus, we choose p_1 and p_2 to be 3 and 5, respectively, meaning that every 3rd time is selected, starting with T5 (=1 sec). As before, dropping the letter S in parameter p_6 ensures one coordinate frame for each time chosen.

Figures 8 and 9 are examples of T - versus - r temperature plots at approximately 1/4 and 3/4 of the way through the window (hence, p_5 = 6 and 16), respectively. Note that setting $p_1 = p_2 = 1$ assures that all times are accounted for and are superimposed because of the S in p_6. Letting p_4 be greater than 22 (here, 100) assures that only that one particular surface will be plotted.

Figures 10 and 11 pertain to the annular-shaped window. Setting p_5 = 11 and 21 results in plots for T - versus - r surfaces through the middle and at the exit face of the window, respectively.

Figures 12-14 show multiple X-Y plots for intensity. In Figure 12 we wanted every time included (hence, $p_1 = p_2 = 1$) of T - versus - axial distance plots (hence, p_3 = Y) along the axis (p_5 = 1), with no other radial distances included (p_4 = 100); and all of the time curves are to be superimposed (hence, S in p_6). In this particular example, we let MP = 100. Thus, any p_4 - value $\geq$100 would have ensured that only those surfaces along the axis would be used.

The axial range is covered by NP points; in our particular example, NP = 61. In Figure 13 we wanted to have superimposed time plots of I - versus - ρ' (hence, p_3 = X) at the center of the axial range only (hence, p_4 = 100, p_5 = 31). At a time of 3 sec (that is, p_2 = 7), the Gaussian focal point occurs at a distance of 830 m along the axis (corresponding to p_5 = 34). Figure 14 displays the I - versus - ρ' graph for this time only.

The plot command for the PERSPECTIVE graphs of Figures 15 and 16 use the default values. Even though a 3D plot was drawn for all 9 times, only a few representative cases are presented here.

For contour plots we wanted the time values of 2 and 8 sec only. Hence, p_1 = 3 and p_2 = 6 in Figures 17 and 18. Twenty contour lines are shown (thus, p_3 = 20). Since we wanted each one of these contours to be labeled with its dimensioned temperature value, rather than a normalized value, we chose the code symbol DN for p_4.

In summary, we can say that program DISPLAY is used mainly to provide a "readable" output of functions of two variables. The PERSPECTIVE plot furnishes a very good general "view" of the function in question. CONTOUR also provides a good view as well as quantitative knowledge of the function values. PLOT (either X-cross-sections or Y-cross-sections) supplies the most quantitative output but usually the least satisfactory overall "view" of the function. It should be pointed out that PLOT can be used for functions of one variable.

10.6 Principal Functions and Subroutines

The various operations associated with the DISPLAY program are carried out by the following subprograms, subroutines and functions:

(1) DISPLAY - the main program which calls the other subroutines and functions into execution.

All of the following are subroutines:

(2) PLOTT1 - draws and titles coordinate frame. It also does the scaling, if it isn't supplied by the user.

(3) PARMPLT - prints out parameters at the beginning of each plotting run.

(4) FILL - selects data from DATAIN and places it in a 2D array which, in turn, is passed to PARMPLT for printing.

(5) CFRAME - draws and labels the coordinate frame for contour and perspective plots.

(6) INTERP - converts user commands and command parameters to subroutine control arguments. It also calls subroutine NUMB to decode parameters.

(7) NUMB - used by INTERP to convert display code numbers in the commands to the proper internal representation of numbers in the computer.

(8) ARROW - draws the arrow used above the rectangular coordinate frame in perspective plots.

(9) APLACE - locates arrow-head for perspective view angle.

(10) RD1 - reads TAPE3 and fills up appropriate arrays to be plotted. Finds maxima and minima of the surfaces to be plotted, if necessary.

(11) CLEV - returns various equispaced contour levels.

(12) SKIP - skips over a designated number of records (that is, "surfaces") while reading TAPE3.

(13) FACE - a system library program which draws the PERSPECTIVE plots. It also uses the library subroutines HIDE, DRAW, SORT and PARFIT.

(14) CONTOR - a system library program which draws the CONTOUR plots. It also calls the library subroutines NEIBOR and FOUR as well as the Calcomp plot subroutines.

(15) RJUST - right adjusts all numerical input data. Its Fortran listing is given in Appendix A. 9.

(16) SYMBL - converts a floating point number using a G10. 3 format to a display code.

The complete Fortran listings for each of the above programs, subroutines and functions are given in Appendix E.

10.7 Algorithms

10.7.1 MULTIPLE X-Y DISPLAYS

The PLOT display graphs either $f(\cdot, y_i)(p_3 = X)$ or $f(x_i, \cdot)(p_3 = Y)$ for selected values of y_i or x_i determined by command parameters p_4 and p_5. Specifically:

$$i = p_5 + (k - 1) \cdot p_4 \qquad (k = 1, \ldots\ldots) \quad .$$

Subroutine PLOTT1 is called after each "surface" array has been filled by subroutine RD1. One of two types of coordinate frames are then drawn by PLOTT1 depending on command parameter p_6. If p_6 = NA or NE, the scaling is done by the Calcomp subroutine SCALE and the coordinate axes are drawn by Calcomp subroutine AXIS. If p_6 = DN, then the user-provided scale is used and the coordinate frame is drawn by a set of statements within PLOTT1.

After the coordinate frame has been drawn, the individual surface "cross-sections" represented by $f(\cdot, y_i)$ or $f(x_i, \cdot)$ are plotted as continuous curves. Each curve is labeled with a symbol and symbol table.

10.7.2 PERSPECTIVE DISPLAY

For perspective displays the surface is always normalized in such a way that the display will approximately fill the same plot area regardless of the size of the surface array. The view elevation angle is fixed at approximately 45 degrees while the view azimuthal angle can be any value (given by the parameter p_3 of the PERSPECTIVE command). To produce a better display, the surface is always bordered by zeroes.

The complete perspective display is produced by calls from the main program to four subroutines (which may, in turn, call other subroutines): APLACE, FACE, CFRAME, and ARROW.

After establishing some display constants, a call is made to APLACE which returns the point at which to place the head of the view direction arrow. Next, a surface of constant height is produced and a call made to FACE with a "switch" positioned such that FACE merely returns values of XMIN, YMIN, DX, DY which will be used on subsequent calls to FACE. The surface is next normalized and a "display frame" is plotted. Because of the difficulty of drawing a coordinate frame for a perspective display, a coordinate frame quite similar to that used for the contour display is drawn adjacent to each perspective display. The only difference is that an angle of view arrow is drawn on the frame. This frame, then, is

produced by calls to CFRAME (see Section 10.7.3 on contour display) followed by a call to ARROW which plots an arrow at the point found by APLACE. Finally, FACE is called to produce the perspective display.

10.7.3 CONTOUR DISPLAY

The contour display is produced by calls to three subroutines: CLEV, CFRAME, and CONTOR. Subroutine CLEV returns p_3 contour levels in array ZLEVS, evenly spaced between the surface maximum and minimum (but exclusive of the surface minimum). The surface may or may not be normalized according to parameter p_4.

After CLEV has been called, the contour "frame" is drawn by a call to subroutine CFRAME. An example of the frame produced by CFRAME is shown below (small letters indicate numerical or character-string values which are inserted):

surface title
ALL FUNCTION VALUES HAVE
BEEN SCALED ACCORDING TO -

$$z \Rightarrow \frac{c \cdot (Z - ZMIN)}{(ZMAX - ZMIN)}$$

WHERE ZMAX = zmax
ZMIN = zmin
ARE THE MAX. AND MIN.
VALUES OVER ALL SURFACES
or (VALUES OF THE SURFACE)
parameter = value

The contour map is then drawn by a call to CONTOR. Each contour level line is identified by a unique (mod 13) symbol which is printed next to the contour map along with the contour level value which the symbol represents.

10.8 Batch and Intercom Modes

Attachment 6 gives a typical set of commands for running the DISPLAY program under the Batch mode only. The Intercom mode does not have sufficient capacity to handle most plotting jobs, since approximately 142,000 octal words are required in the central memory to run the program (including the plot software). The program will run under Intercom if the arrays in unlabeled common are changed from (102, 102) to (22, 22).

Running time of the program is determined by the size of the array to be displayed, the number of commands and the command parameters. As an example,

a 100 by 100 array was displayed with CONTOUR (20 levels), PLOT (10 cross-sections) and PERSPECTIVE in a total CP time of 82.8 sec.

10.9 Other Features

Various quantities which might be useful for debugging purposes are outputted on file TAPE6. These quantities are:

1) XMAX, YMAX, PPI, TICU
2) XLEN, YLEN, SCALEX, SCALEY
3) XMIN, YMIN
4) INDEX (8 integers)
5) NDA (if not 0)
6) IND (four lines of integers representing the locations in DATAIN of data to be plotted).
7) 4) and 5) repeated for each DATAIN.
8) Command parameters. (Note that the command itself can be inferred from the form of the parameters.)
9) NP, MP
10) ZMAX, ZMIN, TIM, ZMIN
11) XMAX, YMIN, YMAX, FLAG1 (Note: FLAG1 is a logical variable which is "FALSE" if an end-of-file has been reached on file TAPE3.)
12) 9) and 10) possibly repeated.
13) XMINF, YMINF, DX, DY (Only for perspective display; these are scale parameters.)

In addition to the output which comes directly from program DISPLAY, the subroutine CONTOR (used in contour display) and FACE (used in perspective display) output various quantities (see subroutine listings).

10.10 Program Files

In addition to OUTPUT, which is used only by the operating system to output messages, DISPLAY uses files TAPE4, TAPE6 and TAPE3.

TAPE3: TAPE3 is the main input file containing the surfaces to be displayed as well as the information records containing titles, etc. All records are unformatted. The file structure is as follows:

record 1:	datain-1 (100, 3)	
record 2:	datain-2 (100, 3)	
.	.	These records may be absent (nda=0)
.	.	
.	.	
record nda:	datain-nda (100, 3)	
.	.	
.	.	
.	.	
record nda+1	row 1 of surface 1	
.	.	
.	.	
.	.	
record nda+np	row np of surface 1	
	row 1 of surface 2	
.	.	
.	.	
.	.	
.	row np of surface 2	
.		
	.	
	.	
last record	row np of surface N	

Each "row" of the surface has a record of the form given in the first paragraph of Section 10.2.

TAPE4: TAPE4 is the "card" input file. It uses formatted data as described in the first paragraph of Section 10.3.

TAPE6: TAPE6 is the formatted output file for messages and debug quantities. See Section 10.8 for a partial listing of TAPE6 records.

11. OTHER CAPABILITIES

The computer programs described in this report either have been extended, or are capable of being extended, to various other aspects of the window problem. For example, a program has been written and successfully implemented in fitting the theoretical temperature distribution produced by program TEMP5 to various experimentally measured temperature distributions. More specifically, the temperature at one or more points on the window's surfaces are measured as a function

of time, as a source is turned on, then off. The theoretical temperature rise and fall is made to approximate the experimental values by a judicious choice of various parameters (such as absorption coefficient, β, and the theoretical boundary conditions, as given by the h_i and g_i) which describe the thermal properties of the window material. The details of this procedure are contained in Technical Memorandum No. 16 by T.B. Barrett, Parke Mathematical Labs. (Oct. 1973).

Another phenomenon which could definitely affect the diffraction pattern in the far field would be multiple internal reflections of the laser beam within the window.[15] Analysis shows that this condition can be handled effectively if each exponential term in Eqs. (25), (26) and (27) is replaced according to the prescription:

$$\exp(ik\Phi^{\gamma}) \rightarrow t_1 t_2 \exp\{ik\Phi^{\gamma}\}/[1 - r_1 r_2 \exp\{2ik(\Phi^{\gamma} + \Phi')\}] \tag{87}$$

in which Φ^{γ} is still determined by Eq. (18) and the extra phase factor Φ' is given by:

$$\Phi'(\rho) = nL + \Delta L_b(\rho) \quad , \tag{88}$$

where ΔL_b is the amount by which the window bulges when heated by the beam. The t's and r's are amplitude transmission and reflection coefficients, respectively. The subscript 1 refers to the window's entrance face, and 2 to its exit face. They are given by Weil:[8]

$$\begin{aligned} t_1 &= 2/(n+1) \\ t_2 &= 2n/(n+1) \\ r_1 &= r_2 = -(n-1)/(n+1) \quad . \end{aligned} \tag{89}$$

Another capability inherent in the program is allowing an axial (that is, z) dependency in the volume heating source term Q. There are two alternate ways in which this could be effected:

(1) Replace subroutine GAUSS (refer to Appendix A.5) by another, which is also to be called GAUSS. This new subroutine is to have the same arguments as before (see card #7120 in Appendix A.5). However, now the Q-array will contain

15. Bendow, B., Gianino, P.D., Hordvik, A., and Skolnik, L.H. (1973) Optics Commun. 7:219; and Skolnik, L.H., Bendow, B., Gianino, P.D., and Cross, E.F. (1974) AFCRL-TR-74-0085 (III), p. 967.

the appropriate z-dependence. The variable names in the argument are defined in the glossary in Table 3.

(2) If the Batch mode is to be employed, an alternate procedure would be to insert the following data cards in the GETDATA stack. Referring to Attachment 1B, use

IQ 2

ICARD 4

on line 34, and, after the blank card on line 35 insert the desired Q-array using format 7F10.3. At present, Q is dimensioned (82, 22).

Other possible extensions of the computer program include:

(1) Temperature dependence of some of the material parameters, such as refractive index n, thermal conductivity K and thermal lensing parameters S_i^{γ}.

(2) Multiple layers, or coatings, on the window to eliminate reflections and/or to compensate the thermal lensing.[16]

(3) Time dependence in the volume source term Q.

Trying to account for an angular (that is, θ) asymmetry in the beam's cross-section would not be a feasible extension, for it would necessitate a complete rewriting of the program, expanding it from a 2-dimensional to a 3-dimensional treatment.

16. Bendow, B., and Gianino, P.D. (1975) Appl. Optics 14:277; and Bendow, B., Gianino, P.D., Flannery, M., and Marburger, J. (1975) Proc. of the Fourth Annual Conf. on Infrared Laser Window Materials, C.R. Andrews and C.L. Strecker (Editors), Advanced Research Project Agcy., Arlington, Virginia, p. 299.

References

VOLUME I

1. Sparks, M. (1971) J. Appl. Phys. 42:5029; and, Jasperse, J.R., and Gianino, P.D. (1972) J. Appl. Phys. 43:1686.
2. Bendow, B., Jasperse, J.R., and Gianino, P.D. (1972) Optics Commun. 5:98.
3. Bendow, B., and Gianino, P.D. (1972) AFCRL-72-0322, unpublished.
4. Bendow, B., and Gianino, P.D. (1973) J. of Electronic Mater. 2:87.
5. Bendow, B., and Gianino, P.D. (1973) Appl. Optics 12:710.
6. Bendow, B., and Gianino, P.D. (1973) Appl. Phys. 2:1.
7. Gianino, P.D., and Bendow, B. (1973) Appl. Phys. 2:71.
8. Weil, R. (1970) J. Appl. Phys. 41:3012: and Bendow, B., Hordvik, A., Lipson, H., and Skolnik, L. (1972) AFCRL-72-0404, unpublished, p. 12.
9. Carslaw, H.S., and Jaeger, J.C. (1959) Conduction of Heat in Solids, 2nd edition, Oxford Press, London, p. 19.
10. Born, M., and Wolf, E. (1964) Principles of Optics, 2nd (revised) edition, Macmillan Co., New York, p. 437.

VOLUME II

11. Carnahan, B., Luther, H.A., and Wilkes, J.O. (1969) Applied Numerical Methods, Wiley and Sons, Inc., New York.
12. Parke, N.G., III (1971) Technical Memorandum No. 4, Parke Mathematical Laboratories, Inc., Carlisle, Massachusetts, unpublished.
13. von Rosenberg, D.U. (1969) Methods for the Numerical Solution of Partial Differential Equations, American-Elsevier Publishing Co., Inc., New York.

14. Ralston, A., and Wilf, H. S. (1967) Mathematical Methods for Digital Computers, Vol. II, Wiley and Sons, Inc., New York.

15. Bendow, B., Gianino, P. D., Hordvik, A., and Skolnik, L. H. (1973) Optics Commun. 7:219; and Skolnik, L. H., Bendow, B., Gianino, P. D., and Cross, E. F. (1974) AFCRL-TR-74-0085 (III), p. 967.

16. Bendow, B., and Gianino, P. D. (1975) Appl. Optics 14:277; and Bendow, B., Gianino, P. D., Flannery, M., and Marburger, J. (1975) Proc. of the Fourth Annual Conf. on Infrared Laser Window Materials, C. R. Andrews and C. L. Strecker (Editors), Advanced Research Project Agcy., Arlington, Virginia, p. 299.

Attachment 1

Commands to Run TEMP5 and TIKIRK

Here, the goal is to initiate and run a specified TEMP5 and a TIKIRK calculation, then print out and catalog the results on permanent file (PF). Before beginning this job, one should have previously stored in the computer the following two PF's: TEBX (the PFN for the TEMP5 program), and, TIBX (the PFN for the TIKIRK program described in Sections 6.1 - 6.6). A third ancillary program, T3D, whose use is optional, is available to be added to the previous two. Given the PFN of T3BX, it prints out functions F1 and F2 (see Eqs. (16) and (17), Volume I). The Fortran listing for program T3D is given in Appendix F. Incidentally, the third letter (B) in each of the above PFN's means that the PF is in binary form.

A. Intercom Commands

The following is a typical set of Intercom commands:

Line No.	Computer types out..	Operator Response	Comments
1	COMMAND-	ETL(450)	Extends time length to 450 octal sec.
2	COMMAND-	CONNECT(TAPE4,TAPE5)	

Line No.	Computer types out..	Operator Response	Comments
	.....Start of TEMP5 calculation.....		
3	COMMAND-	ATTACH(TEB, TEBX, ID=GIANINO)	TEB is LFN for TEBX.
4	COMMAND-	TEB	
5	READ DATA FILE-3?	N	N=no (because you want to create a new TAPE3)
6	DEFAULTS LISTED?-	N	Customary answer.
7	NAME-VALUE MODE?-	Y	Y=yes.
8	NAME VALUE....	BETA .003 I7 1 etc. - - - - - - - - -	(i) Enter the new TEMP5 variables here. See Table 2 Volume II. (ii) The name of the variable starts under N in NAME; its numerical value under V in VALUE.
9	DATA INPUT COMPLETE WORK OF DATINIT COMPLETE... WORK OF CYLTMP COMPLETE... END OF RUN EXIT ----CP SECS EXECUTION... .....TEMP5 Calculation has been completed.....		
10	COMMAND	CONNECT(OUTPUT)	Lines 10-13 cause F1 and F2 to be printed out at remote terminal facility (AU). They may be omitted, if desired.
11	COMMAND-	ATTACH(T3B, T3BX, ID=GIANINO)	T3B is LFN for T3BX.
12	COMMAND-	T3B,,A	A is merely a surrogate name.
	END T3D ..CP SECS EXECUTION..		
13	COMMAND-	ROUTE,A,ST=RMT, FID=GIAAU,DC=PR.	
14	COMMAND-	REQUEST(PAT,*PF)	Lines 14-17 create a new TAPE3-type PF. We arbitrarily give it the PFN of T3NX. PAT is merely a LFN.

Line No.	Computer types out..	Operator Response		Comments
15	COMMAND-	REWIND(TAPE3)		
16	COMMAND-	COPY(TAPE3, PAT)		
17	COMMAND-	CATALOG(PAT, T3NX, ID=GIANINO, RP=999)		
18	COMMAND-	ROUTE, TAPE6, ST=RMT, FID=GIAAU, DC=PR.		Causes the information on TAPE6, containing the TEMP5 results, to be printed out at remote terminal facility (AU). Also clears TAPE6.

(ASIDE: If all that is desired is the temperature distribution in the window, this would be a convenient place to stop).

.....Start of TIKIRK calculation.....

19	COMMAND-	ATTACH(TIB, TIBX, ID=GIANINO)		TIB is LFN for TIBX.
20	COMMAND-	TIB		
21	READ DATA FILE-3?	Y		Because you want to use information from the "new" TAPE3, viz. T3NX.
22	DEFAULTS LISTED?-	N		
23	NAME-VALUE MODE?-	Y		
24	NAME VALUE...	(space, return)		Do not enter any value here; just hit the "space" and "return" keys.
	DATA INPUT COMPLETE			
25	READ DATA FILE-7?-	N		Answer here is always "no".
26	DEFAULTS LISTED?-	N		
27	NAME-VALUE MODE?-	Y		
28	NAME VALUE...	X1 NP - - -	800 26 - - -	(i) Enter the new TIKIRK variables here. See Table 4, Volume II. (ii) See Comment (ii) on Line 8.
29	DATA INPUT COMPLETE			TIKIRK calculation has been completed.

Line No.	Computer types out..	Operator Response	Comments
	NEW VALUE OF TAU IS... - - - -		Prints out the 10 values of TAU corresponding to time values T1 through T10. See Eq. (13), Volume I.
30	COMMAND-	ROUTE, TAPE6, ST=RMT, FID=GIAAU, DC=PR.	Routes the TAPE6 information, containing the TIKIRK results, to remote terminal (AU) for printout.
31	COMMAND-	REQUEST(PIT, *PF)	Lines 31-34 create a new TAPE7-type PF. We arbitrarily give it the PFN of TKX. PIT is merely a LFN. This new PF can be used as the input in the DISPLAY program for plotting the intensity.
32	COMMAND-	REWIND(TAPE7)	
33	COMMAND-	COPY(TAPE7, PIT)	
34	COMMAND-	CATALOG(PIT, TKX, ID=GIANINO, RP=999)	

B. Batch Commands

Card No.	Command	Comments
1	Job ID with T=2000, core memory=170K	
2	ATTACH(TEB, TEBX, ID=GIANINO)	
3	REQUEST(TAPE3, *PF)	Assigns TAPE3 to permanent file.
4	TEB(INPUT, OUTPUT)	Loads and executes TEBX.
5	REWIND(TAPE6)	
6	COPY(TAPE6, OUTPUT)	Prints output.
7	CATALOG(TAPE3, T3NX, ID=GIANINO, RP=999)	
8	ATTACH(TIB, TIBX, ID=GIANINO)	
9	REQUEST(TAPE7, *PF)	
10	REWIND(TAPE6)	
11	TIB(INPUT)	
12	REWIND(TAPE6)	
13	COPY(TAPE6, OUTPUT)	

Card No.	Command	Comments
14	CATALOG(TAPE7, TKX, ID=GIANINO, RP=999)	
15	EXIT, S.	All commands after EXIT, S. are executed if and only if TEB terminates abnormally, e.g., exceeds time limit.
16	REWIND(TAPE6)	
17	COPY(TAPE6, OUTPUT)	
18	CATALOG(TAPE3, T3NX, ID=GIANINO, RP=999)	
19	CATALOG(TAPE7, TKX, ID=GIANINO, RP=999)	
20	7/8/9	EOR Card.
21	N	
22	N	
23	Y	
24A	BETA .003	Name starts in col. 1; numerical value starts in col. 11.
B	I7 1	
-	- -	
-	- -	
-	- -	
25	-----	Blank card.
26	7/8/9	EOR card.
27	Y	
28	N	
29	Y	
30	------	Blank card.
31	N	
32	N	
33	Y	
34A	X1 800	Name starts in col. 1; numerical value starts in col. 11.
B	NP 26	
-	- -	
-	- -	
-	- -	
35	-----	Blank card.
36	6/7/8/9	EOF card.

If only temperature distribution in the window is desired, the above series of commands would be reduced to the following:

Card No.	Command
C1	Job ID with T=60, corememory=60K
C2	ATTACH(TEB, TEBX, ID=GIANINO)
C3	REQUEST(TAPE3,*PF)
C4	TEB(INPUT,OUTPUT)
C5	REWIND(TAPE6)
C6	COPY(TAPE6,OUTPUT)
C7	CATALOG(TAPE3, T3NX, ID=GIANINO, RP=999)
C8	EXIT,S.
C9	REWIND(TAPE6)
C10	COPY(TAPE6,OUTPUT)
C11	CATALOG(TAPE3, T3NX, ID=GIANINO, RP=999)
C12	7/8/9
C13	N
C14	N
C15	Y
C16A	BETA .003
B	I7 1
-	- -
-	- -
-	- -
C17	(blank card)
C18	6/7/8/9

Attachment 2

Commands to Modify a Temperature Distribution

Assume that a TEMP5 calculation has already been performed and the results cataloged under the PFN of T3NX (cf. lines 14-17 in Attach. 1A). Further, assume that one wants to change only a few of the input variables and then calculate a new temperature distribution. The following commands indicate how the previous results may be utilized.

A. Intercom Commands

Line No.	Computer types out..	Operator Response	Comments
1	COMMAND-	ETL(450)	
2	COMMAND-	CONNECT(TAPE4, TAPE5)	
3	COMMAND-	REWIND(TAPE6)	Clears TAPE6.
4	COMMAND-	REWIND(TAPE7)	Clears TAPE7.
5	COMMAND-	ATTACH(PET, T3NX, ID=GIANINO)	Attaches former TAPE3 with LFN of PET.
6	COMMAND-	COPY(PET, TAPE3)	Copies "former" TAPE3 onto a "new" TAPE3, which is given the LFN of TAPE3. Thus, the original file (T3NX) is still intact and available for future use.

Line No.	Computer types out..	Operator Response	Comments
7	COMMAND-	ATTACH(TEB, TEBX, ID=GIANINO)	
8	COMMAND-	TEB	
9	READ DATA FILE-3?-	N	
10	DEFAULTS LISTED?-	N	
11	NAME-VALUE MODE?-	Y	
12	NAME VALUE.....	SIG .4 PWR 1E7 - - - - - -	Enter those TEMP5 variables which are to be changed. Also, see Comment (ii) on Line 8, Attach. 1A.
13 et seq.	Same as line 9, et seq., Attach. 1A. However, when one gets to the equivalent of line 17, Attach. 1A, one should give the new PF a new PFN, say, T3QX.		

B. Batch Commands

Card No.	Command	Comments
1	Job ID with T=1000, CM=60K	
2	ATTACH(PET, T3NX, ID=GIANINO)	Attaches old TAPE3.
3	REQUEST(TAPE, *PF)	If new TAPE3 is to be permanent.
4	COPY(PET, TAPE3)	
5	RETURN(PET)	Return if you want to use at a later date. Otherwise, could purge it.
6	ATTACH(TEB, TEBX, ID=GIANINO)	
7	TEB(INPUT, OUTPUT)	
8	CATALOG(TAPE3, T3QX, ID=GIANINO, RP=999)	The new TAPE3 is given the PFN of T3QX.
9	REWIND(TAPE6)	
10	COPY(TAPE6)	
11	7/8/9	EOR card.
12	Y	
13	N	

Card No.	Command		Comments
14	Y		
15A	SIG	.4	Enter those TEMP5 variables which are to be changed. Also, see comment on line 34, Attach. 1B.
B	PWR	1E7	
	-	-	
	-	-	
	-	-	
16	(blank card)		
17	6/7/8/9		EOF card.

Attachment 3

Commands to Recalculate TIKIRK, Given TEMP5 Results

There are certain input variables which will affect the computed spatial intensity pattern (and possibly even the dimensioned temperature distribution in the window), but not the normalized temperature distribution. The input parameters which belong to this category are most of the variables found in Table 4 and those in Table 2 which are accompanied by the code letter K in the Useage Code column (Col. #6).

As in Attach. 2, we assume that a previous TEMP5 calculation is available under the PFN of T3NX. Now, we desire to observe the effects which changes in the above-mentioned variables have on the computed intensity pattern. For each set of changes made, the TIKIRK program must be executed. The following gives a representative series of typical commands necessary to implement this procedure.

A. Intercom Commands

Line No.	Computer types out...	Operator Response	Comments
1-4	Same as in Attach. 2A.		
5	COMMAND-	ATTACH(TAPE3, T3NX, ID=GIANINO)	Attaches PF with LFN of TAPE3.

Line No.	Computer types out...	Operator Response	Comments

6-21 Repeat lines 19 through 34, inclusive, from Attach. 1A. However, on line 24 enter the desired TEMP5 variables, if any, which you want to change and which do not affect the normalized temperature distribution in the window. (Note that TEMP5 parameters are being entered at this stage, even though the TIKIRK program currently has control.)

On line 28 enter only those TIKIRK variables whose magnitudes are to be different from the default values.

On line 34 select a different PFN for the new TAPE7-type PF just completed.

.....End of TIKIRK calculation.....

If you want to change the variables again and repeat the TIKIRK calculation:

22 COMMAND- TIB

23 Repeat lines 21 through 34, inclusive, from Attach. 1A, subject to the same comments as noted in line 6 above.

Each time the variables are to be changed and the TIKIRK calculation is to be recomputed, repeat lines 22 and 23 above.

B. Batch Commands

If run as a separate problem, assume that the TEMP5 PF having PFN of T3NX has already been created.

Card No.	Command
1	Job ID with T=2000, CM=170K
2	ATTACH(TIB, TIBX, ID=GIANINO)
3	ATTACH(TAPE3, T3NX, ID=GIANINO)
4	REQUEST(TAPE7, *PF)
5	TIB(INPUT)
6	REWIND(TAPE6)
7	COPY(TAPE6, OUTPUT)
8	CATALOG(TAPE7, TKX, ID=GIANINO)

Plus cards #26-#36, inclusive, of Attach. 1B.

Attachment 4

Commands to Produce a TAPE8 for Temperature Plots

The temperature distribution information contained on the file called TAPE3 is not in the proper format for plotting purposes. Rather, this information must be transferred in the appropriate format to another file, called TAPE8, which is suitable for plotting by the DISPLAY program. This attachment gives all of the commands required to generate and catalog a TAPE8 file, provided that the temperature results are already on a TAPE3-type PF. We assume that this latter file has already been created and given the PFN of T3NX.

A. Intercom Commands

Line No.	Computer types out...	Operator Response	Comments
1	COMMAND-	ETL(450)	
2	COMMAND-	CONNECT(TAPE4, TAPE5)	
3	COMMAND-	REWIND(TAPE6)	
4	COMMAND-	REWIND(TAPE7)	
5	COMMAND-	ATTACH(TAPE3, T3NX, ID=GIANINO)	
6	COMMAND-	ATTACH(TIB, TIBX, ID=GIANINO)	

Line No.	Computer types out..	Operator Response		Comments
7	COMMAND-	TIB		
8	READ DATA FILE=3?-	Y		
9	DEFAULTS LISTED?-	N		
10	NAME-VALUE MODE?-	Y		
11	NAME VALUE....	(space, return)		
12	READ DATA FILE-7?-	N		
13	DEFAULTS LISTED?-	N		
14	NAME-VALUE MODE?-	Y		
15	NAME VALUE.....	NP	1	(i) Only the data shown here can be entered. (ii) See Comment (ii) on line 8, Attach. 1A. (iii) Only those temperatures will be plotted which correspond to the (dimensioned) times given by T1, T2, etc. Linear interpolation is done, if necessary.
		MP	1	
		IPRNT	2	
		T1	time value #1 (in sec)	
		-		
		-		
		-		
		-		
		T10	time value #10(in sec)	
16	COMMAND-	REQUEST(DOG,*PF)		Lines 16-19 create a new TAPE8-type PF whose PFN is TEMPX and whose LFN is DOG.
17	COMMAND-	REWIND(TAPE8)		
18	COMMAND-	COPY(TAPE8,DOG)		
19	COMMAND-	CATALOG(DOG, TEMPX, ID=GIANINO, RP=999)		

ASIDE: If you want to calculate the diffraction pattern concomitant with the temperature distribution contained in the permanent file T3NX as mentioned above, refer to Attach. 3A., starting with line #22.

B. Batch Commands

Card No.	Command	Comments
1	Job card, T=20, CM=60K	
2	ATTACH(TAPE3, T3NX, ID=GIANINO)	
3	REQUEST(TAPE8,*PF)	
4	ATTACH(TIB, TIBX, ID=GIANINO)	

Card No.	Command		Comments
5	TIB(INPUT)		
6	CATALOG(TAPE8, TEMPX, ID=GIANINO		
7	7/8/9		EOR card.
8	Y		
9	N		
10	Y		
11	----		Blank card.
12	N		
13	N		
14	Y		
15	NP	1	For cards #15-27, see comments, line 15, Attach. 4A.
16	MP	1	
17	IPRNT	2	
18	T1	time value #1	On cards #18-27, all time values are in seconds.
-	-		
-	-		
-	-		
-	-		
-	-		
27	T10	time value #10	
28	-----		Blank card.
29	6/7/8/9		EOF card.

Attachment 5

Commands for Running Alternate TIKIRK Program

Assume that the alternate TIKIRK program, introduced in Section 9.7, Volume II, has been stored in the computer and is available for use, rather than the original TIKIRK program (as discussed in Sections 9.1-9.6). This alternate program is also given the PFN of TIBX. Let us further assume that we want to utilize option #3, which has the PFN of IK1B. The revised commands to run this alternate program are:

A. Intercom Commands

Line No.	Computer types out..	Operator Response	Comments
1-19	Same as Attach. 1A.		
20	COMMAND-	ATTACH(IKB, IK1B, ID=GIANINO)	IKB is the LFN for this file.
21	COMMAND-	XEQ	Indicates the following commands are loader commands.
22	OPTION =	LOAD=TIB, IKB	Causes loading of the main program TIKIRK and subroutine IKIRK.

Line No.	Computer types out..	Operator Response	Comments
23	OPTION =	EXECUTE=TIKIRK	Initiates execution.
24-37	Same as lines 21-34, inclusive, Attach. 1A.		

The above commands would be the same for the other two options (#1 and 2), except that on line 20 the appropriate PFN would be entered in place of IK1B. However, the LFN of IKB must be maintained.

B. Batch Commands.

Card No.	Command
1	Job card, T=2000, CM=170K
2	ATTACH(TIB, TIBX, ID=GIANINO)
3	ATTACH(TAPE3, T3NX, ID=GIANINO)
4	ATTACH(IKB, IK1B, ID=GIANINO)
5	REQUEST(TAPE7, *PF)
6	LOAD(TIB, IKB)
7	EXECUTE(TIKIRK)
8	REWIND(TAPE6)
9	COPY(TAPE6)
10	CATALOG(TAPE7, TKX, ID=GIANINO, RP=999)
11	7/8/9
12	Y
13	N
14	Y
15	(blank card)
16	N
17	N
18	Y

Card No.	Command	
19A	X1	800
B	NP	26
-	-	-
-	-	-
20	(blank card)	
21	6/7/8/9	

Attachment 6

Batch Commands for Running DISPLAY Program

(1) Intensity Plots

Let us assume that a TAPE7-type file, having the PFN of T7X and containing the intensity information to be plotted, has already been created.

Card No.	Command	Comments
1	Job ID, T=300, CM=170000	
2	ATTACH(DISB, DISBX, ID=GIANINO)	Attaches the DISPLAY program, whose PFN is DISBX and whose LFN is DISB.
3	ATTACH(TAPE3, T7X, ID=GIANINO)	Attaches the permanent file T7X, using the LFN of TAPE3.
4	ATTACH(PEN, ONLINEPEN)	Cards #4-13 control the plotting process.
5	LIBRARY(PEN)	
6	LDSET(PRESET=ZERO)	
7	DISB(INPUT)	

Card No.	Command	Comments
8	REWIND(TAPE6)	
9	COPY(TAPE6)	
10	DISPOSE, PLOT, *OL.	
11	EXIT.	
12	REWIND(TAPE6)	
13	COPY(TAPE6)	
14	7/8/9	
15	Data card #1	See Section 10.3 for details on data cards.
16	Data card #2	
17	Data card #3	Use the values pertaining to TAPE7.
18A	Data card #4A	Cards 18A and 18B contain the sequence nos. of those TEMP5 and TIKIRK parameters, respectively, which are to be listed with the plots.
B	Data card #4B	
19	Plot command cards (one or more)	See Section 10.4 for details on plot command cards.
20	6/7/8/9	

(2) Temperature Plots

Suppose that a TAPE8-type file has already been created, having the PFN of TEMPX and containing the temperature distribution within the window. Then, the commands to obtain temperature plots are the same as in Part (1), above, with the following exceptions:

Card No.	Command	Comments
3	ATTACH(TAPE3, TEMPX, ID=GIANINO)	Again, the PF is given the LFN of TAPE 3.
17	Data card #3	Use the values pertaining to TAPE8.
18B	(Remove data card #4B)	

If it is desired to have any of the above plots drawn with red ink, card #10 above would be replaced by the following:

Card No.	Command	Comments
10	DISPOSE, PLOT, *PL. RED INK PLEASE	The word RED starts in col. 21.

Appendix A

Fortran Listings for TEMP5 Program

A.1 Main Program TEMP5

```
      PROGRAM TEMP5(TAPE4=128,TAPE5=128,TAPE3,TAPE6=128,OUTPUT=128)
C
C     TEMP5 IS THE PROGRAM NAME ASSIGNED TO A PROGRAM BEING ADAPTED
C     FROM PROGRAM TEMP4 FOR THE PURPOSE OF REDUCTION OF EXPERIMENTAL
C     DATA ( U VS T ) TO  DETERMINE H AND BETA. JUNE 1972 BY N.G.PARKE.
C     THE MEANING OF THE SYMBOLS AND PARAMETERS FOR THIS PROGRAM
C     ARE DESCRIBED IN THE COMMENT CARDS WITH SUBROUTINE DTAFT WHICH IS
C     CALLED BY THIS PROGRAM AFTER IT HAS READ IN  THE INITIAL
C     PARAMETERS AND DATA.  A PARAMETER AND DATA DECK IS PUNCHED
C     OUT BY DTAFT.  DTAFT ALSO PRINTS OUT ANY DEBUG DATA INDICATED
C     BY PARAMETERS I1 - I7.
C
C     SUBROUTINE DTAFT IS A MODIFICATION  OF SUBROUTINE TMP88
C     FOR AECRL BY N.G.PARKE IN  JUNE 1972
C
C     THIS  SUBROUTINE  WAS ORIGINALLY MODIFIED ON 4 APRIL 1972 TO SHIFT
C     THE DATA TO EVEN INCREMENTS, USING SPLINE INTERPOLATION AND
C     THE CHOSEN ARBITRARY BOUNDARY CONDITIONS.  THIS MODIFICATION
C     WAS MADE BY N.G.PARKE.
C
C     THE PARENT PROGRAM WAS TEMP4 FOR CALCULATING THE UNSTEADY HEAT
C     CONDUCTION IN A FINITE CYLINDER SUBJECT TO GENERAL BOUNDARY
C     CONDITIONS ON ALL ZED AND RHO SURFACES. IT THEN COMPUTES THE
C     INTEGRALS F1 AND F2, REQUIRED BY DR. BENDOW AND PUNCHES THE
C     INPUT PARAMETERS AND F1 AND F2. IF I1 = 1.
C
C     THE CYLINDER IS HEATED BY A VOLUME SOURCE DISTRIBUTION.
C     CYLINDRICAL SYMMETRY IS ASSUMED. I.E., THE PARABOLIC
C     PARTIAL DIFFERENTIAL EQUATION IS IN CYLINDRICAL COORDINATES
C     WITH THE ANGLE VARIABLE MISSING.
C
C     THE NET OF SPACE POINTS IS SHIFTED HALF AN INCREMENT FROM
C     BOUNDARIES WHICH ARE RECTANGULAR IN CYLINDRICAL COORDINATES.
C     THIS NET IS ROUNDED BY A FICTITIOUS SET OF POINTS THAT
C     ALLOW THE EASY WRITING OF GENERAL BOUNDARY CONDITIONS.
C     TEMP4 IS A PROGRAM WHICH IS BASED ON
C     THE IMPLICIT ALTERNATING DIFFERENCE METHOD OF INTEGRATING
C     A PARABOLIC PARTIAL DIFFERENTIAL EQUATION. IT IS CALLED
C     THE I.A.D. METHOD FOR SHORT.
C
C     IN ADDITION TO THE USUAL LIBRARY OF MATHEMATICAL
C     AND SYSTEM SUBROUTINES, TEMP4 REQUIRED THE SUBROUTINES TRIDAG
C     AND GAUSS, AS WELL AS THE SSP ROUTINE QSF FOR INTEGRATING.
C
C
C     THE THEORETICAL BASIS FOR THIS PROGRAM IS FULLY DOCUMENTED
C     IN TM NO. 5, NATHAN GRIER PARKE III, PARKE MATHEMATICAL LABS,
C     INC., ONE RIVER ROAD, CARLISLE, MASS.,01741, NOVEMBER 1971.
C     MODIFICATIONS FOR VERSION TEMP4 WILL APPEAR IN TM NO.7.
C
C     INPUT AND INITIALIZATION CONTROL CHARACTERS
C     I0 = 0 .=. TEMP DISTRIB U INITIALIZED TO U0
C     I0 = 1 .=. TEMP INITIAL DISTRIBUTION READ IN ON <ICARD<
C     I0 = 0 .=. SOURCE DISTRIB Q INITIALIZED TO ZERO
C     I0 = 1 .=. CALCULATE AND INITIALIZE Q AS A TRUNCATED GAUSSIAN
C                DISTRIBUTION.
C     I0 = 2 .=. SOURCE DISTRIBUTION READ IN ON ICARD.
C
C     USE OF THE BOUNDARY CONDITION PARAMETERS G AND H*
C
C     A PERFECT INSULATING BOUNDARY IS CHARACTERIZED BY G = 0, H = 0
C     NEWTONS LAW OF COOLING IS CHARACTERIZED BY G = 0, H = FILM COEFF
C     GIVEN HEAT INPUT IS CHARACTERIZED BY G = INPUT, H = 0
C     GIVEN TEMPERATURE IS CHARACTERIZED BY G/H = TEMP AND
C     BOTH G AND H VERY LARGE, E.G., G = TEMP * E25,  H = 1.E25.
C
C
C
C     HF(X,Y) AND GF(X,Y,7) ARE STATEMENT FUNCTIONS OF THE FORM
C     USED IN EQUATIONS (19),...,(22) IN TM NO. 5 TO  APPLY THE
C     THE GENERAL BOUNDARY CONDITIO S RE CALCULATING THE  FICTITIOUS
C     POINTS IN THE U AND USTAR ARRAYS.....
C
```

```
C.....ASSIGN LOGICAL NUMBERS TO ICARD,IKEY,IPRINT AND ITYPE USING      000830
C     INPUT DEVICE 1. OR CHANGE THE FIRST READ CARD TO REQIRED DEVICE NO 000840
C     THIS MAKES THE PROGRAM EASILY  TRANSFERABLE  TO  COMPUTERS WITH   000850
C     DIFFERENT LOGICAL DEVICE NUMBER ASSIGNMENTS.                      000860
C                                                                       000870
C     PARAMETERS I1,....I7 FOR CONTROLLING THE PRESENCE AND ABSENCE OF  000880
C     OUTPUT    1 .=. OUTPUT    2 .=. INHIBIT.                          000890
C     I1 .=. PUNCH AND PRINT F1,F2, AND PARAM                           000900
C     I2 .=. PRINT TAU,MOA,MU,NN,NU,ICNT,ICNTR                          000910
C     I3 .=. PRINT KK,A,B,C,D,UPRIM,ALSO INITIAL VALUES OF U,USTAR ETC  000920
C     I4 .=. PRINT U AND D AFTER INITIAL DATA READ IN OR COMPUTED       000930
C     I5 .=. PRINT I,I,UFIN(I,J,K).                                     000940
C     I6 .=. PUNCH UFIN AND PARAMETERS.                                 000950
C     I7 .=. PRINT I,NU(I,J) ON HALF INCREMENT SHIFTED LATTICE.         000960
C     THE HALF INCREMENT SHIFTED LATTICE IS THE ONE USED BY US TO       000970
C     WRITE THE PROGRAM EASILY FOR GENERAL BOUNDARY CONDITIONS.         000980
C                                                                       000990
C     SUBROUTINE SPLNI(N,M,EPS,X,Y,T,SS,SS1,SS2,QUA)                    001000
C     THIS IS A MODIFICATION OF SSP SUBROUTINE SPLIE TO MAKE QUA AN     001010
C     ARRAY TO MAKE THE INTEGRAL A FUNCTION OF X                        001020
C     QUA = INTEGRAL OF SS FROM X(1) TO X(N) - QUA IS AN ARRAY          001030
C                                                                       001040
C     FIND THIRD ORDER SPLINE FCT  FOR A FCT Y(X) GIVEN AT THE          001050
C     POINTS (X(I),Y(I))                                                001060
C     FOLLOWING CHAP.2 OF VOL2 OF RALSTON + WILF                        001070
C                                                                       001080
C     N = NO.OF GIVEN DATA POINTS                                       001090
C     M = NO.OF SPECIFIED ARGUMENTS T(I) FOR WHICH THE SPLINE           001100
C       SS,ITS FIRST DER,SS1 AND SECOND DER.SS2 ARE TO BE COMPUTED      001110
C     EPS = ERROR TOLERANCE  IN ITERATIVE STEPS                         001120
C     X= ARRAY OF STRICTLY INCREASING ABSCISSAS                         001130
C     Y = ARRAY OF FCT VALUES                                           001140
C     T = ARRAY OF DESIRED ABSCISSAS                                    001150
C     SS = ARRAY OF SPLINE VALUES , SS1,SS2 DERIVATIVES                 001160
C     LIMITATIONS  N  NOT LARGER THAT 50                                001170
C                                                                       001180
C     THE ADDITION OF THE SPLINE SUBROUTINE HAS MADE IT NECESSARY TO    001190
C     INTRODUCE SOME NEW SYMBOLS AND ARRAYS                             001200
C     EPS .=. ERROR TOLERANCE IN THE ITERATIVE STEPS                    001210
C     X(22) .=. WORK SPACE FOR ARRAY OF STRICTLY INCREASING ABSCISSAS   001220
C     Y(22) .=. WORK SPACE FOR ARRAY OF FCT VALUES                      001230
C     XX(32) .=. WORK SPACE  FOR DESIRED ABSCISSAS                      001240
C     SS(32) .=. SPLINE VALUE OF U                                      001250
C     SS1 .=. SPLINE VALUE OF FIRST DERIVATIVE OF U                     001260
C     SS2(22) .=. SPLINE SECOND DERIVATIVE OF U                         001270
C     QUA .=. INTEGRAL OF SS FROM X(1),X(NN)                            001280
C     USPLIN(22,22) .=. TEMPORARY WORK SPACE. USED BETWEEN RHO-SPLINING 001290
C     AND ZED SPLINING                                                  001300
C     MS .=. NO. OF GIVEN DATA POINTS                                   001310
C     NS .=. NO. OF SPLINE INTERPOLATED ARGUMENTS                       001320
C                                                                       001330
C                                                                       001340
C -------------------- PROGRAM STATEMENTS ---------------------------- 001350
C                                                                       001360
C     WRITE <PROGRAM TEMPS - A PROGRAM FOR CALLING A SUBROUTINE CYLTMP, 001370
C     WHICH USES PROGRAM TEMP4 MODIFIED  AND DETERMINES H AND BETA FROM 001380
C     EXPERIMENTAL RUNS OF TEMPERATURE VS TIME.<                        001390
C                                                                       001400
      WRITE(6,215)                                                      001410
      WRITE(6,216)                                                      001420
      WRITE(6,217)                                                      001430
      WRITE(6,218)                                                      001440
C                                                                       001450
      CALL DATINIT                                                      001460
      WRITE(5,300)                                                      001470
      CALL CYLTMP                                                       001480
      WRITE(5,210)                                                      001490
      WRITE(5,320)                                                      001500
      CALL EXIT                                                         001510
C                                                                       001520
  210 FORMAT(6E12.5)                                                    001530
  215 FORMAT(1H0,*PROGRAM TEMPS -  A PROGRAM FOR CALLING A SUBROUTINE*) 001540
  216 FORMAT(1H0,*CYLTMP, WHICH  USES PROGRAM TEMP4 MODIFIED TO*)       001550
  217 FORMAT(1H0,*DETERMINE H AND BETA FROM EXPERIMENTAL RUNS OF *)     001560
  218 FORMAT(1H0,*TEMPERATURE VS TIME.*)                                001570
  300 FORMAT(1H0,* WORK OF DATINIT COMPLETE - RETURNED TO TEMPS*)       001580
  310 FORMAT(1H0,*WORK OF CYLTMP COMPLETE - RETURNED TO TEMPS*)         001590
  320 FORMAT(1H0,*END OF RUN*)                                          001600
C                                                                       001610
      END                                                               001620
```

A.2 Subroutine DATINIT

```
      SUBROUTINE DATINIT                                                    001630
C                                                                           001640
      REAL LMDA, MU                                                         001650
      COMMON A(82),B(82),C(82),D(82),UPRIM(82),RHO(82),RFIN(82),            001660
     *U(82,22),USTAR(82,22),Q(82,22),UFIN(82,22),USPLN(82,22),              001670
     *ZFIN(22),ZED(22),F(4),G(4),H(4),LMDA,MU,NN,M1,N1,M2,N2,NF             001680
C                                                                           001690
C                                                                           001700
      REAL NX,K,I0                                                          001710
      INTEGER R1,Z1,R2                                                      001720
      INTEGER DATAIN                                                        001730
      DIMENSION DATAIN(100,3)                                               001740
      COMMON/BLOCK1/                                                        001750
     *      I1,I2,I3,I4,I5,I6,I7,M,N,MI,NI,ICNT,IU,IQ,N0,NMX,IRUN,         001760
     *ICARD,IPRINT,IPNCH,ITAP3,ITAP4,RHO1,RHO2,ZED1,ZED2,DTAU0,             001770
     *TAUMX,TAUOFF,SIG   ,Q0,U0,     EPS,G1(4),H1(4),MATER,NX,BETA,         001780
     *K,LAMBDA,S1R,S1T,S2R,S2T,                                             001790
     *DEN,CP,R,EXPER,PW,R1,Z1,R2,IPLOT,PROBNO,TICU,XLEN,YLEN,SCALEX,        001800
     *SCALEY1,SCALEY2,XTITLE(5),YTITLE1(5),YTITLE2(5),NAME                  001810
      EQUIVALENCE (I1,DATAIN)                                               001820
      DATA (DATAIN(I,1),I=1,92)/7*2,80,20,3*1,0,1,2,11,100,4,6,6,3,         001830
     *4,0,.1,,-.5546,1.1092,.0035,5,,5,,.1292,2*0,,.001,4*0,,0,,3*.015,     001840
     *3HKCL,1,47,4,8E-4,,0653,10,6,,-.34E-5,,05E-5,,1E-5,-,1E-5,1,98,       001850
     *,691,1,258,1H1,24,7,81,11,1,17,4H7204,,5,20,,9,,30,,,1,,1,            001860
     *10HTIME(SECON,3HDS),3*1H ,10HTEMP-DEGC ,10H ABOVE AMB,                001870
     *3*1H ,10HMEAN TEMP ,10H ABOVE AMB,3*1H ,7HBARRETT,                    001880
     *10HRADIAL DIS,10HTANCE,RHO-,4H(CM),1H ,1H ,10HAXIAL DIST,             001890
     *10HANCE,Z-(CM,1H),1H ,1H /                                            001900
      DATA (DATAIN(I,2),I=1,92)/2HI1,2HI2,2HI3,2HI4,2HI5,                  001910
     *2HI6,2HI7,1HM,1HN,2HMI,2HNI,4HICNT,2HIU,2HIQ,2HN0,3HNMX,             001920
     *4HIRUN,5HICARD,6HIPRINT,5HIPNCH,5HITAP3,5HITAP4,4HRHO1,               001930
     *5HRHO12,4HZED1,5HZED12,5HDTAU0,5HTAUMX,6HTAUOFF,3HSIG,                001940
     *2HQ0,2HU0,3HEPS,5HG1(1),5HG1(2),5HG1(3),5HG1(4),5HH1(1),              001950
     *5HH1(2),5HH1(3),5HH1(4),8HMATERIAL,8HREF.IND.,4HBETA,                 001960
     *9HTHER.COND,6HLAMBDA,3HS1R,3HS1T,3HS2R,3HS2T,                         001970
     *          7HDENSITY,9HSPEC.HEAT,6HRADIUS,5HEXPER,                     001980
     *3HPWR     ,3HR1#,3HZ1#,3HR2#,4HPLOT,2N,6HPROBNO,4HTICU,               001990
     *4HXLEN,4HYLEN,7HX-SCALE,8HY1-SCALE,8HY2-SCALE,6HXTITLE,               002000
     *1H2,1H3,1H4,1H5,7HYTITLE1,1H2,1H3,1H4,1H5,7HYTITLE2,1H2,              002010
     *1H3,1H4,1H5,8HOPERATOR,3HXT1,3HXT2,3HXT3,                             002020
     *3HXT4,3HXT5,3HYT1,3HYT2,3HYT3,3HYT4,3HYT5/                            002030
      DATA (DATAIN(I,3),I=1,92)/22*0,19*1,-1,11*1,-1,1,4*0,                 002040
     *-1,6*1,26*-1/                                                         002050
      I1=I2=I3=I4=I5=I6=I7=2                                                002060
      ICARD=5                                                               002070
      INDIC=0                                                               002080
      IPRINT=6                                                              002090
      ITAP3=3                                                               002100
      ITAP4=4                                                               002110
      M=80                                                                  002120
      N=20                                                                  002130
      MI=NI=1                                                               002140
      ICNT=1                                                                002150
      IU=0                                                                  002160
      IQ=1                                                                  002170
      N0=2                                                                  002180
      NMX=11                                                                002190
      IRUN=100                                                              002200
      RHO1=0.                                                               002210
      RHO12=1.                                                              002220
      ZED1=-.5546                                                           002230
      ZED12=1.1092                                                          002240
      DTAU0=.0035                                                           002250
      TAUMX=5.0                                                             002260
      TAUOFF=5.0                                                            002270
      SIG=.1292                                                             002280
      Q0=U0=0.                                                              002290
      EPS=.001                                                              002300
      G1(1)=G1(2)=G1(3)=G1(4)=0.                                            002310
      H1(1)=0.                                                              002320
      H1(2)=H1(3)=H1(4)=.0113                                               002330
      MATER=10HKCL                                                          002340
      NX=1.47                                                               002350
      BETA=4.8E-4                                                           002360
```

```
      K=.0653                                                          000370
      DEN=1.980                                                        000380
      CP=.691                                                          000390
      R=1.258                                                          000400
      EXPER=10H#2                                                      000410
      T0=24.7                                                          000420
      R1=81                                                            000430
      Z1=11                                                            000440
      R2=9                                                             000450
      IPLOT=1                                                          000460
      NAME=10HBARRETT                                                  000470
      PROBNO=10H7204                                                   000480
      TICK=.5                                                          000490
      XLEN=20.                                                         000500
      YLEN=9E0                                                         000510
      SCALEX=12E0                                                      000520
      SCALEY1=.2E0                                                     000530
      SCALEY2=.2E0                                                     000540
      XTITLE(1)=10HTIME(SECON                                          000550
      XTITLE(2)=10HDS)                                                 000560
      YTITLE1(1)=10HTEMP-DEGC                                          000570
      YTITLE1(2)=10H ABOVE AMB                                         000580
      YTITLE2(1)=10HMEAN TEMP.                                         000590
      YTITLE2(2)= 10HABOVE AMB.                                        000600
 1210 CALL GETDATA (DATAIN,92.4,5.6,3.100,300,INDIC)                   000610
      WRITE(ITAP3) DATAIN                                              000620
      NF=0                                                             000630
      M1=M+1                                                           000640
      N1=N+1                                                           000650
      M2=M1+1                                                          000660
      N2=N1+1                                                          000670
C                                                                      000680
C.....CALCULATE THE INCREMENT ARRAY, E(I)..                            000690
C                                                                      000700
      RM=M                                                             000710
      RN=N                                                             000720
      E(1)=RHO12/RM                                                    000730
      E(2)=E(1)                                                        000740
      E(3)=ZED12/RN                                                    000750
      E(4)=E(3)                                                        000760
      DRHO=E(1)                                                        000770
      DZED=E(3)                                                        000780
      NSEQ=0                                                           000790
C                                                                      000800
C.................INITIALIZE U,USTAR,Q,UPRIM,A,B,C,D.....................000810
C                                                                      000820
      DO 20 I = 1,M2                                                   000830
      UPRIM(I) = 0.                                                    000840
      A(I) = 0.                                                        000850
      B(I) = 0.                                                        000860
      C(I) = 0.                                                        000870
      D(I) = 0.                                                        000880
      DO 20 J = 1,N2                                                   000890
      U(I,J) = U0                                                      000900
      USTAR(I,J) = 0.                                                  000910
   20 Q(I,J) = 0.                                                      000920
C                                                                      000930
C     USTAR,Q,UPRIM,A,B,C,D, INITIALIZED TO ZERO.  U = U0              000940
      DO 21 I = 1,M1                                                   000950
      DO 21 J = 1,N1                                                   000960
   21 UFIN(I,J) = 0                                                    000970
C                                                                      000980
C     UFIN HAS BEEN INITIALIZED TO ZERO                                000990
C                                                                      001000
C     PUTS ZETA - COORD IN ZFIN(J)                                     001010
C                                                                      001020
      DO 17 J = 1,N1                                                   001030
      JJ = J - 1                                                       001040
      ZJ = JJ                                                          001050
   17 ZFIN(J) = ZED1 + ZJ*DZED                                         001060
C                                                                      001070
C     PUTS RHO-COORD IN RFIN(I)                                        001080
C                                                                      001090
      DO 18 I=1,M1                                                     001100
      II = I - 1                                                       001110
```

```
150               RI = II                                                  003120
                18 RHO(I) = RHO1 + RI*DRHO                                 003130
                C                                                          003140
                C     PUTS HALF-INC RHO-COORD IN RHO(I)                    003150
                C                                                          003160
155             C     DO 46 I = 1,M2                                       003170
                      II = 2*I-3                                           003180
                      RI = II                                              003190
                      RI = RI/2.                                           003200
                   46 RHO(I) = RHO1 + RI*DRHO                              003210
160             C     WRITE(IPRINT,290)(I,RHO(I),I=1,M2)                   003220
                C                                                          003230
                C                                                          003250
                C     PUTS HALF-INC ZED-COORD IN ZED(J)                    003240
                C     DO 55 J = 1,N2                                       003260
165             C     JJ = 2*J-3                                           003270
                      RJ = JJ                                              003280
                      RJ = RJ/2.                                           003290
                   55 ZED(J) = ZED1 + RJ*DZED                              003300
                C     WRITE(IPRINT,292) (I,ZED(J),J=1,N2)                  003310
170             C                                                          003320
                C     DEBUG PRINTOUT CONTROLLED BY I3                      003330
                C                                                          003340
                C     GO TO (2,3), I3                                      003350
                   2 WRITE(IPRINT,260) U                                   003360
175             C     WRITE(IPRINT,260) USTAR                              003370
                C     WRITE(IPRINT,260) Q                                  003380
                C     WRITE(IPRINT,260) UPRIM                              003390
                C     WRITE(IPRINT,260) A                                  003400
                C     WRITE(IPRINT,260) B                                  003410
180             C     WRITE(IPRINT,260) C                                  003420
                C     WRITE(IPRINT,260) D                                  003430
                C     WRITE(IPRINT,260) UFIN                               003440
                   3 IF(TU.EQ.0) GO TO 32                                  003450
                C     READ(ICARD,225) U                                    003460
185             C  32 IF(TQ.EQ.0) GO TO 34                                 003470
                C     IF(TQ.EQ.1) GO TO 33                                 003480
                C     READ(ICARD,225) Q                                    003490
                C     GO TO 34                                             003500
                C  33 CALL GAUSS(IPRINT,SIG,RHO,M2,N2,Q,QQ)                003510
190             C  34 GO TO (35,36), I4                                    003520
                C  35 WRITE(IPRINT,260) U                                  003530
                C     WRITE(IPRINT,260) ((Q(I,J),I=2,M1),J=2,N1)           003540
                   36 NN = 0                                               003550
                C                                                          003560
195             C     WRITE(IPRINT,280)                                    003570
                C                                                          003580
                C     RETURN                                               003590
                C                                                          003600
                  210 FORMAT(6E12.5)                                       003610
200               220 FORMAT(7I5,T71,I5,I5)                                003620
                  225 FORMAT(7E10.3)                                       003630
                  260 FORMAT(1H0,10(E10.3,3X))                             003640
                  280 FORMAT(1H0,*END OF DATINII*)                         003650
                  290 FORMAT(1H0,*I,RHO(I) =*,5(I5,F8.4))                  003660
205               292 FORMAT(1H0,*J,ZED(J) =*,5(I5,F8.4))                  003670
                  330 FORMAT(1I5,T71,I5,I5)                                003680
                  331 FORMAT(7I5,T71,I5,I5)                                003690
                  335 FORMAT(5E12.5,T71,I5,I5)                             003700
                  336 FORMAT(4E12.5,T71,I5,I5)                             003710
210               END                                                      003720
```

A.3 Subroutine CYLTMP

```
      SUBROUTINE CYLTMP                                                 003730
      REAL LMDA, MU                                                     003740
      COMMON/BLOCK1/                                                    003750
     +       I1,I2,I3,I4,I5,I6,I7,M,N,MI,NI,ICNT,IU,IQ,N0,NMX,TRUN,    003760
     +ICARD,IPRINT,IPNCH,ITAP3,ITAP4,RH01,RH012,ZED1,ZED12,DTAUU,       003770
     +TAUMX,TAUOFF,SIG  ,U0,     EPS,G1(4),H1(4),MATER,NX,BETA,         003780
     +K,LAMBDA,S1R,S1I,S2R,S2I,                                         003790
     +DEN,CP,R,EXPER,PW,R1,Z1,R2,IPLOT,PROBNO,TICU,XLEN,YLEN,SCALEX,    003800
     +SCALEY1,SCALFY2,XTITLE(5),YTITLE1(5),YTITLE2(5),NAME              003810
      COMMON A(82),B(82),C(82),D(82),UPRIM(82),RHO(82),RFIN(82),        003820
     +U(82,22),USTAR(82,22),Q(82,22),UFIN(82,22),USPLN(82,22),          003830
     +ZFIN(22),ZED(22),E(4),G(4),H(4),LMDA,MU,NN,MI,NI,M2,N2,NF         003840
      DIMENSION F1(82),F2(82),X(82),Y(82),SS(82),SS1(82),SS2(82),       003850
     +QUA(82),YU(82),RRR(82),XR(82),ZZZ(22),XZ(22),T(1)                 003860
      HF(X,Y) = (2.-X*Y)/(2.+X*Y)                                       003870
      GF(X,Y,Z) = 2.*Y*Z/(2.+X*Y)                                       003880
      DO 12 I = 1,4                                                     003890
      G(I) = G1(I)                                                      003900
   12 H(I) = H1(I)                                                      003910
      WRITE(IPRINT,255) E                                               003920
      DRHO = E(1)                                                       003930
      DZED = E(3)                                                       003940
C                                                                       003950
C......NOW WE RELOAD ARRAYS H AND G , USING FUNCTIONS HF AND GF         003960
C                                                                       003970
      DO 10 I = 1,4                                                     003980
      G(I) = GF(H(I),E(I),G(I))                                         003990
   10 H(I) = HF(H(I),E(I))                                              004000
C                                                                       004010
      DO 3 I=1,M1                                                       004020
    3 RRR(I)=RFIN(I)                                                    004030
      DO 4 I=1,N1                                                       004040
    4 ZZZ(I)=ZFIN(I)                                                    004050
C......WE NOW WRITE OUT THE NEW VALUES OF ARRAYS G AND H.               004060
C                                                                       004070
      WRITE(IPRINT,250) G                                               004080
      WRITE(IPRINT,245) H                                               004090
      TAU = 0.                                                          004100
      ICNTR = 0                                                         004110
      NF = 0                                                            004120
      TFIN = 0.0                                                        004130
      MS=2                                                              004140
      IF (TAUOFF .GE. TAUMX) MS=1                                       004150
C                                                                       004160
C ..........  MAIN ENTRY FOR NEW I.A.D. CYCLE ..........                004170
C                                                                       004180
   55 DTAU=DTAU0                                                        004190
   40 IF(NN.GE.NMX) GO TO 45                                            004200
      IF (NN .EQ. 0) GOTO 45                                            004210
C                                                                       004220
C       CONDITIONAL CALCULATION OF DTAU, WHEN NN.LT.NMX                 004230
C                                                                       004240
      RNN = NN                                                          004250
      RN0 = N0                                                          004260
      REX = RNN/RN0                                                     004270
      DTAU = DTAU0*2.**REX                                              004280
C                                                                       004290
C..........INCREMENT TAU,NN,ICNTR..............                        004300
C                                                                       004310
   45 GOTO (330,340,360,330) MS                                         004320
  340 IF (TAU+2E0*DTAU .LT. TAUOFF) GOTO 330                            004330
      MS=3                                                              004340
      DTAU=(TAUOFF-TAU)/2E0                                             004350
      GOTO 330                                                          004360
  360 MS=4                                                              004370
      DO 350 J=1,N2                                                     004380
      DO 350 I=1,M2                                                     004390
      Q(I,J)=0E0                                                        004400
  350 CONTINUE                                                          004410
      G(1)=G(2)=G(3)=G(4)=0.                                            004420
      NN=0                                                              004430
      GOTO 55                                                           004440
  330 TAU=TAU+DTAU                                                      004450
      NN = NN + 1                                                       004460
      ICNTR = ICNTR + 1                                                 004470
      LMDA = DTAU/DRHO**2                                               004480
      MU = DTAU/DZED**2                                                 004490
      GO TO (47,48),I2                                                  004500
   47 WRITE(IPRINT,265)TAU,LMDA,MU,NN,N0,ICNT,ICNTR                     004510
```

```
      48 DO 50 I = 2,M1                                                 004520
         U(I,1) = H(3)*U(I,2)+G(3)                                      004530
         U(I,N2) = H(4)*U(I,N1)+G(4)                                    004540
         A(I) = LMDA*(1.-E(I)/2.)/RHO(I))                               004550
         B(I) = -(2.*LMDA+1.)                                           004560
      50 C(I) = LMDA*(1.+E(I)/2./RHO(I))                                004570
C                                                                       004580
C     COMPLETE THE BORDERING OF THE COMPUTATION LATTICE                 004590
C                                                                       004600
      DO 13 J = 2,N1                                                    004610
      U(1,J) = H(1)*U(2,J)+G(1)                                         004620
   13 U(M2,J) = H(2)*U(M1,J)+G(2)                                       004630
C                                                                       004640
C ...TOUCH UP OF COEFFICIENTS B(I).....                                 004650
      B(2) = B(2) + A(2)*H(1)                                           004660
      B(M1) = B(M1) + C(M1)*H(2)                                        004670
      DO 70 J = 2,N1                                                    004680
      DO 60 I = 2,M1                                                    004690
   60 D(I) = -MU*(U(I,J+1)-2.*U(I,J)+U(I,J-1))-DTAU*Q(I,J)-U(I,J)       004700
C                                                                       004710
C.... TOUCH UP OF THE D COEFFICIENTS.....                               004720
C                                                                       004730
      D(2) = D(2) - A(2)*G(1)                                           004740
      D(M1) = D(M1) - C(M1)*G(2)                                        004750
      CALL TRIDAG(2,M1,A,B,C,D,UPRIM)                                   004760
C                                                                       004770
C....WE HAVE JUST SOLVED THE TRIDIAGONAL EQUATIONS.....                 004780
C....THE RESULTS ARE STORED IN UPRIM.. THE CONTENS OF UPRIM..           004790
C....MUST NOW BE TRANSFERED TO COLUMN J OF ARRAY USTAR.....             004800
C                                                                       004810
      DO 63 KK = 2,M1                                                   004820
      GO TO (63,64), I3                                                 004830
   63 WRITE(IPRINT,285) KK,A(KK),B(KK),C(KK),D(KK),UPRIM(KK)            004840
   64 DO 70 K = 2,M1                                                    004850
   70 USTAR(K,J) = UPRIM(K)                                             004860
C                                                                       004870
C                                                                       004880
C     CALCULATE U ON BOUNDARY AND EVEN LATTICE POINTS BY                004890
C     BENDOW INTERNAL SPLINE EXTRAPOLATION TO BOUNDARY POINTS.          004900
C                                                                       004910
      IF (ICNTR.GT.1) GO TO 16                                          004920
      NF = NF+1                                                         004930
      TFIN = TAU - DTAU                                                 004940
C                                                                       004950
C     BENDOW MODIFICATION OF SPLINE INTERPOLATION BEGINS AT THIS POINT  004960
C     IT CONSISTS OF EXTRAPOLATIONS TO THE BOUNDARY, USING SLOPE SSU    004970
C     AND VALUE SS AT INTERIOR POINTS RHO(2),RHO(M1),ZED(2),ZED(N1).    004980
C     OTHER CALCULATIONS ARE MADE BY THIRD ORDER SPLINE INTERPOLATION   004990
C     AND THE ACCURACY OF THE INTERPOLATION IS CONTROLLED BY THE        005000
C     CHOICE OF OF EPS                                                  005010
C                                                                       005020
C      THE FIRST STEP IN THE PROCESS IS SPLINE INTERPOLATION OF U(I,J)  005030
C     RELATIVE TO RHO-VALUES. INDEX I.                                  005040
C                                                                       005050
      DO 160 I = 2,M1                                                   005060
  160 X(I-1) = RHO(I)                                                   005070
      DO 161 J = 1,M1                                                   005080
  161 XR(J)=RRR(J)                                                      005090
      XR(1) = RHO(2)                                                    005100
      XR(M1) = RHO(M1)                                                  005110
      DO 162 J = 2,N1                                                   005120
      DO 163 I = 2,M1                                                   005130
  163 Y(I-1) = U(I,J)                                                   005140
C                                                                       005150
C     IF(I.GT.3) GO TO 1                                                005160
C     WRITE(IPRINT,210)                                                 005170
C     WRITE(IPRINT,200) X                                               005180
C     WRITE(IPRINT,200) Y                                               005190
C     WRITE(IPRINT,200) XR                                              005200
C   1 CALL SPLNI(M,M1,EPS,X,Y,XR,SS,SS1,SS2,QUA)                        005210
      CALL SPLNI(M,M1,EPS,X,Y,XR,SS,SS1,SS2,QUA)                        005220
C                                                                       005230
C     SS AND SS1 ARE USED AT THE END POINTS IN THE BB VERSION.          005240
C                                                                       005250
      DO 164 KS = 1,M1                                                  005260
```

```
                  164 USPLN(KS,J) = SS(KS)                                          005270
155               C     EXTRAPOLATION TO BOUNDARY, USING SS AND SS1                 005280
                  C     USPLN (1,J) = SS(1)+(RFIN(1)-RHO(2))*SS1(1)                 005290
                  C     USPLN (M1,J) = SS(M1)+(RFIN(M1)-RHO(M1))*SS1(M1)            005300
                        162 CONTINUE                                                005310
                  C                                                                 005320
160               C     WE NOW HAVE EVEN  RHO-INTERPOLATED VALUES OF U.  WE NEED EVEN 005330
                  C     ZED-INTERPOLATED VALUES OF U TO INSERT IN UFIN(I,J,K).      005340
                  C                                                                 005350
                  C     THE FOLLOWING STEP DOES SPLINE INTERPOLATION IN THE ZED-DIRECTION 005360
                  C                                                                 005370
165               C                                                                 005380
                        DO 166 I = 2,N1                                             005390
                  166 X(I-1) = ZED(I)                                               005400
                        DO 167 J = 1,N1                                             005410
                  167 X7(J)=Z77(J)                                                  005420
170                     X7(1) = ZED(2)                                              005430
                        X7(N1) = ZED(N1)                                            005440
                        DO 168 I = 1,M1                                             005450
                        DO 169 J = 2,N1                                             005460
                  169 Y(J-1) = USPLN(I,J)                                           005470
175               C                                                                 005480
                  C     IF(I.GT.3) GO TO 2                                          005490
                  C     WRITE(IPRINT,210)                                           005500
                  C     WRITE(IPRINT,200) X                                         005510
                  C     WRITE(IPRINT,200) Y                                         005520
180               C     WRITE(IPRINT,200) X7                                        005530
                  C   2 CALL SPLNI(N,N1,EPS,X,Y,X4,SS,SS1,SS2,QUA)                  005540
                        CALL SPLNI(N,N1,EPS,X,Y,X4,SS,SS1,SS2,QUA)                  005550
                  C                                                                 005560
                  C     THE INTERPOLATED RESULT IS NOW STORED IN UFIN               005570
185               C                                                                 005580
                        DO 170 J = 1,N1                                             005590
                  170 UFIN(I,J) = SS(J)                                             005600
                  C     EXTRAPOLATION TO BOUNDARY, USING SS AND SS1                 005610
                        UFIN(I,1) = SS(1)+(ZFIN(1)-ZED(2))*SS1(1)                   005620
190                     UFIN(I,N1) = SS(N1)+(ZFIN(N1)-ZED(N1))*SS1(N1)              005630
                        168 CONTINUE                                                005640
                  C                                                                 005650
                  C     THIS COMPLETES THE CALCULATION OF UFIN(I,J) FOR THE CURRENT 005660
                  C     VALUE OF K = NF BY THIRD ORDER SPLINE INTERPOLATION AND     005670
195               C     EXTRAPOLATION TO THE BOUNDARIES.                            005680
                  C                                                                 005690
                        DO 130 I = 1,M1                                             005700
                        DO 132 J = 1, N1                                            005710
                  132 YU(J) = UFIN(I,J)                                             005720
200                     T(1) = ZFIN(1)                                              005730
                         CALL SPLNI(N1,1,EPS,ZFIN,YU,1,SS,SS1,SS2,QUA)              005740
                  130 F1(I) = QUA(N1)                                               005750
                  C                                                                 005760
                  C     THIS COMPLETES THE CALCULATION OF F1                        005770
205               C                                                                 005780
                        DO 136 I = 1,M1                                             005790
                  136 YU(I) = F1(I)*RFIN(I)                                         005800
                        T(1) = RFIN(1)                                              005810
                        CALL SPLNI(M1,1,EPS,RFIN,YU,T,SS,SS1,SS2,QUA)               005820
210                     DO 138 II = 2,M1                                            005830
                  138 F2(II) = QUA(II)/RFIN(II)**2                                  005840
                        F2(1) = 0.5*F1(1)                                           005850
                  C                                                                 005860
                  C     THIS COMPLETES THE CALCULATION OF F2                        005870
215               C                                                                 005880
                        WRITE(IT,3) NF,TFIN,RFIN,ZFIN,UFIN,F1,F2                    005890
                  C                                                                 005900
                  C     IF T1 = 1 TYPE OUT F1 AND F2.                               005910
                  C     IF T5 = 1 TYPE OUT UFIN.                                    005920
220               C                                                                 005930
                        IF(T1.EQ.2.AND.T5.EQ.2) GOTO 370                            005940
                        IF(T1.NE.1) GOTO 371                                        005950
                        WRITE(IPRINT,38 )                                           005960
                        DO 373 I=1,M1,5                                             005970
225                 373 WRITE (IPRINT,34) I,F1(I),F2(I)                             005980
                    371 IF(T5.NE.1) GOTO 370                                        005990
                        WRITE(IPRINT,382)                                           006000
                        DO 374 I=1,M1,5                                             006010
                    374 WRITE(IPRINT,383) I,(UFIN(I,J),J=1,N1,5)
```

```
      370 CONTINUE                                                      006020
230   C     THE RECORD OF THE SPLINE CALCULATION AND INTEGRATION HAVE BEEN 006030
      C     WRITTEN AS THE NEXT RECOR ON TAPE 3.                        006040
      C..... THIS IS THE END OF THE FIRST HALF OF THE I.A.D. CYCLE..    006050
      C..... THE INTERMEDIATE ARRAY USTAR HAS BEEN CALCULATED.......    006060
235   C                                                                 006070
        16 TAU = TAU + DTAU                                             006080
           NN = NN + 1                                                  006090
      C                                                                 006100
      C.........APPLY THE RHO BOUNDARY CONDITIONS..........             006110
240   C                                                                 006120
           DO 80 J = 2,N1                                               006130
           USTAR(1,J)=H(1)*USTAR(2,J)+G(1)                              006140
           USTAR(M2,J)=H(2)*USTAR(M1,J)+G(2)                            006150
           A(J) = MU                                                    006160
245        B(J) = -(2.*MU+1.)                                           006170
        80 C(J) = MU                                                    006180
      C                                                                 006190
      C.....COEFFICIENT B TOUCH UP....                                  006200
      C                                                                 006210
250        B(2) = B(2) + A(2)*H(3)                                      006220
           B(N1) = B(N1) + C(N1)*H(4)                                   006230
      C                                                                 006240
      C......CALCULATION OF COEFFICIENTS D ........                     006250
      C                                                                 006260
255        DO 100 I = 2,M1                                              006270
           DO 90 J = 2,N1                                               006280
           D(J) = -LMDA*(USTAR(I+1,J)-2.*USTAR(I,J)+USTAR(I-1,J))       006290
           D(J) = D(J)- LMDA*DRHO/2./RHO(I)*(USTAR(I+1,J)-USTAR(I-1,J)) 006300
        90 D(J) = D(J) - DTAU*Q(I,J) - USTAR(I,J)                       006310
260   C                                                                 006320
      C.... TOUCH UP OF THE D COEFFICIENTS....                          006330
      C                                                                 006340
           D(2) = D(2) - A(2)*G(3)                                      006350
           D(N1) = D(N1) - C(N1)*G(4)                                   006360
265   C                                                                 006370
      C.....CALL TRIDAG TO SOLVE THE TRIDIAGONAL SYSTEM OF EQUATIONS..  006380
      C......THE RESULT WILL BE RETURNED IN UPRIM ARRAY.............    006390
      C                                                                 006400
           CALL TRIDAG(2,N1,A,B,C,D,UPRIM)                              006410
270        DO 93 KK = 2,N1                                              006420
           GO TO (93,94),I3                                             006430
        93 WRITE(IPRINT,285)KK,A(KK),B(KK),C(KK),D(KK),UPRIM(KK)        006440
        94 DO 100 J = 2,N1                                              006450
       100 U(I,J) = UPRIM(J)                                            006460
275        IF(ICNTR.LT.ICNT) GO TO 40                                   006470
      C                                                                 006480
      C.............AT EACH RETURN TO 40 BEGINS ANOTHER I.A.D. CYCLE...006490
      C                                                                 006500
           ICNTR = 0                                                    006510
280        GO TO (111,112),I7                                           006520
       111 CONTINUE                                                     006530
           WRITE(IPRINT,270) TAU,DTAU                                   006540
           DO 110 I = 2,M1,MT                                           006550
           WRITE(IPRINT,275) I,RHO(I)                                   006560
285    110 WRITE(IPRINT,320)(U(I,J),J=2,N1,NI)                         006570
       112 CONTINUE                                                     006580
           IF(TAU.LE.TAUMX) GO TO 40                                    006590
      C    WHEN TAU EXCEEDS TAUMX, THIS SUBROUTE RETURNS TO THE MAIN PROGRAM 006600
      C    TEMP5, AFTER PRINTING <TAU.GT.TAUMX - CYLTMP RET TO TEMP5<   006610
290   C.....................END OF I.A.D. CYCLE ..................       006620
           WRITE(IPRINT,11)                                             006630
           RETURN                                                       006640
        11 FORMAT(1H0,*TAU.GT.TAUMX - CYCLTMP RET TO TEMP5*)            006650
       200 FORMAT(1H0,10F10.4)                                          006660
       210 FORMAT(1H0,*DEBUG% X,Y,XX*)                                  006670
295    245 FORMAT(1H0,*H1,H2,H3,H4 =*,4(E10.3,3X))                      006680
       250 FORMAT(1H0,*G1,G2,G3,G4 =*,4(E10.3,3X))                      006690
       255 FORMAT(1H0,*E1,E2,E3,E4 =*,4(E10.3,3X))                      006700
       265 FORMAT(1H0,*TAU,LMDA,MU,NN,N0,ICNT,INCTR =*,3(E10.3,3X)/415) 006710
       270 FORMAT(1H0,*TAU = *,F10.3,*DTAU = *,E10.3)                   006720
       275 FORMAT(1H0,*I,RHO =*I5,F10.4)                                006730
       285 FORMAT(1H0,I5,3X,5(E10.3,3X))                                006740
       320 FORMAT(1H ,10(E10.3,3X))                                     006750
       380 FORMAT (/,1H0,*    I*,7X,*F1(I)*,8X,*F2(I)*,/)               006760
```

```
305          381 FORMAT (1H0,I5,2(3X,E10.3))                                    000770
             382 FORMAT (/,1H0,*    I*,10X,*UFIN(I,J) FROM J= 1 TO N1 *//)      000780
             383 FORMAT (1H0,I5,10(3X,E10.3),/,1H0,10X,9(3X,E10.3),/,           000790
                +1H0,23X,4(3X,E10.3))                                           000800
                 END                                                            000810
```

A.4 Subroutine TRIDAG

```
1                SUBROUTINE TRIDAG (IF,L,A,B,C,D,V)                            000820
                                                                               000830
C                                                                              000840
C                 THE SUBROUTINE FOR SOLVING A SYSTEM OF LINEAR SIMULTANEOUS   000850
C                EQUATIONS HAVING  A TRIDIAGONAL COEFFICIENT MATRIX.           000860
5               C THE EQUATIONS ARE NUMBERED FROM IF THROUGH L, AND THEIR      000870
C                SUB-DIAGONAL, DIAGONAL, AND SUPER-DIAGONAL COEFFICIENTS ARE   000880
C                STORED IN ARRAYS A,B,C.  THE COMPUTED SOLUTION VECTOR         000890
C                V(IF),.....,V(L) IS STORED IN ARRAY V.                        000900
C                                                                              000910
10               DIMENSION A(1),B(1),C(1),D(1),V(1),BETA(82),GAMMA(82)         000920
C                                                                              000930
C                ....COMPUTE ARRAYS BETA AND GAMMA.....                        000940
C                                                                              000950
                 BETA(IF) = B(IF)                                              000960
15               GAMMA(IF) = D(IF)/BETA(IF)                                    000970
                 IFP1 = IF + 1                                                 000980
                 DO 1 I = IFP1,L                                               000990
                 BETA(I) = B(I) - A(I)*C(I-1)/BETA(I-1)                        001000
               1 GAMMA(I) = (D(I)-A(I)*GAMMA(I-1))/BETA(I)                     001010
20               C                                                             001020
C                ....COMPUTE FINAL SOLUTION VECTOR V .....                     001030
C                                                                              001040
                 V(L) = GAMMA(L)                                               001050
                 LAST = L - IF                                                 001060
25               DO 2 K = 1,LAST                                               001070
                 I = L-K                                                       001080
               2 V(I) = GAMMA(I) - C(I)*V(I+1)/BETA(I)                         001090
                 RETURN                                                        001100
               C                                                               001110
30               END
```

A.5 Subroutine GAUSS

```
      SUBROUTINE GAUSS(IPRINT,SIG,RHO,M,N,Q,QQ)                         007120
      DIMENSION RHO(1),Q(M,N)                                           007130
      IF(SIG.EQ.0.) GO TO 20                                            007140
      SIG2 = SIG**2                                                     007150
      IF(QQ.GE..001) GO TO 5                                            007160
 1000 QQ=.5/SIG2                                                        007170
    5 M1 = M-1                                                          007180
      N1=N-1                                                            007190
      DO  10 I = 2,M1                                                   007200
      RHO2 = RHO(I)**2                                                  007210
      QTEST=.5*RHO2/SIG2                                                007220
      IF (QTEST .GT. 220.) 1020,1030                                    007230
 1020 QQ=0.                                                             007240
      GOTO 1040                                                         007250
 1030 QQ=QQ*EXP(-QTEST)                                                 007260
 1040 CONTINUE                                                          007270
      DO 10 J = 2,N1                                                    007280
   10 Q(I,J) = QQ                                                       007290
      RETURN                                                            007300
   20 WRITE(IPRINT,100)                                                 007310
  100 FORMAT(1H0,*SIG = 0. DEFAULT OPTION IS  Q = 0*)                   007320
      RETURN                                                            007330
      END                                                               007340
```

A.6 Subroutine SPLNI

```
      SUBROUTINE SPLNI(N,M,EPS,X,Y,T,SS,SS1,SS2,QUA)                    007350
C                                                                       007360
C     FIND THIRD ORDER SPLINE FCT  FOR A FCT Y(X) GIVEN AT THE          007370
C     POINTS (X(I),Y(I))                                                007380
C     FOLLOWING CHAP.9 OF VOL2 OF RALSTON + WILF                        007390
C                                                                       007400
C     N = NO.OF GIVEN DATA POINTS                                       007410
C     M = NO.OF SPECIFIED ARGUMENTS T(I) FOR WHICH THE SPLINE           007420
C       SS=ITS FIRST DER,SS1 AND SECOND DER,SS2 ARE TO BE COMPUTED      007430
C     EPS = ERROR TOLERANCE  IN ITERATIVE STEPS                         007440
C     X= ARRAY OF STRICTLY INCREASING ABSCISSAS                         007450
C     Y = ARRAY OF FCT VALUES                                           007460
C     T = ARRAY OF DESIRED ABSCISSAS                                    007470
C     SS = ARRAY OF SPLINE VALUES , SS1,SS2 DERIVATIVES                 007480
C     QUA = ARRAY OF VALUES OF INTEGRAL FROM X(1) TO X(N)               007490
C     LIMITATIONS N NOT LARGER THAN 50                                  007500
C                                                                       007510
      DIMENSION X(1),Y(1),T(1),SS(1),SS1(1),SS2(1),QUA(1)               007520
      DIMENSION H (82),H2(82),DELY(82),B(82),DELSY(82)                  007530
      DIMENSION S2(82),C(82),S3(82)                                     007540
      DATA OMEGA/1.0717968/                                             007550
      N1=N-1                                                            007560
    3 DO 51 I=1,N1                                                      007570
      H(I)=X(I+1)-X(I)                                                  007580
   51 DELY(I)=(Y(I+1)-Y(I))/H(I)                                        007590
    4 DO 52 I=2,N1                                                      007600
      H2(I)=H(I-1)+H(I)                                                 007610
      B(I)= .5*H(I-1)/ H2(I)                                            007620
      DELSY(I)=(DELY(I)-DELY(I-1))/H2(I)                                007630
      S2(I)=2.*DELSY(I)                                                 007640
   52 C(I)=3.*DELSY(I)                                                  007650
      S2(1)=0.                                                          007660
      S2(N)=0.                                                          007670
    5 ETA=0.                                                            007680
    6 DO 10 I=2,N1                                                      007690
    7 W=(C(I)-B(I)*S2(I-1)-(.5-B(I))*S2(I+1)-S2(I))*OMEGA               007700
    8 IF (ABS(W)-ETA) 10,10,9                                           007710
    9 ETA=ABS(W)                                                        007720
   10 S2(I)=S2(I)+W                                                     007730
```

```
      13   IF (ETA-EPS) 14,5,5                                              007740
      14   DO 53 I=1,N1                                                     007750
      53   S3(I)=(S2(I+1)-S2(I))/H(I)                                       007760
      15   DO 61 J=1,M                                                      007770
      16   I=1                                                              007780
      54   IF (T(J)-X(1)) 58,17,55                                          007790
      55   IF (T(J)-X(N)) 57,59,59                                          007800
      56    IF (T(J)-X(I)) 60,17,57                                         007810
      57   I=I+1                                                            007820
           GO TO 56                                                         007830
      58   WRITE (6,44) J                                                   007840
      44   FORMAT (I4,24HTH ARGUMENT OUT OF RANGE)                          007850
           GO TO 61                                                         007860
      59   I=N                                                              007870
      60   I=I-1                                                            007880
      17   HT1=T(J)-X(I)                                                    007890
           HT2=T(J)-X(I+1)                                                  007900
           PROD=HT1*HT2                                                     007910

           SS2(J)=S2(I)+HT1*S3(I)                                           007920
           DELSS=(S2(I)+S2(I+1)+SS2(J))/6.                                  007930
           SS(J)=Y(I)+HT1*DELY(I)+PROD*DELSS                                007940
           SSI(J)=DELY(I)+(HT1+HT2)*DELSS+PROD*S3(I)/6.                     007950
      61   CONTINUE                                                         007960
        20 QUA(1) = 0.0                                                     007970
           DO 62 I=1,N1                                                     007980
        62 QUA(I+1)=QUA(I)+.5*H(I)*(Y(I)+Y(I+1))-H(I)**3*(S2(I)+S2(I+1))/24. 007990
           RETURN                                                           008000
           END                                                              008010
```

A.7 Subroutine GETDATA

```
            SUBROUTINE GETDATA(DATAIN,NV,IIN,IOUT1,IOUT2,IIN1,ISIZE         008020
           +,ISIZET,INDIC)                                                  008030
C THE MAIN PURPOSE OF THIS SUBROUTINE IS TO INPUT CHARACTER STRING OR       008040
C NUMERICAL DATA IN A "CONVERSATIONAL" MODE I.E. FOR INPUTTING DATA         008050
C TO PROGRAMS BEING RUN UNDER INTERCOM.                                     008060
C IT ALSO MAY BE USED FOR BATCH PROCESSING-IN WHICH CASE THE DATA           008070
C SHOULD APPEAR 6 VALUES TO A CARD. DATA WHICH IS NOT TO BE CHANGED         008080
C SHOULD BE REPLACED BY BLANKS. FOR BATCH ALL OR SOME OF THE DATA MAY       008090
C BE DEFAULTED BY USING AN EOR AFTER THE LAST DATA TO BE INPUTTED.          008100
C THE SUBROUTINE ASSUMES THAT DEFAULT VALUES HAVE BEEN ASSIGNED             008110
C AND WILL PRINT OUT THESE DEFAULT VALUES BEFORE ASKING FOR DATA INPUT.     008120
C IT ASKS FOR NEW VALUES BY PRINTING OUT THE "NAMES" OF THE DATA AND THEN   008130
C SKIPPING A LINE. VALUES TO BE ASSIGNED TO THE NAMES SHOULD BE             008140
C ENTERED STARTING IN THE SAME COLUMN AS THE START OF THE NAME.             008150
C EACH DATUM IS ASSIGNED 10 COLUMNS AND UP TO 6 ITEMS MAY BE INPUTTED       008160
C IN A SINGLES ROW.                                                         008170
C ARGUMENTS**********************                                       008180
C DATAIN  (DIMENSION (NV,3) WHERE NV IS THE TOTAL # OF DATA                 008190
C    TO BE INPUTTED)                                                        008200
C DATAIN HOLDS THE FOLLOWING INFORMATION ABOUT EACH DATUM-                  008210
C NAME,VALUE,CODE WHERE-                                                    008220
C NAME => NAME BY WHICH THE DATUM IS IDENTIFIED TO THE USER (IT MAY OR      008230
C MAY NOT BE EQUAL TO THE FORTRAN VARIABLE NAME TO BE ASSIGNED TO THE       008240
C DATUM.)                                                                   008250
C VALUE => NUMERICAL OR CHARACTER STRING VALUE TO BE ASSIGNED (THE DATUM)   008260
C CODE => HOW THE DATUM IS TO BE INTERPRETED                                008270
C   -1 => CHARACTER STRING                                                  008280
C    0 => INTEGER                                                           008290
C    1 => FLOATING POINT NUMBER                                             008300
C NV TOTAL NUMBER OF DATUM TO BE INPUTTED                                   008310
C IIN FILE NO. FOR INPUTTING                                                008320
C IOUT1 PRIMARY OUTPUT FILE                                                 008330
C IOUT2 SECONDARY OUTPUT FILE                                               008340
C ISIZE =>  SIZE OF FIRST DIMENSION OF DATAIN                               008350
```

```
      DIMENSION DATAIN(ISIZE),IA(6)                                     000360
      COMMON/SENSE/IINNN,IOUTNN,INDICC                                  000370
      INTEGER DATAIN,F                                                  000380
      EXTERNAL SSWTCH                                                   000390
      CALL ERRSET(KOUNT,20000)                                          000400
      KOUNT1=KOUNT                                                      000410
      IF(INDIC.NE.0) 200,210                                            000420
  200 ISW=2                                                             000430
      LL=0                                                              000440
      L=INDIC-1                                                         000450
      GOTO 1055                                                         000460
  210 CONTINUE                                                          000470
      ISW=1                                                             000480
      IOUL=00000000000000000053B                                        000490
      IINNN=IIN                                                         000500
      IOUTNN=IOUT1                                                      000510
      IOUTT=IOUT1                                                       000520
      ISW=1                                                             000530
      IBLANK=1RH                                                        000540
      CALL SSWTCH(IIN1,ISW3,10HREAD DATA ,5HFILE-),RETURNS(1060)        000550
      IF (ISW3 .EQ. 1) 1300,1290                                        000560
 1300 WRITE (IOUT1,17) IIN1                                             000570
      REWIND IIN1                                                       000580
      READ(IIN1) DATAIN                                                 000590
      REWIND IIN1                                                       000600
      IF (EOF(IIN1)) 1400,1290                                          000610
 1400 WRITE(IOUT1,24) IIN1                                              000620
 1290 CONTINUE                                                          000630
      CALL SSWTCH(0,ISW4,10HDEFAULTS L,5HISTED),RETURNS(1060)           000640
      IF (ISW4 .NE. 1) GOTO 1150                                        000650
      WRITE(IOUT1,1)                                                    000660
 1140 DO 110 I=1,NV                                                     000670
      II=ISIZE+I                                                        000680
      III=I                                                             000690
      II2=II+ISIZE                                                      000700
      IF (DATAIN(II2)) 1020,1030,1040                                   000710
 1020 WRITE(IOUTT,2) DATAIN(II),DATAIN(III)                             000720
      GOTO 110                                                          000730
 1030 WRITE (IOUTT,3) DATAIN(II),DATAIN(III)                            000740
      GOTO 110                                                          000750
 1040 WRITE(IOUTT,4) DATAIN(II),DATAIN(III)                             000760
  110 CONTINUE                                                          000770
      GOTO (1150,1130) ISW                                              000780
 1150 CALL SSWTCH(0,ISW5,10HNAME-VALUE,5H MODE),RETURNS(1060)           000790
      IF (ISW5 .EQ. 1) 1270,1050                                        000800
 1050 L=1                                                               000810
      ISW=2                                                             000820
      LL=0                                                              000830
 1055 L=L+LL                                                            000840
      IF (L .GT. NV) GOTO 1060                                          000850
 1310 WRITE (IOUT1,18) DATAIN(ISIZE+L)                                  000860
      LL=1                                                              000870
      DO 100 J=1,6                                                      000880
      IA(J)=1RH                                                         000890
  100 CONTINUE                                                          000900
      READ(IIN,10) (IA(J), I=1,6)                                       000910
      IF (EOF(IIN)) 1320,1070                                           000920
 1320 INDICC=1                                                          000930
      GOTO 1060                                                         000940
 1070 IF (IA(1) .EQ. IBLANK) GOTO 1055                                  000950
      DO 180 J=1,6                                                      000960
      DO 180 K=1,10                                                     000970
      IF (MXGETX(IA(J),K,1) .EQ. IOUL) GOTO 1270                        000980
  180 CONTINUE                                                          000990
      DO 190 J=1,6                                                      000000
      JR=L+J-1                                                          000010
      F=DATAIN(JR+2*ISIZE)                                              000020
      IF (F) 1090,1100,1110                                             000030
 1090 IF(IA(J) .NE. IBLANK) DECODE(10,11,IA(J)) DATAIN(JR)              000040
      GOTO 1080                                                         000050
 1100 CALL RJUST(IA(J))                                                 000055
      IF (IA(J) .NE. IBLANK) DECODE(10,12,IA(J)) DATAIN(JR)             000060
      GOTO 1080                                                         000070
 1110 CALL RJUST(IA(J))                                                 000075
      IF (IA(J) .NE. IBLANK) DECODE(10,13,IA(J)) DATAIN(JR)             000080
 1080 IF (IA(J) .EQ. IBLANK) GOTO 1081                                  000090
  190 CONTINUE                                                          000100
 1081 LL=J-1                                                            000110
      IF(INDIC.NE.0) LL=1000                                            000120
      IF (KOUNT .EQ. KOUNT1) GOTO 1055                                  000130
```

```
115         KOUNT1=KOUNT                                                      009140
            WRITE(IOUT1,25)                                                   009150
1270        WRITE(IOUT1,23)                                                   009160
1250        WRITE(IOUT1,8)                                                    009170
            DO 150 I=1,6                                                      009180
120         IA(I)=10H                                                         009190
150         CONTINUE                                                          009200
            READ(IIN,10) (IA(I),I=1,6)                                        009210
            IF (EOF(IIN)) 1330,1085                                           009220
1330        INDICC=1                                                          009230
            GOTO 1060                                                         009240
125         II=IA(1)                                                          009250
1085        IF (II .EQ. IBLANK) GOTO 1060                                     009260
            DO 130 I=1,NV                                                     009270
            J=I*ISIZE                                                         009280
            IF (II .EQ. DATAIN(J)) GOTO 1160                                  009290
130         CONTINUE                                                          009300
            WRITE(IOUT1,16)                                                   009310
            GOTO 1270                                                         009320
1160        F=DATAIN(J+ISIZE)                                                 009330
            JJ=J-ISIZE                                                        009340
135         IF (F) 1170,1180,1190                                             009350
1170        DO 160 I=2,6                                                      009360
            IF (IA(I) .EQ. IBLANK .AND. I.GT.2) GOTO 1240                     009370
            DECODE(10,11,IA(I)) DATAIN(JJ+I-2)                                009380
140         CONTINUE                                                          009390
160         GOTO 1240                                                         009400
1180        CALL RJUST(IA(2))                                                 009405
            DECODE(10,12,IA(2)) DATAIN(JJ)                                    009410
            GOTO 1240                                                         009420
145         CALL RJUST(IA(2))                                                 009425
1190        DECODE(10,13,IA(2)) DATAIN(JJ)                                    009430
1240        IF (KOUNT .EQ. KOUNT1) GOTO (1250,1310) ISW                       009440
            KOUNT1=KOUNT                                                      009450
            WRITE(IOUT1,25)                                                   009460
            GOTO 1250                                                         009470
150         WRITE (IOUT1,14)                                                  009480
1060        IF (IOUT2 .EQ. 0) GOTO 1130                                       009490
            WRITE (IOUT2,15)                                                  009500
            IOUT=IOUT2                                                        009510
            ISN=2                                                             009520
155         GOTO 1140                                                         009530
1130        INDIC=INDICC                                                      009540
            RETURN                                                            009550
            RETURN                                                            009560
1           FORMAT(/,* THE DEFAULT INPUT DATA ARE*)                           009570
160         FORMAT(1X,A10,*=*,A10,*$*)                                        009580
3           FORMAT(1X,A10,*=*,I10)                                            009590
4           FORMAT(1X,A10,*=*,G16.6)                                          009600
5           FORMAT(///)                                                       009610
6           FORMAT(/,* ENTER DATA. START IN COL. BENEATH START OF NAME*)      009620
165         FORMAT(8X,6A10)                                                   009630
8           FORMAT(8X)                                                        009640
10          FORMAT(6A10)                                                      009650
11          FORMAT(A10)                                                       009660
12          FORMAT(I10)                                                       009670
170         FORMAT(E10.0)                                                     009680
14          FORMAT(/,* DATA INPUT COMPLETE*)                                  009690
15          FORMAT(/,* THE INPUT DATA VALUES ARE*,/)                          009700
16          FORMAT(1X,* TRY AGAIN*)                                           009710
17          FORMAT(/,1X,* READING DATA FROM *,I5)                             009720
175         FORMAT(1X,A10,*-*)                                                009730
23          FORMAT(8X,*NAME       VALUE.....*)                                009740
24          FORMAT(1X,* FILE*,I5,* IS EMPTY*)                                 009750
25          FORMAT(1X,* WRONG DATA TYPE-TRY AGAIN*)                           009760
            END
```

A.8 Subroutine SSWTCH

```
      SUBROUTINE SSWTCH(I,J,M1,M2),RETURNS(M)                          009770
      COMMON/SENSE/IIN,IOUT,INDIC                                      009780
      IF (I .EQ. 0) GOTO 100                                           009790
      WRITE(IOUT,1) M1,M2,I                                            009800
      GOTO 110                                                         009810
  100 WRITE(IOUT,2) M1,M2                                              009820
  110 READ(IIN,3) JJ                                                   009830
      J=2                                                              009840
      IF(JJ .EQ. 1HY) J=1                                              009850
      INDIC=EOF(IIN)                                                   009860
      IF (INDIC) 1000,1010                                             009870
 1000 RETURN M                                                         009880
 1010 RETURN                                                           009890
    1 FORMAT(1X,A10,A5,I2,*?-*)                                        009900
    2 FORMAT(1X,A10,A5,*?-*)                                           009910
    3 FORMAT(A1)                                                       009920
      END                                                              009930
```

A.9 Subroutine RJUST

```
      SUBROUTINE RJUST(L)                                              009940
      DIMENSION LC(9)                                                  009950
      DATA (LC=7755555555555555558,777755555555555555558,              009960
     +77777755555555555558,7777777555555555558,                        009970
     +777777777755555555558,777777777775555555558,                    009980
     +7777777777777775555558,77777777777777755558,                    009990
     +77777777777777777558)                                           010000
      LR=1                                                             010010
      IBITS=0                                                          010020
      DO 100 I=1,9                                                     010030
      IBITS=IBITS+6                                                    010040
      LR=MASK(IBITS).OR.LR                                             010050
      IF (LR.EQ.LC(I)) GOTO 110                                        010060
  100 CONTINUE                                                         010070
      GOTO 120                                                         010080
  110 L=SHIFT(L,IBITS)                                                 010090
  120 RETURN                                                           010100
      END                                                              010110
```

Appendix B

Fortran Listings for TIKIRK Program, Options No. 1 and No. 2 Only

B.1 Main Program TIKIRK

```
      PROGRAM TIKIRK(TAPE4=80/80,TAPE5=80/80,TAPE3,TAPE7,TAPE8,      000100
     +TAPE6=80/80,OUTPUT=80)                                         000110
C THIS PROGRAM CAN BEST BE DESCRIBED AS THE I/O INTERFACE FOR FUNCTION 000120
C SUBROUTINE IKIRK WHICH COMPUTES THE KIRKHOFF INTENSITY FUNCTION AS  000130
C DESCRIBED IN AFCRL-72-0565.                                         000140
C    THE INPUT FALLS INTO THREE CLASSES%                              000150
C  1) INPUT HAVING TO DO WITH PROPERTIES OF THE WINDOW MATERIAL AND   000160
C THE LASER BEAM,NAMELY% (QUANTITIES CGS UNLESS OTHERWISE INDICATED)  000170
C SIG     => VALUE OF SIGMA IN GAUSSIAN BEAM                          000180
C LAMBDA  => WAVELENGTH OF THE LIGHT BEAM IN MICRONS                  000190
C P       => TOTAL BEAM POWER                                         000200
C Q       => WINDOW RADIUS                                            000210
C BETA    => BULK ABSORPTION COEFFICIENT                              000220
C K       => THERMAL CONDUCTIVITY                                     000230
C NX      => INDEX OF REFRACTION                                      000240
C S1R     => S SUB-1,SUB-RHO                                          000250
C S1T     => S SUB-1,SUB-THETA                                        000260
C S2R     => S SUB-2,SUB-RHO                                          000270
C S2T     => S SUB-2,SUB-THETA                                        000280
C T       => TIME AT WHICH IKIRK IS TO BE EVALUATED                   000290
C  2) INPUT HAVING TO DO WITH THE EVALUATION DOMAIN OF THE FUNCTION   000300
C IKIRK,NAMELY%                                                       000310
C      X0   => GAUSSIAN FOCAL DISTANCE (METERS)                       000320
C      X1   => MINIMUM X-VALUE FOR FUNCTION EVALUATION (METERS)       000330
C      X2   => MAXIMUM X-VALUE FOR FUNCTION EVALUATION (METERS)       000340
C      RHOP1  => MINIMUM RADIUS VALUE FOR FUNCTION EVALUATION         000350
C      RHOP2  => MAXIMUM RADIUS VALUE FOR FUNCTION EVALUATION         000360
C      MP   => NUMBER OF EVALUATION POINTS IN THE RADIAL DIRECTION    000370
C      NP   => NUMBER OF EVALUATION POINTS IN THE AXIAL (X) DIRECTION 000380
C      TIM  => ARRAY (UP TO 10) OF TIME VALUES FOR FUNCTION EVALUATION 000390
C         (TIME VALUES SHOULD BE IN INCREASING SEQUENCE)              000400
C      UMIN  => MINIMUM U-VALUE FOR FUNCTION EVALUATION (SEE MODE)    000410
C      UMAX  => MAXIMUM U-VALUE FOR FUNCTION EVALUATION               000420
C      VMIN  => MINIMUM V-VALUE FOR FUNCTION EVALUATION               000430
C      VMAX  => MAXIMUM V-VALUE FOR FUNCTION EVALUATION               000440
C   3) INPUT HAVING TO DO WITH PROGRAM CONTROL,NAMELY%                 000450
C      EPS1  => ERROR VALUE FOR INTERPOLATION OF THE TEMPERATURE      000460
C FUNCTION OUTPUTTED BY TEMPS AND INTERPOLATED BY IBM SCI. SUB. ALI.  000470
C      MINT  => NUMBER OF TEMPERATURE FUNCTION POINTS TO BE USED IN   000480
C THE INTERPOLATION (DEFAULT=6)                                       000490
C      IPRNT => USED TO CONTROL DEBUG OUTPUT (1 CAUSES DEBUG OUTPUT)  000500
C               (2 CAUSES WINDOW TEMPERATURE DISTRIBUTION SUITABLE FOR 000510
C               DISPLAY TO BE OUTPUT)                                 000520
C      NGAUS => NUMBER OF FUNCTION VALUES FOR GAUSSIAN INTEGRATION    000530
C      MODE  => IF MODE=1 THEN THE INTENSITY FUNCTION IS EVALUATED AT 000540
C EQUI-SPACED X AND RHOP-PRIME VALUES; IF MODE=2 IT IS EVALUATED AT   000550
C EQUI-SPACED U AND V VALUES.                                         000560
C      I2      => IF 1 USE IKIRK, IF 2 USE IKIRKP                     000570
C               (NOTE THAT IKIRKP SHOULD ONLY BE USED ON THE AXES     000580
C               FOR CONSTANT TEMPERATURE WINDOW)                      000590
C ALL THE ABOVE MENTIONED DATA IS OBTAINED BY TWO CALLS TO THE        000600
C INTERACTIVE INPUT SUBROUTINE GETDATA DESCRIBED IN PML TM-16. IN THE  000610
C FIRST CALL ALL DATA IN THE FIRST CATEGORY IS OBTAINED. IN THE       000620
C SECOND CALL ALL DATA IN  THE SECOND AND THIRD CATEGORIES ARE        000630
C OBTAINED.  AN EXCEPTION TO THIS IS I2 (CONTROLS USE OF              000640
C IKIRK AND IKIRKP) WHICH IS OBTAINED ON THE FIRST CALL TO GETDATA.   000650
C  FOR A LISTING OF DEFAULT INPUT DATA IT IS RECOMMENDED THAT         000660
C IIKIRK BE RUN INTERACTIVELY UNDER INTERCOM AFTER GIVING THE COMMAND  000670
C CONNECT(TAPE4,TAPE5).                                               000680
C THE MAIN OUTPUT OF IIKIRK IS A SEQUENCE OF UNFORMATTED RECORDS      000690
C OF INTENSITY VALUES WITH CORRESPONDING DOMAIN VALUES. EACH RECORD   000700
C CONSISTS OF THE FOLLOWING SEQUENCE OF VALUES%                       000710
C RECORD NO., NUMBER(MP) OF INTENSITY VALUES IN THE AXIAL DIRECTION,  000720
C AXIAL COORDINATE X OR U, TIME VALUE (T) IN SECONDS, NUMBER (NP)     000730
C OF INTENSITY VALUES IN THE RADIAL DIRECTION, MINIMUM RADIAL         000740
C COORDINATE RHOP1 OR VMIN, MAXIMUM RADIAL COORDINATE RHOP2 OR        000750
C VMAX, MP INTENSITY VALUES.                                          000760
C FOR EACH VALUE OF T, NP RECORDS ARE OUTPUTTED CORRESPONDING TO THE  000770
C NP X EVALUATION POINTS. THE RECORD NUMBER RUNS FROM 1 TO NP FOR     000780
C EACH TIME VALUE.                                                    000790
C THERE ARE SIX FILES ASSOCIATED WITH THIS PROGRAM (NOT INCLUDING FILE 000800
C @OUTPUT@). THE FILES ARE REFERRED TO IN THE PROGRAM AND ASSOCIATED   000810
C SUBROUTINES AS IT3,IT4,IT5,IT6,IT7,IT8. THE FILE VALUES ARE IN TURN 000820
C ASSIGNED TO THE USUAL FORTRAN @TAPEN@ BY A DATA STATEMENT AND PROGRAM 000830
```

```
C ASSOCIATION IT IS NECESSARY TO CHANGE  EITHER OR BOTH THE DATA AND      000850
C PROGRAM STATEMENTS.                                                    000860
C THESE FILES SERVE THE FOLLOWING PURPOSES%                              000870
C    IT3 => FILE OUTPUTTED BY PROGRAM TEMP5                              000880
C    IT4 => @INTERACTIVE@ INPUT FILE (SEE GETDATA)                       000890
80   C    IT5 => @INTERACTIVE@ OUTPUT FILE (SEE GETDATA)                 000900
C    IT6 => LISTING OF ALL INPUT PARAMETERS AND DEBUG OUTPUT             000910
C    IT7 => UNFORMATTED INTENSITY VALUES. ALSO MAY BE USED TO INSERT     000920
C    IT8 => UNFORMATTED TEMPERATURE DISTRIBUTION VALUES                  000930
C           SUITABLE FOR DISPLAY PURPOSES                                000940
85   C        PREASSIGNED DATA IN CATEGORIES 2 AND 3                     000950
C IN ADDITION TO THE ABOVE FACTS, THE USER SHOULD BE AWARE OF TWO        000960
C PROGRAM @CONSTANTS@, THE FIRST UNDER THE VARIABLE NAME NT IS THE NUM-  000970
C BER OF TIME VALUES PERMITTED. AT PRESENT THIS IS SET TO 10( (THE       000980
90   C DIMENSION OF THE TIME ARRAY TIM). ALSO NOTE THAT ALL THE TIME      000990
C DEFAULT VALUES ARE ZERO EXCEPT THE FIRST AND THAT THE PROGRAM STOPS    001000
C AS SOON AS A SUCCEEDING TIME VALUE IS LESS THAN THE PRECEEDING         001010
C TIME VALUE.                                                            001020
C THE SECOND @CONSTANT@ HAS TO DO WITH THE SIZE OF THE RECORD OUTPUTTED  001030
95   C BY TEMP5. THE SUBROUTINE RTAPE3 READS THE TEMP5 OUTPUT UNDER THE  001040
C ASSUMPTION THAT ALL @RADIAL@ ARRAYS ARE OF DIMENSION 82 AND @AXIAL@    001050
C ARRAYS ARE OF DIMENSION 22.SEE COMMENTS WITHIN SUBROUTINE RTAPE3.      001060
C THE INTENSITY FUNCTION IKIRK IS DEFINED EXPLICITLY AS A FUNCTION OF    001070
C THE NONDIMENSIONAL VARIABLES U AND V AND IMPLICITLY AS A FUNCTION      001080
100  C OF NON-DIMENSIONAL TIME TAU THROUGH THE TIME DEPENDANT FUNCTIONS   001090
C PHI-THETA AND PHI-RHO AS DEFINED IN THE ABOVE REFERENCE. THESE         001100
C VARIABLES ARE PASSED THROUGH AN ARGUMENT LIST. ALL @ARGUMENTS@         001110
C REQUIRED FOR EVALUATION OF IKIRK ARE PASSED THROUGH BLOCK COMMON       001120
C .PHIBLK. THESE PARAMETERS ARE%                                         001130
105  C    CS1R   => C*S1R (SEE IKIRK COMMENTS)                           001140
C    CS2R   => C*S2R          "                                         001150
C    CS1I   => C*S1I          "                                         001160
C    CS2I   => C*S2I          "                                         001170
C    XS     => STARTING ARGUMENT FOR FUNCTIONS F1,F2 (SEE FUNCTION      001180
110  C              PHI COMMENTS)                                        001190
C    DX     => INTERVAL BETWEEN EQUI-SPACED ARGUMENTS OF F1,F2.         001200
C    NF     => NUMBER OF VALUES OF F1,F2                                001210
C    MNNT   => MINT (SEE INPUT DATA)                                    001220
C    EPP    => EPS1          "                                         001230
115  C    F1(200)=> HOLDS VALUES OF F1 FROM TEMP5                        001240
C    F2(200)=> HOLDS VALUES OF F2 FROM TEMP5                            001250
C    HOLD   => STORES PHI-THETA (PHI-RHO AND PHI-THETA ARE EVALUATED    001260
C                   SIMULTANEOUSLY)                                     001270
C    A      => 1/SQRT(2)/STD (=ALPHA IN THE ABOVE REFERENCE)            001280
120  C    KE     => WAVE NUMBER                                          001290
C    TLAST  => STORES TIME VALUE READ FROM TEMP5 RECORD                 001300
C    TNEXT  =>          "                                               001310
C    IERR   => ERROR INDICATOR FOR RTAPE3 (INDICATES OUT OF RANGE       001320
C              TIME OR OUT OF SEQUENCE TIME)                            001330
C                                                                       001340
125  C    IP     => DEBUG OUTPUT @SWITCH@                                001350
C    MP1    => =NF                                                     001360
C    ISW    => SWITCH FOR GAUSSIAN INTEGRATION. WHEN ISW=1 THEN THE     001370
C         X-VALUES FOR GAUSSIAN INTEGRATION ARE FOUND.                  001380
C    NGAUSS => NUMBER OF POINTS USED IN THE GAUSSIAN INTEGRATION        001390
130  C         (NOTE THAT IF NGAUSS IS CHANGED THEN THE GAUSSIAN INTEGRATION 001400
C         SUBROUTINE MUST ALSO BE CHANGED.)                             001401
C    RAD    => WINDOW RADIUS                                           001402
C    NP1    => NUMBER OF TEMPERATURE SAMPLES IN AXIAL DIRECTION        001403
C              (USED FOR OUTPUTTING DISPLAY COMPATIBLE                  001404
135  C              TEMPERATURE DATA)                                    001405
C    C3     => CONSTANT TO DIMENSIONALIZE TEMPERATURE DATA             001406
C              FOR DISPLAY                                              001407
C    C1     => CONSTANT USED TO DIMENSIONALIZE TIME FOR                001410
      REAL K,IKIRK,LAMBDA,NX,KE                                         001420
140   REAL IKIRKP                                                       001430
      LOGICAL S                                                         001440
      DIMENSION BUF(100)                                                001450
      COMMON/IFILES/IT3,IT4,IT5,IT6,IT7,IT8                             001460
      COMMON/PHIBLK/CS1R,CS2R,CS1I,CS2I,XS,DX,NF,MNNT,EPP,F1(200),      001470
     *F2(200),HOLD,A,KE,TLAST,TNEXT,IERR,IP,MP1,ISW,NGAUSS,             001480
145   *RAD,NP1,C3,C1                                                    001490
      INTEGER DATAIN(100,3),DATAIN1(100,3)                              001500
      COMMON/BLOCK2/X1,X1,X2,RHOP1,RHOP2,MP,NP,TIM(10),EPS1             001508
C              TEMPERATURE DISPLAY
```

```
      +,MINT,IPRNT,NGAUS,MODE,UMIN,UMAX,VMIN,VMAX,DUM(20),MSKIP,NSKIP
      COMMON/BLOCK1/
     +       I1,I2,I3,I4,I5,I6,I7,M,N,MI,NI,ICNT,IU,IQ,N0,NMX,TRUN,
     +ICARD,IPRINT,IPNCH,ITAP3,ITAP4,RHO1,RHO12,ZED1,ZED12,DTAUU,
     +TAUMX,TAUOFF,SIG  ,UU,UU,     EPS,GI(4),MI(4),MATER,NX,BETA,
     +K,LAMBDA,S1R,S1T,S2R,S2T,
     +DEN,CP,R,EXPER,PW,R1,Z1,R2,IPLOT,PROBNO,TICU,XLEN,YLEN,SCALEX,
     +SCALEY1,SCALEY2,XTITLE(5),YTITLE1(5),YTITLE2(5),NAME
      EQUIVALENCE (I1,DATAIN),(X0,DATAIN1)
      DATA (DATAIN(I,1),I=1,92)/7*2,80,20,3*1,0,1,2,11,100,4,6.6,3,
     +4,0,1,.-.5546,1.1092,.0035,5 ,5,.1292,2*0,..001,4*0,.0,.3*.015,
     +3HKCL,1.47,4.8E-4,.0453,10,6,-.34E-5,.05E-5,.1E-5,-.1E-5,1.98,
     +,.69,1.258,1H1,24,7,81,11,1,1,4H7204,.5,20,,9,,30,,,1,,1.
     +10HTIME(SECON3HDS),3*1H ,10H1EMP-DEGC ,10H ABOVE AMB,
     +3*1H ,10HMEAN TEMP ,10H ABOVE AMB,3*1H ,7HBARRETT,
     +10HRADIAL DIS,10HTANCE,RHO-,4H(CM),1H ,1H ,10HAXIAL DIST.
     +10HANCE,7-(CM),1H, ,1H /
      DATA (DATAIN(I,2),I=1,92)/2HI1,2HI2,2HI3,2HI4,2HI5,
     +2HI6,2HI7,1HM,1HN,2HMF,2HNI,4HICNT,2HIU,2HIQ,2HN0,3HNMX,
     +4HIRUN,5HICARD,6HIPRINT,5HIPNCH,5HITAP3,5HITAP4,4HRHO1,
     +5HRHO12,4HZED1,5HZED12,6HDTAUU,5HTAUMX,6HTAUOFF,3HSIG,
     +2HUU,2HUU,3HEPS,5HGI(1),5HGI(2),5HGI(3),5HGI(4),5HMI(1),
     +5HMI(2),5HMI(3),5HMI(4),8HMATERIAL,8HREF.IND.,4HBETA,
     +9HTHER.COND,6HLAMBDA,3HS1R,3HS1T,3HS2R,3HS2T,
     +          7HDENSITY,9HSPEC.HEAT,6HRADIUS,5HEXPER,
     +3HPWR      ,3HR1,,3H71,,3HR2,,3HP1,T?1Y,2N,6HPROBNO,4HTICU,
     +4HXLEN,4HYLEN,7HX-SCALE,8HY1-SCALE,8HY2-SCALE,6HXTITLE,
     +1H2,1H3,1H4,1H5,7HYTITLE1,1H2,1H3,1H4,1H5,7HYTITLE2,1H2,
     +1H3,1H4,1H5,8HOPERATOR,
     +3HXT1,3HXT2,3HXT3,3HXT4,3HXT5,3HYT1,3HYT2,3HYT3,3HYT4,
     +3HYT5/
      DATA (DATAIN(I,3),I=1,92)/22*0,19*1,-1,11*1,-1,1,4*0,
     +-1,6*1,24*-1/
      DATA (DATAIN1(I,1),I=1,48)/1500,,1000,,2000,,0,,2,,100,100,10,,
     +9*-1,,.001,6,1,24,2,-40,,40,,0,,10,,
     +10HKIRKHOFF I,10HNTENSITY F,9HUNCTION,I,1H ,1H ,10HTIME IN SE,
     +5HCONDS,1H ,1H ,1H ,10HNON-DIMENS,10HIONAL RADI,10HAL DISTANC,
     +3HE,V,1H ,10HNON-DIMENS,10HIONAL AXIA,10HL DISTANCE,2H,U,1H ,5,5/
      DATA (DATAIN1(I,2),I=1,48)/2HX0,2HX1,2HX2,5HRHOP1,5HRHOP2,2HMP,
     +2HNO,2HT1,2HT2,2HT3,2HT4,2HT5,2HT6,2HT7,2HT8,2HT9,3HT10,
     +4HEPS1,4HMINT,5HIPRNT,5HNGAUS,4HMODE,4HUMIN,4HUMAX,
     +4HVMIN,4HVMAX,3HST1,3HST2,3HST3,3HST4,3HST5,3HPT1,3HP12,
     +3HP13,3HPT4,3HPT5,3HXT1,3HXT2,3HXT3,3HXT4,3HXT5,3HYT1,
     +3HYT2,3HYT3,3HYT4,3HYT5,5HMSKIP,5HNSKIP/
      DATA (DATAIN1(I,3),I=1,48)/5* ,2*0.11*1,4*0,4*1,20*-1,2*0/
      DATA IT3,IT4,IT5,IT6,IT7,IT8/3,4,5,6,7,8/
      INDIC=0
      S=.F.
      ISW=1
      NT=10
      CALL GETDATA(DATAIN,92,4,5,6,3,100,300,INDIC)
      CALL GETDATA(DATAIN1,48,4,5,6,7,100,300,INDIC)
      IF (MODE .NE. 1) GOTO 150
      DATAIN1(37,1)=10HRADIAL DIS
      DATAIN1(38,1)=10HTANCE,RHO-
      DATAIN1(39,1)=10HRIME (CM)
      DATAIN1(40,1)=10H
      DATAIN1(42,1)=10HAXIAL DIST
      DATAIN1(43,1)=10HANCE,X (CM
      DATAIN1(44,1)=1H)
150   WRITE (IT7) DATAIN
      WRITE (IT7) DATAIN1
      IF (MODE .EQ. 1) S=.T.
      REWIND IT3
      READ(IT3)
      TLAST=TNEXT=0.
      KF=6.28318E4/LAMBDA
      EPP=EPS1
      IP=IPRNT
      NGAUSS=NGAUS
      XS=0.
      DX=1./M
      MMNT=MINT
      MP1=M+1
      NF=MP1
      IF (S) GOTO 120
      XMAX=VMAX
      XMIN=VMIN
      GOTO 140
```

```
      120 XMIN=RHOP1                                                    000310
          XMAX=RHOP2                                                    000320
      230                                                               000330
      140 CONTINUE                                                      000340
          A=.5/SIG                                                      000350
          C=PI*(NX**2+1)/2./NX                                          000360
          C=C*R*BETA/3.14159/K                                          000370
      235                                                               000380
          CS1P=C*S1R                                                    000390
          CS2P=C*S2R                                                    000400
          CS1T=C*S1T                                                    000410
          CS2T=C*S2T                                                    000420
          RSQ=R*R                                                       000430
      240                                                               000440
          X0=X0*100.                                                    000450
          X1=X1*100.                                                    000460
          X2=X2*100.                                                    000470
          C1=K/RSQ/DEN/CP                                               000480
          DELX=(X2-X1)/(AMAX0(1,NP-1))                                  000490
      245                                                               000500
          DELRHO=(RHOP2-RHOP1)/(AMAX0(1,MP-1))                          000510
          DELV=(VMAX-VMIN)/(AMAX0(1,MP-1))                              000520
          DELU=(UMAX-UMIN)/(AMAX0(1,NP-1))                              000530
          RAD=R                                                         000540
          NP1=N+1                                                       000550
      250                                                               000560
          C3=C/R                                                        000570
          X0I=1./X0                                                     000580
          NREC=0                                                        000590
          IT=1                                                          000600
          TR=T=TIM(1)                                                   000610
      255                                                               000620
          IF (T .LT. 0.) GOTO 2000                                      000630
          IF(IP.EQ.2)  WRITE(IT8) DATAIN                                000640
      1000 T=T+C1                                                       000650
          WRITE(IT5,2) T                                                000655
          IF (T .GT. TAUMX) GOTO 2000                                   000660
      260                                                               000670
          X1T=X1                                                        000680
          X1I=1./X1T                                                    000690
          U=UMIN                                                        000700
          DO 100 I=1,NP                                                 000710
          CX=(X0*X1I)**2                                                000720
      265                                                               000730
          IF (S) U=KE*RSQ*(X0I-X1I)                                     000740
          RHOP=RHOP1                                                    000750
          V=VMIN                                                        000760
          DO 110 J=1,MP                                                 000770
          IF (S) V=KE*R*X1I*RHOP                                        000774
      270                                                               000776
          GOTO (220,200) I2                                             000780
      220 BUF(J)=CX*IKIRKP(U,V,T)                                       000790
          GOTO 210                                                      000800
      200 BUF(J)=CX*IKIRK(U,V,T)                                        000810
      210 RHOP=RHOP+DELRHO                                              000820
      275                                                               000830
          V=V+DELV                                                      000840
      110 CONTINUE                                                      000850
          P=U                                                           000860
          IF(S) P=X1T                                                   000870
          WRITE (IT7) I,NP,P,TR,MP,XMIN,XMAX,(BUF(J),J=1,MP)            000880
      280                                                               000890
          CALL PRT(BUF,MODE,I,IT,MP,NP,MSKIP,NSKIP,RHOP1,               000900
         +DELRHO,VMIN,DELV,X1T,U,TR,T,IT6)                              000910
          NREC=NREC+1                                                   000920
          X1T=X1T+DELX                                                  000930
          X1I=1./X1T                                                    000940
      285                                                               000950
          U=U+DELU                                                      000960
      100 CONTINUE                                                      000970
          IT=IT+1                                                       000980
          IF (IT .GT. NT) GOTO 2000                                     000990
          TR=T=TIM(IT)                                                  001000
      290                                                               001010
          IF (T .GT. TIM(IT-1)) GOTO 1000                               001020
      2000 WRITE(IT5,1) NREC                                            001030
      1   FORMAT(1X,* THE NUMBER OF RECORDS IS=*,I10)                   001040
      2   FORMAT(1X,* NEW VALUE OF TAU IS*,E13.5)                       001050
      3   FORMAT(1X,I6,4E12.4)                                          001060
      295                                                               001070
      4   FORMAT(1X,4E13.5)                                             001080
        5 FORMAT(1X,I6,4G12.4,3(/,5X,5G13.5))                           001090
          END                                                           001100
```

B.2 Function IKIRK

```
      REAL FUNCTION IKIRK(U,V,T)                                        000290
C FUNCTION IKIRK IS THE KIRKHOFF INTENSITY FUNCTION DESCRIBED IN        000290
C -BENDOW,B. AND GIANINO,P. @OPTICAL PERFORMANCE EVALUATION OF          000300
C INFRARED TRANSMITTING WINDOWS@ AFCRL-72-0565. ASSUMING A GAUSSIAN     000310
C SHAPED UNPOLARIZED SOURCE. THE INTENSITY FUNCTION CAN BE WRITTEN%     000320
C  IKIRK(U,V)=2(A!2/(1-EXP(-A!2)))!2*(I(0,1,DX)(FW*FX)!!2+             000330
C                    I(0,1,DX)(FW*FY)!!2                               000340
C WHERE%                                                                000350
C     FW(X,U)=EXP(-(A*X)!2)*EXP(-I*U*X!2/2)                             000360
C     FX(X,V)=X*J0(X*V)*EXP(I*K*PHIR(X))-FZ(X,V)                        000370
C     FY(X,V)=X*J0(X*V)*EXP(I*K*PHIT(X))+FZ(X,V)                        000380
C     FZ(X,V)=J1(X*V)*(EXP(I*K*PHIR(X))-EXP(I*K*PHIT(X)))/(V)           000390
C     A=1/SQRT(2)/SIG**2                                                000400
C     K=WAVE NO. (OMEGA/C)                                              000410
C   NOTATION%                                                           000420
C     ! => EXPONENTIATION                                               000430
C     I => @IMAGINARY@                                                  000440
C     I(0,1,DX)(.) MEANS INTEGRATION  OF THE FUNCTION WITHIN () W.R.T.X 000450
C     OVER THE INTERVAL (0,1).                                          000460
C J0 AND J1 ARE BESSEL FUNCTIONS OF THE FIRST KIND,ZEROTH AND FIRST     000470
C ORDER RESPECTIVELY.                                                   000480
C PHIR(X) AND PHIT(X) ARE THE FUNCTIONS PHI-SUPERSCRIPT-RHO AND PHI-    000490
C SUPERSCRIPT-THETA RESPECTIVELY IN THE ABOVE REFERENCE.                000500
C THESE FUNCTIONS ARE GIVEN BY%                                         000510
C PHIR(X)=C*S1R*F1(X)+4*C*S2R*F2(X)                                     000520
C PHIT(X)=C*S1T*F1(X)+4*C*S2T*F2(X)                                     000530
C WHERE%                                                                000540
C     C=B!3*P0*BETA/KT                                                  000550
C     B=WINDOW RADIUS (CM)                                              000560
C     P0=MEAN INCIDENT POWER DENSITY (WATTS/CM!2)                       000570
C     BETA=BULK ABSORPTION COEFFICIENT (1/CM)                           000580
C     KT=THERMAL CONDUCTIVITY (WATTS/(CM DEGC)                          000590
C     S1R,S2R,S1T,S2T ARE MATERIAL CONSTANTS DEFINED IN THE ABOVE REF.  000600
C     F1,F2 ARE THE FUNCTIONS DELTRAP-PRIME(X) AND                      000610
C     (1/X!2)I(0,X,DS)(DELTRAP-PRIME(S))                                000620
C     GIVEN IN THE ABOVE REFERENCE AND WHICH ARE PROVIDED AT SELECTED   000630
C     ARGUMENTS BY PROGRAM @TEMPS@.                                     000640
C*************************************************************************000650
      COMMON/PHIBLK/CS1R,CS2R,CS1T,CS2T,XS,DZ,NF,MINT,EPS,F1(200),      000660
     +F2(200),HOLD,A,K ,TLAST,TNEXT,IERR,IP,MP1,ISW,N,                  000670
     +RAD,NP1,C3,C1                                                     000680
      COMMON/IFILES/IT3,IT4,IT5,IT6,IT7,IT8                             000690
      COMPLEX Q1,EXPR,EXPT,FX,FY,F7,FW                                  000700
      REAL J0,J1                                                        000710
      REAL K                                                            000720
      DIMENSION XA(100),YR(100),YI(100),ZR(100),ZI(100)                 000730
      A2=A*A                                                            000740
      IF (A2 .GT. 220.) 1020,1030                                       000750
 1020 CONST=2.*A2*A2                                                    000760
      GOTO 1040                                                         000770
 1030 CONST=2.*(A2/(1.-EXP(-A2)))**2                                    000780
 1040 UD2=U/2E0                                                         000790
C IF ISW=1 THEN THE ARRAY OF POINTS FOR GAUSSIAN INTEGRATION            000800
C MUST BE FOUND                                                         000810
      GOTO (1050,1060) ISW                                              000820
 1050 ISW=2                                                             000830
      CALL DQG24A(0E0,1E0,XA)                                           000840
      IF (IP .EQ. 1) WRITE(IT6,1) (XA(I),I=1,N)                         000850
 1060 CALL RTAPE3(T)                                                    000860
      IF (IERR .NE. 0) GOTO 2000                                        000870
      DO 100 I=1,N                                                      000880
      X=XA(I)                                                           000890
      X2=X*X                                                            000900
      XV=X*V                                                            000910
      FW=EXP(-A2*X2)*CEXP(CMPLX(0E0,-UD2*X2))                           000920
      EXPR=CEXP(CMPLX(0E0,K*PHI(X)))                                    000930
      EXPT=CEXP(CMPLX(0E0,K*HOLD))                                      000940
      IF (V .EQ. 0.) FZ=(EXPR-EXPT)*X/2.                                000950
      IF (V .NE. 0.) FZ=(EXPR-EXPT)*J1(XV)/V                            000960
      Q1=X*J0(XV)                                                       000970
      FX=Q1*EXPR-FZ                                                     000980
      FY=Q1*EXPT+FZ                                                     000990
      Q1=FW*FX                                                          001000
      YR(I)=REAL(Q1)                                                    001010
```

```
75          YI(I)=AIMAG(Q1)                                              003720
            Q1=FW*FY                                                     003730
            ZR(I)=REAL(Q1)                                               003740
            ZI(I)=AIMAG(Q1)                                              003750
100         CONTINUE                                                     003760
80          1   FORMAT(5(1X,G12.5))                                      003770
            CALL DOG24B(0E0,1E0,YR,YRI)                                  003780
            CALL DOG24B(0E0,1E0,YI,YII)                                  003790
            CALL DOG24B(0E0,1E0,ZR,ZRI)                                  003800
            CALL DOG24B(0E0,1E0,ZI,ZII)                                  003810
85          IKIPK=CONST*(YRI*YRI+YII*YII+ZRI*ZRI+ZII*ZII)                003820
2000 RETURN                                                              003830
            END                                                          003840
```

B.3 Function PHI

```
1           FUNCTION PHI(X)                                              003850
C IF SWT=.F. THEN PHIR(X)=C*S1R*F1(X)+4*S2R*F2(X) IS FOUND               003860
C IF SWT=.T. THEN PHII(X)=C*S1I*F1(X)+C*S2I*F2(X) IS FOUND               003870
C WHERE C=8.3*3.0*BETA/KAPPA (SEE CALLING SUBROUTINE)                    003880
5           C THE PROGRAM IS DESIGNED TO BE CALLED IN THE ORDER .F., .T. FOR A   003890
C GIVEN VALUE OF X IN ORDER TO ELIMINATE DOUBLE CALLS TO THE INTERPOLATI 003900
C ROUTINE                                                                003910
C THE ARRAYS F1,F2 CONTAIN THE FUNCTION VALUES TO BE INTERPOLATED%       003920
C F1(1=XS),F1(2=XS+DX),....,F1(NF), ETC AS PROVIDED BY TEMP5.            003930
10          C THE IBM SCI. SUB. ATSE (P. 251) RETURNS MINT FUNCTION VALUES AND   003940
C ARGUMENTS TO BE USED FOR INTERPOLATION IN ARRAYS ARG,VAL RESPECTIVELY. 003950
C GIVEN THE ARGUMENT X,ETC.                                              003960
C (THE ARGUMENT ICOL IN ATSE IS 1 IF THE FUNCTION IS STORED IN A         003970
C 1-DIMENSIONAL ARRAY)                                                   003980
15          C THE IBM SCI. SUB. ALI(P.241) DOES AITKEN-LAGRANGE INTERPOLATION    003990
C ON (ARG,VAL) AND RETURNS THE RESULTING VALUE Y. EPS IS AN ABSOLUTE     004000
C ERROR FIGURE AND IER IS AN ERROR @FLAG@.                               004010
            COMMON/PHIBLK/CS1R,CS2R,CS1I,CS2I,XS,DX,NF,MINT,EPS,F1(200),     004020
           *F2(200),HOLD,A,KE,TLAST,TNEXT,IERR,IP,MP1,ISW,N              004030
20          *RAD,NP1,C3,C1                                               004040
C HOLD IS USED TO STORE PHI(.T.,X)                                       004050
            DIMENSION ARG(20),VAL(20)                                    004060
1010 CALL ATSE(X,XS,DX,F1,NF,1,ARG,VAL,MINT)                             004070
            CALL ALI(X,ARG,VAL,Y1,MINT,EPS,IER)                          004080
            CALL ATSE(X,XS,DX,F2,NF,1,ARG,VAL,MINT)                      004090
25          CALL ALI(X,ARG,VAL,Y2,MINT,EPS,IER)                          004100
            PHI=CS1R*Y1+4E0*CS2R*Y2                                      004110
            HOLD=CS1I*Y1+4E0*CS2I*Y2                                     004120
2000 RETURN                                                              004130
30          END                                                          004140
```

B.4 Function J0

```
      REAL FUNCTION J0(X)                                                004150
C J0 IS THE BESSEL FUNCTION OF THE FIRST KIND,ZEROTH ORDER. SEE           004160
C HANDBOOK OF MATHEMATICAL FUNCTIONS-AMS 55. FOR VALUES OF THE ARGUMENT   004170
C X<=5 EQUATION 9.1.12 IS USED. OTHERWISE 9.4.3 IS USED.                  004180
      DIMENSION FACT(20)                                                 004190
      DATA MT/0/                                                         004200
      IF(X.GT.5.) GO TO 1                                                004210
      IF(MT.NE.0) GO TO 2                                                004220
      MT=1                                                               004230
      FACT(1)=1.0                                                        004240
      DO 3 I=2,20                                                        004250
      FACT(I)=FACT(I-1)*FLOAT(I*1)                                       004260
    3 CONTINUE                                                           004270
      DO 4 I=2,20                                                        004280
      FACT(I)=1.0/FACT(I)                                                004290
    4 CONTINUE                                                           004300
    2 CONTINUE                                                           004310
      ANSP=0.0                                                           004320
      ANSN=0.0                                                           004330
      ARG=0.25*X*X                                                       004340
      ARGU=ARG                                                           004350
      DO 5 I=1,19,2                                                      004360
      ANSN=ANSN+ARGU*FACT(I)                                             004370
      J=I+1                                                              004380
      ARGU=ARGU*ARG                                                      004390
      ANSP=ANSP+ARGU*FACT(J)                                             004400
      ARGU=ARGU*ARG                                                      004410
    5 CONTINUE                                                           004420
      J0=1.0+(ANSP-ANSN)                                                 004430
      RETURN                                                             004440
    1 CONTINUE                                                           004450
      TOX=3.0/X                                                          004460
      F7=.79788456-.00000077*TOX-.0055274*TOX**2-.00009512*TOX**3+       004470
     1.00137237*TOX**4-.00072805*TOX**5+.00014476*TOX**6                 004480
      THZ=X-.78539816-.04166397*TOX-.00003954*TOX**2+.00262573*TOX**3-    004490
     1.00054125*TOX**4-.00029333*TOX**5+.00013558*TOX**6                 004500
      J0=F7*COS(THZ)/SQRT(X)                                             004510
      RETURN                                                             004520
      END                                                                004530
```

B.5 Function J1

```
      REAL FUNCTION J1(X)                                                004540
C J1 IS THE BESSEL FUNCTION OF THE FIRST KIND,FIRST ORDER. SEE            004550
C HANDBOOK OF MATHEMATICAL FUNCTIONS-AMS 55. FOR VALUES OF THE            004560
C ARGUMENT X<=10 EQUATION 9.1.10 IS USED. OTHERWISE 9.4.4 IS USED.        004570
      DIMENSION FACT(20)                                                 004580
      DATA MT/0/                                                         004590
      IF(X.GT.10.) GO TO 1                                               004600
      IF(MT.NE.0) GO TO 2                                                004610
      MT=1                                                               004620
      FACT(1)=2.0                                                        004630
      DO 3 I=2,20                                                        004640
      FACT(I)=FACT(I-1)*FLOAT(I*(I+1))                                   004650
    3 CONTINUE                                                           004660
      DO 4 I=1,20                                                        004670
      FACT(I)=1.0/FACT(I)                                                004680
    4 CONTINUE                                                           004690
    2 CONTINUE                                                           004700
      ANSP=0.0                                                           004710
      ANSN=0.0                                                           004720
```

```
      ARGS=0.5*X                                                        00004730
      ARG=0.25*X*X                                                      00004740
      ARGU=ARG                                                          00004750
      DO 5 I=1,19,2                                                     00004760
      ANSN=ANSN+ARGU*FACT(I)                                            00004770
      I=I+1                                                             00004780
      ARGU=ARGU*ARG                                                     00004790
      ANSD=ANSD+ARGU*FACT(I)                                            00004800
      ARGU=ARGU*ARG                                                     00004810
    5 CONTINUE                                                          00004820
      J1=ARGS*(1.0+(ANSD-ANSN))                                         00004830
      RETURN                                                            00004840
    1 CONTINUE                                                          00004850
      TOX=3.0/X                                                         00004860
      F1=.79788456+.00000156*TOX+.01659667*TOX**2+.00017105*TOX**3      00004870
     1-.00249511*TOX**4+.00113653*TOX**5-.00020033*TOX**6               00004880
      TH1=X-2.35619449+.12499612*TOX+.0000565U*TOX**2-.00637879*TOX**3  00004890
     1+.00074348*TOX**4+.00007924*TOX**5-.00029166*TOX**6               00004900
      J1=F1*COS(TH1)/SQRT(X)                                            00004910
      RETURN                                                            00004920
      END                                                               00004930
```

B.6 Subroutine RTAPE3

```
      SUBROUTINE RTAPE3(T)                                              00004940
      COMMON/PHTBLK/CS1P,CS2P,CS1T,CS2T,XS,DX,NF,MINT,EPS,F1(200),      00004950
     +F2(200),HOLD,A,KF,TLAST,TNEXT,IERR,IP,MP1,ISW,N,                  00004960
     +RAD,NP1,C3,C1                                                     00004970
      COMMON/TETLES/IT3,IT4,IT5,IT6,IT7,IT8                             00004980
      DIMENSION FIM(82),F2M(82),F1P(82),F2P(82),RFIN(82),ZFIN(22),      00004990
     +UFINP(82,22),UFINM(82,22)                                         00005000
      DATA ISW1/0/                                                      00005010
C RTAPE3 CAN BE USED AS A GENERAL PURPOSE SUBROUTINE FOR LINEARLY       00005020
C INTERPOLATING FUNCTION VALUES 'BETWEEN RECORDS', I.E. ASSUME          00005030
C RECORD N INCLUDES THE INFORMATION TN,F(1;TN),F(2;TN).....**           00005040
C AND RECORD N+1 INCLUDES TN+1,F(1;TN+1),F(2;TN+1).....                 00005050
C THEN IF TN[=T[=TN+1 IS GIVEN, THE QUANTITIES F(1;T),F(2;T)**.*******  00005060
C ARE RETURNED WHERE*                                                   00005070
C  F(1;T)=F(1;TN)+C*(F(1;TN+1)-F(1;TN)); C=(T-TN)/(TN+1-TN), ETC.       00005080
C IT IS ASSUMED THAT THE PARAMETER I INCREASES WITH INCREASING RECORD   00005090
C NUMBER AND NO FILE REWINDS ARE PERMITTED. I.E. RTAPE3 SHOULD BE CALLED 00005100
C WITH INCREASING VALUES OF T ONLY.                                     00005110
C RTAPE3 RECOGNIZES TWO ERROR CONDITIONS WHICH SHOULD BE CHECKED FOR IN 00005120
C THE CALLING PROGRAM. IF IERR=0 THEN NO ERROR HAS OCCURRED. IF IERR=1  00005130
C THEN THE TIME VALUE IS LESS THAN THE TIME VALUE OF THE PRECEDING CALL 00005140
C TO RTAPE3.IF IERR=2 THEN THE TIME VALUE IS GREATER THAN THE TIME VALUE 00005150
C ASSOCIATED WITH THE LAST INPUT RECORD.                                00005160
C T  THE PRESENT VERSION OF RTAPE3 EACH OUTPUT RECORD (FROM TEMP5) IS   00005170
C ASSUMED TO BE IN THE FORM*                                            00005180
C   NF,TFIN,RFIN(82),ZFIN(22),UFIN(82,22),F1(82),F2(82)                00005190
C WHERE TFIN IS THE TIME VALUE AND F1(82),F2(82) ARE THE DESIRED        00005200
C FUNCTION VALUES CORRESPONDING TO TFIN.                                00005210
C RTAPE3 ALSO OUTPUTS THE DIMENSIONALIZED (AND LINEARLY INTERPOLATED    00005220
C IN TIME) WINDOW TEMPERATURE FUNCTIN IN A FORM SUITABLE FOR USE WITH   00005230
C DISPLAY, PROVIDED IPRINT=2.                                           00005240
      IERR=0                                                            00005250
      TY=T/C1                                                           00005255
      IF (ISW1) 1115,1100                                               00005260
 1100 ISW1=1                                                            00005270
      READ(IT3) NG,TLAST,RFIN,ZFIN,UFINM,F1M,F2M                        00005280
      TT=TLAST                                                          00005290
      RMIN=RFIN(1)*RAD                                                  00005300
      RMAX=RFIN(MP1)*RAD                                                00005310
      IF (EOF(IT3)) 1110,1120                                           00005320
 1110 IERR=2                                                            00005330
      GOTO 2000                                                         00005340
 1120 IF (T-TLAST) 1130,1140,1140                                       00005350
 1130 IERR=1                                                            00005360
```

```
45                GOTO 2000                                                    00537
             1140 READ(IT3) NG,TNEXT,RFIN,ZFIN,UFINP,F1P,F2P                   00538
             1115 IF (T .LT. TT) 1000,1010                                     00540
             1000 IERR=1                                                       00541
                  GOTO 2000                                                    00542
50           C IF T .EQ. LASTT THEN NOTHING MORE TO DO                        00543
             1010 IF (T .EQ. TT) GOTO 2000                                     00544
             C IF T .LT. TNEXT THEN A READ IS NOT REQUIRED                    00545
                  IF (T .LT. TNEXT) GOTO 1020                                 00546
             1030 READ(IT3) NG,TLAST,RFIN,ZFIN,UFINM,F1M,F2M                   00547
55                IF (EOF(IT3)) 1040,1050                                     00548
             1040 IERR=2                                                       00549
                  GOTO 2000                                                    00550

             1050 IF (T .LE. TLAST) GOTO 1080                                  00551
                  READ(IT3) NG,TNEXT,RFIN,ZFIN,UFINP,F1P,F2P                  00552
60                IF (EOF(IT3)) 1060,1070                                     00553
             1060 IERR=2                                                       00554
                  GOTO 2000                                                    00555
             1070 IF (T .LE. TNEXT) GOTO 1020                                  00556
                  GOTO 1030                                                    00557
65           1020 C=(T-TLAST)/(TNEXT-TLAST)                                    00558
                  DO 100 I=1,MP1                                              00559
                  F1(I)=F1M(I)+C*(F1P(I)-F1M(I))                              00560
                  F2(I)=F2M(I)+C*(F2P(I)-F2M(I))                              00561
             100  CONTINUE                                                    00562
70                GOTO(2000,120) IP                                           00563
              120 DO 130 I=1,NP1                                              00564
                  U=RAD*ZFIN(I)                                               00565
                  DO 140 J=1,MP1                                              00566
                  RFIN(J)=C3*(UFINM(J,I)+C*(UFINP(J,I)-UFINM(J,I)))           00567
75            140 CONTINUE                                                    00568
                  WRITE(IT6) I,NP1,U,TM,MP1,RMIN,RMAX,(RFIN(J),J=1,MP1)       00569
              130 CONTINUE                                                    00570
                  GOTO 2000                                                   00571
             1080 TTEMP=TLAST                                                 00572
80                TLAST=TNEXT                                                 00573
                  TNEXT=TTEMP                                                 00574
                  DO 110 I=1,MP1                                              00575
                  FT=F1M(I)                                                   00576
                  F1M(I)=F1P(I)                                               00577
85                F1P(I)=FT                                                   00578
                  FT=F2M(I)                                                   00579
                  F2M(I)=F2P(I)                                               00580
                  F2P(I)=FT                                                   00581
              110 CONTINUE                                                    00582
90                GOTO(1020,150) IP                                           00583
              150 DO 160 I=1,NP1                                              00584
                  DO 160 J=1,MP1                                              00585
                  FT=UFINM(J,I)                                               00586
                  UFINM(J,I)=UFINP(J,I)                                       00587
95                UFINP(J,I)=FT                                               00588
              160 CONTINUE                                                    00589
                  GOTO 1020                                                   00590
             2000 TT=T                                                        00591
                  RETURN                                                      00592
100               END                                                         00593
```

```
      SUBROUTINE ALI(X,ARG,VAL,Y,NDIM,EPS,IER)                          005940
C                                                                       005950
C     ******************************************************************005960
C                                                                       005970
C     SUBROUTINE ALI                                                    005980
C                                                                       005990
C     PURPOSE                                                           006000
C        TO INTERPOLATE FUNCTION VALUE Y FOR A GIVEN ARGUMENT VALUE     006010
C        X USING A GIVEN TABLE (ARG,VAL) OF ARGUMENT AND FUNCTION       006020
C        VALUES.                                                        006030
C                                                                       006040
C     USAGE                                                             006050
C        CALL ALI (X,ARG,VAL,Y,NDIM,EPS,IER)                            006060
C                                                                       006070
C     DESCRIPTION OF PARAMETERS                                         006080
C        X      - THE ARGUMENT VALUE SPECIFIED BY INPUT.                006090
C        ARG    - THE INPUT VECTOR (DIMENSION NDIM) OF ARGUMENT         006100
C                 VALUES OF THE TABLE (NOT DESTROYED).                  006110
C        VAL    - THE INPUT VECTOR (DIMENSION NDIM) OF FUNCTION         006120
C                 VALUES OF THE TABLE (DESTROYED).                      006130
C        Y      - THE RESULTING INTERPOLATED FUNCTION VALUE.            006140
C                                                                       006150
C        NDIM   - AN INPUT VALUE WHICH SPECIFIES THE NUMBER OF          006160
C                 POINTS IN TABLE (ARG,VAL).                            006170
C        EPS    - AN INPUT CONSTANT WHICH IS USED AS UPPER BOUND        006180
C                 FOR THE ABSOLUTE ERROR.                               006190
C        IER    - A RESULTING ERROR PARAMETER.                          006200
C                                                                       006210
C     REMARKS                                                           006220
C        (1) TABLE (ARG,VAL) SHOULD REPRESENT A SINGLE-VALUED           006230
C            FUNCTION AND SHOULD BE STORED IN SUCH A WAY, THAT THE      006240
C            DISTANCES ABS(ARG(I)-X) INCREASE WITH INCREASING           006250
C            SUBSCRIPT I. TO GEN-RATE THIS ORDER IN TABLE (ARG,VAL),    006260
C            SUBROUTINES ATSG, AISM OR ATSE COULD BE USED IN A          006270
C            PREVIOUS STAGE.                                            006280
C        (2) NO ACTION BESIDES ERROR MESSAGE IN CASE NDIM LESS          006290
C            THAN 1.                                                    006300
C        (3) INTERPOLATION IS TERMINATED EITHER IF THE DIFFERENCE       006310
C            BETWEEN TWO SUCCESSIVE INTERPOLATED VALUES IS              006320
C            ABSOLUTELY LESS THAN TOLERANCE EPS, OR IF THE ABSOLUTE     006330
C            VALUE OF THIS DIFFERENCE STOPS DIMINISHING, OR AFTER       006340
C            (NDIM-1) STEPS. FURTHER IT IS TERMINATED IF THE            006350
C            PROCEDURE DISCOVERS TWO ARGUMENT VALUES IN VECTOR ARG      006360
C            WHICH ARE IDENTICAL. DEPENDENT ON THESE FOUR CASES,        006370
C            ERROR PARAMETER IER IS CODED IN THE FOLLOWING FORM         006380
C              IER=0 - IT WAS POSSIBLE TO REACH THE REQUIRED            006390
C                      ACCURACY (NO ERROR).                             006400
C              IER=1 - IT WAS IMPOSSIBLE TO REACH THE REQUIRED          006410
C                      ACCURACY BECAUSE OF ROUNDING ERRORS.             006420
C              IER=2 - IT WAS IMPOSSIBLE TO CHECK ACCURACY BECAUSE      006430
C                      NDIM IS LESS THAN 3, OR THE REQUIRED ACCURACY    006440
C                      COULD NOT BE REACHED BY MEANS OF THE GIVEN       006450
C                      TABLE. NDIM SHOULD BE INCREASED.                 006460
C              IER=3 - THE PROCEDURE DISCOVERED TWO ARGUMENT VALUES     006470
C                      IN VECTOR ARG WHICH ARE IDENTICAL.               006480
C                                                                       006490
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     006500
C                                                                       006510
C        NONE                                                           006520
C                                                                       006530
C     METHOD                                                            006540
C        INTERPOLATION IS DONE BY MEANS OF AITKENS SCHEME OF            006550
C        LAGRANGE INTERPOLATION. ON RETURN Y CONTAINS AN INTERPOLATED    006560
C                                                                       006570
C        FUNCTION VALUE AT POINT X, WHICH IS IN THE SENSE OF REMARK     006580
C        (3) OPTIMAL WITH RESPECT TO GIVEN TABLE. FOR REFERENCE, SEE    006590
C        F.B.HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS.            006600
C        MCGRAW-HILL, NEW YORK/TORONTO/LONDON, 1956, PP.49-50.          006610
C                                                                       006620
C     ******************************************************************006630
C                                                                       006640
C                                                                       006650
      DIMENSION ARG(1),VAL(1)                                           006660
      IER=2                                                             006670
```

```
      DELT2=0.                                                          000680
75    IF(NDIM-1)9,7,1                                                   000690
      C                                                                 000700
      C   START OF AITKEN-LOOP                                          000710
      C                                                                 000720
      1 DO 6 J=2,NDIM                                                   000730
80    DELT1=DELT2                                                       000740
      IEND=J-1                                                          000750
      DO 2 I=1,IEND                                                     000760
      H=ARG(I)-ARG(J)                                                   000770
      IF(H)2,13,2                                                       000780
85    2 VAL(J)=(VAL(I)*(X-ARG(J))-VAL(J)*(X-ARG(I)))/H                  000790
      DELT2=ABS(VAL(J)-VAL(IEND))                                       000800
      IF(I-2)6,6,3                                                      000810
      3 IF(DELT2-EPS)10,10,4                                            000820
      4 IF(I-5)6,5,5                                                    000830
      5 IF(DELT2-DELT1)6,11,11                                          000840
      6 CONTINUE                                                        000850
      C   END OF AITKEN-LOOP                                           000860
      C                                                                 000870
      7 J=NDIM                                                          000880
      8 Y=VAL(J)                                                        000890
      9 RETURN                                                          000900
      C                                                                 000910
      C   THERE IS SUFFICIENT ACCURACY WITHIN NDIM-1 ITERATION STEPS    000920
      10 IER=0                                                          000930
      GOTO 8                                                            000940
      C                                                                 000950
      C   TEST VALUE DELT2 STARTS OSCILLATING                           000960
      C                                                                 000970
      11 IER=1                                                          000980
      12 J=IEND                                                         000990
      GOTO 8                                                            001000
      C                                                                 001010
      C   THERE ARE TWO IDENTICAL ARGUMENT VALUES IN VECTOR ARG         001020
      13 IER=3                                                          001030
      GOTO 12                                                           001040
      END
```

B.8 Subroutine ATSE

```
      SUBROUTINE ATSE (X,ZS,DZ,F,IROW,ICOL,ARG,VAL,NDIM)                001050
      C                                                                 001060
      C ...............................................................  001070
      C                                                                 001080
      C                                                                 001090
      C   SUBROUTINE ATSE                                              001100
      C                                                                 001110
      C                                                                 001120
      C   PURPOSE                                                       001130
      C      NDIM POINTS OF A GIVEN TABLE WITH EQUIDISTANT ARGUMENTS ARE 001140
      C      SELECTED AND ORDERED SUCH THAT                             001150
      C      ABS(ARG(I)-Y).GE.ABS(ARG(J)-X) IF I.GT.J.                  001160
      C                                                                 001170
      C   USAGE                                                         001180
      C      CALL ATSE (X,ZS,DZ,F,IROW,ICOL,ARG,VAL,NDIM)               001190
      C                                                                 001200
      C   DESCRIPTION OF PARAMETERS                                     001210
      C      X      - THE SEARCH ARGUMENT.                              001220
      C      ZS     - THE STARTING VALUE OF ARGUMENTS.                  001230
      C      DZ     - THE INCREMENT OF ARGUMENT VALUES.                 001240
      C      F      - IN CASE ICOL=1, F IS THE VECTOR OF FUNCTION VALUES 001250
      C               (DIMENSION IROW).                                 001260
      C               IN CASE ICOL=2, F IS AN IROW BY 2 MATRIX. THE FIRST 001270
      C               COLUMN SPECIFIES THE VECTOR OF FUNCTION VALUES AND 001280
      C               THE SECOND THE VECTOR OF DERIVATIVES.             001290
      C      IROW   - THE DIMENSION OF EACH COLUMN IN MATRIX F.         001300
      C      ICOL   - THE NUMBER OF COLUMNS IN F (I.E. 1 OR 2).         001310
      C      ARG    - THE RESULTING VECTOR OF SELECTED AND ORDERED      001320
      C               ARGUMENT VALUES (DIMENSION NDIM).                 001330
      C      VAL    - THE RESULTING VECTOR OF SELECTED FUNCTION VALUES
```

```
C                 (DIMENSION NDIM) IN CASE ICOL=1. IN CASE ICOL=2.      00007340
C                 VAL IS THE VECTOR OF FUNCTION AND DERIVATIVE VALUES   00007350
C                 (DIMENSION 2*NDIM) WHICH ARE STORED IN PAIRS (I.E.    00007360
C                 EACH FUNCTION VALUE IS FOLLOWED BY ITS DERIVATIVE     00007370
C                 VALUE).                                               00007380
C        NDIM     - THE NUMBER OF POINTS WHICH MUST BE SELECTED OUT OF  00007390
C                 THE GIVEN TABLE.                                      00007400
C                                                                       00007410
C                                                                       00007420
C     REMARKS                                                           00007430
C        NO ACTION IN CASE IROW LESS THAN 1.                            00007440
C        IF INPUT VALUE NDIM IS GREATER THAN IROW, THE PROGRAM          00007450
C        SELECTS ONLY A MAXIMUM TABLE OF IROW POINTS.  THEREFORE THE    00007460
C        USER OUGHT TO CHECK CORRESPONDENCE BETWEEN TABLE (ARG,VAL)     00007470
C        AND ITS DIMENSION BY COMPARISON OF NDIM AND IROW, IN ORDER     00007480
C        TO GET CORRECT RESULTS IN FURTHER WORK WITH TABLE (ARG,VAL).   00007490
C        THIS TEST MAY BE DONE BEFORE OR AFTER CALLING                  00007500
C        SUBROUTINE ATSE.                                               00007510
C                                                                       00007520
C        SUBROUTINE ATSE ESPECIALLY CAN BE USED FOR GENERATING THE      00007530
C        TABLE (ARG,VAL) NEEDED IN SUBROUTINES ALI, AHI, AND ACFI.      00007540
C                                                                       00007550
C     SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                     00007560
C        NONE                                                           00007570
C                                                                       00007580
C     METHOD                                                            00007590
C        SELECTION IS DONE BY COMPUTING THE SUBSCRIPT J OF THAT         00007600
C        ARGUMENT, WHICH IS NEXT TO X.                                  00007610
C        AFTERWARDS NEIGHBOURING ARGUMENT VALUES ARE TESTED AND         00007620
C        SELECTED IN THE ABOVE SENSE.                                   00007630
C                                                                       00007640
C     ..................................................................00007650
C                                                                       00007660
C                                                                       00007670
C                                                                       00007680
      DIMENSION F(1),ARG(1),VAL(1)                                      00007690
      IF(IROW-1)19,17,1                                                 00007700
C                                                                       00007710
C     CASE DZ=0 IS CHECKED OUT                                          00007720
    1 IF(DZ)2,17,2                                                      00007730
    2 N=NDIM                                                            00007740
C                                                                       00007750
C     IF N IS GREATER THAN IROW, N IS SET EQUAL TO IROW.                00007760
      IF(N-IROW)4,4,3                                                   00007770
    3 N=IROW                                                            00007780
C                                                                       00007790
C     COMPUTATION OF STARTING SUBSCRIPT J.                              00007800
    4 J=(Y-ZS)/DZ+1.5                                                   00007810
      IF(J)5,5,6                                                        00007820
    5 J=1                                                               00007830
    6 IF(J-IROW)8,8,7                                                   00007840
    7 J=IROW                                                            00007850
C                                                                       00007860
C     GENERATION OF TABLE ARG,VAL IN CASE DZ.NE.0.                      00007870
    8 II=J                                                              00007880
      JL=0                                                              00007890
      JR=0                                                              00007900
      DO 16 I=1,N                                                       00007910
C                                                                       00007920
      ARG(I)=ZS+FLOAT(II-1)*DZ                                          00007930
      IF(ICOL-2)9,10,10                                                 00007940
    9 VAL(I)=F(II)                                                      00007950
      GOTO 11                                                           00007960
   10 VAL(2*I-1)=F(II)                                                  00007970
      III=II+IROW                                                       00007980
      VAL(2*I)=F(III)                                                   00007990
   11 IF(I+JR-IROW)12,15,12                                             00008000
   12 IF(I-JL-1)13,14,13                                                00008010
   13 IF((ARG(I)-X)*DZ)14,15,15                                         00008020
   14 JR=JR+1                                                           00008030
      II=I+JR                                                           00008040
      GOTO 16                                                           00008050
   15 JL=JL+1                                                           00008060
      II=I-JL                                                           00008070
   16 CONTINUE                                                          00008080
      RETURN
```

```
C
C     CASE DZ=0                                                          008100
   17 ARG(1)=ZS                                                          008110
      VAL(1)=F(1)                                                        008120
      IF(TCOL-2)19,19,18                                                 008130
   18 VAL(2)=F(2)                                                        008140
   19 RETURN                                                             008150
      END                                                                008160
```

B.9 Subroutine DQG24A

```
      SUBROUTINE DQG24A(XL,XU,XA)                                        008170
                                                                         008180
C     ..................................................................  008190
C                                                                        008200
C        SUBROUTINE DQG24                                                008210
C                                                                        008220
C        PURPOSE                                                         008230
C           TO COMPUTE INTEGRAL(FCT(X), SUMMED OVER X FROM XL TO XU)     008240
C                                                                        008250
C        USAGE                                                           008260
C           CALL DQG24 (XL,XU,FCT,Y)                                     008270
C           PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT                 008280
C                                                                        008290
C        DESCRIPTION OF PARAMETERS                                       008300
C           XL     - DOUBLE PRECISION LOWER BOUND OF THE INTERVAL.       008310
C           XU     - DOUBLE PRECISION UPPER BOUND OF THE INTERVAL.       008320
C           FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION   008330
C                    SUBPROGRAM USED.                                    008340
C           Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.      008350
C                                                                        008360
C        REMARKS                                                         008370
C           NONE                                                         008380
C                                                                        008390
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                   008400
C           THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)     008410
C           MUST BE FURNISHED BY THE USER.                               008420
C                                                                        008430
C        METHOD                                                          008440
C           EVALUATION IS DONE BY MEANS OF 24-POINT GAUSS QUADRATURE     008450
C           FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 47        008460
C           EXACTLY. FOR REFERENCE, SEE                                  008470
C           V.I.KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS,            008480
C           MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340.    008490
C                                                                        008500
C     ..................................................................  008510
C                                                                        008520
C                                                                        008530
C                                                                        008540
      DIMENSION XA(1)                                                    008550
C                                                                        008560
      A=.5D0*(XU+XL)                                                     008570
      B=XU-XL                                                            008580
      C=.49759360999851068D0*B                                           008590
      XA(1)=A+C                                                          008600
      XA(2)=A-C                                                          008610
      C=.48736427798565475D0*B                                           008620
      XA(3)=A+C                                                          008630
      XA(4)=A-C                                                          008640
      C=.46913727600176639D0*B                                           008650
      XA(5)=A+C                                                          008660
      XA(6)=A-C                                                          008670
      C=.44320776350220052D0*B                                           008680
      XA(7)=A+C                                                          008690
      XA(8)=A-C                                                          008700
      C=.41000999298695146D0*B                                           008710
      XA(9)=A+C                                                          008720
      XA(10)=A-C                                                         008730
```

```
      C=.9700620957892771900*R                                          008750
      XA(11)=A+C                                                        008760
60    XA(12)=A-C                                                        008770
      C=.9240482596848779000*R                                          008780
      XA(13)=A+C                                                        008790
      XA(14)=A-C                                                        008800
      C=.9271173569441977000*R                                          008810
65    XA(15)=A+C                                                        008820
      XA(16)=A-C                                                        008830
      C=.9168967538130225700*R                                          008840
      XA(17)=A+C                                                        008850
      XA(18)=A-C                                                        008860
70    C=.9575213398480816900*R                                          008870
      XA(19)=A+C                                                        008880
      XA(20)=A-C                                                        008890
      C=.9555943373688150-1*R                                           008900
      XA(21)=A+C                                                        008910
75    XA(22)=A-C                                                        008920
      C=.9202844643130281D0-1*R                                         008930
      XA(23)=A+C                                                        008940
      XA(24)=A-C                                                        008950
      RETURN                                                            008960
80    END
```

B.10 Subroutine DQG24B

```
1     C                                                                 008970
      SUBROUTINE DQG24B(XL,XU,FCT,Y)                                    008980
      C ..................................................................008990
      C                                                                 009000
      C                                                                 009010
5     C    SUBROUTINE DQG24                                             009020
      C                                                                 009030
      C    PURPOSE                                                      009040
      C       TO COMPUTE INTEGRAL(FCT(X), SUMMED OVER X FROM AL TO XU)  009050
      C                                                                 009060
10    C    USAGE                                                        009070
      C       CALL DQG24 (XL,XU,FCT,Y)                                  009080
      C       PARAMETER FCT REQUIRES AN EXTERNAL STATEMENT              009090
      C                                                                 009100
      C    DESCRIPTION OF PARAMETERS                                    009110
15    C       XL     - DOUBLE PRECISION LOWER BOUND OF THE INTERVAL.    009120
      C       XU     - DOUBLE PRECISION UPPER BOUND OF THE INTERVAL.    009130
      C       FCT    - THE NAME OF AN EXTERNAL DOUBLE PRECISION FUNCTION009140
      C                SUBPROGRAM USED.                                 009150
      C       Y      - THE RESULTING DOUBLE PRECISION INTEGRAL VALUE.   009160
20    C                                                                 009170
      C    REMARKS                                                      009180
      C       NONE                                                      009190
      C                                                                 009200
      C    SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                009210
25    C       THE EXTERNAL DOUBLE PRECISION FUNCTION SUBPROGRAM FCT(X)  009220
      C       MUST BE FURNISHED BY THE USER.                            009230
      C                                                                 009240
      C    METHOD                                                       009250
      C       EVALUATION IS DONE BY MEANS OF 24-POINT GAUSS QUADRATURE  009260
30    C       FORMULA, WHICH INTEGRATES POLYNOMIALS UP TO DEGREE 47     009270
      C       EXACTLY. FOR REFERENCE, SEE                               009280
      C       V.I.KRYLOV, APPROXIMATE CALCULATION OF INTEGRALS,         009290
      C       MACMILLAN, NEW YORK/LONDON, 1962, PP.100-111 AND 337-340. 009300
      C                                                                 009310
35    C ..................................................................009320
      C                                                                 009330
      C                                                                 009340
      C                                                                 009350
      DIMENSION FCT(1)
```

```
40          C                                                                   000360
                  B=XU-XL                                                       000370
                  I=1                                                           000380
                  Y=.5170614899993509D-2*(FCT(I)+FCT(I+1))                      000390
                  I=I+2                                                         000400
45                Y=Y+.1424569431446932D-1*(FCT(I)+FCT(I+1))                    000410
                  I=I+2                                                         000420
                  Y=Y+.2213871940870990D-1*(FCT(I)+FCT(I+1))                    000430
                  I=I+2                                                         000440
                  Y=Y+.2964929245771839D-1*(FCT(I)+FCT(I+1))                    000450
50                I=I+2                                                         000460
                  Y=Y+.3667324070554015D-1*(FCT(I)+FCT(I+1))                    000470
                  I=I+2                                                         000480
                  Y=Y+.4308508076597663D-1*(FCT(I)+FCT(I+1))                    000490
                  I=I+2                                                         000500
55                Y=Y+.4881932605205694D-1*(FCT(I)+FCT(I+1))                    000510
                  I=I+2                                                         000520
                  Y=Y+.5372213505798289D-1*(FCT(I)+FCT(I+1))                    000530

                  I=I+2                                                         000540
                  Y=Y+.5775283402682801D-1*(FCT(I)+FCT(I+1))                    000550
60                I=I+2                                                         000560
                  Y=Y+.6083523646391696D-1*(FCT(I)+FCT(I+1))                    000570
                  I=I+2                                                         000580
                  Y=Y+.6291872817341414D-1*(FCT(I)+FCT(I+1))                    000590
                  I=I+2                                                         000600
65                Y=B*(Y+.6396909767337607D-1*(FCT(I)+FCT(I+1)))                000610
                  RETURN                                                        000620
                  END                                                           000630
```

B.11 Function IKIRKP

```
1                 REAL FUNCTION IKIRKP(U,V,T)                                 000640
            C     APRIL 11,1974                                               000650
            C     THIS FUNCTION COMPUTES THE KIRKHOFF INTENSITY FUNCTION      000660
            C     ALONG THE U=0 AND/OR V=0 AXIS OF THE PLANE.                 000670
5           C     SEE COMMENTS IN IKIRK AND COMPUTE FOR BACKGROUND INFO.      000680
            C     IT IS VALID ONLY FOR CONSTANT TEMPERATURE WINDOWS.          000690
                  DIMENSION AA(10)                                            000700
                  COMMON/IFILES/IT3,IT4,IT5,IT6,IT7,IT8                       000710
                  COMMON/VICTOR/VV                                            000720
10                COMMON/PUIBLK/CS1R,CS2R,CS1T,CS2T,XS,DX,NF,MNNT,EPP,F1(200),
                 +F2(200),HOLD,ALPHA,KF,TLAST,TNEXT,IERR,IP,MP1,ISW,NGAUSS,
                 +RAD,NP1,C3,C1                                               000740
                  COMPLEX AJ0Q,CRH0Q,CTHETAQ,A1Q,A2Q,A3Q,A4Q,A6Q,A7Q           000750
                  REAL KE                                                     000765
15                DATA (AA(I),I=1,10)/1.0,-.99999999958,.4999999206,-.16666B3019,
                 +.0416573475,-.0083013598,.0013298820,-.0001413161,0.,0./    000780
                  VV=V                                                        000790
                  CALL RTAPE3(T)                                              000800
                  IF(IERR.NE.0) GOTO 2000                                     000810
20                CTHETAQ=CS1T*F1(1)+4.*CS2T*F2(1)                            000820
                  CTHETAQ=CTHETAQ*KE                                          000825
                  CRH0Q=CS1R*F1(1)+4.*CS2R*F2(1)                              000830
                  CRH0Q=CRH0Q*KE                                              000835
                  AJ0Q=(0.,1.)                                                000840
25                A2Q=CEXP(AJ0Q*CRH0Q)                                        000850
                  A3Q=CEXP(AJ0Q*CTHETAQ)                                      000860
                  A1Q=A2Q-A3Q                                                 000870
                  A=-(ALPHA**2)                                               000880
                  IF (A .LT. -1000) GOTO 110                                  000900
30                EXPA=EXP(A)                                                 000910
                  A5=2.*(A/(1.-EXPA))**2                                      000920
                  GOTO 120                                                    000930
              110 EXPA=0.                                                     000940
                  A5=2.*A*A                                                   000950
```

```
      120  B=U/2.                                                          000960
           IF(V.EQ.0) GOTO 200                                             000970
           C=(ALPHA/V)**2                                                  000980
           IF(U.EQ.0) GOTO 100                                             000990
      200  SINB=SIN(B)                                                     010000
           COSB=COS(B)                                                     010010
           ASQ=A**2                                                        010020
           BSQ=B**2                                                        010030
           A40REAL=.5*((EXPA*(A*COSB+B*SINB))-A)/(ASQ+BSQ)                 010040
           A40IMAG=-.5*((EXPA*(A*SINB-B*COSB))+B)/(ASQ+BSQ)                010050
           A40=CMPLX(A40REAL,A40IMAG)                                      010060
           A60=A10/2.*A40                                                  010070
           IKIRKP=A5*((CABS(A20*A40-A60))**2+(CABS(A30*A40+A60))**2)       010080
           RETURN                                                          010090
      100  IF (-A .GT. .693) GOTO 1000                                     010095
           CALL COMPUTE(C,AA,FF1,FF2,IER)                                  010100
           IF(IER.EQ.2) GOTO 1000                                          010110
           A70=A10*FF2                                                     010120
           IKIRKP=A5*((CABS(A20*FF1-A70))**2+(CABS(A30*FF1+A70))**2)       010130
           RETURN                                                          010140
      1000 WRITE (IT5,10)                                                  010150
      10   FORMAT(/* ALPHA**2 IS OUT OF RANGE*)                           010160
      2000  RETURN                                                        010170
            END                                                           010180
```

B.12 Subroutine COMPUTE

```
            SUBROUTINE COMPUTE (C,AA,F0,F1,IER)                           010190
      C     THIS SUBROUTINE RETURNS THE APPROXIMATION TO THE INTEGRAL       010200
      C     F0= (1/V)**2* INTEGRAL (Y*J0(Y)*E**(-CY**2))DY) OVER (0,V)      010210
      C     F1=(1/V)**2*INTEGRAL(J1(Y)*E**(-CY**2))DY) OVER (0,V) GIVEN BY  010220
      C     F0=(1/V)**2*SUM(A(I)*(C**I)*(INTEGRAL(Y**(2I+1)*J0(Y)DY)))  AND 010230
      C     F1=(1/V)**2*SUM(A(I)*(C**I)*INTEGRAL( Y**(2I)*J1(Y)DY))) OVER(0,V) 010240
      C     FOR I=0,1,...,N                                                 010250
      C     WHERE THE A(I) ARE GIVEN (FOR EXAMPLE) IN NBS 55 P. 71          010260
      C     NOTE THAT FOR THE                                               010270
      C     APPROXIMATIONS TO BE VALID, 0<C<=.693                           010280
            COMMON/VICTOR/V                                               010290
            DIMENSION AA(1)                                               010300
            REAL JI                                                       010310
            IER=1                                                         010320
            I=0                                                           010330
            F0=F1=0.                                                      010340
      110   IF(AA(I+1).EQ.0.) GOTO 2000                                   010350
            F0=F0+AA(I+1)*(C**I)*JI(0,I)                                  010360
            F1=F1+AA(I+1)*(C**I)*JI(1,I)                                  010370
            I=I+1                                                         010380
            GOTO 110                                                      010390
      2000  F0=F0/V**2                                                    010400
            F1=F1/V**2                                                    010410
            RETURN                                                        010420
      1000  IER=2                                                         010430
            RETURN                                                        010440
            END                                                           010450
```

B.13 Function JI

```
      C                                                                 010460
            REAL FUNCTION JI(T,N)                                       010470
            REAL N11,N22                                                010480
            COMMON/VICTOR/V                                             010490
      C     IF T=1. THEN THIS FUNCTION RETURNS%                         010500
      C     INTEGRAL((T**2N)JI(T)DT) OVER(0,V)                          010510
      C     IF T=0. THEN THIS FUNCTION RETURNS                          010520
      C     INTEGRAL((T**(2*N+1))J0(T)DT) OVER (0,V)                    010530
      C     SEE LUKE,Y.L. "INTEGRALS OF BESSEL FUNCTIONS"               010540
      C     MCGRAW @ HILL 1962 (P.51)                                   010550
      C     OR IBS 55 (P.480)                                           010560
            II=I+1                                                      010570
            GOTO (110,100) II                                           010580
          100 IF(N.NE.0) GOTO 120                                       010590
              JI=1.-BESJF(0)                                            010600
              RETURN                                                    010610
          120 N1=-N                                                     010620
              JI=2.*BESJF(2)                                            010630
              EXP=2*N                                                   010640
              K21=2                                                     010650
              GOTO 130                                                  010660
          110 N1=-(N+1)                                                 010670
              JI=BESJF(1)                                               010680
              EXP=2*N+1                                                 010690
              K21=1                                                     010700
          130 N11=N22=1                                                 010710
              N2=N+2                                                    010720
          140 N1=N1+1                                                   010730
              IF(N1.EQ.0) GOTO 2000                                     010740
              K21=K21+2                                                 010750
              N11=N11*N1                                                010760
              N22=N22*N2                                                010770
              N2=N2+1                                                   010780
              JI=JI+K21*N11/N22*BESJF(K21)                              010790
              GOTO 140                                                  010800
         2000 JI=V**EXP/(N+1)*JI                                        010810
              RETURN                                                    010820
              END                                                       010830
```

B.14 Function BESJF

```
      C                                                                 010840
            FUNCTION BESJF (M)                                          010850
            COMMON/VICTOR/V                                             010860
            COMMON/TFILES/IT3,IT4,IT5,IT6,IT7,IT8                       010870
            N=M                                                         010880
            D=.001                                                      010890
            CALL BESJ(V,N,BJ,D,IER)                                     010900
            IER=IER+1                                                   010910
            GOTO (10,20,30,40,50) IER                                   010920
          10  BESJF=BJ                                                 010930
              RETURN                                                    010940
          20  WRITE (IT5,200)                                          010950
         200  FORMAT(* ORDER OF BES FUN NEG,PROGRAM STOP*)              010960
              GOTO 1000                                                 010970
          30  WRITE (IT5,300)                                          010980
         300  FORMAT(* ARG OF BES FUN NEG OR ZERO,PROGRAM STOP*)        010990
              GOTO 1000                                                 011000
          40  WRITE (IT5,400)                                          011010
         400  FORMAT(* ACCURACY OF BES FUN NOT OBTAINED,PROGRAM CONTINUES*) 011020
              GOTO 10                                                   011030
          50  WRITE(IT5,500)                                           011040
         500  FORMAT(* RANGE ERROR IN BES FUN. RANGE ADJUSTED*)         011050
              GOTO 10                                                   011060
         1000 STOP                                                      011070
              END                                                       011080
```

B.15 Function BESJ

```
      SUBROUTINE BESJ(X,N,BJ,D,IER)                                     011130
C     ...............................................................   011140
C                                                                       011150
C        SUBROUTINE BESJ                                                011160
C                                                                       011170
C        PURPOSE                                                        011180
C           COMPUTE THE J BESSEL FUNCTION FOR A GIVEN ARGUMENT AND ORDER 011190
C                                                                       011200
C        USAGE                                                          011210
C           CALL BESJ(X,N,BJ,D,IER)                                     011220
C                                                                       011230
C        DESCRIPTION OF PARAMETERS                                      011240
C           X  -THE ARGUMENT OF THE J BESSEL FUNCTION DESIRED           011250
C           N  -THE ORDER OF THE J BESSEL FUNCTION DESIRED              011260
C           BJ -THE RESULTANT J BESSEL FUNCTION                         011270
C           D  -REQUIRED ACCURACY                                       011280
C           IER-RESULTANT ERROR CODE WHERE                              011290
C              IER=0  NO ERROR                                          011300
C              IER=1  N IS NEGATIVE                                     011310
C              IER=2  X IS NEGATIVE OR ZERO                             011320
C              IER=3  REQUIRED ACCURACY NOT OBTAINED                    011330
C              IER=4  RANGE OF N COMPARED TO X NOT CORRECT (SEE REMARKS) 011340
C                                                                       011350
C        REMARKS                                                        011360
C           N MUST BE GREATER THAN OR EQUAL TO ZERO, BUT IT MUST BE     011370
C           LESS THAN                                                   011380
C              20+10*X-X** 2/3   FOR X LESS THAN OR EQUAL TO 15         011390
C              90+X/2            FOR X GREATER THAN 15                  011400
C                                                                       011410
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED                  011420
C           NONE                                                        011430
C                                                                       011440
C        METHOD                                                         011450
C           RECURRENCE RELATION TECHNIQUE DESCRIBED BY H. GOLDSTEIN AND 011460
C           R.M. THALER,"RECURRENCE TECHNIQUES FOR THE CALCULATION OF   011470
C           BESSEL FUNCTIONS",M.T.A.C.,V.13,PP.102-108 AND I.A. STEGUN  011480
C           AND M. ABRAMOWITZ,"GENERATION OF BESSEL FUNCTIONS ON HIGH   011490
C           SPEED COMPUTERS",M.T.A.C.,V.11,1957,PP.255-257              011500
C                                                                       011510
C     ...............................................................   011520
C                                                                       011530
C                                                                       011540
      BJ=.0                                                             011550
      IF(D)10,20,20                                                     011560
   10 IER=1                                                             011570
      RETURN                                                            011580
   20 IF(X)30,30,31                                                     011590
   30 IER=2                                                             011600
      RETURN                                                            011610
   31 IF(X-15.)32,32,34                                                 011620
   32 NTEST=20.+10.*X-X** 2/3                                           011630
      GO TO 36                                                          011640
   34 NTEST=90.+X/2.                                                    011650
   36 IF(N-NTEST)40,38,38                                               011660
   38 IER=4                                                             011670
      N=NTEST-1                                                         011680
      GOTO 41                                                           011690
   40 IER=0                                                             011700
   41 N1=N+1                                                            011710
      BPREV=.0                                                          011720
C                                                                       011730
C     COMPUTE STARTING VALUE OF M                                       011740
C                                                                       011750
      IF(X-5.)50,60,60                                                  011760
   50 MA=X+6.                                                           011770
      GO TO 70                                                          011780
   60 MA=1.4*X+60./X                                                    011790
   70 MB=N+IFIX(X)/4+2                                                  011800
      MZERO=MAXO(MA,MB)                                                 011810
C                                                                       011820
C     SET UPPER LIMIT OF M                                              011830
C                                                                       011840
      MMAX=NTEST                                                        011850
  100 DO 190 M=MZERO,MMAX,3                                             011860
```

```
      C                                                                 011870
      C     SET F(M),F(M-1)                                             011880
      C                                                                 011890
            FM1=1.0E-28                                                 011900
            FM=.0                                                       011910
            ALPHA=.0                                                    011920
            IF(N-(M/2)*2)120,110,120                                    011930
        110 JT=-1                                                       011940
            GO TO 130                                                   011950
        120 JT=1                                                        011960
        130 M2=M-2                                                      011970
            DO 160 K=1,M2                                               011980
            MK=M-K                                                      011990
            BMK=2.*FLOAT(MK)*FM1/X-FM                                   012000
            FM=FM1                                                      012010
            FM1=BMK                                                     012020
            IF(MK-N-1)150,140,150                                       012030
        140 BJ=BMK                                                      012040
        150 JT=-JT                                                      012050
            S=1+JT                                                      012060
            S=1.*JT                                                     012070
        160 ALPHA=ALPHA+BMK*S                                           012080
            BMK=2.*FM1/X-FM                                             012090
            IF(N)180,170,180                                            012100
        170 BJ=BMK                                                      012110
        180 ALPHA=ALPHA+BMK                                             012120
            BJ=BJ/ALPHA                                                 012130
            IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190                      012140
        190 BPREV=BJ                                                    012150
            IER=3                                                       012160
        200 RETURN                                                      012170
            END                                                         012180
```

B.16 Subroutine GETDATA

```
            SUBROUTINE GETDATA(DATAIN,NV,IIN,IOUT1,IOUT2,IIN1,ISIZE     012190
           *,ISIZET,INDIC)                                              012200
      C THE MAIN PURPOSE OF THIS SUBROUTINE IS TO INPUT CHARACTER STRING OR 012210
      C NUMERICAL DATA IN A @CONVERSATIONAL@ MODE I.E. FOR INPUTTING DATA 012220
      C TO PROGRAMS BEING RUN UNDER INTERCOM.                           012230
      C IT ALSO MAY BE USED FOR BATCH PROCESSING-IN WHICH CASE THE DATA 012240
      C SHOULD APPEAR 6 VALUES TO A CARD. DATA WHICH IS NOT TO BE CHANGED 012250
      C SHOULD BE REPLACED BY BLANKS. FOR BATCH ALL OR SOME OF THE DATA MAY 012260
      C BE DEFAULTED BY USING AN EOR AFTER THE LAST DATA TO BE INPUTTED. 012270
      C THE SUBROUTINE ASSUMES THAT DEFAULT VALUES HAVE BEEN ASSIGNED   012280
      C AND WILL PRINT OUT THESE DEFAULT VALUES BEFORE ASKING FOR DATA INPUT. 012290
      C IT ASKS FOR NEW VALUES BY PRINTING OUT THE @NAMES@ OF THE DATA AND THEN 012300
      C SKIPPING A LINE. VALUES TO BE ASSIGNED TO THE NAMES SHOULD BE   012310
      C ENTERED STARTING IN THE SAME COLUMN AS THE START OF THE NAME.   012320
      C EACH DATUM IS ASSIGNED 10 COLUMNS AND UP TO 6 ITEMS MAY BE INPUTTED 012330
      C IN A SINGLES ROW.                                               012340
      C ARGUMENTS**********                                             012350
      C DATAIN  (DIMENSION (NV,3) WHERE NV IS THE TOTAL # OF DATA       012360
      C     TO BE INPUTTED)                                             012370
      C DATAIN HOLDS THE FOLLOWING INFORMATION ABOUT EACH DATUM-        012380
      C NAME,VALUE,CODE WHERE-                                          012390
      C NAME => NAME BY WHICH THE DATUM IS IDENTIFIED TO THE USER (IT MAY OR 012400
      C MAY NOT BE EQUAL TO THE FORTRAN VARIABLE NAME TO BE ASSIGNED TO THE 012410
      C DATUM.)                                                         012420
      C VALUE => NUMERICAL OR CHARACTER STRING VALUE TO BE ASSIGNED (THE DATUM) 012430
      C CODE => HOW THE DATUM IS TO BE INTERPRETED                      012440
      C    -1 => CHARACTER STRING                                       012450
      C     0 => INTEGER                                                012460
      C     1 => FLOATING POINT NUMBER                                  012470
      C NV TOTAL NUMBER OF DATUM TO BE INPUTTED                         012480
      C IIN FILE NO. FOR INPUTTING                                      012490
      C IOUT1 PRIMARY OUTPUT FILE                                       012500
      C IOUT2 SECONDARY OUTPUT FILE                                     012510
      C ISIZE =>  SIZE OF FIRST DIMENSION OF DATAIN                     012520
```

```
      DIMENSION DATAIN(ISIZE),IA(6)                                     012530
      COMMON/SENSE/IINNN,IOUTNN,INDICC                                  012540
      INTEGER DATAIN,F                                                  012550
      EXTERNAL SSWTCH                                                   012560
      CALL ERRSET(KOUNT,21000)                                          012570
      KOUNT1=KOUNT                                                      012580
      IF(INDIC.NE.0) 200,210                                            012590
  200 ISW=2                                                             012600
      LL=0                                                              012610
      L=INDIC-1                                                         012620
      GOTO 1055                                                         012630
  210 CONTINUE                                                          012640
      ISW=1                                                             012650
      IOUT=000000000000100053B                                          012660
      IINNN=IIN                                                         012670
      IOUTNN=IOUT1                                                      012680
      IOUTT=IOUT1                                                       012690
      ISW=1                                                             012700
      IBLANK=1H                                                         012710
      CALL SSWTCH(IIN1,ISW3,10HREAD DATA ,5HFILE-),RETURNS(1060)        012720
      IF (ISW3 .EQ. 1) 1300,1290                                        012730
 1300 WRITE (IOUT1,17) IIN1                                             012740
      REWIND IIN1                                                       012750
      READ(IIN1) DATAIN                                                 012760
      REWIND IIN1                                                       012770
      IF (EOF(IIN1)) 1400,1290                                          012780
 1400 WRITE(IOUT1,24) IIN1                                              012790
 1290 CONTINUE                                                          012800
      CALL SSWTCH(0,ISW4,10HDEFAULTS L,5HISTED),RETURNS(1060)           012810
      IF (ISW4 .NE. 1) GOTO 1140                                        012820
      WRITE(IOUT1,1)                                                    012830
 1140 DO 110 I=1,NV                                                     012840
      II=ISIZE+I                                                        012850
      II1=I                                                             012860
      II2=II+ISIZE                                                      012870
      IF (DATAIN(II2)) 1020,1030,1040                                   012880
 1020 WRITE(IOUTT,2) DATAIN(II),DATAIN(II1)                             012890
      GOTO 110                                                          012900
 1030 WRITE (IOUTT,3) DATAIN(II),DATAIN(II1)                            012910
      GOTO 110                                                          012920
 1040 WRITE(IOUTT,4) DATAIN(II),DATAIN(II1)                             012930
  110 CONTINUE                                                          012940
      GOTO (1150,1130) ISW                                              012950
 1150 CALL SSWTCH(0,ISW5,10HNAME-VALUE,5H MODE),RETURNS(1060)           012960
      IF (ISW5 .EQ. 1) 1270,1050                                        012970
 1050 L=1                                                               012980
      ISW=2                                                             012990
      LL=1                                                              013000
 1055 L=L+LL                                                            013010
      IF (L .GT. NV) GOTO 1060                                          013020
 1310 WRITE (IOUT1,18) DATAIN(ISIZE+L)                                  013030
      LL=1                                                              013040
      DO 100 J=1,6                                                      013050
      IA(J)=1OH                                                         013060
  100 CONTINUE                                                          013070
      READ(IIN,10) (IA(J),I=1,6)                                        013080
      IF (EOF(IIN)) 1320,1070                                           013090
 1320 INDICC=1                                                          013100
      GOTO 1060                                                         013110
 1070 IF (IA(1) .EQ. IBLANK) GOTO 1055                                  013120
      DO 180 J=1,6                                                      013130
      DO 180 K=1,10                                                     013140
      IF (MXGETX(IA(J),K,1) .EQ. IOUL) GOTO 1270                        013150
  180 CONTINUE                                                          013160
      DO 190 J=1,6                                                      013170
      JR=L+J-1                                                          013180
      F=DATAIN(JR+2*ISIZE)                                              013190
      IF (F) 1090,1100,1110                                             013200
 1090 IF (IA(J) .NE. IBLANK) DECODE(10,11,IA(J)) DATAIN(JR)             013210
      GOTO 1080                                                         013220
 1100 CALL RJUST(IA(J))                                                 013230
      IF (IA(J) .NE. IBLANK) DECODE(10,12,IA(J)) DATAIN(JR)             013240
      GOTO 1080                                                         013250
 1110 CALL RJUST(IA(J))                                                 013255
      IF (IA(J) .NE. IBLANK) DECODE(10,13,IA(J)) DATAIN(JR)             013260
      GOTO 1080                                                         013270
  190 CONTINUE                                                          013280
 1080 LL=1                                                              013290
      IF(INDIC.NE.0) LL=1000                                            013300
      IF (KOUNT .EQ. KOUNT1) GOTO 1055                                  013310
```

```
      KOUNT1=KOUNT                                                      013320
      WRITE(IOUT1,25)                                                   013330
1270  WRITE(IOUT1,23)                                                   013340
1250  WRITE(IOUT1,8)                                                    013350
      DO 150 I=1,6                                                      013360
      IA(I)=10H                                                         013370
150   CONTINUE                                                          013380
      READ(IIN,10) (IA(I),I=1,6)                                        013390
      IF (EOF(IIN)) 1330,1085                                           013400
1330  INDICC=1                                                          013410
      GOTO 1060                                                         013420
1085  II=IA(1)                                                          013430
      IF (II .EQ. IBLANK) GOTO 1060                                     013440
      DO 130 I=1,NV                                                     013450
      J=I*ISIZE                                                         013460
      IF (II .EQ. DATAIN(J)) GOTO 1160                                  013470
130   CONTINUE                                                          013480
      WRITE(IOUT1,16)                                                   013490
      GOTO 1270                                                         013500
1160  F=DATAIN(J+ISIZE)                                                 013510
      JJ=J-ISIZE                                                        013520
      IF (F) 1170,1180,1190                                             013530
1170  DO 160 I=2,6                                                      013540
      IF (IA(I) .EQ. IBLANK) GOTO 1240                                  013550
      DECODE(10,11,IA(I)) DATAIN(JJ+I-2)                                013560
160   CONTINUE                                                          013580
      GOTO 1240                                                         013590
1180  CALL RJUST(IA(2))                                                 013600
      DECODE(10,12,IA(2)) DATAIN(JJ)                                    013610
      GOTO 1240                                                         013615
1190  CALL RJUST(IA(2))                                                 013620
      DECODE(10,13,IA(2)) DATAIN(JJ)                                    013630
1240  IF (KOUNT .EQ. KOUNT1) GOTO (1250,1310) ISW                       013640
      KOUNT1=KOUNT                                                      013650
      WRITE(IOUT1,25)                                                   013660
      GOTO 1250                                                         013670
1060  WRITE (IOUT1,14)                                                  013680
      IF (IOUT2 .EQ. 0) GOTO 1130                                       013690
      WRITE (IOUT2,15)                                                  013700
      IOUT1=IOUT2                                                       013710
      ISW=2                                                             013720
      GOTO 1140                                                         013730
1130  INDICC=INDICC                                                     013740
      RETURN                                                            013750
1     FORMAT(/,* THE DEFAULT INPUT DATA ARE*)                           013760
2     FORMAT(1X,A10,*=@*,A10,*@*)                                       013770
3     FORMAT(1X,A10,*=*,I10)                                            013780
4     FORMAT(1X,A10,*=*,G16.6)                                          013790
5     FORMAT(///)                                                       013800
6     FORMAT(/,* ENTER DATA, START IN COL. BENEATH START OF NAME*)      013810
7     FORMAT(8X,6A10)                                                   013820
8     FORMAT(8X)                                                        013830
10    FORMAT(6A10)                                                      013840
11    FORMAT(A10)                                                       013850
12    FORMAT(I10)                                                       013860
13    FORMAT(E10.0)                                                     013870
14    FORMAT(/,* DATA INPUT COMPLETE*)                                  013880
15    FORMAT(/,* THE INPUT DATA VALUES ARE*,/)                          013890
16    FORMAT(1X,* TRY AGAIN*)                                           013900
17    FORMAT(/,1X,* READING DATA FROM *,I5)                             013910
18    FORMAT(1X,A10,*-*)                                                013920
23    FORMAT(8X,*NAME       VALUE.....*)                                013930
24    FORMAT(1X,*,FILE*,I5,* IS EMPTY*)                                 013940
25    FORMAT(1X,* WRONG DATA TYPE-TRY AGAIN*)                           013950
      END
```

B.17 Subroutine PRT

```
      SUBROUTINE PRT(BUF,MODE,II,IT,MP,NP,MSKIP,NSKIP,RHOPI,
     *DELRHO,VMIN,DELV,XIT,U,TR,T,T16)                                  015000
      DIMENSION R(100),BUF(1),D(2)                                      015010
      DATA ITT/0/                                                       015020
      IF (ITT .EQ. IT) GOTO 100                                         015030
      WRITE(IT4,1) TR,T                                                 015040
      DO 130 I=1,MP,MSKIP                                               015050
      GOTO (110,120) MODE                                               015060
  110 R(I)=RHOPI+DELRHO*(I-1)                                           015070
      GOTO 130                                                          015080
  120 R(I)=VMIN+DELV*(I-1)                                              015090
  130 CONTINUE                                                          015100
      GOTO (170,180) MODE                                               015110
  170 WRITE(IT4,2) (R(I),I=1,MP,MSKIP)                                  015120
      GOTO 100                                                          015130
  180 WRITE(IT4,3) (R(I),I=1,MP,MSKIP)                                  015140
  100 GOTO (140,150) MODE                                               015150.
  140 ENCODE(20,4,D) XIT                                                015160
      GOTO 160                                                          015170
  150 ENCODE(20,5,D) U                                                  015180
  160 IF (MOD(II-1,NSKIP) .EQ. 0) WRITE(IT6,6) D,                       015190
     *(BUF(J),I=1,MP,MSKIP)                                             015200
      ITT=IT                                                            015205
 1000 RETURN                                                            015210
    1 FORMAT(//* TIME(SECONDS)= *G12.4* TAU= *G12.4/                    015220
     **---------------------------------------*)                       015230
    2 FORMAT(/* INTENSITIES EVALUATED AT RHO-PRIME=*/(5X,5G13.5))       015240
    3 FORMAT(/* INTENSITIES EVALUATED AT V=*/(5X,5G13.5))               015250
    4 FORMAT(* X= *G13.5)                                               015260
    5 FORMAT(* U= *G13.5)                                               015270
    6 FORMAT(/2A10/(5X,5G13.5))                                         015280
      END                                                               015290
                                                                        015300
```

Appendix C

The Evaluation of $I'(u,0,t)$ and $I'(0,v,t)$ Used in the IKIRKP Option

1. INTRODUCTION

Consider the circumstance when the temperature distribution is constant throughout the window but is a function of time only, viz, w(t). Then, returning to Eq. (16), Volume I, we see that:

$$F1(\rho,t) = \int_{\zeta_1}^{\zeta_2} w(t)\, d\zeta = (\zeta_2-\zeta_1)\, w(t) = w(t)\, L/a = F1(t).$$

In a similar fashion, Eq. (17), Volume I, becomes:

$$F2(\rho,t) = \rho^{-2} \int_0^{\rho} d\rho\, \rho L\, w(t)/a = L\, w(t)/2a = F2(t).$$

Equation (18), Volume I, simplifies to:

$$\Phi^{\gamma}(\rho,t) = a\Delta T_c \left[S_1^{\gamma}\, w(t)\, L/a + 4\, S_2^{\gamma}\, w(t)\, L/2a\right]$$
$$= L\, w(t)\, \Delta T_c \left[S_1^{\gamma} + 2S_2^{\gamma}\right] \equiv d^{\gamma}(t).$$

2. EVALUATION OF I'(u,0,t)

For the condition v = 0 and u arbitrary, then $J_o(0)=1$ and $\lim_{v\to 0} J_1(\rho v)/\rho v = 1/2$. Equations (25-27), Volume I, reduce to:

$$f_z(\rho,0,t) = \rho\left[\exp(ikd^{\rho}) - \exp(ikd^{\theta})\right]/2 \equiv \rho A_1(t)/2\,,$$

$$f_x(\rho,0,t) = \rho \exp(ikd^{\rho}) - f_z \equiv \rho A_2(t) - \rho A_1(t)/2,$$

$$f_y(\rho,0,t) = \rho \exp(ikd^{\theta}) + f_z \equiv \rho A_3(t) + \rho A_1(t)/2.$$

Then, from Eq. (23), Volume I,

$$I'(u,0,t) = 2\alpha^4\left(1-\exp(-\alpha^2)\right)^{-2}\left\{\left|\int_0^1 \exp(-\alpha^2\rho^2)\exp(-iu\rho^2/2)(\rho A_2 - \rho A_1/2)\,d\rho\right|^2\right.$$
$$\left. + \left|\int_0^1 \exp(-\alpha^2\rho^2)\exp(-iu\rho^2/2)\,(\rho A_3 + \rho A_1/2)\,d\rho\right|^2\right\},$$

$$I'(u,0,t) = 2\alpha^4\left(1-\exp(-\alpha^2)\right)^{-2}\left\{\left|(A_2-A_1/2)\int_0^1 \rho\,\exp(-\alpha^2\rho^2)\exp(-iu\rho^2/2)\,d\rho\right|^2\right.$$

$$\left.+\left|(A_3+A_1/2)\int_0^1 \rho\exp(-\alpha^2\rho^2)\exp(-iu\rho^2/2)\,d\rho\right|^2\right\}.$$

Expressing the integral in terms of sin and cos, we find:

$$\int_0^1 \rho\exp(-\alpha^2\rho^2)\exp(-iu\rho^2/2)\,d\rho = \int_0^1 \rho\exp(-\alpha^2\rho^2)\cos(u\rho^2/2)\,d\rho$$

$$-i\int_0^1 \rho\exp(-\alpha^2\rho^2)\sin(u\rho^2/2)\,d\rho\,.$$

Let $a = -\alpha^2$, $b = v/2$ and $y = \rho^2$, to get:

$$\frac{1}{2}\int_0^1 e^{ay}\cos by\,dy - \frac{i}{2}\int_0^1 e^{ay}\sin by\,dy = \frac{1}{2}\left\{\frac{e^a(a\cos b+b\sin b)-a}{a^2+b^2}\right\} -$$

$$\frac{i}{2}\left\{\frac{e^a(a\sin b-b\cos b)+b}{a^2+b^2}\right\} \equiv A_4.$$

Thus, we get the final result:

$$I'(u,0,t) = 2\alpha^4\left(1-\exp(-\alpha^2)\right)^{-2}\left\{\left|(A_2-A_1/2)A_4\right|^2+\left|(A_3+A_1/2)A_4\right|^2\right\}.$$

3. EVALUATION OF I'(0,v,t)

Consider when u = 0 and v is arbitrary. Then,

$$f_w(\rho) = \exp(-\alpha^2\rho^2)$$

$$f_z(\rho,v,t) = v^{-1}J_1(\rho v)A_1(t)$$

$$f_x(\rho,v,t) = \rho J_o(\rho v)A_2(t) - f_z$$

$$f_y(\rho,v,t) = \rho J_o(\rho v)A_3(t) + f_z\,.$$

Inserting these functions into Eq. (23), Volume I, the integral terms become:

$$\left| A_2 \int_0^1 \rho J_o(\rho v) \exp(-\alpha^2\rho^2)\, d\rho - A_1 v^{-1} \int_0^1 J_1(\rho v) \exp(-\alpha^2\rho^2)\, d\rho \right|^2$$

$$+ \left| A_3 \int_0^1 \rho J_o(\rho v) \exp(-\alpha^2\rho^2)\, d\rho + A_1 v^{-1} \int_0^1 J_1(\rho v) \exp(-\alpha^2\rho^2)\, d\rho \right|^2 .$$

Thus, the integrals to consider are:

$$\int_0^1 \rho J_o(\rho v) \exp(-\alpha^2\rho^2)\, d\rho \quad ; \quad v^{-1} \int_0^1 J_1(\rho v) \exp(-\alpha^2\rho^2)\, d\rho$$

Let $y = \rho v$ and obtain:

$$v^{-2} \int_0^v y J_o(y) \exp(-\alpha^2 y^2/v^2)\, dy; \quad v^{-2} \int_0^v J_1(y) \exp(-\alpha^2 y^2/v^2)\, dy .$$

In order to perform these integrations, consider a polynomial approximation to $\exp(-\alpha^2 y^2/v^2)$. On p. 71 of Abramowitz and Stegun,[1] various approximations to e^{-x} are given, valid for the domain $0 \le x \le 0.693$. They all have the generic form: $e^{-x} = 1 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots$. Since $\alpha^2 y^2/v^2$ is a maximum at $y = v$ (where $\rho = 1$) for any value of v, we get the condition that $0 \le \alpha^2 \le 0.693$. Using the polynomial approximations mentioned above, the typical n-th term from each of the above two integrals will have the form:

$$a_n \alpha^{2n} v^{-(2n+2)} \int_0^v y^{2n+1} J_o(y)\, dy; \quad a_n \alpha^{2n} v^{-(2n+2)} \int_0^v y^{2n} J_1(y)\, dy.$$

On p. 480 of Abramowitz and Stegun[1] or p. 51 of Luke,[2] we find:

$$J_{i_o}^{2n+1}(v) \equiv \int_0^v y^{2n+1} J_o(y)\, dy = \frac{v^{2n+1}}{n+1} \sum_{k=o}^{\infty} \frac{(2k+1)(-n)_k}{(n+2)_k} J_{2k+1}(v)$$

---

1. Abramowitz, M., and Stegun, I.A., Editors (1964) Handbook of Mathematical Functions, National Bureau of Standards, Washington, D. C.
2. Luke, Y. L. (1962) Integrals of Bessel Functions, McGraw-Hill Co., New York.

$$J_{i_1}^{n}(v) \equiv \int_o^v y^n J_1(y)\, dy = \frac{v^{2n}}{n+1} \sum_{k=o} \frac{(2k+2)(1-n)_k}{(n+2)_k} J_{2k+2}(v) \qquad (n \neq 0)$$

and

$$\int_o^v J_1(y)\, dy = 1 - J_o(v)$$

where:

$(a)_o = 1$

$(a)_k = a \cdot (a+1) \cdot (a+2) \ldots (a+k-1)$

Since $(-n)_k = 0$ for $k = n+1$ and $(1-n)_k = 0$ for $k = n$, the series terminates.

The above results have been coded in a function subroutine JI (i, n) which returns $J_{i_o}^{2n+1}(v)$ if i = 0 and $J_{i_1}^{n}(v)$ if i = 1. It uses the IBM Scientific Subroutine BESJ to compute the $J_{2k+1}(v)$ and $J_{2k+2}(v)$.

Appendix D

Fortran Listings for the Main Programs in the Alternate TIKIRK Package Containing All Three Options

D.1 Modified Main Program TIKIRK

```
      PROGRAM TIKIRK(TAPE4=65,TAPE5=65,TAPE7=513,TAPE7=513,TAPE8=513,
     +TAP=6=65,OUTPUT=65)
C THIS PROGRAM CAN BEST BE DESCRIBED AS THE I/O INTERFACE FOR FUNCTION
C SUBROUTINE IKIRK WHICH COMPUTES THE KIRKHOFF INTENSITY FUNCTION AS
C DESCRIBED IN AFCRL-72-0565.
C    THE INPUT FALLS INTO THREE CLASSES%
C  1) INPUT HAVING TO DO WITH PROPERTIES OF THE WINDOW MATERIAL AND
C THE LASER BEAM,NAMELY% (QUANTITIES CGS UNLESS OTHERWISE INDICATED)
C SIG     =>  VALUE OF SIGMA IN GAUSSIAN BEAM
C LAMBDA  =>  WAVELENGTH OF THE LIGHT BEAM IN MICRONS
C PW      =>  TOTAL BEAM POWER
C P       =>  WINDOW RADIUS
C BETA    =>  BULK ABSORPTION COEFFICIENT
C K       =>  THERMAL CONDUCTIVITY
C NX      =>  INDEX OF REFRACTION
C S1P     =>  S SUB-1,SUB-PHO
C S1T     =>  S SUB-1,SUB-THETA
C S2P     =>  S SUB-2,SUB-PHO
C S2T     =>  S SUB-2,SUB-THETA
C T       =>  TIME AT WHICH IKIRK IS TO BE EVALUATED
C  2) INPUT HAVING TO DO WITH THE EVALUATION DOMAIN OF THE FUNCTION
C IKIRK,NAMELY%
C     XC    => GAUSSIAN FOCAL DISTANCE (METERS)
C     X1    => MINIMUM X-VALUE FOR FUNCTION EVALUATION (METERS)
C     X2    => MAXIMUM X-VALUE FOR FUNCTION EVALUATION (METERS)
C     RHOP1  => MINIMUM RADIUS VALUE FOR FUNCTION EVALUATION
C     RHOP2  => MAXIMUM RADIUS VALUE FOR FUNCTION EVALUATION
C     MP    => NUMBER OF EVALUATION POINTS IN THE RADIAL DIRECTION
C     NP    => NUMBER OF EVALUATION POINTS IN THE AXIAL (X) DIRECTION
C     TIM   => ARRAY (UP TO 10) OF TIME VALUES FOR FUNCTION EVALUATION
C        (TIME VALUES SHOULD BE IN INCREASING SEQUENCE)
C     UMIN  => MINIMUM U-VALUE FOR FUNCTION EVALUATION (SEE MODE)
C     UMAX  => MAXIMUM U-VALUE FOR FUNCTION EVALUATION
C     VMIN  => MINIMUM V-VALUE FOR FUNCTION EVALUATION
C     VMAX  => MAXIMUM V-VALUE FOR FUNCTION EVALUATION
C  3) INPUT HAVING TO DO WITH PROGRAM CONTROL,NAMELY%
C     EPST  => ERROR VALUE FOR INTERPOLATION OF THE TEMPERATURE
C FUNCTION OUTPUTTED BY TEMPS AND INTERPOLATED BY IBM SCI. SUB. ALI.
C     MINT  => NUMBER OF TEMPERATURE FUNCTION POINTS TO BE USED IN
C THE INTERPOLATION (DEFAULT=6)
C     IPRNT => USED TO CONTROL DEBUG OUTPUT (1 CAUSES DEBUG OUTPUT)
C              (2 CAUSES WINDOW TEMPERATURE DISTRIBUTION SUITABLE FOR
C              DISPLAY TO BE OUTPUT)
C     NGAUS => NUMBER OF FUNCTION VALUES FOR GAUSSIAN INTEGRATION
C     MODE  => IF MODE=1 THEN THE INTENSITY FUNCTION IS EVALUATED AT
C EQUI-SPACED X AND RHOO-PRIME VALUES% IF MODE=2 IT IS EVALUATED AT
C EQUI-SPACED U AND V VALUES.
C     IP    => IF 1 USE IKIRK. IF 2 USE IKIRKP
C              (NOTE THAT IKIRKP SHOULD ONLY BE USED ON THE AXES
C              FOR CONSTANT TEMPERATURE WINDOW)
C ALL THE ABOVE MENTIONED DATA IS OBTAINED BY TWO CALLS TO THE
C INTERACTIVE INPUT SUBROUTINE GETDATA DESCRIBED IN PML TM-16. IN THE
C FIRST CALL ALL DATA IN THE FIRST CATEGORY IS OBTAINED. IN THE
C SECOND CALL ALL DATA IN  THE SECOND AND THIRD CATEGORIES ARE
C OBTAINED.  AN EXCEPTION TO THIS IS IP (CONTROLS USE OF
C IKIRK AND IKIRKP) WHICH IS OBTAINED ON THE FIRST CALL TO GETDATA.
C  FOR A LISTING OF DEFAULT INPUT DATA IT IS RECOMMENDED THAT

C TIKIRK BE RUN INTERACTIVELY UNDER INTERCOM AFTER GIVING THE COMMAND
C CONNECT(TAPE4,TAPE5).
C THE MAIN OUTPUT OF TIKIRK IS A SEQUENCE OF UNFORMATTED RECORDS
C OF INTENSITY VALUES WITH CORRESPONDING DOMAIN VALUES. EACH RECORD
C CONSISTS OF THE FOLLOWING SEQUENCE OF VALUES%
C RECORD NO., NUMBER(MP) OF INTENSITY VALUES IN THE AXIAL DIRECTION,
C AXIAL COORDINATE X OR U, TIME VALUE (T) IN SECONDS, NUMBER (NP)
C OF INTENSITY VALUES IN THE RADIAL DIRECTION, MINIMUM RADIAL
C COORDINATE RHOP1 OR VMIN, MAXIMUM RADIAL COORDINATE RHOP2 OR
C VMAX, MP INTENSITY VALUES.
C FOR EACH VALUE OF T, NP RECORDS ARE OUTPUTTED CORRESPONDING TO THE
C NP X EVALUATION POINTS, THE RECORD NUMBER RUNS FROM 1 TO NP FOR
C EACH TIME VALUE.
C THERE ARE SIX FILES ASSOCIATED WITH THIS PROGRAM (NOT INCLUDING FILE
C 'OUTPUT'). THE FILES ARE REFERRED TO IN THE PROGRAM AND ASSOCIATED
C SUBROUTINES AS IT3,IT4,IT5,IT6,IT7,IT8. THE FILE VALUES ARE IN TURN
C ASSIGNED TO THE USUAL FORTRAN 'TAPEN' BY A DATA STATEMENT AND PROGRAM
```

```
C STATEMENT SUCH THAT IT3 (=> TAPE3,ETC. THUS TO CHANGE THE                    000840
C ASSOCIATION IT IS NECESSARY TO CHANGE  EITHER OR BOTH THE DATA AND           000850
C PROGRAM STATEMENTS.                                                          000860
C THESE FILES SERVE THE FOLLOWING PURPOSES%                                    000870
C   IT3 => FILE OUTPUTTED BY PROGRAM TEMP5                                     000880
C   IT4 => @INTERACTIVE@ INPUT FILE (SEE GETDATA)                              000890
C   IT5 => @INTERACTIVE@ OUTPUT FILE (SEE GETDATA)                             000900
C   IT6 => LISTING OF ALL INPUT PARAMETERS AND DEBUG OUTPUT                    000910
C   IT7 => UNFORMATTED INTENSITY VALUES. ALSO MAY BE USED TO INSERT            000920
C   IT8 => UNFORMATTED TEMPERATURE DISTRIBUTION VALUES                         000930
C          SUITABLE FOR DISPLAY PURPOSES                                       000940
C       PREASSIGNED DATA IN CATEGORIES 2 AND 3                                 000950
C IN ADDITION TO THE ABOVE FACTS, THE USER SHOULD BE AWARE OF TWO              000960
C PROGRAM @CONSTANTS@. THE FIRST UNDER THE VARIABLE NAME NT IS THE NUM-        000970
C BER OF TIME VALUES PERMITTED. AT PRESENT THIS IS SET TO 100 (THE             000980
C DIMENSION OF THE TIME ARRAY TIM). ALSO NOTE THAT ALL THE TIME                000990
C DEFAULT VALUES ARE ZERO EXCEPT THE FIRST AND THAT THE PROGRAM STOPS          001000
C AS SOON AS A SUCCEEDING TIME VALUE IS LESS THAN THE PRECEEDING               001010
C TIME VALUE.                                                                  001020
C THE SECOND @CONSTANT@ HAS TO DO WITH THE SIZE OF THE RECORD OUTPUTTED        001030
C BY TEMP5. THE SUBROUTINE RTAPE3 READS THE TEMP5 OUTPUT UNDER THE             001040
C ASSUMPTION THAT ALL @RADIAL@ ARRAYS ARE OF DIMENSION 82 AND @AXIAL@          001050
C ARRAYS ARE OF DIMENSION 22.SEE COMMENTS WITHIN SUBROUTINE RTAPE3.            001060
C THE INTENSITY FUNCTION IKIRK IS DEFINED EXPLICITLY AS A FUNCTION OF          001070
C THE NONDIMENSIONAL VARIABLES U AND V AND IMPLICITLY AS A FUNCTION            001080
C OF NON-DIMENSIONAL TIME TAU THROUGH THE TIME DEPENDANT FUNCTIONS             001090
C PHI-THETA AND PHI-RHO AS DEFINED IN THE ABOVE REFERENCE. THESE               001100
C VARIABLES ARE PASSED THROUGH AN ARGUMENT LIST. ALL @PARAMETERS@              001110
C REQUIRED FOR EVALUATION OF IKIRK ARE PASSED THROUGH BLOCK COMMON             001120
C .PHIBLK. THESE PARAMETERS ARE%                                               001130
C   CS1R   => C*S1R (SEE IKIRK COMMENTS)                                       001140
C   CS2R   => C*S2R          "                                                 001150
C   CS1I   => C*S1I          "                                                 001160
C   CS2I   => C*S2I          "                                                 001170
C   XS     => STARTING ARGUMENT FOR FUNCTIONS F1,F2 (SEE FUNCTION              001180
C             PHI COMMENTS)                                                    001190
C   DX     => INTERVAL BETWEEN EQUI-SPACED ARGUMENTS OF F1,F2.                 001200
C   NF     => NUMBER OF VALUES OF F1,F2                                        001210
C   MNNT   => MINT (SEE INPUT DATA)                                            001220
C   EPP    => EPS1          "                                                  001230
C   F1(200)=> HOLDS VALUES OF F1 FROM TEMP5                                    001240
C   F2(200)=> HOLDS VALUES OF F2 FROM TEMP5                                    001250
C   HOLD   => STORES PHI-THETA (PHI-RHO AND PHI-THETA ARE EVALUATED            001260
C                SIMULTANEOUSLY)                                               001270
C   A      => 1/SQRT(2)/SIG (=ALPHA IN THE ABOVE REFERENCE)                    001280
C   KE     => WAVE NUMBER                                                      001290
C   TLAST  => STORES TIME VALUE READ FROM TEMP5 RECORD                         001300
C   TNEXT  =>          "                                                       001310
C   IERR   => ERROR INDICATOR FOR RTAPE3 (INDICATES OUT OF RANGE               001320
C             TIME OR OUT OF SEQUENCE TIME)                                    001330
C   IP     => DEBUG OUTPUT @SWITCH@                                            001340
C   MP1    => =NF                                                              001350
C   ISW    => SWITCH FOR GAUSSIAN INTEGRATION. WHEN ISW=1 THEN THE             001360
C        X-VALUES FOR GAUSSIAN INTEGRATION ARE FOUND.                          001370
C   NGAUSS => NUMBER OF POINTS USED IN THE GAUSSIAN INTEGRATION                001380
C        (NOTE THAT IF NGAUSS IS CHANGED THEN THE GAUSSIAN INTEGRATION         001390
C        SUBROUTINE MUST ALSO BE CHANGED.)                                     001400
C   RAD    => WINDOW RADIUS                                                    001410
C   NP1    => NUMBER OF TEMPERATURE SAMPLES IN AXIAL DIRECTION                 001420
C             (USED FOR OUTPUTTING DISPLAY COMPATIBLE                          001430
C             TEMPERATURE DATA)                                                001440
C   C3     => CONSTANT TO DIMENSIONALIZE TEMPERATURE DATA                      001450
C             FOR DISPLAY                                                      001460
C   C1     => CONSTANT USED TO DIMENSIONALIZE TIME FOR                         001470
      REAL K,IKIRK,LAMBDA,NX,KE                                                001480
      REAL IKIRKP,IKIRKI                                                       001490
      LOGICAL S                                                                001500
      DIMENSION BUF(100)                                                       001510
      COMMON/IFILES/IT3,IT4,IT5,IT6,IT7,IT8                                    001520
      COMMON/PHIBLK/CS1R,CS2R,CS1I,CS2I,XS,DX,NF,MNNT,EPP,F1(200),             001530
     +F2(200),HOLD,A,KE,TLAST,TNEXT,IERR,IP,MP1,ISW,NGAUSS,                    001540
     +RAD,NP1,C3,C1,IERM,ERRORM                                                001550
      INTEGER DATAIN(100,3),DATAIN1(100,3)                                     001560
      COMMON/BLOCK2/X0,X1,X2,RHOP1,RHOP2,MP,NP,TIM(10),EPS1                    001570
C              TEMPERATURE DISPLAY                                             001580
     +,MINT,IPRNT,NGAUS,MODE,UMIN,UMAX,VMIN,VMAX,DUM(20),MSKIP,NSKIP           001590
      COMMON/BLOCK1/                                                           001600
     +      I1,I2,I3,I4,I5,I6,I7,M,N,MI,NI,ICNT,IU,IQ,NO,NMA,TRUN,            001610
     +ICARD,IPRINT,IPNCH,ITAP3,ITAP4,RHO1,RHO12,ZED1,ZED12,DTAU0,             001620
     +TAU,X,TAUOFF,SIG  ,00,U0,     EPS,G1(4),H1(4),MATER,NX,METÄ,            001630
```

```
      +K,LAMBDA,S1R,S1I,S2R,S2I,                                          001640
      +DEN,CR,R,EXPER,PW,R1,Z1,R2,IPLOT,PROBNO,TICU,XLEN,YLEN,SCALEX,     001650
      +SCALEY1,SCALEY2,XTITLE(5),YTITLE1(5),YTITLE2(5),NAME               001660
       EQUIVALENCE (I1,DATAIN),(XU,DATAIN1)                              001670
       DATA (DATAIN(I,1),I=1,92)/7*2.80,20,3*1,0,1,2,11,100,4,6,6,3,     001680
     +4.0,.1,.,.5546,.1,1092.,.0035,5.,.5,.,.1292,2*0.,.001,4*0.,.0,.3*.015,  001690
     +3HRCL,1,47,4,8E-4,,0653,10,6,-,34E-5,,05E-5,.1E-5,-.1E-5,.1,98,    001700
     +,691,1,2*8,1H1,24,7,91,11*1,1,4H7204,,5,20,,9,,30,,,1,,1,         001710
     +10HTIME(SECON,3HDS),3*1H ,10H(EMP-DEGC ,10H ABOVE AMB,             001720
     +3*1H ,10HMEAN TEMP ,10H ABOVE AMB,3*1H ,7HBARRETT,                 001730
     +1HRADIAL DIS,10HTANCE,RHO-,4H(CM),1H ,1H ,10HAXIAL DIST,           001740
     +10HANCE,X-(CM,1H),1H ,1H-/                                         001750
       DATA (DATAIN(I,2),I=1,92)/2HI1,2HI2,2HI3,2HI4,2HI5,              001760
     +2HI6,2HI7,1HM,1HN,2HNI,2HNI,4HICNT,2HIU,2HIQ,2HNO,3HNMX,          001770
     +4HIOU,5HICARD,6HIPRINT,5HIPNCH,5HITAP3,5HITAP4,4HRHO1,            001780
     +5HRHO2,4HZED1,5HZED2,5HUTAIU,5HTAUMX,6HTAUOFF,3HSIG,              001790
     +2HDX,2HUU,3HEPS,5HG1(1),5HG1(2),5HG1(3),5HG1(4),5HH1(1),          001800
     +5HH1(2),5HH1(3),5HH1(4),8HMATERIAL,8HDEF.IND.,4HBETA,             001810
     +5HTHED,COND,6HLAMBDA,3HS1R,3HS1I,3HS2R,3HS2I,                     001820
     +                7HDENSITY,9HSPEC.HEAT,6HRADIUS,5HEXPER,           001830
     +3HPWR      ,3HR1E,3H71*,3HR2*,9HPLOTTIY,2N,6HPROBNO,4HTICU,       001840
     +4HXLEN,4HYLEN,7HX-SCALE,8HY1-SCALE,8HY2-SCALE,6HXTITLE,           001850
     +1H2,1H3,1H4,1H5,7HYTITLE1,1H2,1H3,1H4,1H5,7HYTITLE2,1H2,          001860
     +1H3,1H4,1H5,8HOPERATOR,                                           001870
     +3HXT1,3HXT2,3HXT3,3HXT4,3HXT5,3HYT1,3HYT2,3HYT3,3HYT4,            001880
     +3HYT5/                                                            001890
       DATA (DATAIN(I,3),I=1,92)/22*0,19*1,-1,11*1,-1,1,4*0,            001900
     +-1,5*1,26*-1/                                                     001910
       DATA (DATAIN1(I,1),I=1,48)/1500.,1000.,2000.,0.,2.,100,100,10.,  001920
     +90.,.001,6.1,24,2.,-40.,40.,0.,10.,                               001930
     +10HRELPROBE I,10HNTENSITY F,9HUNCTION,I,1H ,1H ,10HTIME IN SE,    001940
     +5HCONDS,1H ,1H ,1H ,10HNON-DIMENS,10HIONAL RADI,10HAL DISTANC,    001950
     +3HE,V,1H ,10HNON-DIMENS,10HIONAL AXIA,10HL DISTANCE,2H=U,1H ,5,5/ 001960
       DATA (DATAIN1(I,2),I=1,48)/2HA0,2HX1,2HX2,5HRHOP1,5HRHOP2,2HMP,   001970
     +2HNP,2HT1,2HT2,2HT3,2HT4,2HT5,2HT6,2HT7,2HT8,2HT9,3HT10,          001980
     +4HEPS1,4HMINT,5HIPRNT,5HNGAUS,4HMODE,4HUMIN,4HUMAX,               001990
     +4HVMIN,4HVMAX,3HST1,3HST2,3HST3,3HST4,3HST5,3HPT1,3HPT2,          002000
     +3HPT3,3HPT4,3HPT5,3HYT1,3HXT2,3HXT3,3HXT4,3HXT5,3HYT1,            002010
     +3HYT2,3HYT3,3HYT4,3HYT5,5HMSKIP,5HNSKIP/                          002020
       DATA (DATAIN1(I,3),I=1,48)/5*1,2*0,11*1,4*0,4*1,20*-1,2*0/       002030
       DATA (IT3,IT4,IT5,IT6,IT7,IT8/3,4,5,6,7,8/                       002040
       DATA (ERM,ERROR)/0,1E-10/                                        002050
       INDTC=0                                                          002060
       S=.F.                                                            002070
       IS%=1                                                            002080
       NT=10                                                            002090
       CALL GETDATA(DATAIN,92,4,5,6,3,100,300,INDIC)                    002100
       CALL GETDATA(DATAIN1,48,4,5,6,7,100,300,INDIC)                   002110
       IF (MODE .NE. 1) GOTO 150                                        002120
       DATAIN1(37,1)=10HRADIAL DIS                                      002130
       DATAIN1(38,1)=10HTANCE,RHO-                                      002140
       DATAIN1(39,1)=10HRTME (CM)                                       002150
       DATAIN1(42,1)=10HAXIAL DIST                                      002160
       DATAIN1(43,1)=10HANCE,X (RE                                      002170
       DATAIN1(44,1)=10HLATIVE TO                                       002180
       DATAIN1(45,1)=10HGAUSS FOCU                                      002190
       DATAIN1(46,1)=7HS) (CM)                                          002200
  150  WRITE (IT7) DATAIN                                               002210
       WRITE (IT7) DATAIN1                                              002220
       IF (MODE .EQ. 1) S=.T.                                           002230
       REWIND IT3                                                       002240
       READ(IT3)                                                        002250
       TLAST=TNEXT=0.                                                   002260
       KF=4.2831AE4/LAMBDA                                              002270
       EPS=EPS1                                                         002280
       IP=IPRNT                                                         002290
       NGAUSS=NGAUS                                                     002300
       XS=0.                                                            002310
       DX=1./M                                                          002320
       MMINT=MINT                                                       002330
       MP1=M+1                                                          002340
       NF=MP1                                                           002350
       IF (S) GOTO 120                                                  002360
       XMAX=VMAX                                                        002370
       XMIN=VMIN                                                        002380
       GOTO 140                                                         002390
  120  XMIN=RHOP1                                                       002400
       XMAX=RHOP2                                                       002410
  140  CONTINUE                                                         002420
       A=.F/SIG                                                         002430
```

```
      C=P0*(NX**2+1)/2./NX                                              00002440
      C=C*R*BETA/3.14159/K                                              00002450
      CS1D=C*S1R                                                        00002460
      CS2D=C*S2R                                                        00002470
      CS1T=C*S1T                                                        00002480
      CS2T=C*S2T                                                        00002490
      RSQ=R*R                                                           00002500
      X0=X0*100.                                                        00002510
      X1=X1*100.                                                        00002520
      X2=X2*100.                                                        00002530
      C1=K/RSQ/DEN/CP                                                   00002540
      DELX=(X2-X1)/(AMAX0(1,NP-1))                                      00002550
      DELRHO=(RHOP2-RHOP1)/(AMAX0(1,MP-1))                              00002560
      DELV=(VMAX-VMIN)/(AMAX0(1,MP-1))                                  00002570
      DELU=(UMAX-UMIN)/(AMAX0(1,NP-1))                                  00002580
      RAD=R                                                             00002590
      NP1=N+1                                                           00002600
      C3=C/R                                                            00002610
      X0I=1./X0                                                         00002620
      NREC=1                                                            00002630
      IT=1                                                              00002640
      TR=T=TIM(1)                                                       00002650
      IF (T .LT. 0.) GOTO 2000                                          00002660
      IF(IP.EQ.2)  WRITE(ITR) DATAIN                                    00002670
 1000 T=T+C1                                                            00002680
      WRITE(ITE,2) T                                                    00002690
      IF (T .GT. TAUMX) GOTO 2000                                       00002700
      X1T=X1                                                            00002710
      X1I=1./X1T                                                        00002720
      U=UMIN                                                            00002730
      DO 100 I=1,NP                                                     00002740
      CX=(X0*X1I)**2                                                    00002745
      IF (S) U=KE*RS0*(X0I-X1I)                                         00002750
      RHOP=RHOP1                                                        00002760
      V=VMIN                                                            00002770
      DO 110 J=1,MP                                                     00002780
      IF (S) V=KE*R*X1I*RHOP                                            00002790
      BUF(J)=CX*IKIRK(U,V,T)                                            00002800
      IF (IERR .NE. 0) GOTO 2000                                        00002810
  110 RHOP=RHOP+DELRHO                                                  00002820
      V=V+DELV                                                          00002830
  100 CONTINUE                                                          00002840
      WRITE (ITT) I,NP,U,TR,MP,XMIN,XMAX,(BUF(J),J=1,MP)                00002850
      CALL PRT(BUF,MODE,I,IT,MP,NP,NSKIP,NSKIP,RHOP1,                   00002860
     +DELRHO,VMIN,DELV,X1T,U,TR,T,IT6)                                  00002870
      NREC=NREC+1                                                       00002880
      X1T=X1T+DELX                                                      00002890
      X1I=1./X1T                                                        00002900
      U=U+DELU                                                          00002910
  120 CONTINUE                                                          00002920
      IT=IT+1                                                           00002930
      IF (IT .GT. NT) GOTO 2000                                         00002940
      TR=T=TIM(IT)                                                      00002950
      IF (T .GT. TIM(IT-1)) GOTO 1000                                   00002960
 2000 WRITE(ITS,1) NREC                                                 00002970
      WRITE(ITS,6) IERM,ERRORM                                          00002980
    1 FORMAT(1X,* THE NUMBER OF RECORDS IS=*,I10)                       00002990
    2 FORMAT(1X,* NEW VALUE OF TAU IS*,E13.5)                           00003000
    3 FORMAT(1X,I6,4E12.4)                                              00003010
    4 FORMAT(1X,4E13.5)                                                 00003020
    5 FORMAT(1X,I6,4G12.4,3(/,5X,5G13.5))                               00003030
    6 FORMAT(* MAX ERROR PARM= *I4* EST ABS. ERROR INTEG.= *E16.4)      00003040
      END                                                               00003050
```

D.2 Major Function IKIRK Pertaining to Option No. 3 (IKIRK1)

```
      REAL FUNCTION IKIRK(U,V,T)                                        014800
      COMMON/PHIBLK/CS1P,CS2P,CS1T,CS2T,XS,DZ,NF,MINT,EPS,F1(200),      014810
     +F2(200),HOLD,A,K,TLAST,TNEXT,IERP,IP,MP1,ISW,N,                   014820
     +RAD,NP1,C3,C1,IERM,ERRORM                                         014830
      COMMON/UV/U1,V1                                                   014835
      EXTERNAL FREAL1,FIMAG1,FREAL2,FIMAG2                              014840
      DATA AERR,RERR/1E-3,1E-6/                                         014850
      U1=U                                                              014852
      V1=V                                                              014854
      A2=A*A                                                            014860
      IF (A2 .GT. 220.) 1020,1030                                       014870
 1020 CONST=2.*A2*A2                                                    014880
      GOTO 1040                                                         014890
 1030 CONST=2.*(A2/(1.-EXP(-A2)))**2                                    014900
 1040 CALL RTAPE3(T)                                                    014910
      IF (IERR .NE. 0) GOTO 2000                                        014920
      YRI=DCADRE(FREAL1,0.,1.,AERR,RERR,ERROR,IER)                      014930
      IF (ERROR .GT. ERRORM) ERRORM=ERROR                               014940
      IF (IER .GT. IERM) IERM=IER                                       014950
      YII=DCADRE(FIMAG1,0.,1.,AERR,RERR,ERROR,IER)                      014960
      IF (ERROR .GT. ERRORM) ERRORM=ERROR                               014970
      IF (IER .GT. IERM) IERM=IER                                       014980
      ZRI=DCADRE(FREAL2,0.,1.,AERR,RERR,ERROR,IER)                      014990
      IF (ERROR .GT. ERRORM) ERRORM=ERROR                               015000
      IF (IER .GT. IERM) IERM=IER                                       015010
      ZII=DCADRE(FIMAG2,0.,1.,AERR,RERR,ERROR,IER)                      015020
      IF (ERROR .GT. ERRORM) ERRORM=ERROR                               015030
      IF (IER .GT. IERM) IERM=IER                                       015040
      IKIRK=CONST*(YRI*YRI+YII*YII+ZRI*ZRI+ZII*ZII)                     015050
 2000 RETURN                                                            015060
      END                                                               015070
```

D.3 Function FREAL1

```
      FUNCTION FREAL1(X)                                                015080
      COMMON/UV/U,V                                                     015090
      COMMON/PHIBLK/CS1P,CS2P,CS1T,CS2T,XS,DZ,NF,MINT,EPS,F1(200),      015100
     +F2(200),HOLD,A,K,TLAST,TNEXT,IERP,IP,MP1,ISW,N,                   015110
     +RAD,NP1,C3,C1,IERM,ERRORM                                         015120
      COMPLEX Q1,EXPR,EXPT,FX,FY,FZ,FW                                  015130
      REAL J0,J1,K                                                      015140
      A2=A*A                                                            015150
      U2=.5*U                                                           015152
      ISW=1                                                             015154
      GOTO 100                                                          015156
      ENTRY FIMAG1                                                      015158
      ISW=2                                                             015160
      GOTO 100                                                          015162
      ENTRY FREAL2                                                      015164
      ISW=3                                                             015170
      GOTO 100                                                          015172
      ENTRY FIMAG2                                                      015174
      ISW=4                                                             015180
  100 X2=X*X                                                            015182
      XV=X*V                                                            015184
      FW=EXP(-A2*X2)*CEXP(CMPLX(0E0,-U2*X2))                            015190
      EXPR=CEXP(CMPLX(0E0,K*PHI(X)))                                    015200
      EXPT=CEXP(CMPLX(0E0,K*HOLD))                                      015210
      IF (V .EQ. 0.) FZ=(EXPR-EXPT)*X/2.                                015220
      IF (V .NE. 0.) FZ=(EXPR-EXPT)*J1(XV)/V                            015230
      Q1=X*J0(XV)                                                       015240
      GOTO (110,120,130,140) ISW                                        015242
  110 FX=Q1*EXPR-FZ                                                     015250
```

```
      FREAL1=REAL(FW*FX)                                                015255
      RETURN                                                            015257
  120 FX=A1*EXPP-FZ                                                     015260
      FREAL1=AIMAG(FW*FX)                                               015270
      RETURN                                                            015280
  130 FY=A1*EXPT+FZ                                                     015300
      FREAL1=REAL(FW*FY)                                                015310
      RETURN                                                            015320
  140 FY=A1*EXPT+FZ                                                     015340
      FREAL1=AIMAG(FW*FY)                                               015350
      RETURN                                                            015390
      END                                                               015400
```

D.4 Function DCADRE

```
      FUNCTION   DCADRE (F,A,B,AERR,RERR,ERROR,IER)                     015450
C                                                                       015460
C-DCADRE-------S-------LIBRARY 3-------------------------------------- 015470
C                                                                       015480
C   FUNCTION              - INTEGRATE F(X) FROM A TO B, USING CAUTIOUS   015490
C                             ADAPTIVE ROMBERG EXTRAPOLATION.           015500
C   USAGE                 - FUNCTION DCADRE (F,A,B,AERR,RERR,ERROR,IER) 015510
C   PARAMETERS   DCADRE   - ESTIMATE OF THE INTEGRAL OF F(X) FROM A TO B. 015520
C                F        - A SINGLE-ARGUMENT REAL FUNCTION SUBPROGRAM   015530
C                             SUPPLIED BY THE USER.  F MUST BE DECLARED   015540
C                             EXTERNAL IN THE CALLING PROGRAM.            015550
C                A,B      - THE TWO ENDPOINTS OF THE INTERVAL OF         015560
C                             INTEGRATION.  (INPUT)                       015570
C                AERR     - DESIRED ABSOLUTE ERROR IN THE ANSWER.  (INPUT) 015580
C                RERR     - DESIRED RELATIVE ERROR IN THE ANSWER.  (INPUT) 015590
C                ERROR    - ESTIMATED BOUND ON THE ABSOLUTE ERROR OF     015600
C                             THE OUTPUT NUMBER, DCADRE.                  015610
C                IER      - ERROR PARAMETER                              015620
C                           WARNING ERROR(WITH FIX) = 64 + N              015630
C                             N = 1 IMPLIES THAT ONE OR MORE SINGULARITIES 015640
C                               WERE SUCCESSFULLY HANDLED.                015650
C                             N = 2 IMPLIES THAT, IN SOME SUBINTERVAL(S),  015660
C                               THE ESTIMATE OF THE INTEGRAL WAS ACCEPTED  015670
C                               MERELY BECAUSE THE ESTIMATED ERROR WAS     015680
C                               SMALL, EVEN THOUGH NO REGULAR BEHAVIOR     015690
C                               WAS RECOGNIZED.                           015700
C                           TERMINAL ERROR = 128 + N                      015710
C                             N = 3 -- FAILURE DUE TO INSUFFICIENT        015720
C                               INTERNAL WORKING STORAGE.                 015730
C                             N = 4 -- FAILURE.  THIS MAY BE DUE TO TOO   015740
C                               MUCH NOISE IN THE FUNCTION (RELATIVE      015750
C                               TO THE GIVEN ERROR REQUIREMENTS) OR       015760
C                               DUE TO AN ILL-BEHAVED INTEGRAND.          015770
C                             N = 5 INDICATES THAT RERR IS GREATER THAN   015780
C                               0.1, OR RERR IS LESS THAN 0.0, OR RERR    015790
C                               IS TOO SMALL FOR THE PRECISION OF THE     015800
C                               MACHINE.                                  015810
C   PRECISION             - SINGLE                                        015820
C   REQD. IMSL ROUTINES   - UERTST                                        015830
C   LANGUAGE              - FORTRAN                                       015840
C-------------------------------------------------------------------------015850
C   LATEST REVISION       - SEPTEMBER 17, 1974                            015860
C                                                                       015870
      DIMENSION            T(10,10),R(10),AIT(10),DIF(10),RN(4),TS(2049) 015880
      DIMENSION            IREGS(30),BEGIN(30),FINIS(30),EST(30)         015890
      DIMENSION            REGLSV(30)                                    015900
      LOGICAL              H2CONV,AITKEN,RIGHT,REGLAR,REGLSV             015910
      REAL                 LENGTH,JUMPTL                                 015920
      DATA                 AITLOW,H2TOL,AITTOL,JUMPTL,MAXTS,MAXTRL,MXSTGE 015930
     1                     /1.1,.15,.1,.01,2049,10,30/                   015940
      DATA                 RN(1),RN(2),RN(3),RN(4)/                      015950
     1                     .7142005,.3466282,.843751,.1263305/           015960
      DATA                 ZERO,PI,HALF,ONE,TWO,FOUR,FOURPS,TEN,HUN      015970
     1                     /0.0,0.1,0.5,1.0,2.0,4.0,4.5,10.0,100.0/      015980
      ALG402 = ALOG10(TWO)                                              015990
      CADRE = ZERO                                                      016000
      ERROR = ZERO                                                      016010
```

```
      CURFST = ZERO                                                     016020
      VINT = ZERO                                                       016030
      ICP = 0
   60 LENGTH = ABS(B-A)                                                 016050
      IF (LENGTH .EQ. ZERO) GO TO 215                                   016060
      IF (RERR .GT. PI .OR. RERR .LT. ZERO) GO TO 210                   016070
      IF (AERR .EQ. ZERO .AND. (RERR*HUN) .LE. HUN) GO TO 210           016080
      ERRD = RERR                                                       016090
   65 ERRA = ABS(AERR)                                                  016100
      STEPMN = (LENGTH/FLOAT(2**MXSIGE))                                016110
      STEPNM = AMAX1(LENGTH,ABS(A),ABS(B))*TEN                          016120
      STAGE = HALF                                                      016130
   70 ISTAGE = 1                                                        016140
      ENSTZE = ZERO                                                     016150
      PREVER = ZERO                                                     016160
      REGLAR = .FALSE.                                                  016170
C                                          THE GIVEN INTERVAL OF INTEGRATION  016180
   75 C                                      IS THE FIRST INTERVAL CONSIDERED. 016190
      BEG = A                                                           016200
      FBEG = F(BEG)*HALF                                                016210
      TS(1) = FBEG                                                      016220
      IBEG = 1                                                          016230
   80 END = B                                                           016240
      FEND = F(END)*HALF                                                016250
      TS(2) = FEND                                                      016260
      IEND = 2                                                          016270
    5 RIGHT = .FALSE.                                                   016280
C                                          INVESTIGATION OF A PARTICULAR     016290
   85 C                                      SUBINTERVAL BEGINS AT THIS POINT. 016300
C                                                                       016310
   10 STEP = END - BEG                                                  016320
      ASTEP = ABS(STEP)                                                 016330
      IF (ASTEP .LT. STEPMN) GO TO 205                                  016340
   90 IF (STEPNM+ASTEP .EQ. STEPNM) GO TO 205                           016350
      T(1,1) = FBEG + FEND                                              016360
      TABS = ABS(FBEG) + ABS(FEND)                                      016370
      L = 1                                                             016380
      N = 1                                                             016390
   95 HPCONV = .FALSE.                                                  016400
      ATTKEN = .FALSE.                                                  016410
   15 LM1 = L                                                           016420
      L = L + 1                                                         016430
C                                          CALCULATE THE NEXT TRAPEZOID SUM  016440
  100 C                                      T(L,1), WHICH IS BASED ON *N2* + 1 016450
C                                            EQUISPACED POINTS. HERE         016460
C                                            N2 = N*2 = 2**(L-1).            016470
      N2 = N+N                                                          016480
      FN = N2                                                           016490
  105 ISTEP = (IEND - IBEG)/N                                           016500
      IF (ISTEP .GT. 1) GO TO 25                                        016510
      II = IEND                                                         016520
      IEND = IEND + N                                                   016530
      IF (IEND .GT. MAXTS) GO TO 200                                    016540
  110 HOVN = STEP/FN                                                    016550
      III = IEND                                                        016560
      FI = ONE                                                          016570
      DO 20 I=1,N2,2                                                    016580
         TS(III) = TS(II)                                               016590
  115    TS(III-1) = F(END - FI * HOVN)                                 016600
         FI = FI+TWO                                                    016610
         III = III-2                                                    016620
         II = II-1                                                      016630
   20 CONTINUE                                                          016640
  120 ISTEP = 2                                                         016650
   25 ISTEP2 = IBEG + ISTEP/2                                           016660
      SUM = ZERO                                                        016670
      SUMABS = ZERO                                                     016680
      DO 30 I=ISTEP2,IEND,ISTEP                                         016690
  125    SUM = SUM + TS(I)                                              016700
         SUMABS = SUMABS + ABS(TS(I))                                   016710
   30 CONTINUE                                                          016720
      T(L,1) = T(L-1,1)*HALF+SUM/FN                                     016730
      TABS = TABS*HALF+SUMABS/FN                                        016740
  130 ABST = ASTEP*TABS                                                 016750
      N = N2
```

```
C                                  GET PRELIMINARY VALUE FOR *VINT*          016760
C                                    FROM LAST TRAPEZOID SUM AND UPDATE      016770
C                                    THE ERROR REQUIREMENT *ERGOAL*          016780
C                                    FOR THIS SUBINTERVAL.                   016790
      IT = 1                                                                 016800
      VINT = STEP*T(L,1)                                                     016810
      TABTLM = TABS*TEN                                                      016820
      ENSIZE = AMAX1(ENSIZE,ABS(T(L,1)))                                     016830
      ERGL = ASTEP*ENSIZE*TEN                                                016840
      ERGOAL = STAGE*AMAX1(ERRA*EROR*ABS(CUREST+VINT))                       016850
C                                  COMPLETE ROW L AND COLUMN L OF *T*        016860
C                                    ARRAY.                                  016870
      FEXTRP = ONE                                                           016880
      DO 35 I=1,LM1                                                          016890
         FEXTRP = FEXTRP*FOUR                                                016900
         T(I,L) = T(L,I) - T(L-1,I)                                          016910
         T(L,I+1) = T(L,I) + T(I,L)/(FEXTRP-ONE)                             016920
   35 CONTINUE                                                               016930
      ERRER = ASTEP*ABS(T(1,L))                                              016940
C                                  PRELIMINARY DECISION PROCEDURE            016950
C                                    IF L = 2 AND T(2,1) = T(1,1),           016960
C                                    GO TO 135 TO FOLLOW UP THE              016970
C                                    IMPRESSION THAT INTERGRAND IS           016980
C                                    STRAIGHT LINE.                          016990
      IF (L .GT. 2) GO TO 40                                                 017000
      IF (TABS+P1*ABS(T(1,2)) .EQ. TABS) GO TO 135                           017010
      GO TO 15                                                               017020
C                                  CACULATE NEXT RATIOS FOR                  017030
C                                    COLUMNS 1,...,L-2 OF T-TABLE            017040
C                                    RATIO IS SET TO ZERO IF DIFFERENCE      017050
C                                    IN LAST TWO ENTRIES OF COLUMN IS        017060
C                                    ABOUT ZERO                              017070
   40 DO 45 I=2,LM1                                                          017080
         DIFF = ZERO                                                         017090
         IF (TABTLM+ABS(T(I-1,L)).NE.TABTLM) DIFF = T(I-1,LM1)/T(I-1,L)      017100
         T(I-1,LM1) = DIFF                                                   017110
   45 CONTINUE                                                               017120
      IF (ABS(FOUR-T(1,LM1)) .LE. RTOL) GO TO 60                             017130
      IF (T(1,LM1) .EQ. ZERO) GO TO 55                                       017140
      IF (ABS(TWO-ABS(T(1,LM1))) .LT. JUMPTL) GO TO 130                      017150
C                                                                            017160
      IF (L .EQ. 3) GO TO 15                                                 017170
      H2CONV = .FALSE.                                                       017180
      IF (ABS((T(1,LM1)-T(1,L-2))/T(1,LM1)) .LE. AITTOL) GO TO 75            017190
   50 IF (REGLAR) GO TO 55                                                   017200
      IF (L .EQ. 4) GO TO 15                                                 017210
   55 IF (ERRER .GT. ERGOAL .AND. (ERGL+ERRER) .NE. ERGL) GO TO 175          017220
      GO TO 145                                                              017230
C                                  CAUTIOUS ROMBERG EXTRAPOLATION            017240
   60 IF (H2CONV) GO TO 65                                                   017250
      AITKEN = .FALSE.                                                       017260
      H2CONV = .TRUE.                                                        017270
   65 FEXTRP = FOUR                                                          017280
   70 IT = IT + 1                                                            017290
      VINT = STEP*T(L,IT)                                                    017300
      ERRER = ABS(STEP/(FEXTRP-ONE)*T(IT-1,L))                               017310
      IF (ERRER .LE. ERGOAL) GO TO 160                                       017320
      IF (ERGL+ERRER .EQ. ERGL) GO TO 160                                    017330
      IF (IT .EQ. LM1) GO TO 125                                             017340
      IF (T(IT,LM1) .EQ. ZERO) GO TO 70                                      017350
      IF (T(IT,LM1) .LE. FEXTRP) GO TO 125                                   017360
      IF (ABS(T(IT,LM1)/FOUR-FEXTRP)/FEXTRP .LT. AITTOL)                     017370
     1    FEXTRP = FEXTRP*FOUR                                               017380
      GO TO 70                                                               017390
C                                  INTEGRAND MAY HAVE X**ALPHA TYPE          017400
C                                    SINGULARITY                             017410
C                                    RESULTING IN A RATIO OF *SING* =        017420
C                                    2**(ALPHA + 1)                          017430
   75 IF (T(1,LM1) .LT. AITLOW) GO TO 175                                    017440
      IF (AITKEN) GO TO 80                                                   017450
      H2CONV = .FALSE.                                                       017460
      AITKEN = .TRUE.                                                        017470
   80 FEXTRP = T(L-2,LM1)                                                    017480
      IF (FEXTRP .GT. FOUR05) GO TO 65                                       017490
      IF (FEXTRP .LT. AITLOW) GO TO 175                                      017500
      IF (ABS(FEXTRP-T(L-3,LM1))/T(1,LM1) .GT. H2TOL) GO TO 175              017510
      SING = FEXTRP                                                          017520
      FEXTM1 = ONE/(FEXTRP - ONE)                                            017530
      AIT(1) = ZERO
```

```
      DO 85 I=2,L                                                       017540
        AIT(I) = T(I,1) + (T(I,1)-T(I-1,1))*FEXTM1                      017550
        R(I) = T(I,I-1)                                                 017560
        DIF(I) = AIT(I) - AIT(I-1)                                      017570
   85 CONTINUE                                                          017580
      IT = 2                                                            017590
   90 VINT = STEP*AIT(L)                                                017600
      ERRER = ERRER*FEXTM1                                              017610
      IF (ERRER .GT. ERGOAL .AND. (ERGL+ERRER) .NE. ERGL) GO TO 95      017620
      ALPHA = ALOG10(SING)/ALG402 - ONE                                 017630
      IER = MAX0(IER,85)                                                017640
      GO TO 160                                                         017650
   95 IT = IT + 1                                                       017660
      IF (IT .EQ. LM1) GO TO 125                                        017670
      IF (IT .GT. 3) GO TO 100                                          017680
      H2NEXT = FOUR                                                     017690
      SINGNX = SING*SING                                                017700
  100 IF (H2NEXT .LT. SINGNX) GO TO 105                                 017710
      FEXTRP = SINGNX                                                   017720
      SINGNX = SINGNX*SINGNX                                            017730
      GO TO 110                                                         017740
  105 FEXTRP = H2NEXT                                                   017750
      H2NEXT = FOUR*H2NEXT                                              017760
  110 DO 115 I=IT,LM1                                                   017770
        R(I+1) = ZERO                                                   017780
        IF (TABTLM+ABS(DIF(I+1)) .NE. TABTLM) R(I+1) = DIF(I)/DIF(I+1)  017790
  115 CONTINUE                                                          017800
      H2TFEX = -H2TOL*FEXTRP                                            017810
      IF (R(L) - FEXTRP .LT. H2TFEX) GO TO 125                          017820
      IF (R(L-1)-FEXTRP .LT. H2TFEX) GO TO 125                          017830
      ERRER = ASTEP*ABS(DIF(L))                                         017840
      FEXTM1 = ONE/(FEXTRP - ONE)                                       017850
      DO 120 I=IT,L                                                     017860
        AIT(I) = AIT(I) + DIF(I)*FEXTM1                                 017870
        DIF(I) = AIT(I) - AIT(I-1)                                      017880
  120 CONTINUE                                                          017890
      GO TO 90                                                          017900
C                                     CURRENT TRAPEZOID SUM AND RESULTING 017910
C                                       EXTRAPOLATED VALUES DID NOT GIVE  017920
C                                       A SMALL ENOUGH *ERRER*.           017930
C                                       NOTE -- HAVING PREVER .LT. ERRER  017940
C                                       IS AN ALMOST CERTAIN SIGN OF      017950
C                                       BEGINNING TROUBLE WITH IN THE FUNC- 017960
C                                       TION VALUES. HENCE, A WATCH FOR,  017970
C                                       AND CONTROL OF, NOISE SHOULD      017980
C                                       BEGIN HERE.                       017990
  125 FEXTRP = AMAX1(PREVER/ERRER,AITLOW)                               018000
      PREVER = ERRER                                                    018010
      IF (L .LT. 5) GO TO 15                                            018020
      IF (L-IT .GT. 2 .AND. ISTAGE .LT. MXSTGE) GO TO 170               018030
      ERRET = ERRER/(FEXTRP**(MAXTRL-L))                                018040
      IF (ERRET .GT. ERGOAL .AND. (ERGL+ERRET) .NE. ERGL) GO TO 170     018050
      GO TO 15                                                          018060
C                                     INTEGRAND HAS JUMP (SEE NOTES)    018070
  130 IF (ERRER .GT. ERGOAL .AND. (ERGL+ERRER) .NE. ERGL) GO TO 170     018080
C                                     NOTE THAT  2*FN = 2**L            018090
      DIFF = ABS(T(1,L))*(FN+FN)                                        018100
      GO TO 160                                                         018110
C                                     INTEGRAND IS STRAIGHT LINE        018120
C                                       TEST THIS ASSUMPTION BY COMPARING 018130
C                                       THE VALUE OF THE INTEGRAND AT     018140
C                                       FOUR *RANDOMLY CHOSEN* POINTS WITH 018150
C                                       THE VALUE OF THE STRAIGHT LINE    018160
C                                       INTERPOLATING THE INTEGRAND AT THE 018170
C                                       TWO END POINTS OF THE SUB-INTERVAL. 018180
C                                       IF TEST IS PASSED, ACCEPT *VINT*  018190
  135 SLOPE = (FEND-FBEG)*TWO                                           018200
      FBEG2 = FBEG+FBEG                                                 018210
      DO 140 I=1,4                                                      018220
        DIFF = ABS(F(BEG+RN(I)*STEP) - FBEG2-RN(I)*SLOPE)               018230
        IF (TABTLM+DIFF .NE. TABTLM) GO TO 155                          018240
  140 CONTINUE                                                          018250
      GO TO 160                                                         018260
C                                     NOISE MAY BE DOMINANT FEATURE     018270
C                                       ESTIMATE NOISE LEVEL BY COMPARING 018280
C                                       THE VALUE OF THE INTEGRAND AT     018290
```

```
C                                       FOUR *RANDOMLY CHOSEN* POINTS WITH      018300
C                                       THE VALUE OF THE STRAIGHT LINE          018310
C                                       INTERPOLATING THE INTEGRAND AT THE      018320
C                                       TWO ENDPOINTS. IF SMALL ENOUGH,         018330
C                                       ACCEPT *VINT*                           018340
C                                                                               018350
  145 SLOPE = (FEND-FBEG)*TWO                                                   018360
      FBEG2 = FBEG+FBEG                                                         018370
      I = 1                                                                     018380
  150 DIFF = ABS(F(BEG+RN(I)*STEP) - FBEG2-RN(I)*SLOPE)                         018390
  155 ERRER = AMAX1(ERRER,ASTEP*DIFF)                                           018400
      IF (ERRER .GT. ERROAL .AND. (ERGL+ERRER) .NE. EPGL) GO TO 175             018410
      I = I+1                                                                   018420
      IF (I .LE. 4) GO TO 150                                                   018430
      IER = 66                                                                  018440
C                                       INTERGRATION OVER CURRENT SUB-          018450
C                                         INTERVAL SUCCESSFUL                   018460
C                                         ADD *VINT* TO *DCADRE* AND *ERRER*    018470
C                                         TO *ERROR*, THEN SET UP NEXT SUB-     018480
C                                         INTERVAL, IF ANY.                     018490
  160 CADRE = CADRE + VINT                                                      018500
      ERROR = ERROR + ERREP                                                     018510
      IF (RIGHT) GO TO 165                                                      018520
      ISTAGE = ISTAGE - 1                                                       018530
      IF (ISTAGE .EQ. 0) GO TO 220                                              018540
      REGLAR = REGLSV(ISTAGE)                                                   018550
      BEG = BEGIN(ISTAGE)                                                       018560
      END = ENDS(ISTAGE)                                                        018570
      CUREST = CUREST - EST(ISTAGE+1) + VINT                                    018580
      IEND = IBEG - 1                                                           018590
      FEND = TS(IEND)                                                           018600
      IBEG = IBEGS(ISTAGE)                                                      018610
      GO TO 180                                                                 018620
  165 CUREST = CUREST + VINT                                                    018630
      STAGE = STAGE+STAGE                                                       018640
      IEND = IBEG                                                               018650
      IBEG = IBEGS(ISTAGE)                                                      018660
      END = BEG                                                                 018670
      BEG = BEGIN(ISTAGE)                                                       018680
      FEND = FBEG                                                               018690
      FBEG = TS(IBEG)                                                           018700
      GO TO 5                                                                   018710
C                                       INTEGRATION OVER CURRENT SUBINTERVAL    018720
C                                         IS UNSUCCESSFUL. MARK SUBINTERVAL     018730
C                                         FOR FURTHER SUBDIVISION. SET UP       018740
C                                         NEXT SUBINTERVAL.                     018750
  170 REGLAR = .TRUE.                                                           018760
  175 IF (ISTAGE .EQ. MXSTGE) GO TO 205                                         018770
      IF (RIGHT) GO TO 185                                                      018780
      REGLSV(ISTAGE+1) = REGLAR                                                 018790
      BEGIN(ISTAGE) = BEG                                                       018800
      IBEGS(ISTAGE) = IBEG                                                      018810
      STAGE = STAGE*HALF                                                        018820
  180 RIGHT = .TRUE.                                                            018830
      BEG = (BEG+END)*HALF                                                      018840
      IBEG = (IBEG+IEND)/2                                                      018850
      TS(IBEG) = TS(IBEG)*HALF                                                  018860
      FBEG = TS(IBEG)                                                           018870
      GO TO 10                                                                  018880
  185 NNLEFT = IBEG - IBEGS(ISTAGE)                                             018890
      IF (IEND+NNLEFT .GE. MAXTS) GO TO 200                                     018900
      III = IBEGS(ISTAGE)                                                       018910
      II = IEND                                                                 018920
      DO 190 I=III,IBEG                                                         018930
         II = II + 1                                                            018940
         TS(II) = TS(I)                                                         018950
  190 CONTINUE                                                                  018960
      DO 195 I=IBEG,II                                                          018970
         TS(III) = TS(I)                                                        018980
         III = III + 1                                                          018990
  195 CONTINUE                                                                  019000
      IEND = IEND + 1                                                           019010
      IBEG = IEND - NNLEFT                                                      019020
      FEND = FBEG                                                               019030
      FBEG = TS(IBEG)                                                           019040
      ENDS(ISTAGE) = END                                                        019050
      END = BEG                                                                 019060
      BEG = BEGIN(ISTAGE)                                                       019070
      BEGIN(ISTAGE) = END                                                       019080
      REGLSV(ISTAGE) = REGLAR
```

```
      ISTAGE = ISTAGE + 1                                                   019090
      REGLAR = REGLSV(ISTAGE)                                               019100
      EST(ISTAGE) = VINT                                                    019110
      CUREST = CUREST + EST(ISTAGE)                                         019120
      GO TO 5                                                               019130
C                                     FAILURE TO HANDLE GIVEN INTEGRA-      019140
C                                       TION PROBLEM                        019150
C                                                                           019160
  200 IER = 131                                                             019170
      GO TO 215                                                             019180
  205 IER = 132                                                             019190
      GO TO 215                                                             019200
  210 IER = 133                                                             019210
  215 CADRE = CUREST + VINT                                                 019220
  220 DCADRE = CADRE                                                        019230
 9000 CONTINUE                                                              019240
      IF (IER .NE. 0) CALL UERTST (IER,6HDCADRE)                            019250
 9005 RETURN                                                                019260
      END
```

D.5 Subroutine UERTST

```
      SUBROUTINE UERTST (IER,NAME)                                          019270
C                                                                           019280
C-UERTST----------LIBRARY 3-------------------------------------------     019290
C                                                                           019300
C   FUNCTION            - ERROR MESSAGE GENERATION                        019310
C   USAGE               - CALL UERTST (IER,NAME)                          019320
C   PARAMETERS   IER    - ERROR PARAMETER. TYPE + N  WHERE                019330
C                           TYPE= 128 IMPLIES TERMINAL ERROR              019340
C                                 64 IMPLIES WARNING WITH FIX             019350
C                                 32 IMPLIES WARNING                      019360
C                           N   = ERROR CODE RELEVANT TO CALLING ROUTINE  019370
C                NAME   - INPUT SCALAR CONTAINING THE NAME OF THE         019380
C                           CALLING ROUTINE AS A 6-CHARACTER LITERAL      019390
C                           STRING.                                       019400
C   LANGUAGE            - FORTRAN                                         019410
C-----------------------------------------------------------------------  019420
C   LATEST REVISION     - AUGUST 1, 1973                                  019430
C                                                                         019440
      DIMENSION          ITYP(2,4),IBIT(4)                                019450
      INTEGER            WARN,WARF,TERM,PRINTR                            019460
      EQUIVALENCE        (IBIT(1),WARN),(IBIT(2),WARF),(IBIT(3),TERM)     019470
      DATA       ITYP     /10HWARNING   ,10H          ,                   019480
     *                    10HWARNING(WI,10HTH FIX)   ,                    019490
     *                    10HTERMINAL  ,10H          ,                    019500
     *                    10HNON-DEFINE,10HD         /,                   019510
     *           IBIT     / 32,64,128,0/                                   019520
      DATA               PRINTR/6LOUTPUT/                                 019530
      IER2=IER                                                            019540
      IF (IER2 .GE. WARN) GO TO 5                                         019550
C                                     NON-DEFINED                       019560
      IER1=4                                                              019570
      GO TO 20                                                            019580
    5 IF (IER2 .LT. TERM) GO TO 10                                        019590
C                                     TERMINAL                          019600
      IER1=3                                                              019610
      GO TO 20                                                            019620
   10 IF (IER2 .LT. WARF) GO TO 15                                        019630
C                                     WARNING(WITH FIX)                 019640
      IER1=2                                                              019650
      GO TO 20                                                            019660
C                                     WARNING                           019670
   15 IER1=1                                                              019680
C                                     EXTRACT *N*                       019690
   20 IER2=IER2-IBIT(IER1)                                                019700
C                                     PRINT ERROR MESSAGE               019710
      WRITE (PRINTR,25) ((ITYP(I,IER1),I=1,2),NAME,IER2,IER               019720
   25 FORMAT(26H *** T M S L(UERTST) ***  ,2A10,4X,A6,4X,I2,              019730
     1   8H (IER = ,I3,1H))                                               019740
      RETURN                                                              019750
      END                                                                 019760
```

D.6 Major Function IKIRK Pertaining to Option No. 1 (IKIRK)

```
      REAL FUNCTION IKIRK(U,V,T)                                        000100
C FUNCTION IKIRK IS THE KIRKHOFF INTENSITY FUNCTION DESCRIBED IN        000110
C -BENDOW,B. AND GIANINO,P. "OPTICAL PERFORMANCE EVALUATION OF          000120
C INFRARED TRANSMITTING WINDOWS" AFCRL-72-0565. ASSUMING A GAUSSIAN     000130
C SHAPED UNPOLARIAZED SOURCE. THE INTENSITY FUNCTION CAN BE WRITTEN:    000140
C  IKIRK(U,V)=2(A!2/(1-EXP(-A!2)))!2*(1I(0,1,DX)(FW*FX)!!2+            000150
C                    1I(0,1,DX)(FW*FY)!!2                               000160
C WHERE:                                                                000170
C     FW(X,U)=EXP(-(A*X)!2)*EXP(-I*U*X!2/2)                             000180
C     FX(X,V)=X*J0(X*V)*EXP(I*K*PHIR(X))-FZ(X,V)                        000190
C     FY(X,V)=X*J0(X*V)*EXP(I*K*PHIT(X))+FZ(X,V)                        000200
C     FZ(X,V)=J1(X*V)*(EXP(I*K*PHIR(X))-EXP(I*K*PHIT(X)))/(V)           000210
C     A=1/SQRT(2)/SIG**2                                                000220
C     K=WAVE NO. (OMEGA/C)                                              000230
C   NOTATION:                                                           000240
C     ! => EXPONENTIATION                                               000250
C     I => "IMAGINARY"                                                  000260
C     I(0,1,DX)(.) MEANS INTEGRATION  OF THE FUNCTION WITHIN () W.R.T.X 000270
C     OVER THE INTERVAL (0,1).                                          000280
C J0 AND J1 ARE BESSEL FUNCTIONS OF THE FIRST KIND,ZEROTH AND FIRST     000290
C ORDER RESPECTIVELY.                                                   000300
C PHIR(X) AND PHIT(X) ARE THE FUNCTIONS PHI-SUPERSCRIPT-RHO AND PHI-    000310
C S PERSCRIPT-THETA RESPECTIVELY IN THE ABOVE REFERENCE.                000320
C THESE FUNCTIONS ARE GIVEN BY:                                         000330
C PHIR(X)=C*S1R*F1(X)+4*C*S2R*F2(X)                                     000340
C PHIT(X)=C*S1T*F1(X)+4*C*S2T*F2(X)                                     000350
C WHERE:                                                                000360
C     C=R!3*P0*BETA/KT                                                  000370
C     R=WINDOW RADIUS (CM)                                              000380
C     P0="EAN INCIDENT POWER DENSITY (WATTS/CM!2)                       000390
C     BETA=BULK ABSORPTION COEFFICIENT (1/CM)                           000400
C     KT=THERMAL CONDUCTIVITY (WATTS/(CM DEGC)                          000410
C     S1R,S2R,S1T,S2T ARE MATERIAL CONSTANTS DEFINED IN THE ABOVE REF.  000420
C     F1,F2 ARE THE FUNCTIONS DELTRAR-PRIME(X) AND                      000430
C     (1/X!2)I(0,X,DS)(DELTRAR-PRIME(S))                                000440
C     GIVEN IN THE ABOVE REFERENCE AND WHICH ARE PROVIDED AT SELECTED   000450
C     ARGUMENTS BY PROGRAM "TEMP5".                                     000460
C*************************************************************************000470
      COMMON/PHIBLK/CS1R,CS2R,CS1T,CS2T,XS,DZ,NF,MINT,EPS,F1(200),      000480
     +F2(200),HOLD,A,K ,TLAST,TNEXT,IERR,IP,MP1,ISW,N,                  000490
     +RAD,NP1,C3,C1,IERM,ERRORM                                         000500
      COMMON/IFILES/IT3,IT4,IT5,IT6,IT7,IT8                             000510
      COMMON/UV/UU,VV                                                   000520
      COMPLEX Q1,EXPR,EXPT,FX,FY,FZ,FW                                  000530
      REAL J0,J1                                                        000540
      REAL K                                                            000550
      DIMENSION XA(100),YR(100),YI(100),ZR(100),ZI(100)                 000560
      UU=U                                                              000570
      VV=V                                                              000580
      A2=A*A                                                            000590
      IF (A2 .GT. 220.) 1020,1030                                       000600
 1020 CONST=2.*A2*A2                                                    000610
      GOTO 1040                                                         000620
 1030 CONST=2.*(A2/(1.-EXP(-A2)))**2                                    000630
 1040 UD2=U/2E0                                                         000640
C IF ISW=1 THEN THE ARRAY OF POINTS FOR GAUSSIAN INTEGRATION            000650
C M ST BE FOUND                                                         000660
      GOTO (1050,1060) ISW                                              000670
 1050 ISW=2                                                             000680
      CALL DQG24A(0E0,1E0,YA)                                           000690
      IF (IP .EQ. 1) WRITE(IT6,1) (XA(I),I=1,N)                         000700
 1060 CALL RTAPE3(T)                                                    000710
      IF (IERR .NE. 0) GOTO 2000                                        000720
      DO 100 I=1,N                                                      000730
      X=XA(I)                                                           000740
      X2=X*X                                                            000750
      XV=X*V                                                            000760
      FW=EXP(-A2*X2)*CEXP(CMPLX(0E0,-UD2*X2))                           000770
      EXPR=CEXP(CMPLX(0E0,K*PHI(X)))                                    000780
      EXPT=CEXP(CMPLX(0E0,K*HOLD))                                      000790
      IF (V .EQ. 0.) FZ=(EXPR-EXPT)*X/2.                                000800
      IF (V .NE. 0.) FZ=(EXPR-EXPT)*J1(XV)/V                            000810
      Q1=X*J0(XV)                                                       000820
      FX=Q1*EXPR-FZ                                                     000830
```

```
      EY=Q1*EXPT+FZ                                                    000840
      Q1=EW*FX                                                         000850
      YR(T)=REAL(Q1)                                                   000860
      YI(T)=AIMAG(Q1)                                                  000870
      Q1=EW*EY                                                         000880
      ZR(T)=REAL(Q1)                                                   000890
      ZI(T)=AIMAG(Q1)                                                  000900
  100 CONTINUE                                                         000910
    1 FORMAT(5(1X,G12.5))                                              000920
      CALL DQG24R(0E0,1E0,YR,YRI)                                      000930
      CALL DQG24R(0E0,1E0,YI,YII)                                      000940
      CALL DQG24R(0E0,1E0,ZR,ZRI)                                      000950
      CALL DQG24R(0E0,1E0,ZI,ZII)                                      000960
      IKIRK=CONST*(YRI*YRI+YII*YII+ZRI*ZRI+ZII*ZII)                    000970
 2000 RETURN                                                           000980
      END                                                              000990
```

D.7 Major Function IKIRK Pertaining to Option No. 2 (IKIRKP)

```
      REAL FUNCTION IKIRK(U,V,T)                                       009800
C     APRIL 11,1974                                                    009810
C     THIS FUNCTION COMPUTES THE KIRKHOFF INTENSITY FUNCTION           009820
C     ALONG THE U=0 AND/OR V=0 AXIS OF THE PLANE.                      009830
C     SEE COMMENTS IN IKIRK AND COMPUTE FOR BACKGROUND INFO.           009840
C     IT IS VALID ONLY FOR CONSTANT TEMPERATURE WINDOWS.               009850
      DIMENSION AA(10)                                                 009860
      COMMON/FILES/IT3,IT4,IT5,IT6,IT7,IT8                             009870
      COMMON/VECTOR/VV                                                 009880
      COMMON/PHTHLK/CS1R,CS2R,CS1T,CS2T,XS,DX,NF,MNNT,EPP,F1(200),     009890
     +F2(200),HOLD,ALPHA,KE,TLAST,TNEXT,IERR,IP,MP1,ISW,NGAUSS,        009900
     +RAD,NP1,C3,C1                                                    009910
      COMPLEX AJ00,CRH00,CTHETA0,A1Q,A2Q,A3Q,A4Q,A6Q,A7Q               009920
      REAL KE                                                          009930
      DATA (AA(I),I=1,10)/1.,-.99999999958,.4999999206,-.16666E3019,   009940
     +.04165734 75,-.008301 3598,.001329882 0,-.0001413161,0.,0./      009950
      VV=V                                                             009960
      CALL RTADF3(T)                                                   009970
      IF(IERR.NE.0) GOTO 2000                                          009980
      CTHETA0=CS1T*F1(1)+4.*CS2T*F2(1)                                 009990
      CTHETA0=CTHETA0*KE                                               010000
      CRH00=CS1R*F1(1)+4.*CS2R*F2(1)                                   010010
      CRH00=CRH00*KE                                                   010020
      AJ00=(0.,1.)                                                    010030
      A2Q=CEXP(AJ00*CRH00)                                             010040
      A3Q=CEXP(AJ00*CTHETA0)                                           010050
      A1Q=A2Q-A3Q                                                      010060
      A=-(ALPHA**2)                                                    010070
      IF (A .LT. -1000) GOTO 110                                       010080
      EXPA=EXP(A)                                                      010090
      A5=2.*(A/(1.-EXPA))**2                                           010100
      GOTO 120                                                         010110
  110 EXPA=0.                                                          010120
      A5=2.*A*A                                                        010130
  120 B=U/2.                                                           010140
      IF(U.EQ.0) GOTO 200                                              010150
      C=(ALPHA/V)**2                                                   010160
      IF(U.EQ.0) GOTO 100                                              010170
  200 SINB=SIN(B)                                                      010180
      COSB=COS(B)                                                      010190
      ASQ=A**2                                                         010200
      BSQ=B**2                                                         010210
      A4QREAL=.5*((EXPA*(A*COSB+B*SINB))-A)/(ASQ+BSQ)                  010220
      A4QIMAG=-.5*((EXPA*(A*SINB-B*COSB))+B)/(ASQ+BSQ)                 010230
      A4Q=CMPLX(A4QREAL,A4QIMAG)                                       010240
      A6Q=A1Q/2.*A4Q                                                   010250
      IKIRKP=A5*((CABS(A2Q*A4Q-A6Q))**2+(CABS(A3Q*A4Q+A6Q))**2)        010260
      RETURN                                                           010270
  100 IF (-A .GT. .693) GOTO 1000                                      010280
```

```
50          CALL COMPUTE (C,AA,FF1,FF2,IER)                                  010290
            IF(IER.EQ.2) GOTO 1000                                           010300
            A70=A10*FF2                                                      010310
            IKIOK =A5*((CABS(A20*FF1-A70))**2+(CABS(A30*FF1+A70))**2)        010320
            RETURN                                                           010330
55          RETURN                                                           010340
1000  WRITE (IT5,10)                                                         010350
10    FORMAT(/* ALPHA**2 IS OUT OF RANGE*)                                   010360
2000  RETURN                                                                 010370
      END                                                                    010370
```

Appendix E

Fortran Listings for DISPLAY Program

E.1 Main Program DISPLAY

```
      PROGRAM DISPLAY(TAPE4=80/80,TAPE2=80/80,TAPE3,OUTPUT=80)                  000100
C REVISION---JUNE 11,1976                                                       000105
      COMMON Z(103,103),XT(103,103),YT(103,103)                                 000110
      COMMON/BLOCK3/NP2,MP2,XMINF,YMINF,DX,DY,SWI,IT6                           000120
      DIMENSION DATA(3)                                                         000130
      DIMENSION PROGID(3),CODE(7),DATAIN(100,3),PARM(80)                        000140
      DIMENSION IND(40),ZLEVS(50),INDEX(4,2)                                    000150
      INTEGER CCODE                                                             000160
      LOGICAL FLAG,FLAG1,SWI                                                    000170
      DATA IT4,IT6,IT3/4,6,3/                                                   000180
      DATA XS,YS/0.,0./                                                         000190
      DATA (PROGID=7HGIANINO,7HIU 2347,7HDISPLAY)                               000200
      DATA (CODE=1.1,45.,2HMA,1,1)                                              000210
      BLANK=1H                                                                  000220
      IBLANK=1H                                                                 000230
C THE FOLLOWING DATA CARDS MUST BE INPUTTED TO TAILOR THE PROGRAM               000240
C FOR THE USERS PARTICULAR APPLICATION. THE NUMBER IN PARENTHESIS IN            000250
C FRONT OF EACH DATAUM NAME IS THE CARD COLUMN AT WHICH TO START THE            000260
C DATUM. THE NUMBER IN PARENTHESIS FOLLOWING THE DATUM IS THE DEFAULT           000270
C VALUE OF THE DATUM. IF THE DEFAULT VALUE IS TO BE USED LEAVE THE              000280
C CORRESPONDING CARD FIELD BLANK.                                               000290
C CARD 1                                                                        000300
C (1)    XMAX            (100.)     MAXIMUM PLOT LENGTH IN INCHES               000310
C (11)   YMAX            (10.)      MAXIMUM PLOT WIDTH IN INCHES                000320
C (21)   NPI             (10.)      NUMBER OF POINTS/INCH FOR CONTOURS          000330
C (31)   TICU            (.5)       NUMBER OF INCHES BETWEEN TIC-MARKS          000340
C                                   FOR USER DEFINED X-Y PLOTS                  000350
C (41)   XLEN            (10.)      X-Y PLOT COORDINATE FRAME X-SIZE            000360
C (51)   YLEN            (8.)       X-Y PLOT COORDINATE FRAME Y-SIZE            000370
C (61)   SCALEX          (1.)       X-Y PLOT X-SCALE (UNITS/TIC)                000380
C (71)   SCALEY          (1.)       X-Y PLOT Y-SCALE (UNITS/TIC)                000390
C                                                                               000400
C CARD2                                                                         000410
C (01)   XMIN            (0.)       X-Y PLOT XMIN                               000420
C (11)   YMIN            (0.)       Y-Y PLOT YMIN                               000430
C (21)   NAME            (HARRETT)  NAME ON PLOT                                000440
C (31)   PROB. #          (2784)     PLOT PROBLEM NUMBER                         000450
C                                                                               000460
C CARD 3                                                                        000470
C (CARD 3 CONTAINS THE 2-D ARRAY 'INDEX' WHICH IS THE INDEX OF                  000480
C LOCATIONS FOR LABELING INFORMATION ASSUMED TO BE CONTAINED IN THE             000490
C LAST DATAIN ON THE FILE CONTAINING THE SURFACES. IF THERE IS NO               000500
C SUCH DATA THEN THIS CARD MAY BE LEFT BLANK AND THE LABELING WILL              000510
C NOT BE DONE. INDEX CONSISTS OF PAIRS OF NUMBERS WHEREIN THE FIRST             000520
C NUMBER IS THE STARTING LOCATION OF THE LABEL AND THE SECOND NUMBER            000530
C IS THE LENGTH OF THE LABEL.                                                   000540
C IF THE LETTER 'D' IS PLACED IN COLUMN 1 OF CARD 2, THE FOLLOWING              000550
C DEFAULT LOCATIONS ARE USED FOR TITLE INFORMATION-                             000560
C (1)    INDEX(1,1)        (1)        SURFACE TITLE                            000570
C (6)    INDEX(1,2)        (30)        SURFACE TITLE LENGTH (CHARACTERS)       000580
C (11)   INDEX(2,1)        (4)        PARAMETER TITLE                          000590
C (16)   INDEX(2,2)        (30)        PARAMETER TITLE LENGTH                  000600
C (21)   INDEX(3,1)        (7)        X-TITLE                                  000610
C (26)   INDEX(3,2)        (30)        X-TITLE LENGTH                          000620
C (31)   INDEX(4,1)        (12)       Y-TITLE                                  000630
C (36)   INDEX(4,2)        (30)        Y-TITLE LENGTH                          000640
C (41)   NDA               (3)        NUMBER OF 'DATAIN' ARRAYS                000650
C                                                                               000660
C CARD 4-CARD NDA+3                                                             000670
C CARD 4-      CONTAIN THE INDICES OF DATA IN 'DATAIN' WHICH ARE TO             000680
C BE PLOTTED AT THE BEGINNING OF EACH RUN OF DISPLAY. THERE MUST BE             000690
C ONE CARD FOR EACH 'DATAIN' EVEN THOUGH FOR SOME 'DATAIN' NO DATA IS           000700
C TO BE PLOTTED (A BLANK CARD IS ACCEPTABLE). THE INDEX NUMBERS START           000710
C IN COLUMNS 1,3,5,7,.... FOR A TOTAL OF UP TO 40 INDICES PER CARD.             000720
C THE DEFAULT IS A BLANK CARD,I.E. NO 'DATAIN' DATA IS TO BE PLOTTED.           000730
      PXMAX=100.                                                                000740
      PYMAX=11.                                                                 000750
      PP=10.                                                                    000760
      TICU=.5                                                                   000770
      XLEN=10.                                                                  000780
      YLEN=8.                                                                   000790
      SCALEX=SCALEY=1.                                                          000800
      XMIN=X=YMINY=0.                                                           000810
      IND=X(1,1)=1                                                              000820
```

```
      INDEX(1,2)=30                                                     000830
      INDEX(2,1)=4                                                      000840
      INDEX(2,2)=30                                                     000850
      INDEX(3,1)=7                                                      000860
      INDEX(3,2)=30                                                     000870
      INDEX(4,1)=10                                                     000880
      INDEX(4,2)=30                                                     000890
      NDA=2                                                             000900
      READ(IT4,5) (DATA(I),I=1,8)                                       000910
      IF (DATA(1) .NE. BLANK) DECODE(10,6,DATA(1)) PXMAX                000920
      IF (DATA(2) .NE. BLANK) DECODE(10,6,DATA(2)) PYMAX                000930
      IF (DATA(3) .NE. BLANK) DECODE(10,6,DATA(3)) PPI                  000940
      IF (DATA(4) .NE. BLANK) DECODE(10,6,DATA(4)) TICU                 000950
      IF (DATA(5) .NE. BLANK) DECODE(10,6,DATA(5)) XLEN                 000960
      IF (DATA(6) .NE. BLANK) DECODE(10,6,DATA(6)) YLEN                 000970
      IF (DATA(7) .NE. BLANK) DECODE(10,6,DATA(7)) SCALEX               000980
      IF (DATA(8) .NE. BLANK) DECODE(10,6,DATA(8)) SCALEY               000990
      WRITE(IT6,10) PXMAX,PYMAX,PPI,TICU,XLEN,YLEN,SCALEX,SCALEY        001000
      READ(IT4,5) (DATA(I),I=1,8)                                       001010
      IF (DATA(1) .NE. BLANK) DECODE(10,6,DATA(1)) XMINX                001020
      IF (DATA(2) .NE. BLANK) DECODE(10,6,DATA(2)) YMINY                001030
      IF (DATA(3) .NE. BLANK) DECODE(10,4,DATA(3)) PROGID(1)            001040
      IF (DATA(4) .NE. BLANK) DECODE(10,4,DATA(4)) PROGID(2)            001050
                                                                        001060
      WRITE(IT6,10) XMINX,YMINY                                         001070
      READ(IT4,7) (DATA(I),I=1,8),NDAT                                  001080
      II=0                                                              001090
      DO 91 I=1,4                                                       001100
      DO 91 J=1,2                                                       001110
      II=II+1                                                           001120
      IF (II .GT. 1) GOTO 84                                            001130
      IF (DATA(1) .EQ. IHD) GOTO 85                                     001140
   84 IF (DATA(II) .EQ. BLANK) 92,93                                    001150
   92 INDEX(I,J)=0                                                      001160
      GOTO 91                                                           001170
   93 CALL RJUST (DATA(II))                                             001180
      DECODE(10,8,DATA(II)) INDEX(I,J)                                  001190
   91 CONTINUE                                                          001200
   85 WRITE(IT6,14) ((INDEX(I,J),J=1,2),I=1,4)                          001210
C INITIALIZE THE PLOTTER                                                001220
      CALL PLTID3(PROGID,PXMAX,PYMAX,1E0)                               001230
C PLOT DATA BLOCKS                                                      001240
      REWIND IT3                                                        001250
      XPT=0.                                                            001260
      IF (NDAT .EQ. IBLANK) GOTO 94                                     001270
      CALL RJUST(NDAT)                                                  001280
      DECODE(10,8,NDAT) NDA                                             001290
      IF (NDA .EQ. 0) GOTO 70                                           001300
   94 DO 96 I=1,NDA                                                     001310
      READ(IT4,9) (IND(J),J=1,40)                                       001320
      DO 97 J=1,40                                                      001330
      IF (IND(J) .EQ. IBLANK) 98,99                                     001340
   98 IND(J)=0                                                          001350
      GOTO 97                                                           001360
   99 CALL RJUST(IND(J))                                                001370
      DECODE(10,8,IND(J)) IND(J)                                        001380
   97 CONTINUE                                                          001390
      WRITE(IT6,11) NDA,(IND(K),K=1,40)                                 001400
      READ(IT3) DATAIN                                                  001410
      CALL FILL(DATAIN,PARM,IND,100,1,NN)                               001420
      CALL PARMPLT(XPT,0E0,,1E0,PARM,1,NN)                              001430
      XPT=XPT+3.                                                        001440
   96 CONTINUE                                                          001450
   70 CALL PLOT(XPT,0.,-3)                                              001460
C START THE MAIN DISPLAY LOOP. INTERP READS AND INTERPRETS THE DISPLAY  001470
C COMMANDS. IT RETURNS A FLAG VALUE OF .FALSE. IF THE LAST COMMAND HAS  001480
C     BEEN READ.                                                        001490
      FLAG=.T.                                                          001500
      INDIC=1                                                           001510
  100 DO 101 K=1,7                                                      001520
      CODE(K)=BLANK                                                     001530
  101 CONTINUE                                                          001540
      INDTX=0                                                           001550
      XP=.5                                                             001560
      IPEN=2                                                            001570
      CALL INTERP(CCODE,CODE,FLAG,IT4)                                  001580
      IF (.NOT. FLAG .AND. INDIC .EQ. 2) GOTO 2000                      001590
      IF (CCODE .EQ. 1) WRITE(IT6,15) (CODE(K),K=1,5)                   001600
      IF (CCODE .EQ. 2) WRITE(IT6,12) (CODE(K),K=1,4)                   001610
      IF (CCODE .EQ. 3) WRITE(IT6,16) (CODE(K),K=1,7)                   001620
      FLAG1=.T.
```

```
      IF (NDA .LE. 0) GOTO 82                                          00001630
      REWIND IT3                                                       00001640
      DO 81 I=1,NDA                                                    00001650
      READ(IT3)                                                        00001660
   81 CONTINUE                                                         00001670
   82 INDIC=2                                                          00001680
      ZMINA=1E100                                                      00001690
      ZMAXA=-1E100                                                     00001700
      NSTART=CODE(2) .OR. 0                                            00001710
      NSTART=NSTART-1                                                  00001720
      IF (NSTART .LE. 0) GOTO 72                                       00001730
      DO 74 I=1,NSTART                                                 00001740
      CALL RD1(IT3,D1,D2,D3,D4,4,FLAG1,D5,D6,D7,D8,D9)                 00001750
      IF (.NOT. FLAG1) GOTO 100                                        00001760
   74 CONTINUE                                                         00001770
   72 NSKIP=CODE(1) .OR. 0                                             00001780
      NSKIP=NSKIP-1                                                    00001790
      GOTO (110,120,130) CCODE                                         00001800
C 110 => CONTOUR                                                       00001810
C 120 => PERSPECTIVE                                                   00001820
C 130 => PLOT                                                          00001830
C CONTOUR- IF CONTOURS ARE TO BE NORMALIZED OVER ALL SURFACES, FIND THE 00001840
C MAX. AND MIN. OVER ALL SURFACES.                                     00001850
  110 IF (CODE(4) .EQ. 2HON .OR. CODE(6) .EQ. 2HON) GOTO 150           00001860
  140 CALL RD1(IT3,NP,MP,ZMAX,ZMIN,1,FLAG1,TIM,XMIN,XMAX,YMIN,YMAX)    00001870
      WRITE(IT6,13) NP,MP,ZMAX,ZMIN,TIM,XMIN,XMAX,YMIN,YMAX,FLAG1      00001880
      IF (CODE(4) .EQ. 2HNE .OR. CODE(6) .EQ. 2HNE) GOTO 140           00001890
      CALL SKIP(IT3,NSKIP,FLAG1,NP)                                    00001900
      IF (ZMAX .GT. ZMAXA) ZMAXA=ZMAX                                  00001910
      IF (ZMIN .LT. ZMINA) ZMINA=ZMIN                                  00001920
      IF (FLAG1) GOTO 140                                              00001930
  140 REWIND IT3                                                       00001940
      FLAG1=.T.                                                        00001950
      IF (NDA .LE. 0) GOTO 145                                         00001960
      DO 83 I=1,NDA                                                    00001970
      READ (IT3)                                                       00001980
   83 CONTINUE                                                         00001990
  145 IF(NSTART .LE. 0) GOTO 150                                       00001991
      DO 115 I=1,NSTART                                                00001992
      CALL RD1(IT3,DUM1,DUM2,DUM3,DUM4,4,FLAG1,DUM5,DUM6,DUM7,DUM8,    00001994
     +DUM9)                                                            00001996
  115 CONTINUE                                                         00001998
  150 GOTO (155,250,350) CCODE                                         00002000
C NEXT FILL THE Z-ARRAY,DETERMINE CONTOUR LEVELS AND DRAW THE CONTOUR MA 00002010
  155 CALL RD1(IT3,NP,MP,ZMAX,ZMIN,2,FLAG1,TIM,XMIN,XMAX,YMIN,YMAX)    00002020
      IF (.NOT. FLAG1) GOTO 190                                        00002030
      IF (CODE(4) .EQ. 2HON) 170,180                                   00002040
  170 CALL CLEV(CODE(3),ZMAX,ZMIN,ZLEVS)                               00002050
      GOTO 190                                                         00002060
  180 IF (CODE(4) .EQ. 2HNA) 200,210                                   00002070
  200 ZMAX=ZMAXA                                                       00002080
      ZMIN=ZMINA                                                       00002090
  210 C=100./(ZMAX-ZMIN)                                               00002100
      DO 220 J=1,MP                                                    00002110
      DO 220 I=1,NP                                                    00002120
      Z(I,J)=(Z(I,J)-ZMIN)*C                                           00002130
  220 CONTINUE                                                         00002140
      CALL CLEV(CODE(3),100.,0.,ZLEVS)                                 00002150
  190 XLEN1=AMIN1((MP-1)/PPI,PYMAX-1.)                                 00002160
      YLEN1=AMIN1((NP-1)/PPI,PYMAX-1.)                                 00002170
      IF (CODE(5) .EQ. 2HSP .A. INDIX .EQ. 1) GOTO 205                 00002180
      CALL CFRAME(INDEX,ZMIN,ZMAX,CODE(4),NP,MP,TIM,PPI,100.,6.,0.,    00002190
     +XMIN,XMAX,YMIN,YMAX,DATAIN,XLEN1,YLEN1,CCODE)                    00002200
      INDIX=1                                                          00002210
  205 SYM=SYMBL(TIM)                                                   00002220
      NX=INDEX(2,2)                                                    00002230
      IF (NX .EQ. 0) GOTO 222                                          00002240
      CALL SYMBOL(XLEN1+XP,PYMAX-1.1,.1E0,DATAIN(INDEX(2,1),1),0.,NX)  00002250
      GOTO 223                                                         00002260
  222 CALL SYMBOL(XLEN1+XP,PYMAX-1.1,.1E0,9HPARAMETER,0.,9)            00002270
  223 CALL SYMBOL(999E0,999E0,.1E0,SYM,0E0,10)                         00002280
      CALL CONTOR(NP,MP,CODE(3),ZLEVS,BLANK,XLEN1,YLEN1,XP,PYMAX-1.2)  00002290
      IF (CODE(5) .NE. 2HSP) GOTO 215                                  00002300
      XP=XP+4.                                                         00002310
      IPEN=IPEN+1                                                      00002320
      IF (IPEN .GT. 3) IPEN=1                                          00002330
      CALL NEWPEN(IPEN)                                                00002340
      GOTO 196                                                         00002350
  215 CALL PLOT(XLEN1+12.,0.,-3)                                       00002360
  196 IF (FLAG1) 195,106                                               00002370
```

```
      105  CALL SKIP(IT3,NSKIP,FLAG1,NP)                                    00002380
           IF (FLAG1) 155,106                                               00002390
      106  IF (CODE(5) .NE. 2HSP) GOTO 100                                  00002400
           CALL NEWPEN(2)                                                   00002410
           CALL PLOT(XLEN1+XO+8.,.0E0,-3)                                   00002420
           GOTO 100                                                         00002430
C START OF PERSPECTIVE PLOTS. FIRST FIND ZMAX,ZMIN FOR NORMALIZATION IF     00002440
C NORMALIZATION IS OVER ALL SURFACES.                                       00002450
      120  GOTO 110                                                         00002460
      250  PP1=10.                                                          00002470
           IF (MP .GT. NP) GOTO 390                                         00002480
           XLEN1=5.*MP/FLOAT(NP)                                            00002490
           YLEN1=5.                                                         00002500
           GOTO 400                                                         00002510
      390  YLEN1=5.*NP/FLOAT(MP)                                            00002520
           XLEN1=5.                                                         00002530
C THE NEXT TASK IS TO FIND SCALING FACTORS FOR THE DISPLAY BY CALLING       00002540
C FACE WITH A RECTANGLE FUNCTION.                                           00002550
      400  NP1=NP+1                                                         00002560
           MP1=MP+1                                                         00002570
           NP2=NP1+1                                                        00002580
           MP2=MP1+1                                                        00002590
           X2=NP2**2                                                        00002600
           Y2=MP2**2                                                        00002610
           H=SQRT(X2+Y2)                                                    00002620
           THETA1=.785398164                                                00002630
           THETA2=CODE(3)/180.*3.14159265                                   00002640
           X=.5*NP2+H*COS(THETA2)                                           00002650
           Y=.5*MP2+H*SIN(THETA2)                                           00002660
           C1=.5*H                                                          00002670
           CALL APLACE(XP,YP,XLEN1,YLEN1,CODE(3))                           00002680
C THIS SUBROUTINE FINDS THE POINT AT WHICH TO PLACE THE VIEW-ANGLE          00002690
C ARROW                                                                     00002700
           DO 260 K=1,NP2                                                   00002710
           Z(K,1)=Z(K,MP2)=0.                                               00002720
      260  CONTINUE                                                         00002730
           DO 270 K=2,MP1                                                   00002740
           Z(NP2,K)=Z(1,K)=0.                                               00002750
      270  CONTINUE                                                         00002760
           DO 280 I=2,NP1                                                   00002770
           DO 280 J=2,MP1                                                   00002780
           Z(I,J)=C1                                                        00002790
      280  CONTINUE                                                         00002800
           SW1=.F.                                                          00002810
C IF SW1 IS OFF (.F.) THEN FACE RETURNS XMIN,YMIN,DX,DY ONLY AND DOES NO    00002820
C PLOTTING                                                                  00002830
           CALL FACE(X,Y,H,THETA1,THETA2,0)                                 00002840
           SW1=.T.                                                          00002850
           WRITE(IT6,2) XMINF,YMINF,DX,DY                                   00002860
      2    FORMAT(1X,4E15.5)                                                00002870
C NOW FILL UP THE Z-ARRAY                                                   00002880
      290  CALL RD1(IT3,NP,MP,ZMAX,ZMIN,J,FLAG1,TIM,XMIN,XMAX,YMIN,YMAX)   00002890
           IF (.NOT. FLAG1) GOTO 100                                        00002900
           IF (CODE(4) .EQ. 2HNF) GOTO 300                                  00002910
           ZMIN=ZMINA                                                       00002920
           ZMAX=ZMAXA                                                       00002930
C NOW DRAW LABELS,ETC.                                                      00002940
       30  C=C1/(ZMAX-ZMIN)                                                00002950
           DO 310 I=2,MP1                                                   00002960
           DO 310 K=2,NP1                                                   00002970
           Z(K,J)=(Z(K,J)-ZMIN)*C                                           00002980
      310  CONTINUE                                                         00002990
           PP=.1*AMAX0(NP,MP)                                               00003000
           CALL CFRAME(INDEX,ZMIN,ZMAX,CODE(4),NP,MP,TIM,PP,C1,6E0,1E0,     00003010
          +XMIN,XMAX,YMIN,YMAX,DATAIN,XLEN1,YLEN1,CCODE)                    00003020
           CALL ARROW(XP,YP,1.,CODE(3),10HVIEW ANGLE,10)                    00003030
           XPT=XLEN1+4E0                                                    00003040
           YPT=-1E0                                                         00003050
           CALL PLOT(XPT,YPT,-3)                                            00003060
C NOW DRAW PERSPECTIVE DISPLAY                                              00003070
           CALL FACE(X,Y,H,THETA1,THETA2,0)                                 00003080
           CALL PLOT(12E0,0E0,-3)                                           00003090
           IF (FLAG1) 320,100                                               00003100
      320  CALL SKIP(IT3,NSKIP,FLAG1,NP)                                    00003110
           IF (FLAG1) 240,100                                               00003120
C START OF X-Y PLOTS. FIRST FIND ZMAX,ZMIN FOR SCALING IF SCALING IS        00003130
C OVER ALL SURFACES                                                         00003140
      310  GOTO 110                                                         00003150
      350  CALL RD1(IT3,NP,MP,ZMAX,ZMIN,2,FLAG1,TIM,XMIN,XMAX,YMIN,YMAX)   00003160
           IF (.NOT. FLAG1) GOTO 105                                        00003170
```

```
315         IF (CODE(6) .EQ. 2HNF) GOTO 370                                   000180
            IF (CODE(6) .EQ. 2HON) GOTO 370                                   000190
            ZMIN=ZMINA                                                        000200
            ZMAX=ZMAXA                                                        000210
370         CONTINUE                                                          000220
320         IF (CODE(7) .NE. 2HSP) XS=YS=0.                                   000230
            CALL PLOTT1(TITLE,SCALEX,SCALEY,XLEN,YLEN,TIM,NP,MP,IT6,INDEX,    000240
           +DATAIN,XMIN,XMAX,YMIN,YMAX,XS,YS,ZMIN,ZMAX,CODE,XMINX,YMINY)      000250
            IF (FLAG1) 380,105                                                000260
380         CALL SKIP(IT3,NSKIP,FLAG1,NP)                                     000270
325         IF (FLAG1) 350,105                                                000280
105         IF (CODE(7) .EQ. 2HSP) CALL PLOT(XLEN+6.,0.,-3)                   000290
            XS=YS=0.                                                          000300
            GOTO 100                                                          000310
2000        CALL ENDPLT                                                       000320
330         16  FORMAT(1X,2I5,3X,A1,2I5,3X,A2,3XA2)                          000330
            15  FORMAT(1X,3I5,3X,A2,3XA2)                                    000340
            14  FORMAT(1X,8I5)                                               000350
            13  FORMAT(1X,2I10/1X,4G10.2/1X,3G10.2,L10)                      000360
            12  FORMAT(1X,2I5,3X,G10.2,3X,A2,3XA2,3XA2)                      000370
335         11  FORMAT(1X,I5,4(/1X,10I5))                                    000380
            10  FORMAT(1X,4G10.2/1X,4G10.2)                                  000390
            9   FORMAT(40A2)                                                 000400
            8   FORMAT(I10)                                                  000410
            7   FORMAT(9A5)                                                  000420
340         6   FORMAT(E10.0)                                                000430
            5   FORMAT(8A10)                                                 000440
            4   FORMAT(A10)                                                  000450
                END                                                          000460
```

E.2 Subroutine PLOTT1

```
1           SUBROUTINE PLOTT1(TITLE,SCALXX,SCALYY,XLEN,YLEN,TIM,NP,MP,IT6,    000470
           +INDEX,DATAIN,XMINX,XMAXX,YMINY,YMAXY,XSS,YSS,ZMIN,ZMAX,CODE,      000480
           +XMINXX,YMINYY)                                                    000490
            DIMENSION PTITLE(5),TITLE(5),XTITLE(5),YTITLE(5)                  000500
5           DIMENSION INDEX(4,2),DATAIN(100,3),XARRAY(101),YARRAY(101),CODE(7) 000510
            COMMON Z(103,103)                                                 000520
C IN THE FIRST PART OF PLOTT1 THE COORDINATE FRAME IS DRAWN AND               000530
C TITLED. IF CODE(7) IS ON THEN SCALING IS OBTAINED FROM THE USER.            000540
C OTHERWISE SCALING IS DONE BY SUBROUTINE SCALE.                              000550
10          C THE INDEX ARRAY IS USED AS IN SUBROUTINE CFRAME TO OBTAIN VARIOUS 000560
C TITLING INFORMATION. THE PLOT Y-ARRAY IS OBTAINED FROM Z(I,J) AS            000570
C YARRAY(.)=Z(.,N) IF CODE(3)=X AND AS YARRAY(.)=Z(N,.) IF CODE(3)            000580
C =Y. WHERE N STARTS AT CODE(4) AND IS INCREMENTED BY CODE(3). THE             000590
C CALL PLOTT1 MUST SUPPLY THE FOLLOWING ARGUMENTS-                            000600
15          MI=CODE(4) .OR. 0                                                 000610
            IF (CODE(7) .EQ. 2HSP .A. XSS .NE. 0.) GOTO 430                   000620
            XS=YS=0.                                                          000630
370         YMIN=YARRAY(1)=ZMIN                                               000640
            YARRAY(2)=ZMAX                                                    000650
20          CALL SCALE(YARRAY,YLEN,2,1,20.,YMIN1,DELY)                        000660
            IF (CODE(3) .EQ. 1HX) 380,390                                     000670
380         XMIN=XMINX                                                        000680
            XMAX=XMAXX                                                        000690
            PMIN=YMINY                                                        000700
25          PMAX=YMAXY                                                        000710
            DELX=(XMAX-XMIN)/AMAX0(1,MP-1)                                    000720
            DELP=(PMAX-PMIN)/AMAX0(1,NP-1)                                    000730
            DO 382 I=1,MP                                                     000740
            XARRAY(I)=XMIN+(I-1)*DELX                                         000750
30          382 CONTINUE                                                      000760
            NX=INDEX(3,2)                                                     000770
            NN=(NX+9)/10                                                      000780
            IF (NX .NE. 0) GOTO 505                                           000790
            NX=7                                                              000800
35          XTITLE(1)=ZMAX-VALUE                                              000810
            GOTO 500                                                          000820
505         II=INDEX(3,1)                                                     000830
            DO 384 I=1,NN                                                     000840
            XTITLE(I)=DATAIN(II,1)                                            000850
```

```
40                II=I+1                                                  003860
              384 CONTINUE                                                003870
              500 NT=INDEX(4,2)                                           003880
                  NN=(NT+9)/10                                            003890
                  IF (NT .NE. 0) GOTO 515                                 003900
45                NT=7                                                    003910
                  TITLE(1)=7HY-VALUE                                      003920
                  GOTO 510                                                003930
              515 II=INDEX(4,1)                                           003940
                  DO 386 I=1,NN                                           003950
50                TITLE(I)=DATAIN(II,1)                                   003960
                  II=I+1                                                  003970
              386 CONTINUE                                                003980
              510 NPTS=NP                                                 003990
                  NLIN=NP                                                 004000
55                GOTO 400                                                004010
              300 XMIN=YMINY                                              004020
                  XMAX=YMAXY                                              004030
                  PMIN=XMINX                                              004040
                  PMAX=XMAXX                                              004050
60                DELY=(XMAX-XMIN)/AMAX0(1,NP-1)                          004060
                  DELP=(PMAX-PMIN)/AMAX0(1,NP-1)                          004070
                  DO 392 I=1,NP                                           004080
                  XARRAY(I)=XMIN+(I-1)*DELX                               004090
              392 CONTINUE                                                004100
65                NX=INDEX(4,2)                                           004110
                  NN=(NX+9)/10                                            004120
                  IF (NX .NE. 0) GOTO 525                                 004130
                  NX=7                                                    004140
                  XTITLE(1)=7HY-VALUE                                     004150
70                GOTO 520                                                004160
              525 II=INDEX(4,1)                                           004170
                  DO 394 I=1,NN                                           004180
                  XTITLE(I)=DATAIN(II,1)                                  004190
                  II=II+1                                                 004200
75            394 CONTINUE                                                004210
              520 NT=INDEX(3,2)                                           004220
                  NN=(NT+9)/10                                            004230
                  IF (NT .NE. 0) GOTO 535                                 004240
                  NT=7                                                    004250
80                TITLE(1)=7HX-VALUE                                      004260
                  GOTO 530                                                004270
              535 II=INDEX(3,1)                                           004280
                  DO 396 I=1,NN                                           004290
                  TITLE(I)=DATAIN(II,1)                                   004300
85                II=II+1                                                 004310
              396 CONTINUE                                                004320
              530 NPTS=NP                                                 004330
                  NLIN=NP                                                 004340
              400 CALL SCALE(XARRAY,XLEN,NPTS,1,20.,XMIN,DELX)            004350
90                NY=INDEX(1,2)                                           004360
                  NN=(NY+9)/10                                            004370
                  IF (NY .NE. 0) GOTO 545                                 004380
                  NY=7                                                    004390
                  YTITLE(1)=7HZ-VALUE                                     004400
95                GOTO 540                                                004410
              545 II=INDEX(1,1)                                           004420
                  DO 410 I=1,NN                                           004430
                  YTITLE(I)=DATAIN(II,1)                                  004440
                  II=II+1                                                 004450
100           410 CONTINUE                                                004460
              540 NB=INDEX(2,2)                                           004470
                  IF (NB .NE. 0) GOTO 555                                 004480
                  NN=(NB+9)/10                                            004490
                  NB=10                                                   004500
105               BTITLE(1)=10HPARAMETER=                                 004510
                  GOTO 550                                                004520
              555 II=INDEX(2,1)                                           004530
                  DO 420 I=1,NN                                           004540
                  BTITLE(I)=DATAIN(II,1)                                  004550
110               II=II+1                                                 004560
              420 CONTINUE                                                004570
              550 ISYM=0                                                  004580
                  M=CODE(5) .OR. 0                                        004590
                  ISWTCH=3                                                004600
115               IF (CODE(6) .EQ. 2HON) GOTO 430                         004610
                  ISWTCH=2                                                004620
                  SCALEX=DELX                                             004630
                  XMIN=XMINI                                              004640
                  SCALEY=DELY                                             004650
```

```
      YMIN=YMIN1                                                        004660
C START OF MAIN PLOTTING LOOP                                           004670
  430 IF (M .GT. NLIM) GOTO 440                                         004680
      IF (CODE(3) .EQ. 1HX) GOTO 450                                    004690
      DO 460 I=1,NPTS                                                   004700
      YARRAY(I)=Z(I,M)                                                  004710
  440 CONTINUE                                                          004720
      GOTO 470                                                          004730
  450 DO 480 I=1,NPTS                                                   004740
      YARRAY(I)=Z(M,I)                                                  004750
  480 CONTINUE                                                          004760
  470 CURVE=PMIN+(M-1)*DELP                                             004770
      IF (CODE(7) .NE. 2HSP) GOTO 425                                   004780
      TIMT=TIM                                                          004790
      TIM=CURVE                                                         004800
      CURVE=TIMT                                                        004810
      DO 215 I=1,5                                                      004820
      TIMT=PTITLE(I)                                                    004830
      PTITLE(I)=TITLE(I)                                                004840
      TITLE(I)=TIMT                                                     004850
  215 CONTINUE                                                          004860
      TIMT=NB                                                           004870
      NB=NT                                                             004880
      NT=TIMT                                                           004890
      IF (XSS .NE. 0.) GOTO 200                                         004900
  425 GOTO (200,220,210,200) ISWTCH                                     004910
  210 CALL PLOT(XS,YS,3)                                                004920
      ISWTCH=4                                                          004930
      XS=YS=0.                                                          004940
      SCAY=TICU/SCALXX                                                  004950
      SCAY=TICU/SCALYY                                                  004960
      ENCODE(10,1,ISYMB) YMINYY                                         004970
      CALL SYMBOL(-.1.0E0,.1E0,1SYMB,90E0,10)                           004980
      CALL SYMBOL(XS-.2E0,YS+1.2E0,.1E0,YTITLE,90E0,NY)                 004990
      CALL SYMBOL(999E0,999E0,.1E0,10H SCALE IS ,90E0,10)               005000
      ENCODE(10,1,ISYMB) SCALYY                                         005010
      CALL SYMBOL(999E0,999E0,.1E0,1SYMB,90E0,10)                       005020
      CALL SYMBOL(999E0,999E0,.1E0,10H UNITS/TIC,90E0,11)               005030
      ENCODE(10,1,ISYMB) YMINXX                                         005040
      CALL SYMBOL(0E0,-.2E0,.1E0,1SYMB,0E0,10)                          005050
      CALL SYMBOL(1.2E0,-.3E0,.1E0,XTITLE,0E0,NX)                       005060
      CALL SYMBOL(999E0,999E0,.1E0,10H SCALE IS ,0E0,10)                005070
      ENCODE(10,1,ISYMB) SCALXX                                         005080
      CALL SYMBOL(999E0,999E0,.1E0,1SYMB,0E0,10)                        005090
      CALL SYMBOL(999E0,999E0,.1E0,10H UNITS/TIC,0E0,11)                005100
      CALL PLOT(0E0,0E0,3)                                              005110
      YPT=0.                                                            005120
      DO 100 I=1,100000                                                 005130
      N=I                                                               005140
      XPT=I*TICU                                                        005150
      IF (XPT .GT. XLEN) GOTO 1000                                      005160
      CALL PLOT(XPT,YPT,2)                                              005170
      CALL PLOT(XPT,YPT+.1,2)                                           005180
      CALL PLOT(XPT,YPT,2)                                              005190
  100 CONTINUE                                                          005200
 1000 XPT=(N-1)*TICU                                                    005210
      DO 110 I=1,10000                                                  005220
      N=I                                                               005230
      YPT=I*TICU                                                        005240
      IF (YPT .GT. YLEN) GOTO 1010                                      005250
      CALL PLOT(XPT,YPT,2)                                              005260
      CALL PLOT(XPT-.1,YPT,2)                                           005270
      CALL PLOT(XPT,YPT,2)                                              005280
  110 CONTINUE                                                          005290
 1010 YPT=(N-1)*TICU                                                    005300
      XPTT=XPT                                                          005310
      DO 120 I=1,10000                                                  005320
      N=I                                                               005330
      XPT=XPTT-I*TICU                                                   005340
      IF (XPT .LT. XS) GOTO 1020                                        005350
      CALL PLOT(XPT,YPT,2)                                              005360
      CALL PLOT(XPT,YPT-.1,2)                                           005370
      CALL PLOT(XPT,YPT,2)                                              005380
  120 CONTINUE                                                          005390
 1020 YPTT=YPT                                                          005400
      XPT=XS                                                            005410
      CALL PLOT(XS,YPT,2)                                               005420
      CALL PLOT(XS,YPT,3)                                               005430
      DO 130 I=1,10000                                                  005440
      YPT=YPTT-I*TICU                                                   005450
```

```
200         IF (YPT .LT. YS) GOTO 1030                                        005470
            CALL PLOT(XPT,YPT,2)                                              005480
            CALL PLOT(XPT+.1,YPT,2)                                           005490
            CALL PLOT(XPT,YPT,2)                                              005500
          130 CONTINUE                                                        005510
205       1030 CALL PLOT(XS,YS,2)                                             005520
            GOTO 230                                                          005530
          200 CALL PLOT(XS,YS,3)                                              005540
            ISW=CH=1                                                          005550
            XS=YS=0E0                                                         005560
210         CALL AXIS(XS,YS,XTITLE,-NX,XLEN,0E0,XMIN,SCALEX,20.)              005570
            CALL AXIS(XS,YS,YTITLE,NY,YLEN,90E0,YMIN,SCALEY,20.)              005580
        C NEXT PLOT PARAMETER AND CURVE TITLES                                005590
          230 CALL SYMBOL(XS+1E0,YS+YLEN+.2,.1E0,PTTTLE,0E0,NB)               005600
            SYM=SYMBL(TIM)                                                    005610
215         CALL SYMBOL(999E0,999E0,.1E0,SYM,0E0,10)                          005620
            XSS=XLEN+1E0                                                      005630
            YSS=YLEN-1E0                                                      005640
            CALL SYMBOL(XSS,YSS,.1E0,TITLE,0E0,NT)                            005650
        C PLOT CURVE                                                          005660
220       200 CALL PLOT(0E0,0E0,3)                                            005670
            GOTO (240,240,250,250) ISWITCH                                    005680
          240 CALL LINE(XARRAY,YARRAY,NPTS,1,5,ISYM,XMIN,SCALEX,YMIN,         005690
            *SCALEY,.07)                                                      005700
            GOTO 1050                                                         005710
225       250 XQ=YQ=-.1                                                       005720
            XLEM=XLEN+.1                                                      005730
            YLEM=YLEN+.1                                                      005740
            IP=2                                                              005750
            DO 140 I=1,NPTS                                                   005760
230         N=I                                                               005770
            XPT=(XARRAY(I)-XMIN)*SCAX                                         005780
            YPT=(YARRAY(I)-YMIN)*SCAY                                         005790
            IF (XPT .GE. XQ .AND. YPT .GE. YQ .AND. YPT .LE. YLEM) GOTO 1040  005800
          140 CONTINUE                                                        005810
235         WRITE(ITS,2)                                                      005820
            GOTO 1050                                                         005830
          2 FORMAT(1X,* NO POINTS PLOTTED*)                                   005840
          1040 IF (ISYM) 1070,1070,1080                                       005850
          1070 CALL PLOT(XPT,YPT,IP)                                          005860
240         IP=2                                                              005870
            GOTO 1060                                                         005880
          1080 CALL SYMBOL(XPT,YPT,.1,ISYM,0.,-1)                             005890
          1060 N=N+1                                                          005900
            IF (N .GT. NPTS) GOTO 1050                                        005910
245         YPT=(YARRAY(N)-YMIN)*SCAY                                         005920
            IF (YPT .GT. YLEM .OR. YPT .LT. YQ) GOTO 1065                     005930
            XPT=(XARRAY(N)-XMIN)*SCAX                                         005940
            IF (XPT .GT. XLEM .OR. XPT .LT. XQ) GOTO 1065                     005950
            GOTO 1040                                                         005960
250       1065 IP=3                                                           005970
            GOTO 1060                                                         005980
          1050 IF (ISYM .LT. 0) ISYM=15                                       005990
            YSS=YSS-.15                                                       006000
            IF (YSS .GT. 0.) GOTO 260                                         006010
255         YSS=YLEN-1.15                                                     006020
            XSS=XSS+3.                                                        006030
          260 CALL SYMBOL(XSS,YSS,.1E0,ISYM,0E0,-1)                           006040
            CALL SYMBOL(999E0,999E0,.1E0,2H  ,0E0,2)                          006050
            ENCODE(10,1,ISYMB) CURVE                                          006060
260         CALL SYMBOL(999E0,999E0,.1E0,ISYMB,0E0,10)                        006070
            ISYM=MOD(ISYM+1,17)                                               006080
            IF (CODE(7) .EQ. 2HSP) GOTO 560                                   006090
            M=M+MT                                                            006100
            GOTO 430                                                          006110
265       440 XPT=XLEN+6E0                                                    006120
            CALL PLOT(XPT,0E0,-3)                                             006130
          560 RETURN                                                          006140
          1 FORMAT(G10.3)                                                     006150
            END
```

E.3 Subroutine PARMPLT

```
      SUBROUTINE PARMPLT(XS,YS,HT,PARM,NS,N)                            000160
C THIS SUBROUTINE PLOTS N PARAMETERS IN 'PARM' STARTING AT LOCATION     000170
C XS,YS. HT IS THE HEIGHT OF THE LETTERS (INCHES),NS IS THE 'STARTING   000180
C LOCATION' IN PARM AT WHICH TO LOOK FOR DATA FOR PLOTTING.             000190
      DIMENSION PARM(1)                                                 000200
      IF (N .LT. 0) GOTO 2000                                           000210
      XPT=XS                                                            000220
      YPT=YS                                                            000230
      NS1=NS-1                                                          000240
      DO 100 I=1,N,2                                                    000250
      II=NS1+I                                                          000260
      CALL SYMBOL(XPT,YPT,HT,PARM(II),0E0,20)                           000270
      YPT=YPT-2.*HT                                                     000280
  100 CONTINUE                                                          000290
 2000 RETURN                                                            000300
      END                                                               000310
```

E.4 Subroutine FILL

```
      SUBROUTINE FILL(DATAIN,PARM,INDEX,ISIZE,NS,N)                     000320
C THIS SUBROUTINE SELECTS DATA FROM DATAIN(ISIZE,3) AND PLACES IT IN    000330
C 2-D ARRAY PARM STARTING AT NS ACCORDING TO NUMBERS IN INDEX. THE      000340
C PARM VALUES ARE CHACTER-STRINGS SUITABLE FOR PLOTTING.                000350
      DIMENSION DATAIN(1),PARM(1),INDEX(1)                              000360
      EQUIVALENCE (IF,FT)                                               000370
      II=2*NS-1                                                         000380
      DO 100 I=1,40                                                     000390
      K=INDEX(I)                                                        000400
      IF (K .EQ. 0) GOTO 100                                            000410
      II=II+2                                                           000420
      K1=K+ISIZE                                                        000430
      K2=K1+ISIZE                                                       000440
      FT=DATAIN(K2)                                                     000450
      PARM(II)=DATAIN(K1)                                               000460
      IF (IF) 1000,1010,1020                                            000470
 1000 ENCODE (10,1,PARM(II+1)) DATAIN(K)                                000480
      GOTO 100                                                          000490
 1010 ENCODE (10,2,PARM(II+1)) DATAIN(K)                                000500
      GOTO 100                                                          000510
 1020 ENCODE (10,3,PARM(II+1)) DATAIN(K)                                000520
  100 CONTINUE                                                          000530
      N=II                                                              000540
      RETURN                                                            000550
    1 FORMAT(A10)                                                       000560
    2 FORMAT(I10)                                                       000570
    3 FORMAT(G10.3)                                                     000580
      END                                                               000590
```

E.5 Subroutine CFRAME

```
      SUBROUTINE CFRAME(INDEX,ZMIN,ZMAX,CODE,NP,MP,TIM,PPI,C,XS,YS,
     *XMIN,XMAX,YMIN,YMAX,DATAIN,XLEN,YLEN,CCODE)
C INDEX IS A 2-D ARRAY OF INTEGERS (IN SEQUENCE) DENOTING THE
C STARTING LOCATION AND NUMBER OF WORDS IN DATAIN CONTAINING THE
C FOLLOWING INFORMATION-
C 1,1  FUNCTION TITLE
C 1,2  F-TITLE LENGTH (WORDS)
C 2,1  PARAMETER LABEL
C 2,2  P-LABEL LENGTH
C 3,1  X-LABEL
C 3,2  X-LABEL LENGTH
C 4,1  Y-LABEL
C 4,2  Y-LABEL LENGTH
C C IS THE FUNCTION NORMALIZATION CONSTANT
C XS,YS IS THE STARTING POINT OF THE FRAME
C T IS SUBROUTINE ASSUMES THAT THE PRECEDING PLOT WAS TERMINATED AT THE
C LOWER RIGHT CORNER OF ITS ALLOTTED SPACE (AND THIS IS THE NEW ORIGIN)
C
      DIMENSION DATAIN(100,3),INDEX(4,2)
      INTEGER CCODE,CODE
      TICH=5/PPI
C FIRST DRAW TITLE,SCALING INFO,PARAMETER
      NX=INDEX(1,2)
      IF (NX .EQ. 0) GOTO 170
      CALL SYMBOL(1E0,8,5E0,.15E0,DATAIN(INDEX(1,1),1),0E0,NX)
  170 IF (CCODE .EQ. 1) GOTO 215
      SYM=SYMBOL(TIM)
      NX=INDEX(2,2)
      IF (NX .EQ. 0) GOTO 200
      CALL SYMBOL(1.5E0,8,2E0,.1E0,DATAIN(INDEX(2,1),1),0E0,NX)
      GOTO 210
  200 CALL SYMBOL(1.5E0,8,2E0,.1E0,9HPARAMETER,0E0,9)
  210 CALL SYMBOL(999E0,999E0,.1E0,SYM,0E0,10)
  215 IF (CODE .EQ. 2HDW) GOTO 220
      CALL SYMBOL(1E0,8,E0,.1E0,24HALL FUNCTION VALUES HAVE ,0E0,24)
      CALL SYMBOL(1E0,7,8E0,.1E0,25HBEEN SCALED ACCORDING TO=,0E0,25)
      CALL SYMBOL(1,E0,7,4E0,.1E0,1HZ,0E0,1)
      CALL ARROW(1,5,7,5,,3,180,,1H ,0)
      ENCODE(10,1,SYM) C
      CALL SYMBOL(1,6E0,7,6E0,.1E0,SYM,0E0,10)
      CALL SYMBOL(999,E0,999,E0,.1E0,9H*(Z-ZMIN),0E0,9)
      CALL PLOT(1,9E0,7,5E0,3)
      CALL PLOT(3,1E0,7,5E0,2)
      CALL SYMBOL(2E0,7,3E0,.1E0,11H(ZMAX-ZMIN),0E0,11)
      ENCODE(10,1,SYM) ZMAX
      CALL SYMBOL(1E0,7,1E0,.1E0,11HWHERE ZMAX=,0E0,11)
      CALL SYMBOL(999E0,999E0,.1E0,SYM,0E0,10)
      ENCODE(10,1,SYM) ZMIN
    1 FORMAT(G10,3)
      CALL SYMBOL(1E0,6,9E0,.1E0,11H      ZMIN=,0E0,11)
      CALL SYMBOL(999E0,999E0,.1E0,SYM,0E0,10)
      CALL SYMBOL(1E0,6,7E0,.1E0,2)HARE THE MAX. AND MIN.,0E0,20)
      IF (CODE .EQ. 2HNA) 110,120
  110 CALL SYMBOL(1E0,6,5E0,.1E0,24HVALUES OVER ALL SURFACES,0E0,24)
      GOTO 100
  120 CALL SYMBOL(1E0,6,5E0,.1E0,21HVALUES OF THE SURFACE,0E0,21)
  100 CONTINUE
C NEXT SET THE PLOT ORIGIN AT THE ORIGIN OF THE SURFACE COORDINATE
C FRAME,DRAW THE FRAME AND LABEL IT.
  220 CALL PLOT(XS,YS,-3)
C THE COORDINATE FRAME IS DRAWN WITH THE TIC-MARKS EVERY TICH (INCHES).
      ENCODE(10,1,SYM) YMIN
      CALL SYMBOL(0E0,-,2E0,.1E0,SYM,0E0,10)
      NX=INDEX(3,2)
      IF (NX .EQ. 0) GOTO 180
      CALL SYMBOL(0E0,-,5E0,.1E0,DATAIN(INDEX(3,1),1),0E0,NX)
  180 ENCODE(10,1,SYM) YMAX
      XPT=0E0
      YPT=0E0
      CALL PLOT(XPT,YPT,3)
      DO 130 I=1,100000
      N=I
      XPT=I*TICH
      IF (XPT .GT. XLEN) GOTO 1000
```

```
      CALL PLOT(XPT,0E0,2)                                              007340
      CALL PLOT(XPT,,1,2)                                               007350
      CALL PLOT(XPT,0E0,2)                                              007360
  130 CONTINUE                                                          007370
 1000 XPT=N*TICU                                                        007380
      XPT=XLEN                                                          007390
      CALL PLOT(XPT,0E0,2)                                              007400
      CALL SYMBOL(XPT-,5,-,2,,1,SYM,0E0,10)                             007410
      CALL PLOT(XPT,0E0,3)                                              007420
      DO 140 I=1,100000                                                 007430
      N=I                                                               007440
      YPT=I*TICU                                                        007450
      IF (YPT .GT. YLEN) GOTO 1010                                      007460
      CALL PLOT(XPT,YPT,2)                                              007470
      CALL PLOT(XPT-,1,YPT,2)                                           007480
      CALL PLOT(XPT,YPT,2)                                              007490
  140 CONTINUE                                                          007500
 1010 YPTT=N*TICU                                                       007510
      YPT=YLEN                                                          007520
      CALL PLOT(XPT,YPT,2)                                              007530
      DO 150 I=1,100000                                                 007540
      N=I                                                               007550
      XPT=XPTT-I*TICU                                                   007560
      IF (XPT .LT. 0E0) GOTO 1020                                       007570
      CALL PLOT(XPT,YPT,2)                                              007580
      CALL PLOT(XPT,YPT-,1,2)                                           007590
      CALL PLOT(XPT,YPT,2)                                              007600
  150 CONTINUE                                                          007610
 1020 XPT=0E0                                                           007620
      CALL PLOT(XPT,YPT,2)                                              007630
      ENCODE(10,1,SYM) YMAX                                             007640
      YPTS=YPT-,5                                                       007650
      IF (YPT .GT. 9,5) YPTS=9,0                                        007660
      CALL SYMBOL(-,1E0,YPTS,,1E0,SYM,90E0,10)                          007670
      CALL PLOT(0E0,YPT,3)                                              007680
      DO 160 I=1,100000                                                 007690
      YPT=YPTT-I*TICU                                                   007700
      IF (YPT .LT. 0E0) GOTO 1030                                       007710
      CALL PLOT(XPT,YPT,2)                                              007720
      CALL PLOT(XPT,,1,YPT,2)                                           007730
      CALL PLOT(XPT,YPT,2)                                              007740
  160 CONTINUE                                                          007750
 1030 CALL PLOT(0E0,0E0,2)                                              007760
      ENCODE(10,1,SYM) YMIN                                             007770
      CALL SYMBOL(-,1E0,0E0,,1E0,SYM,90,,10)                            007780
      NX=INDEX(4,2)                                                     007790
      IF (NX .EQ, 0) GOTO 190                                           007800
      CALL SYMBOL(-,4E0,0E0,,1E0,DATAIN(INDEX(4,1),1),90,,NX)           007810
  190 CALL PLOT(0E0,0E0,3)                                              007820
      RETURN                                                            007830
      END                                                               007840
```

E.6 Subroutine INTERP

```
      SUBROUTINE INTERP(CCODE,CODE,FLAG,ICARD)                          007850
      COMMON/LET/LETTER(80)                                             007860
      DIMENSION CODE(7)                                                 007870
      LOGICAL FLAG                                                      007880
      INTEGER CCODE,BLANK,C,L,LPAREN,RPAREN,Y,D,E                       007890
      DATA BLANK,C,L,LPAREN,RPAREN,Y,D,E/00A0000000000000000055B,       007900
     +000^0000^0000000^000038,0000000000000000^00148,0000000000000^0000518  007910
     +,00^0000^0000000000005BR,0000A0000000000^00031B,                  007920
     +000^0000^0000000^04R,0000000000^0000^0005B/                       007930
      CCODE=2                                                           007940
      CODE(2)=1 .OR.                                                    007950
      DO 90 I=1,80                                                      007960
      LETTER(I)=BLANK                                                   007970
   90 CONTINUE                                                          007980
```

```
      READ(ICARD,1) (LETTER(I),I=1,80)                                  000790
      IF (EOF(ICARD)) 115,100                                           000800
  115 FLAG=.F.                                                          000810
      GOTO 110                                                          000820
  100 CONTINUE                                                          000830
    1 FORMAT(80R1)                                                      000840
      DO 120 I=1,80                                                     000850
      N=I                                                               000860
      IF (LETTER(I) .NE. BLANK) GOTO 130                                000870
  120 CONTINUE                                                          000880
  130 IF (I .EQ. 81) GOTO 110                                           000890
      IF (LETTER(N) .EQ. C) CCODE=1                                     000900
      IF (LETTER(N+1) .EQ. L) CCODE=3                                   000910
      GOTO (131,134,133) CCODE                                          000920
C SET DEFAULT VALUES OF CODE                                            000930
  131 CODE(1)=1 .OR. 0                                                  000940
      CODE(3)=10 .OR. 0                                                 000950
      CODE(4)=2HNA                                                      000960
      CODE(5)=10H                                                       000970
      GOTO 132                                                          000980
  134 CODE(1)=1 .OR. 0                                                  000990
      CODE(3)=45.                                                       001000
      CODE(4)=2HNA                                                      001010
      GOTO 132                                                          001020
  133 CODE(1)=1 .OR. 0                                                  001030
      CODE(3)=1HX                                                       001040
      CODE(4)=1 .OR. 0                                                  001050
      CODE(5)=1 .OR. 0                                                  001060
      CODE(6)=2HNA                                                      001070
      CODE(7)=10H                                                       001080
  132 DO 140 I=N,80                                                     001090
      M=I                                                               001100
      IF (LETTER(I) .EQ. LPAREN) GOTO 150                               001110
  140 CONTINUE                                                          001120
  150 IF (I .EQ. 81) GOTO 110                                           001130
      CALL NUMB(CODE(1),M,1)                                            001140
C IF M=0 THEN A RIGHT PAREN HAS BEEN FOUND,OTHERWISE IT RETURNS         001150
C THE POSITION OF THE COMMA.                                            001160
      IF (M .EQ. 0) GOTO 110                                            001170
      CALL NUMB(CODE(2),M,1)                                            001180
      IF (M .EQ. 0) GOTO 110                                            001190
      GOTO (160,170,180) CCODE                                          001200
  160 CALL NUMB(CODE(3),M,1)                                            001210
      IF (M .EQ. 0) GOTO 110                                            001220
      CODE(4)=2HNA                                                      001230
      IF (LETTER(M+1) .EQ. D) CODE(4)=2HDN                              001240
      IF (LETTER(M+2) .EQ. E) CODE(4)=2HNE                              001250
      IF (LETTER(M+3) .EQ. 2RB) CODE(5)=2HSP                            001260
      GOTO 110                                                          001270
  170 CALL NUMB(CODE(3),M,2)                                            001280
      IF (M .EQ. 0) GOTO 110                                            001290
      IF (LETTER(M+2) .EQ. E) CODE(4)=2HNE                              001300
      GOTO 110                                                          001310
  180 IF (LETTER(M+1) .EQ. Y) CODE(3)=1HY                               001320
      IF (LETTER(M+2) .EQ. RPAREN) GOTO 110                             001330
      IF (LETTER(M+1) .EQ. 5RB) M=M-1                                   001340
      M=M+2                                                             001350
      CALL NUMB(CODE(4),M,1)                                            001360
      IF (M .EQ. 0) GOTO 110                                            001370
      CALL NUMB(CODE(5),M,1)                                            001380
      IF (M .EQ. 0) GOTO 110                                            001390
      CODE(6)=2HNA                                                      001400
      IF (LETTER(M+1) .EQ. D) CODE(6)=2HDN                              001410
      IF (LETTER(M+2) .EQ. E) CODE(6)=2HNE                              001420
      IF (LETTER(M+3) .EQ. 2RB) CODE(7)=2HSP                            001430
  110 RETURN                                                            001440
      END                                                               001450
```

E.7 Subroutine NUMB

```
      SUBROUTINE NUMB(PARAM,M,IFORMAT)                                  008660
      COMMON/LET/LETTER(80)                                             008670
      INTEGER RPAREN,COMMA                                              008680
      DATA RPAREN,COMMA/00000000000000000052B,00000000000000000056B/    008690
      IPARAM=1HB                                                        008700
C IFORMAT IS THE FORMAT CODE FOR DECODING-1=>INTEGER,2=>FLOAT.          008710
C M IS THE POSITION IN LETTER OF A 'DELIMITER' SUCH AS ( OR *. NUMB     008720
C RETURNS M=0 IF A ) IS FOUND.                                          008730
      M1=M+1                                                            008740
      NUM=0                                                             008750
      DO 100 I=M1,80                                                    008760
      M=I                                                               008770
      IF (LETTER(I) .EQ. RPAREN) 110,120                                008780
  110 M=0                                                               008790
      GOTO 130                                                          008800
  120 IF (LETTER(I) .EQ. COMMA) 130,150                                 008810
  150 NUM=NUM+1                                                         008820
C STORE THE RIGHT MOST CHARACTER FORM LETTER(I) AT CHARACTER            008830
C NUM OF IPARAM.                                                        008840
      IPARAM=MVBITX(LETTER(I),IPARAM,NUM)                               008850
  100 CONTINUE                                                          008860
  130 IF (I .EQ. M1) GOTO 160                                           008870
      IF(IFORMAT.EQ.1)  170,180                                         008880
  170 CALL RJUST(IPARAM)                                                008890
      DECODE (10,1,IPARAM) PARAM                                        008900
    1 FORMAT(I10)                                                       008910
      GOTO 160                                                          008920
  180 CALL RJUST(IPARAM)                                                008925
      DECODE (10,2,IPARAM)PARAM                                         008930
    2 FORMAT(F10.0)                                                     008940
  160 RETURN                                                            008950
      END                                                               008960
```

E.8 Subroutine ARROW

```
      SUBROUTINE ARROW(XP,YP,L,THETAD,LABEL,NC)                         008970
C     THETAD IS THE ARROW ANGLE WRT X-AXIS IN DEGREES                   008980
C     L IS THE LENGTH OF THE ARRAW IN INCHES                            008990
C     THE LOCATION OF THE ARROW TIP (XP,YP) IS RETURNED                 009000
C NC IS THE NUMBER OF CHARACTERS IN THE ARROW LABEL                     009010
      REAL L                                                            009020
      DIMENSION LABEL(1)                                                009030
      THETA=THETAD*3.1415927/180.                                       009040
      XXP=L*COS(THETA)+XP                                               009050
      YYP=L*SIN(THETA)+YP                                               009060
      XL=.2*L*COS(THETA-.1745)+XP                                       009070
      YL=.2*L*SIN(THETA-.1745)+YP                                       009080
      XU=.2*L*COS(THETA+.1745)+XP                                       009090
      YU=.2*L*SIN(THETA+.1745)+YP                                       009100
      CALL PLOT(XP,YP,3)                                                009110
      CALL PLOT(XXP,YYP,2)                                              009120
      CALL PLOT(XP,YP,3)                                                009130
      CALL PLOT(XL,YL,2)                                                009140
      CALL PLOT(XP,YP,3)                                                009150
      CALL PLOT(XU,YU,2)                                                009160
      XU=.25*L*COS(THETA+.0873)+XP                                      009170
      YU=.25*L*SIN(THETA+.0873)+YP                                      009180
      IF (NC .LE. 0) GOTO 1000                                          009190
      CALL SYMBOL(XU,YU,.1,LABEL,THETAD,NC)                             009200
 1000 RETURN                                                            009210
      END                                                               009220
```

E.9 Subroutine APLACE

```
      SUBROUTINE APLACE(XP,YP,XLEN1,YLEN1,THETA2)                       000230
C THIS SUBROUTINE FINDS THE X,Y, COURDINATES (XP,YP) FOR THE TIP        000240
C OF THE ARROW WHICH INDICATES THE VIEW ANGLE (THETA2-DEGREES)          000250
C FOR PERSPECTIVE PLOTS                                                 000260
      X2=.5*XLEN1                                                       000270
      Y2=.5*YLEN1                                                       000280
      PI180=3.1415927/180.                                              000290
      TH=PI180*THETA2                                                   000300
      R=SQRT((X2+.5)**2+(Y2+.5)**2)                                     000310
      XP=X2+R*COS(TH)                                                   000320
      YP=Y2+R*SIN(TH)                                                   000330
      RETURN                                                            000340
      END                                                               000350
```

E.10 Subroutine RD1

```
      SUBROUTINE RD1(IT3,NP,MP,ZMAX,ZMIN,ISW,FLAG,TIM,XMIN,XMAX,YMIN,   000360
     +YMAX)                                                             000370
      COMMON Z(103,103)                                                 000380
      DIMENSION BUF(101)                                                000390
      LOGICAL FLAG                                                      000400
C THIS SUBROUTINE FILLS UP THE 'SURFACE ARRAY' Z(...) AND/OR FINDS THE  000410
C SURFACE MAX. AND MIN. (ZMAX,ZMIN). IT ALSO RETURNS THE SIZE OF THE    000420
C SURFACE (MP X NP) AND THE SURFACE PARAMETER VALUE, TIM. THE LOGICAL   000430
C VARIABLE, FLAG, IS .TRUE. UNTIL AN END-OF-FILE IS REACHED AT WHICH    000440
C POINT IT BECOMES FALSE. EACH UNFORMATTED RECORD OF THE INPUT FILE,IT3, 000450
C IS ASSUMED TO BE OF THE FORM-                                         000460
C     I        ROW NUMBER                                             000470
C     NP       TOTAL NO. OF ROWS                                      000480
C     Y        Y-VALUE CORRESPONDING TO THIS ROW                      000490
C     TIM      SURFACE PARAMETER VALUE (E.G. TIME)                    000500
C     MP       TOTAL NO. OF COLUMNS                                   000510
C     XMIN     VALUE ASSOCIATED WITH THE FIRST COLUMN                 000520
C     XMAX     VALUE ASSOCIATED WITH THE LAST COLUMN                  000530
C     BUF(J)   MP SURFACE ELEMENTS OF THE I-TH ROW                    000540
C ISW ACTS AS A SWITCH TO CONTROL THE ACTION OF RD1 AS FOLLOWS-         000550
C ISW=1 => RETURN ZMAX,ZMIN ONLY                                        000560
C ISW=2 => RETURN SURFACE Z AND ZMAX,ZMIN                               000570
C ISW=3 => RETURN SURFACE Z BORDERED BY ZEROES AND ZMAX,ZMIN            000580
C ISW=4 => SKIP A SURFACE                                               000585
C NOTE THAT THE FIRST RECORD READ IS ASSUMED TO INCLUDE THE FIRST       000590
C 'ROW' OF THE SURFACE                                                  000600
  100 READ(IT3) I,NP,Y,TIM,MP,XMIN,XMAX,(BUF(J),J=1,MP)                 000610
      IF (EOF(IT3)) 110,120                                             000620
  110 FLAG=.F.                                                          000630
      GOTO 1000                                                         000640
  120 IF (ISW .EQ. 4) GOTO 230                                          000645
      I1=I+1                                                            000650
      Z(I1,1)=Z(I1,MP+2)=0.                                             000660
      IF (I .EQ. 1) 210,220                                             000670
  210 ZMIN=ZMAX=BUF(1)                                                  000680
      YMIN=Y                                                            000690
      MP2=MP+2                                                          000700
      DO 90 J=1,MP2                                                     000710
   90 Z(1,J)=0.                                                         000720
  220 IF (I .EQ. NP) YMAX=Y                                             000730
      DO 160 J=1,MP                                                     000740
      ZT=BUF(J)                                                         000750
      IF (ZT .LT. ZMIN) ZMIN=ZT                                         000760
      IF (ZT .GT. ZMAX) ZMAX=ZT                                         000770
      GOTO (160,170,180) ISW                                            000780
  170 Z(I,J)=ZT                                                         000790
      GOTO 160                                                          000800
  180 Z(I1,J+1)=ZT                                                      000810
      GOTO 160                                                          000820
```

```
50          140  CONTINUE                                                  009830
            230  IF (I .LT. NP) GOTO 100                                   009840
                 GOTO (1000,1000,190,1000) ISW                             009850
            190  MP2=MP+2                                                  009860
                 NP2=NP+2                                                  009870
55               DO 200 I=1,MP2                                            009880
            200  Z(NP2,J)=0.                                               009890
           1000  RETURN                                                    009900
                 END                                                       009910
```

E.11 Subroutine CLEV

```
1                SUBROUTINE CLEV(NLEV,ZMAX,ZMIN,ZLEVS)                     009920
                 DIMENSION ZLEVS(1)                                        009930
           C THIS SUBROUTINE RETURNS NLEV EQUI-SPACED CONTOUR LEVELS BETWEEN ZMIN  009940
           C AND ZMAX (BUT NOT INCLUDING ZMIN)                             009950
5                DEL=(ZMAX-ZMIN)/NLEV                                      009960
                 DO 100 I=1,NLEV                                           009970
                 ZLEVS(I)=ZMIN+I*DEL                                       009980
            100  CONTINUE                                                  009990
                 RETURN                                                    010000
10               END                                                       010010
```

E.12 Subroutine SKIP

```
1                SUBROUTINE SKIP(ITAPE,NSKIP,FLAG,N)                       010020
                 LOGICAL FLAG                                              010030
                 IF (NSKIP .EQ. 0) GOTO 120                                010040
                 NREC=N*NSKIP                                              010050
5                DO 100 I=1,NREC                                           010060
                 READ(ITAPE)                                               010070
                 IF (EOF(ITAPE)) 110,100                                   010080
            100  CONTINUE                                                  010090
                 GOTO 120                                                  010100
10          110  FLAG=.F.                                                  010110
            120  RETURN                                                    010120
                 END                                                       010130
```

E.13 Subroutine FACE

```
1                SUBROUTINE FACE(XP1,YP1,ZPP,THETA1,THETA2,NREFIN)         010140
                 DIMENSION R(3,3)                                          010150
                 COMMON/BLOCK3/INUN,JNUN,XMIN,YMIN,DX,DY,SWI,IT6           010160
                 COMMON DATA(103,103,2)                                    010170
5                WRITE(IT6,101)XP1,YP1,ZPP,THETA1,THETA2,INUN,JNUN,NREFIN  010180
            101  FORMAT(3(F6.2,1X),2(F7.4,1X),3(I3,1X))                    010190
                 YES=3HYES                                                 010200
                 YNO=2HNO                                                  010210
                 INUM=INUN                                                 010220
10               JNUM=JNUN                                                 010230
                 YP=YPP=YP1                                                010240
                 XP=XPP=XP1                                                010250
                 XB = ZPP*COS(THETA2)/TAN(THETA1)                          010260
                 YB = ZPP*SIN(THETA2)/TAN(THETA1)                          010270
```

```
      XBAR = XPP-XB                                                     010280
      YBAR = YPP-YB                                                     010290
      XPP = XPP-XBAR                                                    010300
      YPP = YPP-YBAR                                                    010310
      DIS = SQRT(XPP*XPP+YPP*YPP+ZPP*ZPP)                               010320
      R(1,1) = -SIN(THETA2)                                             010330
      R(1,2) = COS(THETA2)                                              010340
      R(1,3) = 0.0                                                      010350
      R(2,1) = -SIN(THETA1)*COS(THETA2)                                 010360
      R(2,2) = -SIN(THETA1)*SIN(THETA2)                                 010370
      R(2,3) = COS(THETA1)                                              010380
      R(3,1) = COS(THETA1)*COS(THETA2)                                  010390
      R(3,2) = COS(THETA1)*SIN(THETA2)                                  010400
      R(3,3) = SIN(THETA1)                                              010410
      DO 30 I=1,INUM                                                    010420
      A = FLOAT(I)                                                      010430
      Y = A-YBAR                                                        010440
      DO 30 J=1,JNUM                                                    010450
      B = FLOAT(J)                                                      010460
      X = B-XBAR                                                        010470
      Z = DATA(I,J,1)                                                   010480
      DATA(I,J,1) = R(1,1)*X + R(1,2)*Y + R(1,3)*Z                      010490
      DATA(I,J,2) = R(2,1)*X + R(2,2)*Y + R(2,3)*Z                      010500
      Z = DIS-(R(3,1)*X + R(3,2)*Y + R(3,3)*Z)                          010510
      DATA(I,J,1) = DATA(I,J,1)/Z                                       010520
20    DATA(I,J,2) = DATA(I,J,2)/Z                                       010530
      INUM1 = INUM/2                                                    010540
      JNUM1 = JNUM/2                                                    010550
      IF (YP .EQ. 1.0) GO TO 45                                         010560
      IF (YP .GE. FLOAT(INUM)) GO TO 30                                 010570
      IF (XP .EQ. 1.0) GO TO 40                                         010580
      IF (XP .GE. FLOAT(JNUM)) GO TO 30                                 010590
30    DO 35 J=1,JNUM                                                    010600
      DO 35 I=1,INUM1                                                   010610
      T1 = DATA(I,J,1)                                                  010620
      T2 = DATA(I,J,2)                                                  010630
      DATA(I,J,1) = DATA(INUM+1-I,J,1)                                  010640
      DATA(I,J,2) = DATA(INUM+1-I,J,2)                                  010650
      DATA(INUM+1-I,J,1) = T1                                           010660
35    DATA(INUM+1-I,J,2) = T2                                           010670
      GO TO 45                                                          010680
39    XP = YES                                                          010690
40    ITEMP = INUM                                                      010700
      INUM = JNUM                                                       010710
      JNUM = ITEMP                                                      010720
      ITEMP = MAX0(INUM,JNUM)                                           010730
      DO 42 I=1,ITEMP                                                   010740
      DO 41 J=1,I                                                       010750
      T1 = DATA(I,J,1)                                                  010760
      T2 = DATA(I,J,2)                                                  010770
      DATA(I,J,1) = DATA(J,I,1)                                         010780
      DATA(J,I,1) = T1                                                  010790
      DATA(I,J,2) = DATA(J,I,2)                                         010800
41    DATA(J,I,2) = T2                                                  010810
42    CONTINUE                                                          010820
      IF (XP .EQ. YES) GO TO 30                                         010830
45    IF (DATA(1,1,1) .LT. DATA(1,JNUM,1)) GO TO 46                     010840
      IF (DATA(INUM,JNUM,1) .LT. DATA(1,JNUM,1)) GO TO 50               010850
      GO TO 99                                                          010860
46    IF (DATA(1,JNUM,1) .LT. DATA(INUM,JNUM,1)) GO TO 50               010870
      GO TO 99                                                          010880
50    DO 55 I=1,INUM                                                    010890
      DO 55 J=1,JNUM1                                                   010900
      T1 = DATA(I,J,1)                                                  010910
      T2 = DATA(I,J,2)                                                  010920
      DATA(I,J,1) = DATA(I,JNUM+1-J,1)                                  010930
      DATA(I,J,2) = DATA(I,JNUM+1-J,2)                                  010940
      DATA(I,JNUM+1-J,1) = T1                                           010950
55    DATA(I,JNUM+1-J,2) = T2                                           010960
99    CALL HIDE                                                         010970
1000  CONTINUE                                                          010980
      RETURN                                                            010990
      END                                                               011000
```

E.14 Subroutine HIDE

```
      SUBROUTINE HIDE                                                   011010
      COMMON DATA(103,103,2)/BLOCK1/TEST(500,2),TEST1(500,2),NUM/BLOCK2/ 011020
     1P(2),Q(2),JCUT                                                    011030
      COMMON/BLOCK3/INUM, JNUM,XMIN,YMIN,DX,DY,SWI,IT6                  011040
      LOGICAL SWI                                                       011050
      SIZE=8.                                                           011060
      KFLAG = 0                                                         011070
      JFLAG = 0                                                         011080
      IF (SWI) GOTO 2000                                                011090
      SMALL = DATA(1,1,1)                                               011100
      BIG = SMALL                                                       011110
      DO 5 J=1,JNUM                                                     011120
      DO 5 I=1,INUM                                                     011130
      IF (DATA(I,J,1) .LT. SMALL) SMALL = DATA(I,J,1)                   011140
    5 IF (DATA(I,J,1) .GT. BIG) BIG = DATA(I,J,1)                       011150
      XMIN = SMALL                                                      011160
      DX = (BIG-SMALL)/SIZE                                             011170
      SMALL = DATA(1,1,2)                                               011180
      BIG = SMALL                                                       011190
      DO 6 I=1,INUM                                                     011200
      DO 6 J=1,JNUM                                                     011210
      IF (DATA(I,J,2) .LT. SMALL) SMALL = DATA(I,J,2)                   011220
    6 IF (DATA(I,J,2) .GT. BIG) BIG = DATA(I,J,2)                       011230
      YMIN = SMALL                                                      011240
      DY = (BIG-SMALL)/SIZE                                             011250
      D = DY                                                            011260
      IF (DX .GE. DY) D=DX                                              011270
      DY = D                                                            011280
      DX = D                                                            011290
      GOTO 2010                                                         011300
 2000 DO 10 I=1,JNUM                                                    011310
      TEST(I,1) = DATA(1,I,1)                                           011320
      TEST(I,2) = DATA(1,I,2)                                           011330
   10 TEST1(I,1) = TEST(I,2)                                            011340
      CALL LINE(TEST,TEST1,JNUM,1,0,0,XMIN,DX,YMIN,DY,0.)               011350
 2010 I = 1                                                             011360
      NUM = JNUM                                                        011370
      ICUT = 0                                                          011380
      JCUT = 2                                                          011390
   20 JCUT = JCUT + 1                                                   011400
      KFLAG = 0                                                         011410
      JFLAG = 0                                                         011420
      IF (JCUT .EQ. JNUM+1) GO TO 100                                   011430
   30 I = 0                                                             011440
      DO 35 I=2,NUM                                                     011450
      IF ((DATA(ICUT,JCUT,1) .LE. TEST(I,1)) .AND. (DATA(ICUT,JCUT,1)   011460
     1.GE. TEST(I-1,1))) GO TO 31                                       011470
      IF ((DATA(ICUT,JCUT,1) .LT. TEST(I,1)) .OR. (DATA(ICUT,JCUT,1)    011480
     1.GT. TEST(I-1,1))) GO TO 35                                       011490
   31 B = (TEST(I,2)-TEST(I-1,2))/(TEST(I,1)-TEST(I-1,1))               011500
      C = TEST(I,2) - B*TEST(I,1)                                       011510
      Y = B*DATA(ICUT,JCUT,1) + C                                       011520
      IF (Y .LE. DATA(ICUT,JCUT,2)) GO TO 35                            011530
      I = J + 1                                                         011540
   35 CONTINUE                                                          011550
      IF (2*(J/2) .EQ. J) GO TO 120                                     011560
      GO TO 400                                                         011570
  100 JCUT = 1                                                          011580
      ICUT = ICUT + 1                                                   011590
      IF (ICUT .NE. INUM+1) GO TO 30                                    011600
      RETURN                                                            011610
  120 IF (JCUT .EQ. 1) GO TO 200                                        011620
      P(1) = DATA(ICUT,JCUT,1)                                          011630
      P(2) = DATA(ICUT,JCUT-1,1)                                        011640
      Q(1) = DATA(ICUT,JCUT,2)                                          011650
      Q(2) = DATA(ICUT,JCUT-1,2)                                        011660
      IF (P(2) .GE. 6.0) GO TO 130                                      011670
      X = P(2)                                                          011680
      Y = Q(2)                                                          011690
      GO TO 200                                                         011700
  130 CALL DRAW(X,Y,K,KFLAG)                                            011710
      IF (K .GT. NUM) GO TO 365                                         011720
  200 P(1) = DATA(ICUT,JCUT,1)                                          011730
      P(2) = DATA(ICUT-1,JCUT,1)                                        011740
```

```
      P(1) = DATA(ICUT, JCUT,2)                                          011750
      P(2) = DATA(ICUT-1, JCUT,2)                                        011760
      IF (P(2) .GE. 6.0) GO TO 210                                       011770
      X1 = P(2)                                                          011780
      Y1 = P(2)                                                          011790
      GO TO 300                                                          011800
  210 CALL DRAW(X1,Y1,J,JFLAG)                                           011810
      IF (J .GT. NUM) GO TO 365                                          011820
  300 IF (JCUT .NE. 1) GO TO 305                                         011830
      K = I                                                              011840
      GO TO 314                                                          011850
  305 CONTINUE                                                           011860
  600 FORMAT (5H HERE)                                                   011870
      IF (KFLAG .EQ. 1) GO TO 314                                        011880
      DO 310 I=1,NUM                                                     011890
      IF ((X .EQ. TEST(I,1)) .AND. (Y .EQ. TEST(I,2)))GO TO 313          011900
  310 CONTINUE                                                           011910
  311 FORMAT (10H 310 ERROR)                                             011920
      GO TO 365                                                          011930
  313 K = I                                                              011940
  314 IF (JFLAG .EQ. 1) GO TO 320                                        011950
      DO 315 I=1,NUM                                                     011960
      IF ((X1 .EQ. TEST(I,1)) .AND. (Y1 .EQ. TEST(I,2))) GO TO 318       011970
  315 CONTINUE                                                           011980
  316 FORMAT (10H 316 ERROR)                                             011990
      GO TO 365                                                          012000
  318 J = I                                                              012010
C     IF IFLAG IS SET, WE ARE LOOKING AT THE BACK OF THE SQUARE PIECE    012020
C WHOSE CORNER IS DATA(ICUT,JCUT).  IN THIS CASE WE DO NOT DRAW THE      012030
C LINE TO DATA(ICUT,JCUT-1), AND WE DO NOT INCLUDE IT IN THE SKYLINE,    012040
C I.E. TEST.                                                             012050
  320 IFLAG = 0                                                          012060
      IF (JFLAG .EQ. 0) J = J + 1                                        012070
      IF (J .LT. K) IFLAG = 1                                            012080
      IF ((J .EQ. K) .AND. (ABS(X1-TEST(J,1)) .GT. ABS(X-TEST(J,1))))   012090
     1IFLAG = 1                                                          012100
      JFLAG = 0                                                          012110
      KFLAG = 0                                                          012120
      II = 0                                                             012130
      DO 330 I=J,NUM                                                     012140
      II = II + 1                                                        012150
      TEST1(II,1) = TEST(I,1)                                            012160
  330 TEST1(II,2) = TEST(I,2)                                            012170
      IF (IFLAG .EQ. 0) GO TO 331                                        012180
      K = J                                                              012190
      X = X1                                                             012200
      Y = Y1                                                             012210
      GO TO 335                                                          012220
  331 IF (JCUT .NE. 1) GO TO 335                                         012230
      K = K - 1                                                          012240
      GO TO 340                                                          012250
  335 TEST(K,1) = X                                                      012260
      TEST(K,2) = Y                                                      012270
  340 TEST(K+1,1) = DATA(ICUT,JCUT,1)                                    012280
      TEST(K+1,2) = DATA(ICUT,JCUT,2)                                    012290
      TEST(K+2,1) = X1                                                   012300
      TEST(K+2,2) = Y1                                                   012310
      K = K + 2                                                          012320
      IF (J .EQ. (NUM + 1)) GO TO 350                                    012330
      DO 345 I=1,II                                                      012340
      K = K + 1                                                          012350
      TEST(K,1) = TEST1(I,1)                                             012360
  345 TEST(K,2) = TEST1(I,2)                                             012370
  641 FORMAT (1H ,3I10)                                                  012380
  350 NUM = K                                                            012390
      IF (JCUT .EQ. 1) GO TO 355                                         012400
      IF (IFLAG .EQ. 1) GO TO 355                                        012410
      P(2) = X                                                           012420
      Q(2) = Y                                                           012430
      CALL LINE(P,Q,2,1,0.0,XMIN,DX,YMIN,DY,0.)                          012440
  355 P(2) = X1                                                          012450
      Q(2) = Y1                                                          012460
      CALL LINE(P,Q,2,1,0.0,XMIN,DX,YMIN,DY,0.)                          012470
      GO TO 20                                                           012480
  400 CONTINUE                                                           012490
  401 FORMAT (4H 400,2X,I10)                                             012500
      IF (JCUT .EQ. 1) GO TO 500                                         012510
      P(1) = DATA(ICUT, JCUT-1,1)                                        012520
      P(2) = DATA(ICUT, JCUT,1)                                          012530
```

```
40          Q(1) = DATA(ICUT, JCUT-1,2)                                    012540
            Q(2) = DATA(ICUT, JCUT,2)                                      012550
            IF (P(1) .GE. 6.0) GO TO 500                                   012560
            CALL DRAW(X,Y,K,KFLAG)                                         012570
            IF (K .GT. NUM) GO TO 545                                      012580
45          P(2) = X                                                       012590
            Q(2) = Y                                                       012600
            CALL LINE(P,Q,2,1,0.0,XMIN,DX,YMIN,DY,0.)                      012610
420         IC = ICUT                                                      012620
            JC = JCUT - 1                                                  012630
50          CALL SORT(IC,JC,K,X,Y)                                         012640
500         P(1) = DATA(ICUT-1, JCUT,1)                                    012650
            P(2) = DATA(ICUT, JCUT,1)                                      012660
            Q(1) = DATA(ICUT-1, JCUT,2)                                    012670
            Q(2) = DATA(ICUT, JCUT,2)                                      012680
55          IF (P(1) .GE. 6.0) GO TO 545                                   012690
            CALL DRAW(X,Y,J,KFLAG)                                         012700
            IF (J .GT. NUM) GO TO 545                                      012710
            P(2) = X                                                       012720
            Q(2) = Y                                                       012730
            CALL LINE(P,Q,2,1,0.0,XMIN,DX,YMIN,DY,0.)                      012740
175         520 IC = ICUT - 1                                              012750
            JC = JCUT                                                      012760
            CALL SORT(IC,JC,J,X,Y)                                         012770
            545 DATA(ICUT,JCUT,1) = DATA(ICUT,JCUT,1) + SIZE               012780
            GO TO 365                                                      012790
180         2510 RETURN                                                    012800
            END                                                            012810
```

E.15 Subroutine DRAW

```
1           SUBROUTINE DRAW(X,Y,K,KFLAG)                                   012820
            COMMON /BLOCK1/TEST(500,2),TEST1(500,2),NUM/BLOCK2/P(2),Q(2),JCUT 012830
            SIZE=0.                                                        012840
            DIS = 0.                                                       012850
5           IF (P(2) .GE. 0.0) GO TO 10                                    012860
            GO TO 15                                                       012870
10          P(2) = P(2) - SIZE                                             012880
15          Y = 15.0                                                       012890
            S = .0001                                                      012900
            B1 = (Q(2)-Q(1))/(P(1)-P(2))                                   012910
            IP = LEGVAR(B1)                                                012920
            IF (IP .NE. 0) GO TO 20                                        012930
            C1 = Q(1) + B1*P(1)                                            012940
20          DO 90 I=2,NUM                                                  012950
            DX = TEST(I,1)-TEST(I-1,1)                                     012960
15          DY = TEST(I,2)-TEST(I-1,2)                                     012970
            DS = S*SQRT(DX*DX+DY*DY)                                       012980
            B2 = (TEST(I-1,2)-TEST(I,2))/(TEST(I,1)-TEST(I-1,1))           012990
            IT = LEGVAR(B2)                                                013000
            IF (IT .NE. 0) GO TO 35                                        013010
20          C2 = TEST(I,2) + B2*TEST(I,1)                                  013020
            IF (IP .NE. 0) GO TO 40                                        013030
            XX = (C2-C1)/(B2-B1)                                           013040
            IF (LEGVAR(XX) .NE. 0) GO TO 90                                013050
            YY = C1+B1*XX                                                  013060
25          IF ((TEST(I-1,1)-DS .LE.XX) .AND. (XX.LE. TEST(I,1)+DS)) GO TO 30 013070
            IF ((TEST(I,1)-DS .LE.XX) .AND. (XX.LE. TEST(I-1,1)+DS)) GO TO 30 013080
            GO TO 90                                                       013090
30          IF ((P(1)-DS .LE.XX) .AND. (XX.LE. P(2)+DS)) GO TO 45          013100
            IF ((P(2)-DS .LE.XX) .AND. (XX.LE. P(1)+DS)) GO TO 45          013110
            GO TO 90                                                       013120
35          IF (IP .NE. 0) GO TO 90                                        013130
            XX = TEST(I,1)                                                 013140
            YY = C1+B1*XX                                                  013150
            IF ((TEST(I-1,2)-DS .LE.YY) .AND. (YY.LE. TEST(I,2)+DS)) GO TO 30 013160
35          IF ((TEST(I,2)-DS .LE.YY) .AND. (YY.LE. TEST(I-1,2)+DS)) GO TO 30 013170
            GO TO 90                                                       013180
            GO TO 90                                                       013190
40          XX = P(1)                                                      013200
            YY = C2+B2*XX
```

```
40          IF ((Q(1)-DS .LE.YY) .AND. (YY.LE. Q(2)+DS)) GO TO 25          013220
            IF ((Q(2)-DS .LE.YY) .AND. (YY.LE. Q(1)+DS)) GO TO 25          013230
            GO TO 90                                                       013240
45        45 DX = P(1)-XX                                                  013250
            DY = Q(1)-YY                                                   013260
            D = SQRT(DX*DX+DY*DY)                                          013270
            IF (D .LT. DIS) GO TO 90                                       013280
            DIS = D                                                        013290
        153 FORMAT (1H ,F10.3)                                             013300
            X=XX                                                           013310
50          Y=YY                                                           013320
            K = I                                                          013330
        90  CONTINUE                                                       013340
            IF (Y .EQ. 15.0) GO TO 110                                     013350
        152 FORMAT (1H ,I10,1X,2F10.3)                                     013360
55          KFLAG = 1                                                      013370
        151 FORMAT (4H 150,2X,I10)                                         013380
            RETURN                                                         013390
        110 K = NUM + 1                                                    013400
        171 FORMAT (15H ERROR AT ICUT=,I2)                                 013410
60          RETURN                                                         013420
            END
```

E.16 Subroutine SORT

```
1           SUBROUTINE SORT(IC,JC,K,X,Y)                                   013430
            COMMON DATA(103,103,2)/BLOCK1/TEST(500,2),TEST1(500,2),NUM     013440
            COMMON/BLOCK3/INUM, JNUM,XMIN,YMIN,DX,DY,SWI,IT6               013450
            DO 10 I=1,NUM                                                  013460
5           IF (TEST(I,1) .NE. DATA(IC,JC,1)) GO TO 10                     013470
            IF (TEST(I,2) .NE. DATA(IC,JC,2)) GO TO 10                     013480
            J = I                                                          013490
            GO TO 20                                                       013500
        10  CONTINUE                                                       013510
10      431 FORMAT (4H 430,2X,4HK = ,I3,I10)                               013520
            GO TO 99                                                       013530
        20  IF (J .LT. K) GO TO 30                                         013540
            IF (J .EQ. K) GO TO 99                                         013550
            KK = J                                                         013560
15          JJ = K                                                         013570
            GO TO 40                                                       013580
        30  KK = K                                                         013590
            JJ = J + 1                                                     013600
        40  II = 0                                                         013610
20          DO 50 I=KK,NUM                                                 013620
            II = II + 1                                                    013630
            TEST1(II,1) = TEST(I,1)                                        013640
        50  TEST1(II,2) = TEST(I,2)                                        013650
            TEST(JJ,1) = X                                                 013660
25          TEST(JJ,2) = Y                                                 013670
            DO 60 I=1,II                                                   013680
            JJ = JJ + 1                                                    013690
            TEST(JJ,1) = TEST1(I,1)                                        013700
        60  TEST(JJ,2) = TEST1(I,2)                                        013710
30          NUM = JJ                                                       013720
        99  RETURN                                                         013730
            END                                                            013740
```

E.17 Subroutine PARFIT

```
      SUBROUTINE PARFIT(Y1,Y2,Y3,YF1,YF2)                                013750
C THIS SUBROUTINE FITS A PARABOLA TO THREE DATA POINTS Y1,Y2,Y3, TAKEN   013760
C  AT EQUAL INTERVALS AND COMPUTES POINTS YF1,YF2, ON THE FITTED PARABOL 013770
C  , AT THE MIDDLE OF THE INTERVALS                                      013780
C                                                                        013790
      A=(Y1+Y3-2.*Y2)/8.                                                 013800
      B=(Y3-Y1)/4.                                                       013810
      C=Y2                                                               013820
      YF1=A-B+C                                                          013830
      YF2=A+B+C                                                          013840
      RETURN                                                             013850
      END                                                                013860
```

E.18 Subroutine CONTOR

```
      SUBROUTINE CONTOR(M,N,NLEVS,HLEVS,BLANK,XLEN,YLEN,XP,YP)           013870
C                                                                        013880
C     DRAW PENPLOT CONTOURS FROM RECTANGULAR GRID INPUT                  013890
C                                                                        013900
C     PARAMETERS                                                         013910
C                                                                        013920
C     Z             THE GIVEN FUNCTION                                 013930
C     M             NUMBER OF ROWS IN Z ARRAY                          013940
C     N             NUMBER OF COLS IN Z ARRAY                          013950
C     XT            TEMPORARY ARRAY     SIZE=(M*N)                     013960
C     YT            TEMPORARY ARRAY     SIZE=(M*N)                     013970
C     NLEVS         NUMBER OF CONTOURS                                 013980
C     HLEVS         CONTOUR VALUES                                     013990
C     BLANK         MSNG DATA CODE                                     014000
C     XLEN          LENGTH OF X-AXIS IN INCHES                         014010
C     YLEN          LENGTH OF Y-AXIS IN INCHES                         014020
C                                                                        014030
C        Z(I,J) ASSUMED TO BE IN ASSENDING ORDER BY X AND Y              014040
C             IE   Z(1,1) = Z(XMIN,YMIN)                                 014050
C                  Z(M,N) = Z(XMAX,YMAX)                                 014060
C                                                                        014070
C                                                                        014080
C     PROGRAM WILL PLOT ANY NUMBER OF CONTOURS                           014090
C           LINES ARE LABELED BY CHARACTERS 0 - 12                       014100
C                CHARACTERS ARE REPEATED AS NECESSARY                    014110
C                                                                        014120
C     USES SUBROUTINES NEIBOR AND FOUR                                   014130
C                                                                        014140
      DIMENSION HLEVS(1)                                                 014150
      COMMON Z(103,103),XT(103,103),YT(103,103)                          014160
      DIMENSION T(150)                                                   014170
      DIMENSION SX(3),SY(3),IC(4)                                        014180
      DIMENSION IO(16),IP(16)                                            014190
      DATA (IO(I),I=1,16,1)/-1,-1,-2,-1,-1,-2,-1,-1,-1,0,0,0,0,0,0,-1/  014200
      DATA(IP(I),I=1,16,1)/-1,-1,-1,0,0,0,1,0,0,1,0,0,0,-1,-1,-1/       014210
C                                                                        014220
      K=75                                                               014230
      XMIN=YMIN=0E0                                                      014240
      IFLAG=1                                                            014250
      YSYMB=YP-.2                                                        014260
      XPLT=XLEN+XP                                                       014270
      JLINE=8                                                            014280
C                                                                        014290
      CALL PLOT (0.0,0.0,3)                                              014300
      XSIZE=FLOAT(N-1)/XLEN                                              014310
      YSIZE=FLOAT(M-1)/YLEN                                              014320
      UD=M*N                                                             014330
C                                                                        014340
C  FIND ZMAX, ZMIN                                                       014350
```

```
      C                                                                 014360
            IMAX=0                                                      014370
            DO 13 I=1,M                                                 014380
            DO 12 J=1,N                                                 014390
            IF (Z(I,J).EQ.BLANK) GO TO 12                               014400
      55    IMAX=IMAX+1                                                 014410
            IF (IMAX.GT.1) GO TO 11                                     014420
            ZMAX=Z(I,J)                                                 014430
            ZMIN=Z(I,J)                                                 014440
            GO TO 12                                                    014450
      60    11  IF (Z(I,J).GT.ZMAX) ZMAX=Z(I,J)                         014460
            IF (Z(I,J).LT.ZMIN) ZMIN=Z(I,J)                             014470
          12    CONTINUE                                                014480
          13    CONTINUE                                                014490
            PRINT  100, ZMIN,ZMAX,NLEVS,(HLEVS(I),I=1,NLEVS)            014500
      65    C                                                           014510
            C  GET H.                                                   014520
            C                                                           014530
            ILEVS=0                                                     014540
          99    ILEVS=ILEVS+1                                           014550
      70    LLIMIT=ILEVS-1                                              014560
            IF (ILEVS.GT.NLEVS) GO TO 999                               014570
            H=HLEVS(ILEVS)                                              014580
            PRINT  200, H                                               014590
            IDONE=0                                                     014600
      75    IF ((H.GE.ZMIN).AND.(H.LE.ZMAX)) GO TO 101                  014610
            PRINT  300                                                  014620
            GO TO 99                                                    014630
            C                                                           014640
            C  PREPARE THE XT-TABLE                                     014650
            C                                                           014660
      80    101   DO 102 I=1,M                                          014670
            DO 102 J=1,N                                                014680
            XT(I,J)=UD                                                  014690
            IF (I.EQ.M) GO TO 102                                       014700
      85    IF (Z(I  ,J).EQ.BLANK) GO TO 102                            014710
            IF (Z(I+1,J).EQ.BLANK) GO TO 102                            014720
            IF (((Z(I,J)-H)*(Z(I+1,J)-H)).GE.0.0) GO TO 102             014730
            XT(I,J)=-ABS(FLOAT(I-1)+((H-Z(I,J))/(Z(I+1,J  )-Z(I,J))))   014740
          102   CONTINUE                                                014750
      90    C                                                           014760
            C  PREPARE THE YT-TABLE                                     014770
            C                                                           014780
            DO 103 I=1,M                                                014790
            DO 103 J=1,N                                                014800
      95    YT(I,J)=UD                                                  014810
            IF (J.EQ.N) GO TO 103                                       014820
            IF (Z(I,J  ).EQ.BLANK) GO TO 103                            014830
            IF (Z(I,J+1).EQ.BLANK) GO TO 103                            014840
            IF (((Z(I,J)-H)*(Z(I,J+1)-H)).GE.0.0) GO TO 103             014850
     100    YT(I,J)=-ABS(FLOAT(J-1)+((H-Z(I,J))/(Z(I  ,J+1)-Z(I,J))))   014860
          103   CONTINUE                                                014870
            C                                                           014880
            DO 201 I1=1,4                                               014890
          201   IC(I1)=0                                                014900
     105    DO 207 I=1,M                                                014910
            DO 207 J=1,N                                                014920
            IF (Z(I,J).NE.H) GO TO 207                                  014930
            C                                                           014940
            C     COUNT ENTRANCES AND EXITS IN SURROUNDING BLOCKS       014950
     110    C                                                           014960
            DO 202 I2=1,16                                              014970
            II=IO(I2)+I                                                 014980
            IF (II.LT.1.OR.II.GT.M) GO TO 202                           014990
            JJ=JP(I2)+J                                                 015000
     115    IF (JJ.LT.1.OR.JJ.GT.N) GO TO 202                           015010
            I1=(I2-1)/4+1                                               015020
            IF((XT(II,JJ).NE.UD) .OR. (YT(II,JJ).NE.UD))IC(I1)=IC(I1)+1 015030
          202   CONTINUE                                                015040
            IHOLD=0                                                     015050
     120    DO 206 I2=1,4                                               015060
            IC(I2)=MOD(IC(I2),2)                                        015070
            IF (IC(I2).EQ.0) GO TO 206                                  015080
            IF (IHOLD.NE.0) GO TO 203                                   015090
            IHOLD=I2                                                    015100
     125    GO TO 206                                                   015110
          203   IF (I2-IHOLD.EQ.1) GO TO 204                            015120
            IF (IHOLD.EQ.1.AND.I2.EQ.4) GO TO 205                       015130
            GO TO 30)                                                   015140
          204   IF (I2+IHOLD.EQ.5) GO TO 205                            015150
```

```
130               I3=I+I2-3                                                    015160
                  XT(I3,J)=-ABS(FLOAT(I3-1)+.001)                              015170
                  GO TO 207                                                    015180
              206 I3=I-1                                                       015190
                  J4=J+4-I2                                                    015200
135               YT(I3,J4)=-ABS(FLOAT(J4)+.001)                               015210
                  GO TO 207                                                    015220
              206 CONTINUE                                                     015230
              207 CONTINUE                                                     015240
                  C                                                            015250
140           301 IL=-30                                                       015260
                  JL=-30                                                       015270
                  KL=-30                                                       015280
                  XF=UD                                                        015290
                  YF=UD                                                        015300
145               C                                                            015310
                  C     COMPILE A LIST                                         015320
                  C                                                            015330
                  302 CONTINUE                                                 015340
                      DO 303 I=1,M                                             015350
150                   DO 303 J=1,N                                           015360
                      KEY=1                                                    015370
                      IF (XT(I,J).NE.UD) GO TO 304                             015380
                      KEY=2                                                    015390
                      IF (YT(I,J).NE.UD) GO TO 304                             015400
155               303 CONTINUE                                                 015410
                  C                                                            015420
                      IF (IDONE.EQ.0) PRINT  400                               015430
                      GO TO 99                                                 015440
                  C                                                            015450
160               304 IX=0                                                     015460
                      IF (NEIBOR(KEY,I,J,KL,IL,JL).LE.0) GO TO 305             015470
                      IX=2                                                     015480
                      T(1)=XF                                                  015490
                      T(2)=YF                                                  015500
165               305 IL=I                                                     015510
                      JL=J                                                     015520
                      KL=KEY                                                   015530
                      LB=1                                                     015540
                      GO TO (306,307), KEY                                     015550
170               306 XF=XT(I,J)                                               015560
                      YF=I-1                                                   015570
                      GO TO 401                                                015580
                  307 XF=I-1                                                   015590
                      YF=YT(I,J)                                               015600
175               C                                                            015610
                  C  ADD A POINT TO THE LIST                                   015620
                  C                                                            015630
                  401 XS=XF                                                    015640
                      YS=YF                                                    015650
180                   ISAVE=I                                                015660
                      JSAVE=J                                                  015670
                      KSAVE=KEY                                                015680
                  402 IF ((IX-K+2).LT.0) GO TO 404                             015690
                      I47=1                                                    015700
185                   GO TO 801                                              015710
                  403 CONTINUE                                                 015720
                      T(1)=T(IX-1)                                             015730
                      T(2)=T(IX)                                               015740
                      IX=2                                                     015750
190               404 IX=IX+2                                                  015760
                      T(IX-1)=ABS(XS)                                          015770
                      T(IX)=ABS(YS)                                            015780
                      IF (I47.EQ.1) GO TO 405                                  015790
                      IF (IX.LT.6) GO TO 406                                   015800
195               405 IF (KSAVE.EQ.1) XT(ISAVE,JSAVE)=UD                       015810
                      IF (KSAVE.EQ.2) YT(ISAVE,JSAVE)=UD                       015820
                  406 IF (IX.EQ.2) GO TO 501                                   015830
                      GO TO (407,409), KEY                                     015840
                  C                                                            015850
200               C  THE POINT IS ON AN HORIZONTAL LINK                        015860
                  C                                                            015870
                  407 IF (XT(I,J).GE.0.0) GO TO 408                            015880
                      XT(I,J)=ABS(XT(I,J))                                     015890
                      GO TO 501                                                015900
205               409 XT(I,J)=UD                                               015910
                      GO TO 501                                                015920
                  C                                                            015930
                  C     THE POINT IS ON A VERTICAL LINK                        015940
                  C                                                            015950
```

```
      409   IF (YT(I,J).GE.0.0) GO TO 410                                  0015960
            YT(I,J)=ABS(YT(I,J))                                           0015970
            GO TO 501                                                      0015980
      410   YT(I,J)=UD                                                     0015990
            C                                                              0016000
            C     INTO BOX - IS THERE A WAY OUT                            0016010
            C                                                              0016020
      501   KH=0                                                           0016030
            ISAVE=I                                                        0016040
            JSAVE=J                                                        0016050
            KSAVE=KEY                                                      0016060
            GO TO (502,601), KEY                                           0016070
      502   CONTINUE                                                       0016080
            IF (LB.EQ.2) GO TO 505                                         0016090
            IF (J.EQ.N) GO TO 701                                          0016100
            IF (XT(I,J+1).EQ.UD) GO TO 503                                 0016110
            KH=KH+1                                                        0016120
            KS=1                                                           0016130
            SX(1)=XT(I,J+1)                                                0016140
            SY(1)=J                                                        0016150
      503   CONTINUE                                                       0016160
            IF (YT(I,J).EQ.UD) GO TO 504                                   0016170
            KH=KH+1                                                        0016180
            KS=2                                                           0016190
            SX(2)=I-1                                                      0016200
            SY(2)=YT(I,J)                                                  0016210
      504   CONTINUE                                                       0016220
            IF (I.EQ.M) GO TO 508                                          0016230
            IF (YT(I+1,J).EQ.UD) GO TO 508                                 0016240
            KH=KH+1                                                        0016250
            KS=3                                                           0016260
            SX(3)=I                                                        0016270
            SY(3)=YT(I+1,J)                                                0016280
            GO TO 508                                                      0016290
      505   CONTINUE                                                       0016300
            IF (J.EQ.1) GO TO 701                                          0016310
            IF (XT(I,J-1).EQ.UD) GO TO 506                                 0016320
            KH=KH+1                                                        0016330
            KS=4                                                           0016340
            SX(4)=XT(I,J-1)                                                0016350
            SY(4)=J-2                                                      0016360
      506   CONTINUE                                                       0016370
            IF (YT(I,J-1).EQ.UD) GO TO 507                                 0016380
            KH=KH+1                                                        0016390
            KS=5                                                           0016400
            SX(5)=I-1                                                      0016410
            SY(5)=YT(I,J-1)                                                0016420
      507   CONTINUE                                                       0016430
            IF (I.EQ.M) GO TO 508                                          0016440
            IF (YT(I+1,J-1).EQ.UD) GO TO 508                               0016450
            KH=KH+1                                                        0016460
            KS=6                                                           0016470
            SX(6)=I                                                        0016480
            SY(6)=YT(I+1,J-1)                                              0016490
      508   IF (KH.EQ.0.OR.KH.EQ.2) GO TO 701                              0016500
            IF (KH.NE.1) CALL FOUR (I,IX,SX,SY,KS,KEY)                     0016510
            GO TO (509,510,511,512,513,514), KS                            0016520
      509   KEY=1                                                          0016530
            LB=1                                                           0016540
            J=J+1                                                          0016550
            GO TO 516                                                      0016560
      510   KEY=2                                                          0016570
            LB=2                                                           0016580
            GO TO 516                                                      0016590
      511   KEY=2                                                          0016600
            I=I+1                                                          0016610
            LB=1                                                           0016620
            GO TO 516                                                      0016630
      512   KEY=1                                                          0016640
            LB=2                                                           0016650
            GO TO 515                                                      0016660
      513   KEY=2                                                          0016670
            LB=2                                                           0016680
            GO TO 515                                                      0016690
      514   KEY=2                                                          0016700
            I=I+1                                                          0016710
            LB=1                                                           0016720
      515   KS=KS-3                                                        0016730
            J=J-1                                                          0016740
      516   XS=SX(KS)                                                      0016750
```

```
      YS=SY(KS)                                                         016760
      IFLAG=2                                                           016770
      GO TO 402                                                         016780
C                                                                       016790
C                                                                       016800
C                                                                       016810
  601 CONTINUE                                                          016820
      IF (LR.EQ.2) GO TO 604                                            016830
      IF (I.EQ.M) GO TO 701                                             016840
      IF (XT(I,J).EQ.UD) GO TO 602                                      016850
      KH=KH+1                                                           016860
      KS=1                                                              016870
      SX(1)=XT(I,J)                                                     016880
      SY(1)=J-1                                                         016890
  602 CONTINUE                                                          016900
      IF (J.EQ.N) GO TO 603                                             016910
      IF (XT(I,J+1).EQ.UD) GO TO 604                                    016920
      KH=KH+1                                                           016930
      KS=2                                                              016940
      SX(2)=XT(I,J+1)                                                   016950
      SY(2)=J                                                           016960
  603 CONTINUE                                                          016970
      IF (I.EQ.M) GO TO 607                                             016980
      IF (YT(I+1,J).EQ.UD) GO TO 607                                    016990
      KH=KH+1                                                           017000
      KS=3                                                              017010
      SX(3)=I                                                           017020
      SY(3)=YT(I+1,J)                                                   017030
      GO TO 607                                                         017040
  604 CONTINUE                                                          017050
      IF (I.EQ.1) GO TO 731                                             017060
      IF (XT(I-1,J).EQ.UD) GO TO 605                                    017070
      KH=KH+1                                                           017080
      KS=4                                                              017090
      SX(4)=XT(I-1,J)                                                   017100
      SY(4)=J-1                                                         017110
  605 CONTINUE                                                          017120
      IF (J.EQ.N) GO TO 606                                             017130
      IF (XT(I-1,J+1).EQ.UD) GO TO 606                                  017140
      KH=KH+1                                                           017150
      KS=5                                                              017160
      SX(5)=XT(I-1,J+1)                                                 017170
      SY(5)=J                                                           017180
  606 CONTINUE                                                          017190
      IF (YT(I-1,J).EQ.UD) GO TO 607                                    017200
      KH=KH+1                                                           017210
      KS=6                                                              017220
      SX(6)=I-1                                                         017230
      SY(6)=YT(I-1,J)                                                   017240
  607 CONTINUE                                                          017250
      IF (KH.EQ.0.OR.KH.EQ.2) GO TO 701                                 017260
      IF (KH.NE.1) CALL ERRP (7,1X,SX,SY,KS,KEY)                        017270
      GO TO (608,609,610,611,612,613), KS                               017280
  608 KEY=1                                                             017290
      LR=2                                                              017300
      GO TO 615                                                         017310
  609 KEY=1                                                             017320
      LR=1                                                              017330
      I=J+1                                                             017340
      GO TO 615                                                         017350
  610 KEY=2                                                             017360
      I=I+1                                                             017370
      LR=1                                                              017380
      GO TO 615                                                         017390
  611 KEY=1                                                             017400
      LR=2                                                              017410
      GO TO 614                                                         017420
  612 KEY=1                                                             017430
      LR=1                                                              017440
      I=J+1                                                             017450
      GO TO 614                                                         017460
  613 KEY=2                                                             017470
      LR=2                                                              017480
  614 I=I-1                                                             017490
      KS=KS-3                                                           017500
  615 XS=SX(KS)                                                         017510
      YS=SY(KS)                                                         017520
      IFLAG=2                                                           017530
      GO TO 402                                                         017540
C                                                                       017550
C  NO MORE POINTS
```

```
      C                                                                 017560
  701 IF (NEIBOR(KEY,I,J,KL,IL,JL).LE.0) GO TO 704                      017570
      IF (IX.LT.6) GO TO 704                                            017580
      IF (T(1).EQ.T(IX-1).AND.T(2).EQ.T(IX)) GO TO 704                  017590
      IX=IX+2                                                           017600
      T(IX-1)=ABS(XF)                                                   017610
      T(IX)=ABS(YF)                                                     017620
      C                                                                 017630
      GO TO (702,703), KEY                                              017640
  702 XT(I,J)=UD                                                        017650
      GO TO 704                                                         017660
  703 YT(I,J)=UD                                                        017670
      GO TO (702,704), IFLAG                                            017680
  704 I47=2                                                             017690
      C                                                                 017700
  801 LLINE=MOD(LLINE,13)                                               017710
      IF (IX.NE.2) GO TO 901                                            017720
      IF (LB.EQ.2) GO TO 902                                            017730
      LB=2                                                              017740
      GO TO 501                                                         017750
  802 IF(KEY .EQ. 1) XT(I,J)=UD                                         017760
      IF(KEY .EQ. 2) YT(I,J)=UD                                         017770
      GO TO 902                                                         017780
      C                                                                 017790
      C                                                                 017800
  901 IDONE=IDONE+1                                                     017810
      CALL LINE (T(2),T(1),IX/2+2,LLINE,LLINE,XMIN,XSIZE,YMIN,YSIZE,.06) 017820
      IF (IDONE.GT.1) GO TO 902                                         017830
      YSYMB=YSYMB-0.2                                                   017840
      IF (YSYMB .LT. 0.) 110,120                                        017850
  110 YSYMB=YB-.2                                                       017860
      XPLT=XPLT+2.                                                      017870
  120 CALL SYMBOL (XPLT ,YSYMB,0.09,LLINE,0.0,-1)                       017880
      CALL SYMBOL(3HOLD,3HOLD,.09,1H=,0.,1)                             017890
      ENCODE(10,1,ISYM) H                                               017900
      CALL SYMBOL(3HOLD,3HOLD,.09,ISYM,0.0,10)                          017910
      PRINT 500, LLINE                                                  017920
  902 IF(I47 .EQ. 1) GO TO 403                                          017930
      IFLAG=1                                                           017940
      I=II                                                              017950
      J=JI                                                              017960
      KEY=KL                                                            017970
      GO TO 301                                                         017980
      C                                                                 017990
    1 FORMAT(G10.3)                                                     018000
  100 FORMAT (*1BEGIN CONTOUR PLOT*,2G13.5,I5/(5X,10G13.5))             018010
  200 FORMAT (* CONTOUR*,G13.5)                                         018020
  300 FORMAT (10X,*OUTSIDE GRID*)                                       018030
  400 FORMAT (10X,*NO DATA*)                                            018040
  500 FORMAT (10X,*PLOTTED CHARACTER*,I3)                               018050
      C                                                                 018060
  999 RETURN                                                            018070
      END                                                               018080
```

E.19 Subroutine NEIBOR

```
      FUNCTION NEIBOR (KA,IA,JA,KB,IB,JB)                              018090
C                                                                      018100
C                                                                      018110
C     NEIBOR=-1  IF NOT NEIGHBORS                                      018120
C     NEIBOR = 1  IF NEIGHBORS                                         018130
C                                                                      018140
C                                                                      018150
      DIMENSION K1(14),K2(14),IT(14),JT(14)                            018160
      DATA K1/1,1,1,1,1,2,2,2,2,2,2,1,2/                               018170
      DATA K2/1,1,2,2,2,2,1,1,1,1,2,2,1,2/                             018180
      DATA IT/0,0,0,0,1,1,0,0,-1,-1, 1,-1,0,0/                         018190
      DATA JT/-1,1,0,-1,-1,0,1,0,1,1,0,0,0,0/                          018200
      ID = IB-IA                                                       018210
      JD = JB-JA                                                       018220
      DO 80 I=1,14                                                     018230
      IF (KA-K1(I)) 80,50,80                                           018240
   50 IF (KB-K2(I)) 80,60,80                                           018250
   60 IF (ID-IT(I)) 80,70,80                                           018260
   70 IF (JD-JT(I)) 80,90,80                                           018270
   80 CONTINUE                                                         018280
      NEIBOR = -1                                                      018290
      GO TO 100                                                        018300
   90 NEIBOR = 1                                                       018310
      RETURN                                                           018320
  100 CONTINUE                                                         018330
      RETURN                                                           018340
      END                                                              018350
```

E.20 Subroutine FOUR

```
      SUBROUTINE FOUR(T,IX,SX,SY,KS,KEY)                               018360
      DIMENSION T(150),SX(3),SY(3)                                     018370
      IF(IX.GE.4) GO TO 498                                            018380
      KS=2                                                             018390
      RETURN                                                           018400
  498 SLOPT=(T(IX)-T(IX-2))/(T(IX-1)-T(IX-3))                          018410
      DO 515 I1=1,3                                                    018420
      XS=ABS(SX(I1))                                                   018430
      YS=ABS(SY(I1))                                                   018440
      SLOPE=(T(IX)-YS)/(T(IX-1)-XS)                                    018450
      IF(SLOPT*SLOPE .LT. 0.0) GO TO 515                               018460
      IF(KEY.EQ.1 .AND. I1.EQ.1) GO TO 515                             018470
      IF(KEY.EQ.2 .AND. I1.EQ.3) GO TO 515                             018480
      IF(KS.LE.1) GO TO 1                                              018490
      KS=I1+3                                                          018500
      RETURN                                                           018510
    1 KS=I1                                                            018520
      RETURN                                                           018530
  515 CONTINUE                                                         018540
      RETURN                                                           018550
      END                                                              018560
```

E.21 Function SYMBL

```
      FUNCTION SYMBL(TIM)                                              018750
      IF (MYGETX(TIM+1.1) .EQ. 5581) GOTO 100                          018760
      ENCODE(10,1,SYMBL) TIM                                           018770
      RETURN                                                           018780
  100 SYMBL=TIM                                                        018790
      RETURN                                                           018800
    1 FORMAT(G10.3)                                                    018810
      END                                                              018820
```

Appendix F

Fortran Listing for Program T3D

```
      PROGRAM T30(TAPE3,OUTPUT)
C REVISION---SEPT 10,1975
C PURPOSE---DISPLAY VALUES OF F1 AND F2 FROM TAPE3 (TEMP5 OUTPUT)
C     USING LINEAR INTERPOLATION BETWEEN TIMES.
C SUBROUTINE---
C     RTAPE3
      INTEGER DATAIN(100,3)
      EQUIVALENCE(I1,DATAIN)
      DATA IT3/3/
      REWIND IT3
      READ(IT3) DATAIN
      TLAST=TNEXT=0.
      IP=1
      DELT=.1*TAUMX
      WRITE 1,NO,NMX,DTAUO,TAUMX
      DO 100 I=1,10
      T=DELT*I
      CALL RTAPE3(T)
      WRITE 2,T
      WRITE 3,(F1(J),J=1,81,10)
      WRITE 3,(F2(J),J=1,81,10)
  100 CONTINUE
    1 FORMAT(1H1,* NO=*I3*, NMX=*I3*, DTAUO=*E13.6* TAUMX=*E13.6/
     +* VALUES OF F1 AND F2 AT 1,11,..,81*)
    2 FORMAT(/* TAU=*E13.6)
    3 FORMAT(/(4E13.6))
      END
```

MISSION
of
Rome Air Development Center

RADC plans and conducts research, exploratory and advanced development programs in command, control, and communications (C^3) activities, and in the C^3 areas of information sciences and intelligence. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

Printed by
United States Air Force
Hanscom AFB, Mass. 01731