22-640

11-27
1-2 077
LA-6382

UC-32

Issued: November 1976

**A Linear Iterative Technique
for Solution of
Nonlinear Thermal Network Problems**

by

C. M. Seabourn

**los alamos
scientific laboratory**

of the University of California

LOS ALAMOS, NEW MEXICO 87545

An Affirmative Action/Equal Opportunity Employer

UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7408-ENG. 36

MASTER

This work was supported by the US Energy Research and Development Administration, Division of Nuclear Research and Applications.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy \$4.00 Microfiche \$3.00

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

UNIVERSITY OF CALIFORNIA
LOS ALAMOS SCIENTIFIC LABORATORY
(CONTRACT W-7405-ENG-36)
P. O. Box 1663
LOS ALAMOS, NEW MEXICO 87544

IN REPLY
REFER TO
MAIL STOP

ISD-6
184

December 13, 1976

To: Holders of LA-6382

Page 26, Section I. GAUSSJ, should be replaced with the following.

```
05330 SUBROUTINE GAUSSJ(ARR,BRR,N)
05340C
05350C      GENERAL GAUSS-JORDAN MATRIX SOLVER
05360C
05370 DIMENSION ARR(N,N),BRR(N)
05380C
05390 DO 60 J=1,N
05400 DIV=ARR(J,J)
05410 DO 30 I=J,N
05420 ARR(J,I)=ARR(J,I)/DIV
05430 30 CONTINUE
05440 BRR(J)=BRR(J)/DIV
05450 DO 50 I=1,N
05460 AMUL=ARR(I,I)
05470 IF(I.EQ.J)GO TO 50
05480 DO 40 K=J,N
05490 ARR(I,K)=ARR(I,K)-AMUL*ARR(J,K)
05500 40 CONTINUE
05510 BRR(I)=BRR(I)-AMUL*BRR(J)
05520 50 CONTINUE
05530 60 CONTINUE
05540 RETURN
05550 END
```

MASTER

CONTENTS

ABSTRACT	1
I. INTRODUCTION	1
II. ASSUMPTIONS	2
A. Heat Transfer Equations	2
B. Thermal Conductivity	2
C. Emissivity	2
III. HEAT BALANCE EQUATIONS	2
A. Linear Systems	2
B. Nonlinear Systems	3
IV. THE REFERENCE NODE(S)	5
A. One Reference Node	5
B. More than One Reference Node	5
V. THE MATRIX SOLVERS	6
VI. SOLUTION METHOD	6
VII. PRINCIPLES FOR FACILITATING SOLUTION	7
VIII. ONE-DIMENSIONAL EXAMPLE PROBLEM	7
A. Fuel Sphere Assembly	7
B. Design Degradation Study	8
IX. THREE-DIMENSIONAL EXAMPLE PROBLEM	8
X. SUMMARY	11
REFERENCES	11
APPENDIX A. KRONOS, FORTRAN Listing of Thermal Conductivity Function Subroutine	25
APPENDIX B. KRONOS, FORTRAN Listings of Gauss-Jordan Matrix Solvers	26
APPENDIX C. Program Listing of Fuel Sphere Assembly Example Problem	27
APPENDIX D. Program Listing of Plutonia Aerodynamic Disk Example Problem	31

MASTER

GLOSSARY OF TERMS

A	Area (cm²)
A (i,j)	i-th row and j-th column term of A matrix
ANODE	Reference node A matrix diagonal modification term
AL	Length of fuel center in PAD (cm)
ALD	Length-to-diameter ratio of fuel cylinder in PAD
B(i)	i-th term of B columnar matrix
C₁, C₂, C₃	Coefficients of ΔT in Eqs. (1a), (2a), and (3a)
C1	Clad thickness in FSA (Fig. 1) (cm)
CK(i,j,k)	Thermal conductivity (W/cm-°K) of material i and between nodes j and k. (Also function subroutine, Appendix A)
CREF	Reference temperature (°C)
D	Diameter (cm)
DFSA	Diameter of FSA (cm)
DFUEL	Diameter of PPO fuel (cm)
DJFG, etc.	Distance from center (J) of fuel in PAD model to point FG (Fig. 6) (cm)
DPAD	Diameter of PAD (cm)
E	Emissivity and geometric term in Eq. (4a)
EDGE	Distance from fuel cavity to edge of PAD (Fig. 3) (cm)
FACE	Graphite thickness from fuel cavity to PAD face (cm)
FSA	Fuel sphere assembly (Fig. 1)
Fuel	PPO, PuO₂
F1	Emissivity of PPO
F2	Emissivity of PICS: first value for inside surface, second value for outside surface
F3	Emissivity of graphite
FSASPHR	Thermal model of FSA (Appendix C)

GAUSSJ	Gauss-Jordan elimination method subroutine (Appendix B)
g_1, g_2, g_3	Geometric terms of Eqs. (1a), (2a), (3a)
G1	Gap between fuel and PICS of FSA (Fig. 1)
G2	Gap between PICS and GIS of FSA (Fig. 1)
GIS	Graphite impact shell
INODE	Beginning reference node in PAD model (166)
ITIMES	Counter for successive intermediate solutions
JNODE	Final reference node in PAD model (190)
k	Thermal conductivity (W/cm-°K)
k_{ij}	Thermal conductivity between nodes i and j (W/cm-°K)
L	Length of cylinder (Eq. 2) (cm)
LSYSL	Modified, general Gauss-Jordan elimination method matrix solver (Appendix B)
LSYSL1	Modified Gauss-Jordan elimination method specific for tridiagonal matrices (Appendix B)
M	Number of fuel cylinders (input valve) in PAD
N	Total number of thermal nodes in a model
N1	Maximum number of iterations permitted for model solution
N2	Iteration counter
P	Power (watts thermal)
PAD	Plutonia aerodynamic disk (Fig. 2)
PADTHRM	Thermal model of PAD (Appendix D)
PDENSTY	Power density (W/cm ³)
PICS	Post impact containment shell of FSA (Fig. 1)
PPO	Pressed plutonium oxide
PW	Total power (W_t) including all fuel elements in PAD
q, Q	Heat output (W_t)
r	Radius (cm)

MASTER

R	Half of separation of two nodes in FSA model fuel (Fig. 1) (cm)
RCAV	Radius of graphite cavity in PAD (Fig. 3) (cm)
RFUEL	Radius of fuel (cm)
RPAD	Radius of PAD (Fig. 3) (cm)
S1	Thickness of FSA GIS (Fig. 1) (cm)
SEPAR	Separation of cavities in PAD graphite (Fig. 3) (cm)
T	Temperature (°K)
T_i	Temperature of node i(°K)
TLIMIT	Temporary precision limit for matrix solution
TREF	Reference temperature (°K)
W_t	Watts thermal
x	Eq. (1) thickness of cross section Δ (cm)
XTLIMIT	Final precision limit required for ultimate matrix solution
YCAV	Distance from center of PAD to fuel cavity (Fig. 3) (cm)
YFUEL	Distance from center of PAD to center of fuel cylinder (Fig. 3) (cm)
ΔT	Differential temperature (°K)
ϵ	Eq. (4a) emissivity constant
Ψ, PSI	Angle subtended by adiabatic planes in PAD model (Fig. 3) (radians)
σ, SIGMA	Stefan-Boltzmann constant, $5.67032 \times 10^{-12} \text{ W/cm}^2 \cdot \text{°K}^4$

A LINEAR ITERATIVE TECHNIQUE FOR SOLUTION OF NONLINEAR THERMAL NETWORK PROBLEMS

by

C. M. Seabourn

ABSTRACT

A method for rapid and accurate solution of linear and/or nonlinear thermal network problems is described. It is a matrix iterative process that converges for nodal temperatures and variations of thermal conductivity with temperature. The method is computer oriented and can be changed easily for design studies.

I. INTRODUCTION

Radioisotopic heat sources are becoming widely used in electrical power sources for spacecraft and ground instrumentation stations, artificial hearts, and a number of other applications. Until now, there has been no quick and simple mathematical method for thermal evaluation of designs, performance, and life degradation of these heat sources. Most thermal computations involved complex, time-consuming computer programs or gross assumptions and approximations to avoid nonlinear equations that were performed by hand. There was clearly a need for a computational method that was accurate, fast, and easily modified to study design perturbations before expensive hardware fabrication.

Radioisotopic heat sources obtain their heat from the radioactive decay of an isotope. Primarily for safety reasons, the isotope is encapsulated in one or more metal containers to prevent its dispersion. Because there are gaps in the assembly, a great deal of the heat generated is transferred out of the capsule by thermal radiation. In computations, this leads to nonlinear heat balance equations (Eq. 4) that are difficult to solve. A method for rapid,

economical solution of such equations is needed for evaluation of designs and performance.

Dusinberre¹ has developed several methods for computing thermal network problems. His methods are good for small, linear systems, but they do not lend themselves easily to computer techniques. When large networks and nonlinear systems are involved, the finite difference methods become cumbersome and time-consuming.

A computer-oriented method for solving linear and/or nonlinear systems has been developed and is offered here. The method is essentially a description of the network in terms of matrices and their iteration through a matrix solver until the desired precision is attained. The method converges simultaneously on the answer temperatures and temperature-variable thermal conductivities within the system. It is fast and lends itself to easy change for design studies.

This method of using linear techniques to solve sets of nonlinear simultaneous equations may be applicable to other types of problems that involve nonlinear, simultaneous equations.

Time dependent functions can be introduced into the matrices for study of dynamic systems.

II. ASSUMPTIONS

A. Heat Transfer Equations

The following equations are assumed to be valid for heat transfer, given small temperature differences and small dimensions.* The imposed conditions are such that thermal conductivity can be assumed constant without affecting the computational accuracy.

1. Solid of Constant Cross Section.

$$q = \frac{k A \Delta T}{x} \quad (1)$$

$$q = C_1 \Delta T = g_1 k \Delta T. \quad (1a)$$

2. Cross Section of Path Proportional to Radius (Long Hollow Cylinders).

$$q = \frac{k 2\pi L \Delta T}{\ln(r_2/r_1)} \quad (2)$$

$$q = C_2 \Delta T = g_2 k \Delta T. \quad (2a)$$

3. Cross Section of Path Proportional to Square of Radius (Hollow Spheres).

$$q = \frac{4\pi k \Delta T}{(1/r_1 - 1/r_2)} \quad (3)$$

$$q = C_3 \Delta T = g_3 k \Delta T. \quad (3a)$$

4. Thermal Radiation.

$$q = \frac{\sigma A (T_1^4 - T_2^4)}{(1/\epsilon_1 + 1/\epsilon_2 - 1.0)} \quad (4)$$

$$q = E (T_1^4 - T_2^4). \quad (4a)$$

B. Thermal Conductivity

For this solution technique, thermal conductivity is assumed to be a function of temperature. The purpose is to demonstrate that the solution technique converges on both the nodal temperatures and thermal conductivities (k) between respective nodes. Depending on the materials and ΔT 's involved, k may also be considered constant.

A special function subroutine (App. A) was devised for this purpose. The function $CK(i,j,k)$, where i designates the material and j and k are the two thermal nodes, computes the thermal conductivity (k) for the heat balance equations and matrix coefficients.

In devising this subroutine, polynomials in T , straight line functions of temperature, or segments of straight lines of $k = f(T_k)$, and the arithmetic average of T_j and T_k were used. This subroutine is not restrictive; any data set for conductivity (table interpolation, polynomials, constants, etc.) may be used.

C. Emissivity

For the method and problems shown here, thermal emissivity is considered constant. However, emissivity may also be considered as variable with temperature. If emissivity is to be used as a variable, a function subroutine like the one described for conductivity should be written and incorporated into the computer program for the model. (Convective heat transfer coefficients can be treated similarly).

III. HEAT BALANCE EQUATIONS

A. Linear Systems

Assume a homogeneous system where Eqs. (1)-(3) apply and where the C 's in the following equations can represent C_1 , C_2 , or C_3 .

Take the following network of thermal nodes.

The heat balance equation for node i is*

$$C_{ij}(T_j - T_i) + C_{ik}(T_k - T_i) + C_{il}(T_l - T_i) + q_i = 0, \quad (5)$$

*If no heat is generated by the nodal volume, q_i will be zero.

or

$$C_{ij}T_j + C_{ik}T_k + C_{il}T_l - \sum_{b=j,k,l} \epsilon_{b,j,k,l} C_{ib}T_i = -q_i \quad (6)$$

Similarly for node j (note: $c_{ij} = c_{ji}$),

$$C_{ij}T_i + C_{jm}T_m + C_{jn}T_n - \sum_{b=i,m,n} \epsilon_{b,i,m,n} C_{jb}T_j = -q_j \quad (7)$$

The corresponding A and B matrix coefficients are

$$A(i,j) = C_{ij} = \epsilon_{ij} k_{ij} \quad (8)$$

$$A(i,k) = C_{ik} = \epsilon_{ik} k_{ik} \quad (9)$$

$$A(i,l) = C_{il} = \epsilon_{il} k_{il} \quad (10)$$

$$A(i,i) = -\sum_{b=j,k,l} \epsilon_{b,j,k,l} C_{ib} = -\sum_{b=j,k,l} \epsilon_{b,j,k,l} \epsilon_{ib} k_{ib} \quad (11)$$

$$A(j,i) = C_{ji} = \epsilon_{ji} k_{ji} \quad (12)$$

$$A(j,m) = C_{jm} = \epsilon_{jm} k_{jm} \quad (13)$$

$$A(j,n) = C_{jn} = \epsilon_{jn} k_{jn} \quad (14)$$

$$A(j,j) = -\sum_{b=i,m,n} \epsilon_{b,i,m,n} C_{jb} = -\sum_{b=i,m,n} \epsilon_{b,i,m,n} \epsilon_{jb} k_{jb} \quad (15)$$

$$B(i) = -q_i \quad (16)$$

$$B(j) = -q_j \quad (17)$$

The above show that for a matrix A made up of linear terms, the matrix is symmetrical about the diagonal, i.e., $A(i,j) = A(j,i)$; the diagonal term is the negative sum of all the linear terms in that row; and the B columnar matrix is composed of constants on the right side of the equality signs (it can be zero).

Reference node equation modifications are required to solve this network, and they will be discussed after the following nonlinear example.

B. Nonlinear Systems

Assume a nodal network as above, but with nodes i and j separated by a gap filled with a conductive gas (or a nonconductive vacuum, $C_{ij} = 0.0$).

The heat balance equations for node i become*

$$C_{ik}(T_k - T_i) + C_{il}(T_l - T_i) + \epsilon_{ij}(T_j^4 - T_i^4) + C_{ij}(T_j - T_i) + q_i = 0 \quad (18)$$

or

$$C_{ik}T_k + C_{il}T_l + \epsilon_{ij}T_j^4 + C_{ij}T_j - \sum_{b=j,k,l} \epsilon_{b,j,k,l} C_{ib}T_i - \epsilon_{ij}T_i^4 = -q_i \quad (19)$$

Rewriting changes this to

$$C_{ik}T_k + C_{il}T_l + (\epsilon_{ij}T_j^3 + C_{ij})T_j - (\epsilon_{ij}T_i^3 + \sum_{b=j,k,l} \epsilon_{b,j,k,l} C_{ib})T_i = -q_i \quad (20)$$

Similarly, for node j,

$$C_{ij}(T_i - T_j) + \epsilon_{ij}(T_i^4 - T_j^4) + C_{jn}(T_n - T_j) + C_{jo}(T_o - T_j) + C_{jp}(T_p - T_j) = -q_j \quad (21)$$

or

$$(C_{ij} + \epsilon_{ij}T_i^3)T_i - (\epsilon_{ij}T_j^3 + C_{ij}T_j) + \sum_{b=n,o,p} \epsilon_{b,i,m,n} C_{jb}T_j + C_{jm}T_m + C_{jn}T_n = -q_j \quad (22)$$

The corresponding matrix coefficients become

$$A(i,j) = \epsilon_{ij}T_j^3 + C_{ij} \quad (23)$$

*Emissivities can also be treated as functions of temperature.

$$A(i,k) = C_{ik}, \quad (24)$$

$$A(i,l) = C_{il}, \quad (25)$$

$$A(j,i) = E_{ij}T_i^3 + C_{ij}, \quad (26)$$

$$A(j,m) = C_{jm}, \quad (27)$$

$$A(j,n) = C_{jn}, \quad (28)$$

$$A(i,i) = -\sum_{b=k,l} C_{ib} - (E_{ij}T_i^3 + C_{ij}), \quad (29)$$

or

$$A(i,i) = -\sum_{b=k,l} C_{ib} - A(j,i), \quad (30)$$

$$A(j,j) = -\sum_{b=m,n} C_{jb} - (E_{ij}T_j^3 + C_{ij}), \quad (31)$$

or

$$A(j,j) = -\sum_{b=m,n} C_{jb} - A(i,j), \quad (32)$$

$$B(i) = -q_i, \quad (33)$$

and

$$B(j) = -q_j, \quad (34)$$

Note that for the nonlinear T^3 terms, $A(i,j) \neq A(j,i)$. It can be shown that all linear terms are symmetrical: $A(a,b) = A(b,a)$.

The above examples show that the matrix terms for a network may be written directly without writing the heat balance equation, by examining the thermal path between the adjacent nodes.

To demonstrate the effect of thermal conductivity on the above coefficients, we backsubstitute $c_i = g_i k_i$ (where g_i is the geometrical term) so that

$$A(i,j) = E_{ij}T_j^3 + g_{ij}k_{ij}, \quad (35)$$

$$A(i,k) = g_{ik}k_{ik}, \quad (36)$$

$$A(i,l) = g_{il}k_{il}, \quad (37)$$

$$A(j,i) = E_{ij}T_i^3 + g_{ij}k_{ij}, \quad (38)$$

$$A(j,m) = g_{jm}k_{jm}, \quad (39)$$

$$A(j,n) = g_{jn}k_{jn}, \quad (40)$$

$$A(i,i) = -\sum_{b=k,l} g_{ib}k_{ib} - A(j,i), \quad (41)$$

and

$$A(j,j) = -\sum_{b=m,n} g_{jb}k_{jb} - A(i,j). \quad (42)$$

We can now consider thermal conductivity as a variable and as a function of the temperatures of the associated nodes.

Use of the thermal conductivity subroutine (Sec. II B) makes the above equations become

$$A(i,j) = E_{ij}T_j^3 + g_{ij} * CK(ij,i,j), \quad (43)$$

$$A(i,k) = g_{ik} * CK(ii,i,k), \quad (44)$$

$$A(i,l) = g_{il} * CK(ii,i,l), \quad (45)$$

$$A(j,i) = E_{ij}T_i^3 + g_{ij}CK(ij,j,i), \quad (46)$$

$$A(j,m) = g_{jm} * CK(jj,j,m), \quad (47)$$

$$A(j,n) = g_{jn} * CK(jj,j,n), \quad (48)$$

$$A(i,i) = -\sum_{b=k,l} g_{ib}A(i,b) - A(j,i), \quad (49)$$

and

$$A(j,j) = -\sum_{b=m,n} g_{jb}A(j,b) - A(i,j), \quad (50)$$

where

ii = index for solid containing nodes i, k, and l,

ij = index for gas between nodes i and j,

and

jj = index for solid containing nodes j, m, and n.

The above example shows that for a thermal network of N nodes, a square A matrix of N x N terms and a columnar matrix for B(N) are generated. The T(N) columnar matrix is implicit for the matrix equation

$$[A_{ik}][T_k] = [B_k]$$

(51)

for $i = 1$ through N and $k = 1$ through N .

Further, nonzero A matrix terms exist only for contiguous nodes. Therefore, most of the A matrix terms are zero (sparse matrix).

IV. THE REFERENCE NODE(S)

The matrices just described represent a set of N equations with N unknowns (N temperatures). If the matrices were left in their present state, an infinite number of solutions would satisfy all equations. For example, set the temperature of node k to a constant T_k . It follows that all the other temperatures would self-adjust so that a valid network solution would be obtained. And that is true for any other temperature T_k , so an infinite number of solutions is possible.

This condition is avoided by choosing one or more reference nodes for which the temperatures are known. This is best done by choosing the heat sink node(s) for reference.

For a reference node, the A matrix diagonal and corresponding B matrix term must be modified by a variable ($ANODE$) to satisfy

$$ANODE \left(\frac{TREF}{ANODE} - T_N \right) = 0.0, \quad (52)$$

or

$$-ANODE * T_N = -TREF, \quad (53)$$

where $TREF$ is the absolute temperature at T_N (predefined) and $1. < ANODE < 2$. The corresponding A and B matrix terms are changed to

$$A(N,N)_{new} = A(N,N)_{old} - ANODE, \quad (54)$$

and

$$B(N) = -TREF. \quad (55)$$

A. One Reference Node

For a thermal network that has only one heat sink reference node (e.g., a one-dimensional network), $ANODE$ can be calculated exactly by

$$ANODE = 1.0 + P/TREF$$

(56)

(i.e., all heat P generated by the system must pass through node N when leaving $T(N) = TREF$.) See the example in Sec. IX.

B. More than One Reference Node

In thermal networks that have more than one reference node (e.g., two- and three-dimensional networks), $ANODE$ cannot be precomputed precisely. Then, $ANODE$ is dimensioned to the existing number of reference nodes. An initial value (e.g., 1.001) is assigned to each $ANODE(i)$. These values are then further modified in the iteration process as follows (see example in Appendix D).

1. Set the initial precision limit ($TLIMIT$) required for solution to a high value (e.g., $TLIMIT = FLOAT(N)$ for a system of N nodes).
2. Solve the network to that limit (below).
3. Update $ANODE(i)$ by using

$$ANODE(i)_{new} = ANODE(i)_{old} * T(i)/TREF. \quad (57)$$

4. Reduce the precision limit for the next solution required and return to step 2.

5. Continue this process until the final precision limit ($XTLIMIT$) has been satisfied.

By this means, each reference node is computed to be equal to the specified reference node temperature ($TREF$). Note that all reference nodes do not have to be at the same temperature. $TREF$ can be dimensioned to the number of reference nodes, giving each its own value. Equation (57) then becomes

$$ANODE(i)_{new} = ANODE(i)_{old} * T(i)/TREF(i). \quad (58)$$

Note that for successive intermediate solutions each reference node's matrix terms are redefined.

$$A(i,i)_{new} = A(i,i)_{old} - ANODE(i), \quad (59)$$

and

$$B(i) = -TREF, \quad (60)$$

or

$$B(i) = -TREF(i). \quad (61)$$

V. THE MATRIX SOLVERS

To solve the matrix equation, Eq. (51), we chose the Gauss-Jordan elimination method^a (Appendix B). Gauss developed this method originally to solve sets of N linear equations with N unknowns by successive elimination, and finally by back-substitution. Jordan later modified the method so that the back-substitution is done simultaneously with the Gaussian elimination. Appendix B, Sec. I is a computer listing of the standard Gauss-Jordan method (GAUSSJ). The solution is returned in the columnar matrix B so that $T(i) = B(i)$ for $i=1$ through N nodes.

For network problems, the standard Gauss-Jordan method is long and time-consuming because it is general and was devised to work with dense matrices. Network problems usually produce very sparse matrices. The author has modified the Gauss-Jordan method for easy application to network problems by checking to see if an unknown (node) is missing (not numbered, or isolated in the system, as in switched electric circuits) and by bypassing the processing of zero terms in the A matrices. These modifications greatly speed the computational process and reduce the computer time required. A computer listing of this modified version is LSYSL (App. B, Sec. I). This method is general and applicable to multidimensional network problems, but it takes advantage of the fact that the matrices are sparse.

A one-dimensional problem in which node k is associated only with nodes $(k-1)$ and $(k+1)$ produces a tridiagonal A matrix (with no nodes missing or isolated). Taking advantage of this, the author further modified the Gauss-Jordan method to be specific for one-dimensional problems such as LSYSL1 (App. B, Sec. III). This method is used in the example model in Sec. VIII. This modification gives a factor of 6 computer time reduction compared to use of the general modified Gauss-Jordan method.

VI. SOLUTION METHOD

Because the Gauss-Jordan matrix solver is for linear equations, the equations for heat transfer encountered were nonlinear (Sec. III), and thermal conductivity variations with temperature must be accounted for, an iterative method was developed to solve the thermal network problems. The iteration process starting with a set of estimated temperatures is used. For each iteration, the thermal conductivities are updated for the new temperatures. For each successive solution, as the precision limit (TLIMIT) is reduced, the modification factor (ANODE) is changed (except for one-dimensional or one reference node problems, (Sec. IV A). This process is repeated until the final precision limit (XTLIMIT) has been reached. Thus, nonlinear equations are solved by use of linear techniques.

A suggested approach to mathematical modeling (see the examples in Apps. C and D) of a thermal network is as follows.

- A. Write a function subroutine for thermal conductivities of materials (see App. A for an example).
- B. Draw the thermal system and place nodes at strategic points in it. Careful numbering of the nodes facilitates later computations for similar nodes in DO loops.
- C. Dimension all A and B arrays and other variables to be used.
- D. Input data statements for system constants (including initial nodal temperature estimates).
- E. Compute geometrical constants and emissivity constants to be used in nodal equations.
- F. Begin the iteration process.
 1. Zero A and B matrix terms.
 2. Define linear A matrix terms for $A(i,j)$ where $i < j$ for all linear terms in the system.
 3. Load the linear symmetrical terms into $A(j,i) = A(i,j)$ using a DO loop. See examples in Apps. C and D.
 4. Load A matrix diagonal terms with linear terms using nested DO loops. See example in App. C for one-dimensional systems and in App. D for multidimensional systems.
 5. Define nonlinear, asymmetrical (emissivity) terms of A matrix.

6. Add nonlinear matrix terms to A matrix diagonal.

7. Modify corresponding A matrix diagonal(s) for reference node(s) (see Sec. IV).

8. Define B matrix with q_{node} .

9. Set reference node temperature(s) by using $B(i) = -TREF$.

10. Call matrix solver.

11. Compute iteration precision.

12. Set new temperatures ($T(i) = B(i)$).

13. If precision limit has not been met, return to step 1.

14. If precision limit (TLIMIT) has been met, reduce TLIMIT.

15. If final precision limit (XTLIMIT) has been met, go to step 18.

16. Recompute ANODE for reference node modification.

17. Return to step 1.

18. Print the results.

See examples in Secs. VIII and IX.

VII. PRINCIPLES FOR FACILITATING SOLUTION

A. For a thermal network involving Eq. (2), it is logical to place a node at the center of the cylinder ($r = 0.0$). This causes a division by zero, which is undefined. In practice, it is adequate to let $r = r_0/50$.⁴ In effect, this is offsetting that node from the center. However, it seems to make the answer accurate enough when XTLIMIT is small enough (see Sec. IX).

B. When Eq. (3) is used with a sphere (a solid sphere is modeled as concentric hollow spheres, see Sec. VIII), it is not possible to put a node at the sphere center for $r = 0$, again because of division by zero. If the nodal separation within the sphere is to be $2 \cdot xR$, then the first (center) node must be offset from the center by $r = R/2$, but the volume for that node must be retained for $r = R$. See the example in Sec. VIII.

C. Note that the computed thermal profile of a heat-generating sphere depends on the minimum number of nodes that should be used for accuracy. An empirical equation,

$$N = \text{INT}(69.52 \cdot (P/PDENSTY)**0.33333) - 3, \quad (62)$$

has been derived for the self-heating sphere.

$N = N_{old} + 4$ for the doubly clad problem in Sec. VIII and the listing in App. C (two nodes for each clad).

Note that if fewer than the minimum number of nodes are used, the computed thermal profile within the sphere is incorrect but the computed surface temperature of the sphere is correct. Thus, if one is not interested in the internal thermal profile of the sphere, he can save array dimensions and computer time by using a minimum of two nodes within the fuel sphere.

VIII. ONE-DIMENSIONAL EXAMPLE PROBLEM

A. Fuel Sphere Assembly

Consider a fuel sphere assembly (FSA) (Fig. 1) suspended in a uniform medium so that the entire outer surface is isothermal. The FSA consists of a solid central sphere of PuO_2 fuel that produces heat by radioactive decay, a hollow iridium-clad sphere (PICS) and, a hollow outer sphere of graphite (GIS). The system is thermally symmetrical about its center (i.e. any radial line drawn from the center of the fuel to the surface of the GIS is adiabatic and

Fig. 1.
Fuel sphere assembly.

equivalent to all other radial lines that can be drawn). Therefore, the problem can be modeled as one-dimensional along a radial line.

The gap between the fuel and PICS is predetermined (G1 cm), as is that between the PICS and GIS (G2 cm). Both gaps are filled with helium. The thicknesses of the PICS (C1 cm) and the GIS (S1 cm) are also predetermined.

The computer program (see App. C) was constructed so that many items can be changed easily. The power (P, W_t) and power density (PDENSTY) of the fuel are used to compute the fuel sphere diameter and the number of thermal nodes required to describe the system accurately (see Sec. VII C for an explanation of node number calculation). Although the model is one-dimensional, the thermal equations used were for concentric hollow spheres, Eq. (3), and radiation between spheres, Eq. (4). It was assumed that sphere masses and areas were concentrated at the radial nodal points.

Four nodes, N-4, N-3, N-2, and N-1, in this problem required radiation, nonlinear terms. This requirement renders the corresponding A matrix asymmetrical and nonlinear.

For this particular problem, the power rating (P) of the fuel was set at 100 W, and the temperature of the reference node N was set at 1214°C (CREF).

An example of the output computations is given in Table I. The compilation and solution required nine iterations (N2) and 17.239 s on a CDC-6600 computer. Computer core space required was 150 000 words. Two-hundred nodes were required to provide an accurate thermal profile of the fuel (see Sec. VII C).

B. Design Degradation Study

Problem: For the model just discussed, determine the GIS surface temperature after the PPO has decayed to 80% of its original power rating. This is done most easily by inserting the following line into the program.

$$00871 \text{ PDENSTY} = 0.80 * \text{PDENSTY}. \quad (63)$$

Thus, all dimensions are calculated for the original power density, but heat output is based on the new value. (Note: ANODE remains unchanged!)

The output results are shown in Table II. Comparison of Tables I and II shows that the GIS surface temperature drops from its original 1214°C to 1195°C when the fuel is degraded. Further, the fuel center temperature drops from 1384.6 to 1337.5°C. Similar studies can determine the useful life of the FSA.

IX. THREE-DIMENSIONAL EXAMPLE PROBLEM

Consider the plutonia aerodynamic disk (PAD) configuration shown in Fig. 2. It is composed of several (M) PuO_2 cylinders of length-to-diameter ratio ALD, and power P (W_t), encapsulated in Pt-Rh cylinders of thickness CLAD, embedded in a circle of cavities about the center of a graphite disk. Some other dimension variables are shown in Fig. 3. Other dimensions are as follows.

GAP1A = axial gap between fuel and clad,

GAP1R = radial gap between fuel and clad,

GAP2A = axial gap between clad and graphite,

GAP2R = radial gap between clad and graphite.

The following adiabatic planes can be passed through the model:

Side center of PAD, passing through the center of the fuel (Fig. 4).

Axially through the PAD center and a fuel cylinder center (Fig. 3).

Axially through the PAD center and along a radial line equidistant between PPO fuel capsules (Fig. 3).

The cylindrical edge is also considered adiabatic. For this problem, heat is considered to be dissipated from the PAD surface over the area of a circle of radius J-FG (Fig. 12, also called DJFG in the program). The rest of the PAD surface is considered adiabatic. The full PAD can then be modeled as a pie-shaped section.

The computer program (PADTHRM, App. D) takes M, P, and ALD as input. From these the PAD and model geometry (Figs. 3, 5, 6) is generated.

Thermal nodes are numbered as in Figs. 7-12. There are 124 nodes that require emissivity (nonlinear) terms, and 25 reference nodes (166-190). There is a total of 198 nodes in the model.

TABLE I
FSA THERMAL PROFILE

ITIMES	1	N2	2		
	2	4			
	3	6			
	4	7			
	5	8			
P	6	DFUEL	9	DFSA	
.100000000000E+03		.356930803055E+01		.633536803055E+01	
.106724271257E+01					
ANODE	1	NODE	1657.78 °K	1384.63 °C	2524.34 °F
	2		1657.78	1384.63	2524.33
	3		1657.77	1384.62	2524.31
	4		1657.75	1384.60	2524.29
	5		1657.73	1384.58	2524.25
	6		1657.71	1384.56	2524.21
	7		1657.68	1384.53	2524.15
	8		1657.64	1384.49	2524.09
	9		1657.60	1384.45	2524.01
	10		1657.55	1384.40	2523.93
	11		1657.50	1384.35	2523.83
	12		1657.44	1384.29	2523.72
	13		1657.38	1384.23	2523.61
	14		1657.31	1384.16	2523.48
	15		1657.23	1384.08	2523.35
	16		1657.15	1384.00	2523.20
	17		1657.07	1383.92	2523.05
	18		1656.97	1383.82	2522.88
	19		1656.88	1383.73	2522.71
	20		1656.77	1383.62	2522.52
	21		1656.66	1383.51	2522.33
	22		1656.55	1383.40	2522.12
	23		1656.43	1383.28	2521.90
	24		1656.30	1383.15	2521.68
	25		1656.17	1383.02	2521.44
	26		1656.04	1382.89	2521.20
	27		1655.89	1382.74	2520.94
	28		1655.75	1382.60	2520.67
	193		1546.01	1272.86	2323.14
	194		1544.75	1271.60	2320.88
	195		1543.48	1270.33	2318.59
	196		1542.20	1269.05	2316.30
	197		1504.70	1231.55	2248.78
	198		1504.66	1231.51	2248.72
	199		1492.34	1219.19	2226.54
	200		1487.15	1214.00	2217.20

TABLE II
FSA DESIGN STUDY

ITIMES	1	N2	2		
	2		3		
	3		6		
	4		7		
P	5	DFUEL	8	DFSA	
.100000000000E+03		.356930803055E+01		.633536803055E+01	
.106724271257E+01					
ANODE	1	NODE	1610.68 °K	1337.53 °C	2439.56 °F
	2		1610.68	1337.53	2439.56
	3		1610.67	1337.52	2439.54
	4		1610.66	1337.51	2439.52
	5		1610.64	1337.49	2439.49
	6		1610.62	1337.47	2439.45
	7		1610.60	1337.45	2439.40
	8		1610.56	1337.41	2439.34
	9		1610.53	1337.38	2439.28
	10		1610.49	1337.34	2439.21
	11		1610.44	1337.29	2439.12
	12		1610.39	1337.24	2439.03
	13		1610.33	1337.18	2438.93
	14		1610.27	1337.12	2438.82
	15		1610.21	1337.06	2438.71
	16		1610.14	1336.99	2438.58
	17		1610.06	1336.91	2438.45
	18		1609.99	1336.84	2438.30
	19		1609.90	1336.75	2438.15
	20		1609.81	1336.66	2437.99
	21		1609.72	1336.57	2437.82
	22		1609.62	1336.47	2437.64
	23		1609.52	1336.37	2437.46
	24		1609.41	1336.26	2437.26
	25		1609.29	1336.14	2437.06
	26		1609.18	1336.03	2436.85
	27		1609.05	1335.90	2436.62
	28		1608.92	1335.77	2436.39
	193		1516.45	1243.30	2269.93
	194		1515.41	1242.26	2268.06
	195		1514.36	1241.21	2266.18
	196		1513.31	1240.16	2264.29
	197		1482.60	1209.45	2209.02
	198		1482.57	1209.42	2208.96
	199		1472.50	1199.35	2190.83
	200		1468.41	1195.26	2183.47

Fig. 2.

Configuration of plutonia aerodynamic disk (PAD).

Table III shows a sample output for the problem $M = 4$, $P = 100$, and $ALD = 1.0$. CREF was set at 800°C . The program requires 150 000 words of core and 71.613 s to be compiled and run on a CDC-6600 computer.

X. SUMMARY

A. A computer-oriented method for solution of nonlinear thermal network problems has been developed and demonstrated. The method is equally applicable to linear systems.

B. It was initially thought that the small magnitude of the Stefan-Boltzmann constant ($\sigma = 5.67032 \times 10^{-12} \text{ Wcm}^{-2} \text{ }^{\circ}\text{K}^{-4}$) multiplier in Eq. (4) made the nonlinear matrix terms so small that solution was nearly linear. However, when σ was set

to a value of 1.0, convergence to solution still obtained. This tends to indicate that the method described may be applicable to other systems of simultaneous nonlinear equations.

C. The method described is for static systems. Dynamic and transient systems may be solved by

1. Incorporating time-dependent terms in the matrix coefficients,
2. Initializing the time to zero,
3. Solving as a static system,
4. Incrementing time by Δt ,
5. Returning to step 3 until the time period of interest is covered.

D. The method described here can be changed easily. By changing dimensions or materials (thermal conductivities), one can study variations of the original design rapidly. Also by changing the reference node temperature(s), one can generate complete thermal profiles with very little effort.

E. Thermal models can be solved to any accuracy warranted or desired within the precision limits of the computer used.

REFERENCES

1. G. M. Dusenberre, *Heat-Transfer Calculations by Finite Differences* (International Textbooks of Mechanical Engineering, Scranton, PA, 1961).
2. W. H. McAdams, *Heat Transmission* (McGraw-Hill, New York, NY, 1954).
3. Ben Noble, *Applied Linear Algebra* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969).
4. J. J. Burns, Private communication.

Fig. 4.
PAD Geometry, A.

Fig. 5.
PAD Geometry, B.

Fig. 6.
Side view cross section of PAD.

Fig. 7.
PAD level one nodal numbers.

Fig. 8.
PAD level two nodal numbers.

Fig. 9.
PAD level three nodal numbers.

Fig. 10.
PAD level four nodal numbers.

Fig. 11.
PAD level five nodal numbers.

Fig. 12.
PAD level six nodal numbers.

TABLE III

PAD THERMAL PROFILE

M P ALD		1 ITIMES		2 N2			
? 4,100.,1.		3	PW	3	DFUEL	RPAD	DPAD
P.	100000000000E+03	.400000000000E+03	.520474203851E+01			.104094840770E+02	
ANODE(I).	100000000000E+03	.311807776811E+01	.311807776811E+01	AL.		.100000000000E+01	ALJD
	.100014198390E+01	.100035564220E+01	.100033191662E+01			.100030152599E+01	
	.100035445105E+01	.100027147212E+01	.100022518852E+01			.100016975622E+01	
	.100021197866E+01	.100113487911E+01	.100112550607E+01			.100108028850E+01	
	.100121714234E+01	.100079259832E+01	.100056527656E+01			.100049707521E+01	
	.100072560757E+01	.100263579735E+01	.100268565267E+01			.100263217682E+01	
	.100294904453E+01	.100172611483E+01	.100107530191E+01			.100105273152E+01	
	.100172083461E+01						
	1 NODE	1099.70°K	826.55°C			1519.79°F	
	2	1093.98	820.83			1509.49	
	3	1093.34	820.19			1508.34	
	4	1093.18	820.03			1508.06	
	5	1095.17	822.02			1511.64	
	6	1097.75	824.60			1516.29	
	7	1099.78	826.63			1519.93	
	8	1099.19	826.04			1518.87	
	9	1098.99	825.84			1518.50	
	10	1087.21	814.06			1497.31	
	11	1086.99	813.84			1496.91	
	12	1087.26	814.11			1497.39	
	13	1089.09	815.94			1500.70	
	14	1092.41	819.26			1506.66	
	15	1095.38	822.23			1512.01	
	16	1095.66	822.51			1512.52	
	17	1095.44	822.29			1512.12	
	18	1086.36	813.23			1495.81	
	19	1086.24	813.09			1495.57	
	20	1086.61	813.46			1496.22	
	21	1086.34	815.19			1499.35	
	22	1091.71	818.56			1505.40	
	23	1094.73	821.58			1510.85	
	24	1095.21	822.06			1511.70	
	25	1094.98	821.83			1511.29	
	26	1086.37	813.22			1495.80	
	27	1086.24	813.09			1495.56	
	28	1086.61	813.46			1496.22	
	29	1088.34	815.19			1499.34	
	30	1091.70	818.55			1505.39	
	31	1094.73	821.58			1510.84	
	32	1095.21	822.06			1511.70	
	33	1094.98	821.83			1511.29	
	34	1084.53	811.38			1492.48	
	35	1084.43	811.28			1492.30	
	36	1084.79	811.64			1492.96	
	37	1086.40	813.25			1495.85	
	38	1090.38	817.23			1503.01	
	39	1094.13	820.98			1509.77	
	40	1094.64	821.49			1510.67	
	41	1094.13	820.98			1509.77	
	42	1084.50	811.35			1492.43	
	43	1084.40	811.25			1492.25	
	44	1084.76	811.61			1492.90	
	45	1086.37	813.22			1495.79	
	46	1090.36	817.21			1502.97	
	47	1094.12	820.97			1509.75	
	48	1094.63	821.48			1510.66	
	49	1094.12	820.97			1509.75	
	50	1084.31	811.16			1492.09	
	51	1084.16	811.01			1491.82	
	52	1084.43	811.28			1492.30	

TABLE III (CONTINUED)

53	1066.14	812.99	1495.36
54	1090.32	817.17	1502.91
55	1094.12	820.97	1509.75
56	1094.62	821.47	1510.65
57	1094.08	820.93	1509.67
58	1085.88	812.73	1494.92
59	1082.79	809.64	1489.36
60	1082.77	809.65	1489.31
61	1082.75	809.60	1489.28
62	1082.77	809.62	1489.32
63	1083.11	809.96	1489.93
64	1083.46	810.31	1490.55
65	1083.51	810.36	1490.65
66	1083.49	810.34	1490.61
67	1081.43	808.28	1486.90
68	1081.35	808.20	1486.75
69	1081.43	808.23	1486.91
70	1082.80	809.65	1489.37
71	1084.87	811.72	1493.09
72	1086.65	813.50	1496.30
73	1085.99	812.64	1495.12
74	1085.36	812.23	1494.02
75	1081.43	808.28	1486.90
76	1081.35	808.20	1486.75
77	1081.43	808.28	1486.91
78	1082.80	809.65	1489.37
79	1084.87	811.72	1493.09
80	1086.65	813.50	1496.30
81	1085.99	812.64	1495.12
82	1085.36	812.23	1494.02
83	1081.43	808.27	1486.89
84	1081.34	808.19	1486.75
85	1081.43	808.26	1486.90
86	1082.79	809.64	1489.36
87	1084.86	811.71	1493.08
88	1086.65	813.50	1496.30
89	1085.99	812.84	1495.11
90	1085.36	812.23	1494.02
91	1079.29	806.14	1483.06
92	1079.37	806.22	1483.20
93	1079.66	806.51	1483.71
94	1080.44	807.29	1485.12
95	1082.87	809.72	1489.50
96	1085.25	812.10	1493.79
97	1084.79	811.64	1492.95
98	1083.88	810.73	1491.31
99	1079.25	806.10	1482.98
100	1079.33	806.18	1483.12
101	1079.62	806.47	1483.64
102	1080.39	807.24	1485.03
103	1082.83	809.68	1489.42
104	1085.22	812.07	1493.73
105	1084.76	811.61	1492.90
106	1083.84	810.69	1491.25
107	1079.03	805.86	1482.58
108	1079.06	805.91	1482.65
109	1079.30	806.15	1483.07
110	1080.12	806.97	1484.51
111	1082.79	805.64	1489.35
112	1085.22	812.07	1493.73
113	1084.75	811.60	1492.88
114	1083.11	810.36	1491.11
115	1083.04	809.89	1489.80
116	1082.79	809.64	1489.36
117	1082.77	809.62	1489.31
118	1082.75	809.60	1489.28
119	1082.77	809.62	1489.32
120	1083.11	809.96	1489.93

TABLE III (CONTINUED)

121	1083.46	810.31	1490.55
122	1083.51	810.36	1490.65
123	1083.49	810.34	1490.61
124	1082.45	809.30	1488.73
125	1082.46	809.31	1488.76
126	1082.46	809.31	1488.76
127	1082.26	809.11	1488.40
128	1082.62	809.47	1489.04
129	1083.01	809.86	1489.74
130	1083.16	810.01	1490.02
131	1083.15	810.00	1489.99
132	1083.03	809.88	1489.78
133	1082.76	809.61	1489.30
134	1082.74	809.59	1489.25
135	1082.71	809.56	1489.22
136	1082.73	809.58	1489.25
137	1083.06	809.91	1489.84
138	1083.40	810.25	1490.45
139	1083.45	810.30	1490.54
140	1083.43	810.28	1490.51
141	1082.44	809.29	1488.72
142	1082.45	809.30	1488.74
143	1082.46	809.31	1488.75
144	1082.26	809.11	1488.40
145	1082.62	809.47	1489.04
146	1083.00	809.85	1489.73
147	1083.16	810.01	1490.01
148	1083.14	809.99	1489.99
149	1074.27	801.12	1474.02
150	1074.21	801.06	1473.91
151	1074.21	801.06	1473.91
152	1074.21	801.06	1473.91
153	1074.22	801.07	1473.93
154	1074.27	801.12	1474.01
155	1074.31	801.16	1474.10
156	1074.32	801.17	1474.11
157	1074.32	801.17	1474.11
158	1074.20	801.05	1473.90
159	1074.20	801.05	1473.89
160	1074.20	801.05	1473.90
161	1074.21	801.06	1473.91
162	1074.26	801.11	1474.00
163	1074.32	801.17	1474.10
164	1074.33	801.18	1474.12
165	1074.32	801.17	1474.11
166	1073.09	799.94	1471.88
167	1072.99	799.84	1471.71
168	1072.98	799.83	1471.70
169	1072.98	799.83	1471.70
170	1072.98	799.83	1471.70
171	1073.04	799.89	1471.80
172	1073.08	799.93	1471.88
173	1073.06	799.93	1471.87
174	1073.06	799.91	1471.83
175	1073.00	799.85	1471.73
176	1072.99	799.84	1471.71
177	1072.98	799.83	1471.69
178	1073.00	799.85	1471.73
179	1073.04	799.89	1471.81
180	1073.10	799.95	1471.90
181	1073.08	799.93	1471.88
182	1073.06	799.91	1471.84
183	1073.45	800.30	1472.53
184	1073.46	800.31	1472.56
185	1073.44	800.29	1472.53
186	1073.54	800.39	1472.71
187	1073.31	800.16	1472.28
188	1073.24	800.09	1472.16
189	1073.24	800.09	1472.16

TABLE III (CONTINUED)

190	1073.37	800.22	1472.39
191	1074.90	801.75	1475.16
192	1075.41	802.26	1476.08
193	1076.04	802.89	1477.19
194	1075.36	802.21	1475.98
195	1073.48	800.33	1472.59
196	1073.24	800.09	1472.16
197	1073.30	800.15	1472.26
198	1073.82	800.67	1473.21

APPENDIX A

KRONOS, FORTRAN LISTING OF THERMAL CONDUCTIVITY FUNCTION SUBROUTINE

```

04890 FUNCTION CK(I,J,K)
04900C
04910C     FUNCTION SUBROUTINE FOR THERMAL CONDUCTIVITY
04920C
04930 DIMENSION C(5)
04940C
04950 COMMON/TEMP/T(1)
04960C
04970 DATA (C(I),I=1,3)/7.583568E-3,-2.734178E-5,1.201841E-7/
04980 DATA (C(I),I=4,5)/-1.304382E-10,4.546199E-14/
04990C
05000C     THERMAL CONDUCTIVITY IN WATTS/CM-DEG K
05010C
05020 X=(T(J)+T(K))/2.
05030 GO TO (10,20,30,40),I
05040C
05050 10 CONTINUE
05060C PFO     (DATA IN DEGREES C)
05070 X=X-273.15
05080 CK=C(1)+C(2)*X+C(3)*X**2+C(4)*X**3+C(5)*X**4
05090 GO TO 999
05100C
05110 20 CONTINUE
05120C PT-RH ALLOY CLAD
05130 CK=1.2226538E-3*X + 1.7911849
05140 GO TO 999
05150C
05160 30 CONTINUE
05170C GRAPHITE
05180 IF(X.GT.1940.)GO TO 999
05190 CK=-2.1967031E-4*X + .61025427
05200 IF(X.GT.811.)GO TO 999
05210 CK=-3.8778386E-4*X + .74658225
05220 GO TO 999
05230C
05240 40 CONTINUE
05250C HELIUM
05260 CK=2.7327313E-6*X + 8.604340E-4
05270C
05280 999 RETURN
05290 END

```


APPENDIX B

KRONOS, FORTRAN LISTINGS OF GAUSS-JORDAN MATRIX SOLVERS

I. GAUSSJ

```
05330 SUBROUTINE GAUSSJ(ARR,BRR,N)
05340C
05350C   GENERAL GAUSS-JORDAN MATRIX SOLVER
05360C
05370 DIMENSION ARR(N,N),BRR(N)
05380C
05390 DO 60 J=1,N
05400 IF(ABS(ARR(J,J)).GT.1.E-20)GO TO 20
05410 ARR(J,J)=1.
05420 GO TO 60
05430 20 CONTINUE
05440 DIV=ARR(J,J)
05450 DO 50 I=1,N
05460 AMUL=ARR(I,J)
05470 IF(ABS(AMUL).LT.1.E-20)GO TO 50
05480 IF(I.EQ.J)GO TO 50
05490 DO 40 K=J,N
05500 ARR(I,K)=ARR(I,K)-AMUL*ARR(J,K)
05510 40 CONTINUE
05520 50 CONTINUE
05530 60 CONTINUE
05540 RETURN
05550 END
```

II. LSYSL

```
05330 SUBROUTINE LSYSL(ARR,BRR,N)
05340C
05350C   MODIFIED, GENERAL GAUSS-JORDAN MATRIX SOLVER
05360C
05370 DIMENSION ARR(N,N),BRR(N)
05380C
05390 DO 60 J=1,N
05400 IF(ABS(ARR(J,J)).GT.1.E-20)GO TO 20
05410 ARR(J,J)=1.
05420 GO TO 60
05430 20 CONTINUE
05440 DIV=ARR(J,J)
05450 DO 30 I=J,N
05460 ARR(J,I)=ARR(J,I)/DIV
05470 30 CONTINUE
05480 BRR(J)=BRR(J)/DIV
05490 DO 50 I=1,N
05500 AMUL=ARR(I,J)
05510 IF(ABS(AMUL).LT.1.E-20)GO TO 50
05520 IF(I.EQ.J)GO TO 50
05530 DO 40 K=J,N
05540 ARR(I,K)=ARR(I,K)-AMUL*ARR(J,K)
05550 40 CONTINUE
05560 BRR(I)=BRR(I)-AMUL*BRR(J)
05570 50 CONTINUE
05580 60 CONTINUE
05590 RETURN
05600 END
```

III. LSYSL1

```
03060 SUBROUTINE LSYSL1(ARR,BRR,N)
03070C
```

```

03080C    NONMALIZED GAUSS-JORDAN FOR TRIDIAGONAL MATRIX
03090C
03100C
03110 DIMENSION ARR(N,N),BRR(N)
03120C
03130 DO 60 J=1,N
03140 L=J+1
03150 IF(L.GT.N)L=N
03160 DIV=ARR(J,J)
03170 DO 30 I=J,L
03180 ARR(J,I)=ARR(J,I)/DIV
03190 30 CONTINUE
03200 BRR(J)=BRR(J)/DIV
03210 DO 50 I=1,N
03220 AMUL=ARR(I,J)
03230 IF(ABS(AMUL).LT.1.E-20)GO TO 50
03240 IF(I.EQ.J)GO TO 50
03250 DO 40 K=J,L
03260 ARR(I,K)=ARR(I,K)-AMUL*ARR(J,K)
03270 40 CONTINUE
03280 BRR(I)=BRR(I)-AMUL*BRR(J)
03290 50 CONTINUE
03300 60 CONTINUE
03310 RETURN
03320 END

```

APPENDIX C

PROGRAM LISTING OF FUEL SPHERE ASSEMBLY EXAMPLE PROBLEM

LASL IDENTIFICATION: LP-0631

```

00100 PROGRAM FSASPHR(INPUT,OUTPUT)
00110C
00120C
00130C    THERMAL MODEL OF FUEL SPHERE ASSEMBLY (ONE-DIMENSIONAL)
00140C
00150C
00160 DIMENSION A(200,200),B(200),Q(200),R1(200)
00170C
00180 COMMON/TEMP/T(200)
00190C
00200 DATA SIGMA/5.6703271E-12/
00210 DATA T/200*1700./
00220C
00230C    P=WATTS THERMAL OF FUEL SPHERE
00240C
00250 P=100.
00260C
00270C    PDENSTY=POWER DENSITY OF FUEL, WATTS/CC
00280C
00290 PDENSTY=4.2
00300C
00310C    CREF=FSA SURFACE REFERENCE NODE TEMPERATURE, DEGREES C
00320C
00330 CREF=1214.
00340C
00350C    COMPUTE NUMBER OF NODES REQUIRED
00360C
00370 N=INT(69.52*(P/PDENSTY)**.33333333)+1
00380 N=MINO(N,200)
00390 N=MAXO(N,6)
00400C

```

```

00410 J1=N-4
00420 J2=J1-1
00430 J3=N-1
00440 N1=100
00450 N2=0
00460 N3=0
00470 ITIMES=0
00480C
00490C    SET INITIAL PRECISION LIMIT
00500C
00510 TLIMIT=FLOAT(N)
00520C
00530C    DEFINE FINAL PRECISION LIMIT REQUIRED
00540C
00550 XTLIMIT=0.1
00560C
00570C    FSA DIMENSION CONSTANTS
00580C
00590 G1=.10795
00600 G2=.04572
00610 C1=.06096
00620 S1=1.1684
00630C
00640C    EMISSIVITY CONSTANTS
00650C
00660 F1=.5
00670 F2=.25
00680 F3=.85
00690 E1=SIGMA/(1./F1+1./F2-1.)
00700 F2=.85
00710 E2=SIGMA/(1./F2+1./F3-1.)
00720C
00730 PI=2.*ATAN2(1.,0.)
00740C
00750C    COMPUTE DIAMETER OF FUEL
00760C
00770 D=EXP(ALOG(6.*P/(PI*PDENSTY)))/3.)
00780C
00790C    2.*R=SEPARATION OF NODES IN FUEL
0800C
00810 R=D/(2.*(2.*FLOAT(J1)-1.))
00820C
00830 X=D/2.
00840 Y=X+G1+C1
00850 A1=4.*PI*X*(X+G1)
00860 A2=4.*PI*Y*(Y+G2)
00870C
00880 X=PDENSTY*4.*PI/3.
00890 DO 10 I=1,J1
00900 Q(I)=X*((2.*FLOAT(I)-1.)*R)**3
00910 10 CONTINUE
00920 DO 20 I=1,J2
00930 L=J1+1-I
00940 Q(L)=Q(L)-Q(L-1)
00950 IF(I.EQ.1)GO TO 20
00960 R1(I)=1./(1./((2.*FLOAT(I)-2.)*R)-1./((2.*FLOAT(I+1)-2.)*R))
00970 20 CONTINUE
00980C
00990 Z6=2./R
01000 X1=1./(Z6-1./(2.*R))
01010 X6=1./(1./(D/2.)-1./(D/2.+G1))
01020 X7=1./(1./{D/2.+G1}-1./(D/2.+G1+C1))
01030 X8=1./(1./{D/2.+G1+C1}-1./(D/2.+G1+C1+G2))
01040 X9=1./(1./{D/2.+G1+C1+G2}-1./(D/2.+G1+C1+G2+S1))
01050C
01060 X=4.*PI
01070 TREF=CREF+273.15
01080C
01090C    COMPUTE REFERENCE NODE MATRIX DIAGONAL CONSTANT
01100C
01110 ANODE=1.+P/TREF
01120C

```

```

01130 30 CONTINUE
01140C
01150C   BEGINNING OF ITERATION LOOP
01160C
01170 40 CONTINUE
01180C
01190C   ZERO B MATRIX AND TRIDIAGONAL TERMS OF A MATRIX
01200C
01210 DO 60 I=1,N
01220 L=I+1
01230 IF(L.GT.N)L=N
01240 M=I-1
01250 IF(M.LT.1)M=1
01260 DO 50 J=M,L
01270 A(I,J)=0.0
01280 50 CONTINUE
01290 B(I)=0.0
01300 60 CONTINUE
01310C
01320C   DEFINE LINEAR SYMMETRICAL TERMS OF MATRIX
01330C
01340 A(1,2)=X*X1*CK(1,1,2)
01350C
01360 DO 70 L=2,J2
01370 A(L,L+1)=X*R1(L)*CK(1,L,L+1)
01380 70 CONTINUE
01390C
01400 A(N-3,N-2)=X*X7*CK(2,N-3,N-2)
01410 A(N-1,N)=X*X9*CK(3,N-1,N)
01420C
01430C   LOAD SYMMETRICAL TERMS OF MATRIX
01440C
01450 DO 80 I=1,J3
01460 J=I+1
01470 A(J,I)=A(I,J)
01480 80 CONTINUE
01490C
01500C   LOAD MATRIX DIAGONAL WITH SYMMETRICAL TERMS
01510C
01520 DO 100 J=1,N
01530 L=J+1
01540 IF(L.GT.N)L=N
01550 M=J-1
01560 IF(M.LT.1)M=1
01570 DO 90 I=M,L
01580 IF(I.EQ.J)GO TO 90
01590 A(J,J)=A(J,J)-A(J,I)
01600 90 CONTINUE
01610 100 CONTINUE
01620C
01630C   DEFINE NON-LINEAR EMISSIVITY TERMS
01640C
01650 A(N-4,N-3)=A1*E1*T(N-3)**3+X*X6*CK(4,N-4,N-3)
01660 A(N-3,N-4)=A1*E1*T(N-4)**3+X*X6*CK(4,N-3,N-4)
01670 A(N-2,N-1)=A2*E2*T(N-1)**3+X*X8*CK(4,N-2,N-1)
01680 A(N-1,N-2)=A2*E2*T(N-2)**3+X*X8*CK(4,N-1,N-2)
01690C
01700C   ADD NON-LINEAR EMISSIVITY TERMS TO DIAGONAL
01710C
01720 A(N-4,N-4)=A(N-4,N-4)-A(N-3,N-4)
01730 A(N-3,N-3)=A(N-3,N-3)-A(N-4,N-3)
01740 A(N-2,N-2)=A(N-2,N-2)-A(N-1,N-2)
01750 A(N-1,N-1)=A(N-1,N-1)-A(N-2,N-1)
01760C
01770C
01780C   ADD REFERENCE NODE TO DIAGONAL
01790C
01800 A(N,N)=A(N,N)-ANODE
01810C
01820C   DEFINE B MATRIX
01830C
01840 DO 110 I=1,J1

```

```

01850 B(I)=-Q(I)
01860 110 CONTINUE
01870C
01880C     SET REFERENCE NODE TEMPERATURE
01890C
01900 B(N)=-TREF
01910C
01920C     CALL TRIDIAGONAL MATRIX SOLVER
01930C
01940 CALL LSYSL1(A,B,N)
01950C
01960C
01970C     INCREMENT ITERATION COUNTER
01980C
01990 N2=N2+1
02000C
02010 Z1=0.0
02020C
02030C     COMPUTE SQUARE OF ACCUMULATED ERROR
02040C
02050 DO 120 I=1,N
02060 Z2=B(I)-T(I)
02070 Z1=Z1+Z2*Z2
02080C
02090C     COMPUTE NEW TEMPERATURE
02100C
02110 T(I)=B(I)
02120 120 CONTINUE
02130C
02140 IF(N2.GT.N1)GO TO 130
02150C
02160C     PRECISION TEST
02170C
02180 IF(SQRT(Z1).GT.TLIMIT)GO TO 40
02190C
02200C     SET SOLUTION PRECISION REQUIRED IN DECRAMENTS
02210C
02220 IF(ITIMES.EQ.0)TLIMIT=TLIMIT/10.
02230 IF(ITIMES.EQ.1)TLIMIT=TLIMIT/10.
02240 IF(ITIMES.GT.3.AND.(N2-N3).LE.10)TLIMIT=XTLIMIT
02250 N3=N2
02260C
02270C     INCREMENT SOLUTION COUNTER
02280C
02290 ITIMES=ITIMES+1
02300C
02310 PRINT,ITIMES,N2
02320C
02330C     TEST FOR FINAL SOLUTION
02340C
02350 IF(SQRT(Z1).LT.XTLIMIT)GO TO 130
02360C
02370C
02380 GO TO 40
02390C
02400C
02410 130 CONTINUE
02420C
02430C
02440 DFSA=D+2.*(G1+G2+C1+S1)
02450 PRINT 900,P,D,DFSA
02460 PRINT 900,ANODE
02470C
02480 DO 140 I=1,N
02490 CENT=T(I)-273.15
02500 FF=9.*CENT/5.+32.
02510 PRINT,I,T(I),CENT,FF
02520 140 CONTINUE
02530C
02540 900 FORMAT(4(E18.12,2X))
02550C
02560 STOP
02570 END

```

APPENDIX D

PROGRAM LISTING OF PLUTONIA AERODYNAMIC DISK EXAMPLE PROBLEM

LASL IDENTIFICATION: LP-0632

```
00100 PROGRAM PADTHRM(INPUT,OUTPUT)
00110C
00120C   THERMAL MODEL OF PAD  (THREE DIMENSIONAL)
00130C
00140 DIMENSION A(198,198),B(198)
00150 DIMENSION ADIST(7,7)
00160 DIMENSION AGRAPH(8)
00170 DIMENSION ALENGTH(8)
00180 DIMENSION ANODE(25)
00190 DIMENSION F(8)
00200 DIMENSION THETA(8)
00210 DIMENSION Y4250(8)
00220C
00230 COMMON/TEMP/T(198)
00240C
00250 DATA ANODE/25*1.0001/
00260 DATA PDENSTY/4.2/
00270 DATA SIGMA/5.6703271E-12/
00280 DATA T/198*850./
00290C
00300C   INPUT
00310C   M=NUMBER OF FUEL ELEMENTS IN PAD
00320C   P=WATTS THERMAL OF EACH FUEL ELEMENT
00330C   ALD=LENGTH TO DIAMETER RATIO OF FUEL CYLINDERS
00340C
00350 READ,M,P,ALD
00360C
00370C   CREF=TEMPERATURE OF REFERENCE NODES, DEGREES C
00380C
00390 CREF=800.
00400C
00410C   PAD DIMENSION CONSTANTS
00420C
00430 CLAD=.0508
00440 EDGE=1.016
00450 FACE=.4445
00460 GAP1A=.01524
00470 GAP1R=.0127
00480 GAP2A=.0381
00490 GAP2R=.0381
00500 SEPAR=.254
00510C
00520 N=198
00530 N1=100
00540 N2=0
00550 N3=0
00560 ITIMES=0
00570C
00580C   SET INITIAL PRECISION LIMIT
00590C
00600 TLIMIT=FLOAT(N)
00610C
00620C   DEFINE FINAL PRECISION LIMIT REQUIRED
00630C
00640 XTLIMIT=0.1
00650C
00660C   SET RANGE OF REFERENCE NODES....INODE THRU JNODE
00670C
00680 INODE=166
00690 JNODE=190
```

```

00700 KNODE=INODE-1
00710C
00720 PI=2.*ATAN2(1.,0.)
00730 PIOTWO=PI/2.
00740 TWOPI=2.*PI
00750C
00760C     DETERMINE PAD GEOMETRY
00770C
00780 DFUEL=EXP(ALOG(4.*P/(PDENSTY*PI*ALD))/3.)
00790 RFUEL=DFUEL/2.
00800 R=RFUEL/5.
00810 AL=ALD*DFUEL
00820 RCAV=RFUEL + GAP1R + CLAD + GAP2R
00830 PSI=PI/FLOAT(M)
00840 DJFG=RCAV+SEPAR/2.
00850 YFUEL=DJFG/SIN(PSI)
00860 YCAV=YFUEL-RCAV
00870 RPAD=YFUEL+RCAV+EDGE
00880 CORD=2.*RPAD*SIN(PSI/6.)
00890 AARC=PI*RPAD**2*(PSI/3.)/TWOPI-RPAD**2*SIN(PSI/3.)/2.
00900 DJA=RCAV+EDGE
00910 DJK=YFUEL*SIN(PSI/3.)
00920 DIK=YFUEL*COS(PSI/3.)
00930 DKB=RPAD-DIK
00940 DJB=SQRT(DKB**2+DJK**2)
00950 THETA(1)=PI-(ASIN(DKB/DJB)+PIOTWO-PSI/3.)
00960 DJL=YFUEL*SIN(2.*PSI/3.)
00970 DIL=YFUEL*COS(2.*PSI/3.)
00980 DLC=RPAD-DIL
00990 DJC=SQRT(DLC**2+DJL**2)
01000 THETA(2)=PI-THETA(1)-(ASIN(DLC/DJC)+PIOTWO-2.*PSI/3.)
01010 DIFG=YFUEL*COS(PSI)
01020 DFGD=RPAD-DIFG
01030 DJD=SQRT(DFGD**2+DJFG**2)
01040 THETA(3)=PI-(THETA(1)+THETA(2))-(ASIN(DFGD/DJD)+PIOTWO-PSI)
01050 THETA(4)=2.*ASIN(DFGD/DJD)/5.
01060 THETA(5)=THETA(4)
01070 THETA(6)=THETA(4)
01080 THETA(7)=(PI-(THETA(1)+THETA(2)+THETA(3)+3.*THETA(4)))/2.
01090 THETA(8)=THETA(7)
01100 DJDE=DJFG/COS(2.*THETA(4))
01110 DJE=DJFG/COS(3.*THETA(4)/2.)
01120 DJEF=DJFG/COS(THETA(4))
01130 DJF=DJFG/COS(THETA(4)/2.)
01140 DJG=DJF
01150 DJGH=DJFG/COS((THETA(4)+THETA(7))/2.)
01160 DJH=DJFG/COS(THETA(4)/2.+THETA(7))
01170 DJHI=DJFG/COS((THETA(4)+3.*THETA(7))/2.)
01180 DJI=YFUEL
01190C
01200 ANGLE=PIOTWO-THETA(1)/2.
01210 AA=1.+TAN(ANGLE)**2
01220 BB=2.*DJI*TAN(ANGLE)
01230 CC=DJI**2-RPAD**2
01240 X=(-BB+SQRT(BB**2-4.*AA*CC))/(2.*AA)
01250 Y=X*TAN(ANGLE)+DJI
01260 DJAB=SQRT(X**2+(Y-DJI)**2)
01270 ANGLE=PIOTWO-THETA(1)-THETA(2)/2.
01280 AA=1.+TAN(ANGLE)**2
01290 BB=2.*DJI*TAN(ANGLE)
01300 X=(-BB+SQRT(BB**2-4.*AA*CC))/(2.*AA)
01310 Y=X*TAN(ANGLE)+DJI
01320 DJBC=SQRT(X**2+(Y-DJI)**2)
01330 ANGLE=PIOTWO-THETA(1)-THETA(2)-THETA(3)/2.
01340 AA=1.+TAN(ANGLE)**2
01350 BB=2.*DJI*TAN(ANGLE)
01360 X=(-BB+SQRT(BB**2-4.*AA*CC))/(2.*AA)
01370 Y=X*TAN(ANGLE)+DJI
01380 DJCD=SQRT(X**2+(Y-DJI)**2)
01390C
01400 Y12=TWOPI/ALOG(50.)
01410 Y210=TWOPI/ALOG(4./2.)

```

```

01420 Y1018=TWOPI/ALOG((RFUEL+GAP1R)/RFUEL)
01430 Y1826=TWOPI/ALOG((RFUEL+GAP1R+CLAD)/(RFUEL+GAP1R))
01440 DJ=(DJFG-RCAY)/4.
01450 Y2634=TWOPI/ALOG(RCAY/(RFUEL+GAP1R+CLAD))
01460 Y3442=TWOPI/ALOG((RCAY+3.*DJ)/(RCAY+DJ))
01470 Y1042=TWOPI/ALOG((RCAY+3.*DJ)/(4.*R))
01480 X=DJFG-DJ
01490 ALENGTH(1)=DJAB
01500 ALENGTH(2)=DJBC
01510 ALENGTH(3)=DJCD
01520 ALENGTH(4)=DJDE
01530 ALENGTH(5)=DJEF
01540 ALENGTH(6)=DJFG
01550 ALENGTH(7)=DJGH
01560 ALENGTH(8)=DJHI
01570 DO 10 I=1,8
01580 F(I)=(ALENGTH(I)+DJFG)/2.
01590 Y4250(I)=TWOPI/ALOG(F(I)/X)
01600 10 CONTINUE
01610C
01620 A1FUEL=PI*R**2
01630 A2FUEL=8.*A1FUEL
01640 A3FUEL=16.*A1FUEL
01650 ACLADR=PI*((RFUEL+GAP1R+CLAD)**2-(RFUEL+GAP1R)**2)/2.
01660 A34GRAP=PI*((RCAY+DJFG)/2.)*2-RCAY**2)
01670 A42GRAP=PI*(DJFG**2-((DJFG+RCAY)/2.)*2)
01680 AFACE99=PI*(DJFG**2-RFUEL**2)
01690 AA=PI*DJFG**2
01700C
01710 AGRAPH(1)=DJA*DJB*SIN(THETA(1))/2.+AARC-AA*THETA(1)/TWOPI
01720 AGRAPH(2)=DJB*DJC*SIN(THETA(2))/2.+AARC-AA*THETA(2)/TWOPI
01730 AGRAPH(3)=DJC*DJD*SIN(THETA(3))/2.+AARC-AA*THETA(3)/TWOPI
01740 AGRAPH(4)=DJD*DJE*SIN(THETA(4))/2.-AA*THETA(4)/TWOPI
01750 AGRAPH(5)=DJE*DJF*SIN(THETA(5))/2.-AA*THETA(5)/TWOPI
01760 AGRAPH(6)=DJF*DJG*SIN(THETA(6))/2.-AA*THETA(6)/TWOPI
01770 AGRAPH(7)=DJG*DJH*SIN(THETA(7))/2.-AA*THETA(7)/TWOPI
01780 AGRAPH(8)=DJH*DI*SIN(THETA(8))/2.-AA*THETA(8)/TWOPI
01790C
01800 X1=AL/4.
01810 X=DFUEL+2.*GAP1R
01820 Y=X+2.*CLAD
01830 Z=Y+2.*GAP2R
01840 ACLAD1=PI*X1*(X-DFUEL)/ALOG(X/DFUEL)
01850 ACLAD2=PI*X1*(Z-Y)/ALOG(Z/Y)
01860C
01870C EMISSIVITY TERMS
01880C
01890 F1=.5
01900 F2=.25
01910 F3=.85
01920 E1=SIGMA/(1./F1+1./F2-1.)
01930 F2=.85
01940 E2=SIGMA/(1./F2+1./F3-1.)
01950C
01960 Q1=PDENSTY*X1
01970C
01980C
01990C
02000 DO 20 I=1,7
02010 X=(THETA(I)+THETA(I+1))/4.
02020 ADIST(I,1)=4.*R*SIN(X)
02030 ADIST(I,2)=8.*R*SIN(X)
02040 ADIST(I,3)=2.*X*(RFUEL+GAP1A)
02050 ADIST(I,4)=2.*X*(RFUEL+GAP1A+CLAD)
02060 ADIST(I,5)=2.*(RCAY+(DJFG-RCAY)/4.)*SIN(X)
02070 ADIST(I,6)=2.*(RCAY+3.*(DJFG-RCAY)/4.)*SIN(X)
02080 Y=F(I)
02090 Z=F(I+1)
02100 ADIST(I,7)=SQRT(Y**2+Z**2-2.*Y*Z*COS(2.*X))
02110 20 CONTINUE
02120C

```



```

02130 TREF=CREF+273.15
02140C
02150C BEGINNING OF ITERATION LOOP
02160C
02170 30 CONTINUE
02180C
02190C ZERO A AND B MATRICES
02200C
02210 DO 50 I=1,N
02220 DO 40 J=1,N
02230 A(J,I)=0.0
02240 40 CONTINUE
02250 B(I)=0.0
02260 50 CONTINUE
02270C
02280C DEFINE LINEAR TERMS OF MATRIX
02290C
02300 DO 60 I=2,9
02310 Y=THETA(I-1)/TWOPI
02320 A(1,I)=X1*CK(1,1,I)*Y12
02330 A(I,I+8)=X1*CK(1,I,I+8)*Y210
02340 A(I+16,I+24)=X1*CK(2,I+16,I+24)*Y1826
02350 A(I+73,I+81)=X1*CK(2,I+73,I+81)*Y1826
02360 A(I+165,I+173)=FACE/2.*CK(3,I+165,I+174)*Y210
02370 A(58,I+57)=X1*CK(1,58,I+57)*Y12
02380 A(I+57,I+65)=X1*CK(1,I+57,I+65)*Y210
02390 A(I,I+57)=A2FUEL*THETA(I-1)/TWOPI*CK(1,I,I+57)/(2.*X1)
02400 A(I+8,I+65)=A3FUEL*THETA(I-1)/TWOPI*CK(1,I+8,I+65)/(2.*X1)
02410 A(I+16,I+73)=ACLADR*THETA(I-1)/TWOPI*CK(2,I+16,I+73)/(2.*X1)
02420 A(I+24,I+81)=ACLADR*THETA(I-1)/TWOPI*CK(2,I+24,I+81)/(2.*X1)
02430 A(I+32,I+40)=X1*CK(3,I+32,I+40)*Y3442
02440 A(I+89,I+97)=X1*CK(3,I+89,I+97)*Y3442
02450 A(I+32,I+89)=A34GRAP*THETA(I-1)/TWOPI*CK(3,I+32,I+89)/(2.*X1)
02460 A(I+40,I+97)=A42GRAP*THETA(I-1)/TWOPI*CK(3,I+40,I+97)/(2.*X1)
02470 A(I+40,I+48)=X1*CK(3,I+40,I+48)*Y4250(I-1)
02480 A(I+48,I+105)=AGRAPH(I-1)*CK(3,I+48,I+105)/(2.*X1)
02490 A(I+97,I+105)=X1*CK(3,I+97,I+105)*Y4250(I-1)
02500 X=CLAD/2.
02510 A(115,I+114)=X*CK(2,115,I+114)*Y12
02520 A(I+114,I+122)=X*CK(2,I+114,I+122)*Y210
02530 A(132,I+131)=X*CK(2,132,I+131)*Y12
02540 A(I+131,I+139)=X*CK(2,I+131,I+139)*Y210
02550 X=FACE/2.
02560 A(149,I+148)=X*CK(3,149,I+148)*Y12
02570 A(166,I+165)=X*CK(3,166,I+165)*Y12
02580 A(I+148,I+156)=X*CK(3,I+148,I+156)*Y210
02590 A(I+173,I+181)=X*CK(3,I+173,I+181)*Y1042
02600 A(I+181,I+189)=X*CK(3,I+181,I+189)*Y4250(I-1)
02610 A(I+114,I+131)=A2FUEL*Y*CK(2,I+114,I+131)/CLAD
02620 A(I+122,I+139)=A3FUEL*Y*CK(2,I+122,I+139)/CLAD
02630 A(I+148,I+165)=A2FUEL*Y*CK(3,I+148,I+165)/FACE
02640 A(I+156,I+173)=A3FUEL*Y*CK(3,I+156,I+173)/FACE
02650 X=FACE+GAP1A+CLAD+GAP2A
02660 A(I+97,I+181)=AFACE99*Y*CK(3,I+97,I+181)/X
02670 A(I+105,I+189)=AGRAPH(I-1)*CK(3,I+105,I+189)/X
02680C
02690 IF(I.EQ.9)GO TO 60
02700C
02710 X=2.*R*X1
02720 W=1./ADIST(I-1,1)
02730 A(I,I1)=X*W*CK(1,I,I+1)
02740 A(I+57,I+58)=X*W*CK(1,I+57,I+58)
02750 X=R*CLAD
02760 A(I+114,I+115)=X*W*CK(2,I+114,I+115)
02770 A(I+131,I+132)=X*W*CK(2,I+131,I+132)
02780 X=R*FACE
02790 A(I+148,I+149)=X*W*CK(3,I+148,I+149)
02800 A(I+165,I+166)=X*W*CK(3,I+165,I+166)
02810 X=2.*R*X1
02820 W=1./ADIST(I-1,2)
02830 A(I+8,I+9)=X*W*CK(1,I+8,I+9)

```

```

02840 A(I+65,I+66)=X*W*CK(1,I+65,I+66)
02850 X=R*CLAD
02860 A(I+122,I+123)=X*W*CK(2,I+122,I+123)
02870 A(I+139,I+140)=X*W*CK(2,I+139,I+140)
02880 X=R*FACE
02890 A(I+156,I+157)=X*W*CK(3,I+156,I+157)
02900 A(I+173,I+174)=X*W*CK(3,I+173,I+174)
02920 W=1./ADIST(I-1,3)
02930 A(I+16,I+17)=X*W*CK(2,I+16,I+17)
02940 A(I+3,I+74)=X*W*CK(2,I+73,I+74)
02950 W=1./ADIST(I-1,4)
02960 A(I+24,I+25)=X*W*CK(2,I+24,I+25)
02980 X=(DJFG-RCAV)*X1/2.
02990 W=1./ADIST(I-1,5)
0300 A(+32,I33)=X*W*CK(3,I+32,I+33)
0301 A(+89,I+0)=X*W*CK(3,I+89,I+90)
0020W=1./ADIST(I-1,6)
03030 A(I+40,I+41)=X*W*CK(3,I+40,I+41)
0040 A(I+97,I+98)=X*W*CK(3,I+97,I+98)
0350 X=(DJFG-RFUEL)*(GAP1A+CLD+GAP2A+FACE)
03060 A(I+181,I+182)=X*CK(3,I+181,I+182)
03070 Y=(ALEGTHI-)+ALEGTH(I))/2.-DJFG
03080 X=Y*X1
03090 =1./ADIST(I-1,7)
03100 A(I+4,I+9)=X*W*CK(3,I+48,I+49)
03110 A(I+105,I+106)=X*W*CK(3,I+105,I+106)
03120 X=Y*(GAP1A+CLAD+GAP2A+FACE)
03130 A(I+189,I+190)=X*W*CK(3,I+189,I+190)
03140 60 CONTINUE
03150C
03160 A(1,58)=A1FUEL/2.*CK(1,1,58)/(2.*X1)
03170 A(115,132)=A1FUEL/2.*C(2,115,132)/CLAD
03180 A(149,166)=A1FUEL/2.*CK(3,149,166)/FACE
03190C
03200C LOAD LINEAR SYMMETRICAL TERMS INTO MATRIX
03210C
03220 DO 80 I=1,N
0323 IF(I.GE.N)GO TO 80
03240 IP1=I+1
03250 DO 70 J=IP1,N
03260 A(J,I)=A(I,J)
03270 70 CONTINUE
03280 80 CONTINUE
03290C
03300C COMPUTE MATRIX DIAGONAL FOR LINEAR TERMS
03310C
03320 DO 100 J=1,N
03330 DO 90 I=1,N
03340 IF(I.EQ.J)GO TO 90
03350 A(J,J)=A(J,J)-A(J,I)
03360 90 CONTINUE
03370 100 CONTINUE
03380C
03390C DEFINE NON-LINEAR (NON-SYMMETRICAL) MATRIX TERMS
03400C
03410 A(58,115)=A1FUEL/2.*(E1*T(115)**3+CK(4,58,115)/GAP1A)
03420 A(115,58)=A1FUEL/2.*(E1*T(58)**3+CK(4,115,58)/GAP1A)
03430 A(132,149)=A1FUEL/2.*(E2*T(149)**3+CK(4,132,149)/GAP2A)
03440 A(149,132)=A1FUEL/2.*(E2*T(132)**3+CK(4,149,132)/GAP2A)
03450C
03460 DO 110 I=1,8
03470 X=THETA(I)/TWOPI
03480 A(I+9,I+17)=ACLAD1*X*E1*T(I+17)**3+X1*CK(4,I+9,I+17)*Y1018
03490 A(I+17,I+9)=ACLAD1*X*E1*T(I+9)**3+X1*CK(4,I+17,I+9)*Y1018
03500 A(I+25,I+33)=ACLAD2*X*E2*T(I+33)**3+X1*CK(4,I+25,I+33)*Y2634
03510 A(I+33,I+25)=ACLAD2*X*E2*T(I+25)**3+X1*CK(4,I+33,I+25)*Y2634
03520 A(I+58,I+115)=A2FUEL*X*(T(I+115)**3+CK(4,I+58,I+115)/GAP1A)
03530 A(I+115,I+58)=A2FUEL*X*(T(I+58)**3+CK(4,I+115,I+58)/GAP1A)
03540 A(I+66,I+74)=ACLAD1*X*T(I+74)**3+X1*CK(4,I+66,I+74)*Y1018
03550 A(I+74,I+66)=ACLAD1*X*T(I+66)**3+X1*CK(4,I+74,I+66)*Y1018
03560 A(I+66,I+123)=A3FUEL*X*(E1*T(I+123)**3+CK(4,I+66,I+123)/GAP1A)

```

```

03570 A(I+123,I+66)=A3FUEL*X*(E1*T(I+66)**3+CK(4,I+123,I+66)/GAP1A)
03580 A(I+82,I+90)=ACLAD2*X*E2*T(I+90)**3+CK(4,I+82,I+90)*Y2634
03590 A(I+90,I+82)=ACLAD2*X*E2*T(I+82)**3+CK(4,I+90,I+82)*Y2634
03600 A(I+132,I+149)=A2FUEL*X*(E2*T(I+149)**3+CK(4,I+132,I+149)/GAP2A)
03610 A(I+149,I+132)=A2FUEL*X*(E2*T(I+132)**3+CK(4,I+149,I+132)/GAP2A)
03620 (I10,I+157)=AFUE*X*E2*T(I157)**3+CK(4,I+140,I+157)/GAP2A)
03630 A(I+17,I14)=3FUEL*X*(E*T(I+140)**3+CK(4,I+157,I+140)/GAP2A)
03650 110 CONTINUE
03660
0670C LOAD NON-LINEAR TERMS INTO MATRIX DGNAL
0680
03690 A(5,58)=A(58,58)-A(115,58)
03700 A(115,115)=A(115,115)-A(58,115)
03710 A(132,132)=A(132,132)-A(149,132)
03720 A(149,149)=A(149,149)-A(132,149)
03730C
03740C
03750 DO 120 I=1,8
03760 A(I+9,I+9)=A(I+9,I+9)-A(I+17,I+9)
03770 A(I+17,I+17)=A(I+17,I+17)-A(I+9,I+17)
03780 A(I+25,I+25)=A(I+25,I+25)-A(I+33,I+25)
03790 A(I+33,I+33)=A(I+33,I+33)-A(I+25,I+33)
03800 A(I+58,I+58)=A(I+58,I+58)-A(I+115,I+58)
03810 A(I+115,I+115)=A(I+115,I+115)-A(I+58,I+115)
03820 A(I+66,I+66)=A(I+66,I+66)-A(I+74,I+66)
03830 A(I+74,I+74)=A(I+74,I+74)-A(I+66,I+74)
03840 A(I+66,I+66)=A(I+66,I+66)-A(I+123,I+66)
03850 A(I+123,I+123)=A(I+123,I+123)-A(I+66,I+123)
03860 A(I+82,I+82)=A(I+82,I+82)-A(I+90,I+82)
03870 A(I+90,I+90)=A(I+90,I+90)-A(I+82,I+90)
03880 A(I+12,I+32)=A(I+132,I+132)-A(I+149,I+132)
03890 A(I+149,I+149)=A(I+149,I+149)-A(I+132,I+149)
03900 A(I+140,I+140)=A(I+140,I+140)-A(I+157,I+140)
03910 A(I+157,I+157)=A(I+157,I+157)-A(I+140,I+157)
03920 120 CONTINUE
03930C
03940C EFINE B ARI
03950C
03960 B(1)=-A1FUEL*Q1/2.
0397 B(58)B(1)
03980C
03990 DO 130 I=2,17
04000 AB=A2FUEL
04010 IF(I.GT.9)AB=A3FUEL
04020 K=I-1
04030 IF(I.GT.9)K=I-9
04040 B(I)=-AB*Q1*THETA(K)/TWOPI
04050 B(I+57)=B(I)
04060 130 CONTINUE
04070C
04080C DEFINE REFERENCE NODES AND CORRESPONDING MATRIX DIAGONAL
04090C
04100 DO 140 I=INODE,JNODE
04110 B(I)=-TREF
04120 A(I,I)=A(I,I)-ANODE(I-KNODE)
04130 140 CONTINUE
04140C
04150C CALL MATRIX SOLVER
04160C
04170 CALL LSYSL(A,B,N)
04180C
04190C
04200 N2=N2+1
04210 Z1=0.
04220C
04230C COMPUTE SQUARE OF ERROR...AND NEW TEMPERATURE
04240C
04250 DO 150 I=1,N
04260 Z2=B(I)-T(I)
04270 Z1=Z1+Z2*Z2
04280 T(I)=B(I)
04290 150 CONTINUE

```

```

04300C
04310 IF(N2.GT.N1)GO TO 170
04320C
04330C    PRECISION TEST
04340C
04350 IF(SQRT(Z1).GT.TLIMIT)GO TO 30
04360C
04370C
04380C    SET PRECISION LIMIT IN DECRAMENTS
04390C
04400 IF(ITIMES.EQ.0)TLIMIT=TLIMIT/10.
04410 IF(ITIMES.EQ.1)TLIMIT=TLIMIT/10.
04420 IF(ITIMES.GT.3.AND.(N2-N3).LE.10)TLIMIT=XTLIMIT
04430 N3=N2
04440C
04450C    INCREMENT SOLUTION COUNTER
04460C
04470 ITIMES=ITIMES+1
04480 PRINT,ITIMES,N2
04490C
04500C    TEST FOR FINAL SOLUTION
04510C
04520 IF(SQRT(Z1).LT.XTLIMIT)GO TO 170
04530C
04540C    COMPUTE NEW REFERENCE NODES" DIAGONAL CONSTANTS
04550C
04560 DO 160 I=INODE,JNODE
04570 ANODE(I-KNODE)=ANODE(I-KNODE)*T(I)/TREF
04580 160 CONTINUE
04590C
04600C
04610 GO TO 30
04620C
04630C
04640 170 CONTINUE
04650C
04660C
04670C
04680 PW=P*FLOAT(M)
04690 DPAD=2.*RPAD
04700 PRINT 900,P,PW,RPAD,DPAD
04710 PRINT 900,P,DFUEL,AL,ALD
04720 PRINT 900,ANODE
04730C
04740C
04750C
04760 DO 180 I=1,N
04770 CENT=T(I)-273.15
04780 FF=9.*CENT/5.+32.
04790 PRINT,I,T(I),CENT,FF
04800 180 CONTINUE
04810C
04820 900 FORMAT(4(E18.12,2X))
04830C
04840 STOP
04850 END

```