A Two dimensional two fluid model for sodium boiling in LMBFR fuel assemblie. Vol. 2

Thesis (Doctorate) - MIT

Mario Roberto Granziera

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusion

A two dimensional computer code for the simulation of sodium boiling transients was developed using the two fluid model of conservation equations. A semi-implicit numerical differencing scheme, capable of handling the problems associated with the ill-posedness implied by the complex characteristic roots of the two fluid model was used, which took advantage of the dumping effect of the exchange terms. The stability of the method was demonstrated theoretically in Section 2.5 and also by the practical results obtained with the model, shown in Chapter 4. The stability of the model imposes an upper limit on the time step size, which is related to the mesh spacing the phase velocity by the expression

$$\Delta t < \max [\Delta z/\mu_2, \Delta r/\mu r]$$

Of particular interest in the development of the model was the identification of the numerical problem used by the strong disparity between the axial and radial dimensions of fuel assemblies used in the current design of Liquid Metal Fast Breeder Reactors. A solution to this problem was found, which used the particular geometry of fuel assemblies to its advantage, reducing drastically the computation time.

Most of the constitutive equations incorporated in the model were

obtained through previous work. In general, adequate models were found for most equations, but for a few of them no satisfactory correlations could be produced. These models involve areas of the sodium technology not yet fully understood, and a substantial effort of development must be done in these areas. These models are identified and discussed in the recommendations of this work.

The models and methods of this work were incorporated into the computer program called NATOF-2D. With this program three series of experiments were simulated in order to demonstrate the model capabilities. The results of this simulation, which were presented in Chapter 4 showed good agreement with the experimental results obtained in the tests. One important capability demonstrated in these simulations was the ability of the model to represent the most severe boiling conditions, including flow reversal.

5.2 Recommendations

A word of caution must be said to the eventual users of NATOF-2D. The purpose of this work was to develop a numerical framework capable of solving the set of conservation equations of fluid flow under severe conditions of transient sodium boiling. In this way, most of the effort put into the work was dedicated to developing and organizing the numerical methods and models for solving this set of equations.

Of course the system of equations of fluid flow is not closed unless the constitutive relations describing the interaction of the fluids with the structural components and with themselves is provided, and a

set of constitutive equations were incorporated into NATOF-2D.

Some judgment was exercised in order to select constitutive equations representative of the sodium beahvior, especially those characterizing the explosive volume change associated with sodim boiling at low pressure. This part of the code development was treated as complimentary to the numerical model construction. Therefore, the constitutive models may not be as realistic as the correct representation of sodium boiling in LMFBR fuel assemblies would require, and the overall results of simulations with NATOF-2D may be improved by the eventual improvement of some of the constitutive models incorporated in the code. Thus, this word of caution.

The relatively superficial treatment of the constitutive models is not incidental. Only recently did the interest in LMFBR safety reach the point where extensive investigation of sodium boiling became justified, and a substantial amount of research is yet to be done. Therefore, the present status of knowledge of the physical phenomena associated with sodium boiling does not lead immediately to significantly accurate models of the constitutive equations involved in sodium boiling. The task of developing these models is not a simple one, requiring a considerable effort in theoretical analysis and experimental work, well beyond the scope of a one person thesis.

But if NATOF-2D cannot claim to be a complete analytical model for sodium boiling simulation, because of the uncertainties contained in the constitutive models, it is an invaluable tool for the development of these models, where they can be implemented and tested against experimental results.

One of the most important benefits which NATOF-2D can provide to the development of sodium boiling is to identify, by the execution of sensitivity analysis, those constitutive models which affect most of the overall results, thus directing the research effort of sodium boiling to the directions which will lead to more fruitful results.

From the experience we had with NATOF-2D calculations, by far the most important model affecting the end results of sodium boiling simulation is the one for the interphase mass exchange rate (which unfortunately is the one that showed the widest disagreement between authors). Therefore, we recommend as a first step in the continuation of the work presented here that a substantial effort be made in developing a dependable model for the interphase mass exchange rate.

Of the same magnitude in importance is the two phase heat transfer coefficients. Here again the presently available models are few and incomplete. Thus a theoretical and experimental work in this area is recommended, in order to acquire a thorough understanding of the sodium boiling curve.

Another area which could be the object of future investigation is the one related to the interphase heat transfer. Although the direct effect of this exchange term on the overall results is not very marked, the relatively simple model incorporated in NATOF-2D could be replaced by a more refined one. The close relationship between this exchange term and the two previously mentioned would make this model a natural by product of the development of the above-mentioned ones.

REFERENCES

1. Griffith, J. D. , "Safety Considerations in Commercial Fast Breeder Reactor Plant Design," MIT summer course — Fast Breeder Reactor Safety, July 1977

2. Hinkle, W. D., LMFBR "Safety and Sodium Boiling, A State of the Art Report," draft, MIT, December 1977

3. Chawla, T. C., and Fauske, H. K., "On the Incoherence in Subassembly Voiding in FTR and its Possible Effects on the Loss of Flow Accident Sequence," Trans Am Nucl Soc 17, p285, November 1973

4. Rowe, D.S., COBRA III-C: "A Digital Computer Program for Steady-State and Transient Thermo-hydraulic Analysis for Rod Bundle Nuclear Fuel Elements," BNWL 1695, March 1973

5. Bohl, W.R., et al, "An Analysis of Transient Undercooling and Transient Overpower Accidents without Scram in the Clinch River Breeder Reactor," ANL/RAS 75-29, July 1975

6. Miao, C., and Theofanous, "A Numerical Simulation of the Two-Dimensional Boiling (Voiding) in LMFBR Fuel Subassemblies," Purdue University

7. Sha, W.T., et al, COMMIX1: "A Three Dimensional Transient Single-Phase Component Computer Program for Thermal Hydraulic Analysis," MIREG-CR 0415, ANL-77-96, September 1978

8. Grand, D., and Basque, G., "Two-Dimensional Calculation of Sodium Boiling in Sub-Assemblies," Service des Transferts Thermique, Centre d'Etude Nucleaires de Grenoble

9. Stewart, H.B., "Fractional Step Methods for Thermo-hydraulic Calculation, "Brookhaven National Laboratory, March 1980

10. Shih, T.A., "The SOBOIL Program, A Transient, Multichannel Two Phase Flow Model for Analysis of Sodium Boiling in LMFBR Fuel Assemblies," Technical Note ST-TN-79008, March 1979

11. Hinkle, W.D., et al, "MIT Sodium Boiling Project FY 1979 Interim Report," draft, 1979

12. Shah, et al, "A Numerical Procedure for Calculating Steady/Unsteady Single-Phase/Two-Phase Three-Dimensional Fluid Flow with Heat Transfer," ANL-CT-79-31, June 1979

13. Ishii, M., "One-Dimensional Drift-Flux Model and Constitutive Equations for Relative Motion Between Phases in Various Two-Phase Flow Regimes," ANL 77-47, October 1977

14. Grolmes, M.A., and Henry, R.E., "Heat Transfer in Nuclear Power Reactors, Part II: Safety of Liquid Metal Cooled Fast Breeder Reactors," Argonne National Laboratory

15. Weber, M., et al, "Reactor Development Program," Progress Report ANL-RDP-78, December 1978.

16. Carter, J.C., et al, SAS1A, "A Computer Code for the Analysis of Fast Reactor Power and Flow Transients," ANL7607, October 1970

17. Dunn, F.E., "The SAS3A LMFBR Accident Analysis Computer Code," ANL/RAS 75-17, 1975

18. The Separate Flow Model of Two Phase Flow, EPRI NP275, December 1976

19. Agrawal, A.K., et al, "Simulation of Transients in Liquid Metal Fast Breeder Reactor Systems," Nuclear Science and Engineering, Vol 64, 480-491, 1977

20. Boure, J.A., and Latrobe, A., "On Well-Posedness of Two Phase Flow Problems," 16th National Heat Transfer Conference, August 1976

21. Reed, Wm. H., and Stewart, H.B., "THERMIT, A Computer Program For Three-Dimensional Thermal-Hydraulic Analysis of Light Water Reactor Cores, MIT, 1978

22. Rivard, W.C. and Torrey, M.D., "Numerical Calculation of Flashing from Long Pipers Using a Two Field Model," LA6104-MS, Los Alamos, November 1975

23. Chen, J.C., "A Proposed Mechanism and Method of Correlation for Convective Boiling Heat Transfer with Liquid Metals," BNL 7319, August 1973

24. Chao, B.T., Sha, W.T., and Soo, S.L., "On Inertial Coupling in Dynamic Equations of Components in a Mixture," Int J Multiphase Flow, Vol 4, pp219-223, 1978

25. Fabic, S., "Computer Codes in Water Reactor Safety: Problems in Modeling of Loss-of-Coolant Accident," I Mech E Conference, Manchester, 13-15 September 1977

26. Chawla, T.C., and Ishii, M., "Equations of Motion for Two-Phase Flow in a Pin Bundle of a Nuclear Reactor," J Heat Mass Transfer, Vol 21, pp1057-1068, 1978

27. Ramshaw, J.D., and Trapp, J.A., "Characteristics, Stability and Short Wavelength Phenomena in Two-Phase Flow Equations Systems," Nuclear Science and Engineering, Vol 66, pp93-102, 1978

28. Murray, S.E., and Smith, L.L., "Two Dimensional Sodium Voiding Analysis with SIMMER-I," LA-NUREG-6342-PR, June 1977

29. Lyczkowsky, R.W., and Solbrig, C.W., "Calculation of the Governing Equations for Seriated Unequal Velocity, Equal Temperature Two-Phase Flow, Nat Heat Transfer Conference 1977

30. Jones, O.C., Jr., and Pradip, S., "Non-Equilibrium Aspects of Water Reactor Safety," Brookhaven National Laboratory

31. Nigmatulin, R.I., "Equations of Hydromechanics and Compression Shock in Two Velocity and Two Temperature Continuum with Phase Transformation," Fluid Dynamics, Vol 2, No 5, 1967

32. Rohsenow, W.M., and Sukhatme, S.P., "Condensation," Massachusetts Institute of Technology

33. Brinkmann, K.J. and deVries, J.E., "Survey of Local Boiling Investigations in Sodium at ECN-Petten," Netherlands Energy Research Foundation ECN

34. Garrison, P.W., "Superheat Simulation Requirements for the Next Generation of LMFBR Codes," Oak Ridge National Laboratory, March 1979

35. Basque, G., Grand, D., and Menant, B., "Theoretical Analysis and Experimental Evidence of Three Types of Thermohydraulic Incoherence in Undisturbed Cluster Geometry," Kalsruhe, 1979

36. Baker, A.N., et al, "SLSF W-1 Experiment Test Predictions," GEFR 00047-9(L), December 1977

37. Knight, D.D., "SLSF W-1 LOD1 Experiments Preliminary Evaluation Data," ST-TN-80015, October 1979

38. Henderson, J.M., "Sodium Loop Safety Facility Test Plan HEDL W-1 SLSF Experiment," Hanford Engineering Development Laboratory, September 1978

39. Thompson, D.H., et al, "SLSF In-Reactor Experiment P3A," Interim Post Test Report, Argonne National Laboratory, November 1977

40. Kraft, T.E., et al, "Simulations of an Unprotected Loss-of-Flow Accident with a 37-Pin Bundle in the Sodium Loop Safety Facility," Argonne National Laboratory

41. Collier, J.G., Convective Boiling and Condensation, McGraw-Hill, United Kingdom, 1972

42. Clark, M., Jr., and Hansen, K.F., Numerical Methods of Reactor Analysis, Academic Press, New York, 1964

43. Richtmeyer, R.D., and Morton, K.W., Differential Methods for Initial Value Problems, Interscience, New York, 1967

44. Wallis, G.B., One Dimensional Two Phase Flow, McGraw-Hill, New York, 1969

45. Courant, R., and Hilbert, D., Methods of Mathematical Physics, Interscience, New York 1962

46. Golden, G.H., and Tokar, J.V., Thermophysical Properties of Sodium, ANL 7323, August 1967

47. Van Wylen, G.J., and Sonntag, R.E., Fundamentals of Classical Thermodynamics, John Wiley & Sons, New York, 1973

48. Kays, W., and London, A.L., Compact Heat Exchanges, McGraw-Hill, New York, 1964

49. Gunter, A.Y., and Shaw, W.A., A General Correlation of Friction Factors for Various Types of Surfaces in Crossflow, ASME Transactions, 67, pp643-660, 1945

50. Autruffe, M.A., Theoretical Study of Thermohydraulic Phenomena for LMFBR Accident Analysis, MIT thesis, September 1978

51. Stewart, H.B., "Stability of Two Phase Flow Calculations Using Two Fluid Model," Journal of Computational Physics, Vol 33, No 2, November 1979

52. Kaiser, A., and Peppler, W., "Sodium Boiling Experiments in an Annular Test Section under Flow Rundown Conditions," KFK 2389, March 1977

53. El Wakil, M.M., "Nuclear Heat Transport," International Textbook Company, 1971

54. Fink, J.K., and Leibowitz, L., "Thermophysical Properties of Sodium," ANL-CEN-RSD-79-1, May 1979

55. Gantmacher, F.R., "The Theory of Matrices," Chelesea Publishing Company, 1977

56. Varga, R.S., "Matrix Iterative Analysis," Prentice Hall, 1962

57. Rohsenow, W.M., and Choi, H., "Heat, Mass and Momentum Transfer," Prentice Hall, 1961

58. Poter, M.C., and Foss, J.F., "Fluid Mechanics," Ronald Press, 1975

59. "CRC Handbook of Chemistry and Physics," 58th Edition, CRC Press, 1978

60. Thompson,D.H.,et al."SLSF In-Reactor Experiment P3A - Interim Posttest Report",ANL/RAS 77-48,November 1977

APPENDIX A - NATOF-2D INPUT DATA MANUAL

In this section the user supplied information necessary to operate NATOF-2D is presented. Before showing the description of the input cards, it is useful to review the array structure of the code. Figure A.1 shows an example of a full assembly and the corresponding cell arrangement in a r-z plane. Quantities appearing in this figure are:

NI = number of mesh cells in the axial direction. It includes two fictitious half-cells in the top and bottom of fuel assembly.

NJ = number of mesh cells in the radial direction.

All dimensioned variables appear in the program with only one index, therefore a single number identifies each cell in full assembly. The cells are numbered from bottom to top and radially from center to hex can.

Figure A.2 shows a cross section of the fuel assembly indicating the numbering of the fuel pins. Fuel pin rows are numbered from center to hex can, and the boundary between cells is indicated by the row number where this boundary lies.

Figure A.3 shows schematically the cell arrangement for the fuel pin heat conduction. The quantities describing this cell arrangement are:

NCF = number of mesh cells in fuel.

NCLD = number of mesh cells in clad.

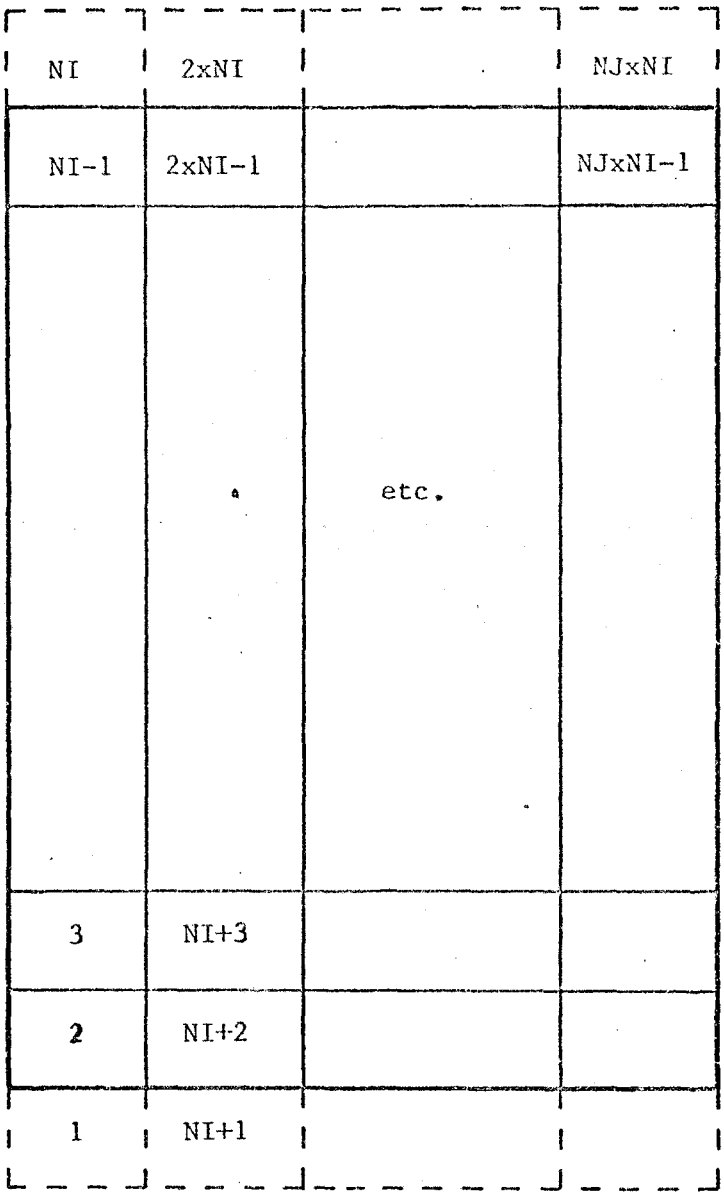

Figure A1. Cell Arrangement in the R-Z Plane

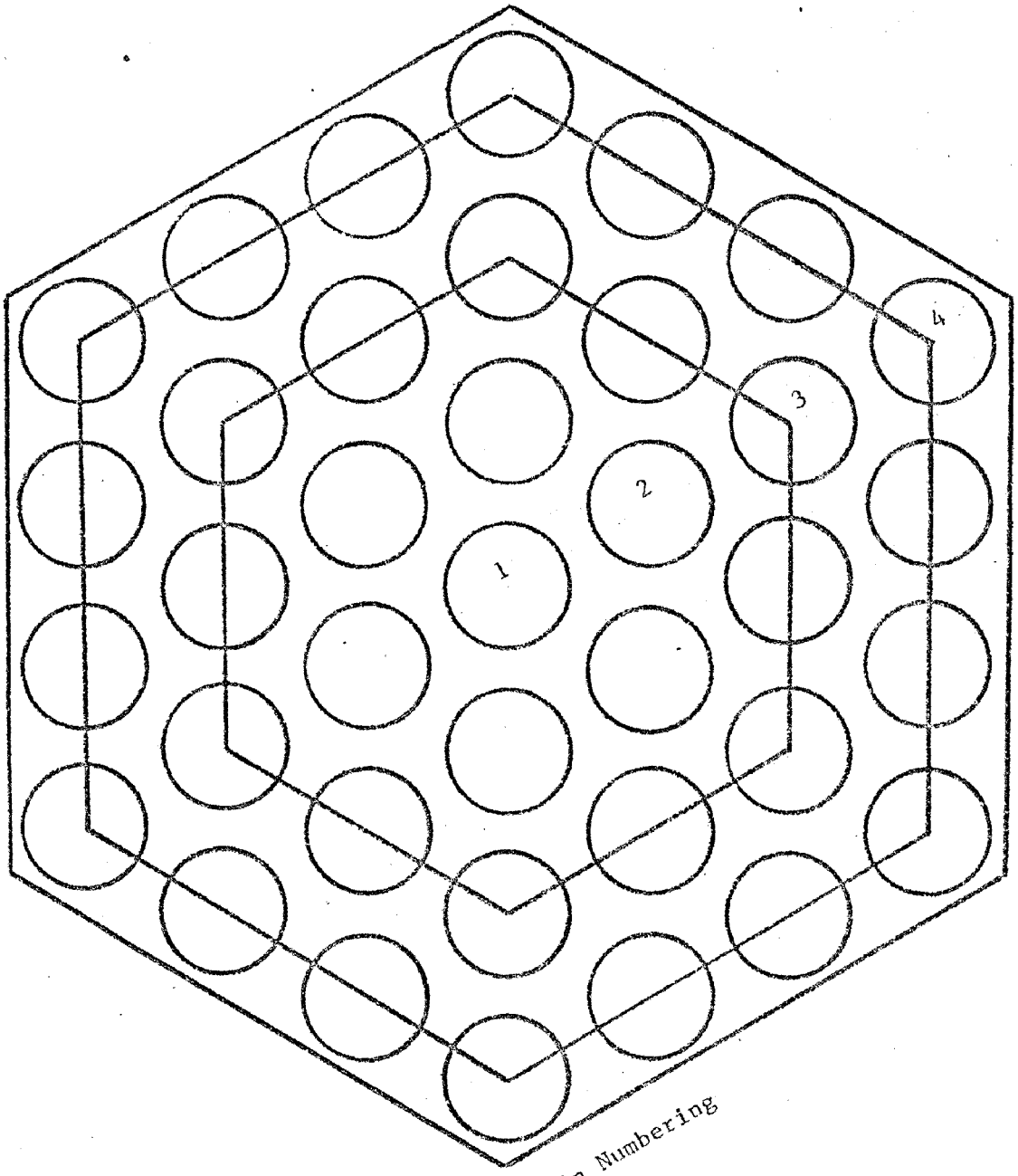

Figure A.2 Fuel Pin Numbering

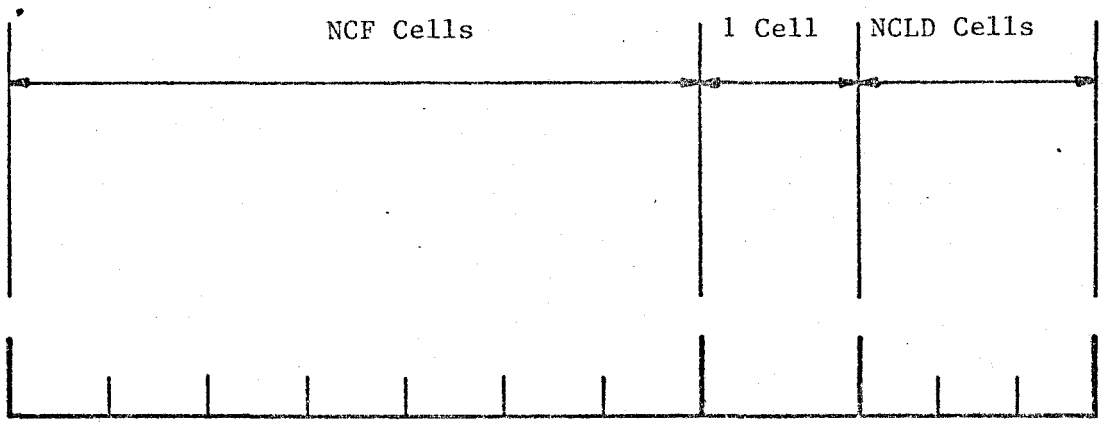

Figure A3. Cell Arrangement for Fuel Pin Heat Conduction

A single cell is assumed by the code for the gap between fuel and clad.

Following is a presentation of the sequence of cards in the input data. Following the list of variables, in parenthesis, is the corresponding format for these variables.

1. General Description of the Problem

1st CARD: NI, NJ, NCF, NCLD (4I5)

NI = number of mesh cells in axial direction

NJ = number of mesh cells in radial direction

NCF = number of mesh cells in fuel

NCLD = number of mesh cells in clad

2nd CARD: NSET, TSET (I5, E15.4)

This card contains information which controls the printed output. the code will print NSET times the flow map, with a time interval TSET. This card can be repeated up to 49 times, so that the time interval between prints can be varied to reflect the desired degree of information at each time. Following these cards, a card containing only zeros in the position corresponding to NSET must be placed, to indicate the end of this subset.

3rd CARD: ITM, IGAUSS, DTMAX, EPS1, EPS2 (2I10, 3E15.9)

ITM = maximum number of iterations in the Newton iterative solution.

IGAUSS = maximum number of iterations in the pressure problem solution.

DTMAX = maximum value for the time step increment.

EPS1 = convergence criterion for the Newton iteration.

EPS2 = convergence criterion for the pressure problem.

EPS1 and EPS2 are criteria on the absolute value of the pressure. Their unit is N/m2.

2. Boundary Conditions

The next group of cards contains information governing the boundary conditions of the problem as a function of time. The simulation time is divided in up to 50 segments in which different functions can be prescribed for the boundary conditions. For a generic time segment L, the formulas used by the program for the boundary condition are:

$X = (X_1(L)*DTIME + X_2(L))*\exp(OMX(L)*DTIME) + X_3(L)$

where:

DTIME = TIME - TB(L-1)

L = Index of current time segment

TB(L) = Time at the end of segment L

X_1, X_2, X_3, OMX = Input parameters

and X stands for:

PNB = Pressure at the bottom of fuel assembly (N/m^2)

PNT = Pressure at the top of fuel assembly (N/m^2)

ALB = Void fraction at the inlet of fuel assembly.

TVB = Vapor temperature at inlet (°K).

TLB = Liquid temperature at inlet (°K).

HNW = Power density in fuel pins (W/m^3)

In order to save time, the code has an option to eliminate the exponential part in the formula to calculate the boundary condition. Thus, whenever the logical parameter LP is .TRUE., the boundary conditions are calculated as:

$$X = X_1(L)*DTIME + X_2(L)$$

1st CARD:	LP, TB	(L1, F15.5)
2nd CARD:	PNB1, PND2, PNB3, OMP	(4E15.9)
3rd CARD:	PNT1, PNT2, PNT3, OMT	(4E15.9)
4th CARD:	ALB1, ALB2, ALB3, OMA	(4E15.9)
5th CARD:	TVB1, TVB2, TVB3, OMV	(4E15.9)
6th CARD:	TLB1, TLB2, TLB3, OML	(4E15.9)
7th CARD:	HNB1, HNB2, HNB3, OMH	(4E15.9)

This group of seven cards can be repeated for as much as the number of segments desired. To indicate the end of this subset, a card containing only a 'F' in the first position must be placed following the data.

3. Geometric Description of the Problem

1st CARD: NROW, PITCH, D, E (I5, 3E15.9)

NROW = Number of rows of fuel pins in fuel assembly.

PITCH = Distance between fuel pin centerlines (m).

D = Fuel pin diameter (m).

E = Minimum distance between fuel pin surface and hex can wall (m).

(see Figure A.2)

2nd CARD: N(J), J = 1, 20 (20I4)

N(J) is the row number where the boundary between cell J and cell J + 1 lies.

(see Figure A.2)

3rd CARD: LDATA, DZ(K) (L1, 5E15.9)

In this group of cards the axial mesh spacing DZ are written sequentially from 1 to NI, five per card. The logical parameter LDATA must have a .TRUE. value in each card where DZ is written. Following this group of cards, a card containing an 'F' in the first position must be placed to indicate the end of this set of data.

4th CARD: LDATA, CAN(K) (L1, 5E15.9)

The same arrangement of the previous group of cards.

CAN = Heat capacity of the hex can per unit area, for each axial mesh cell ($J/m^2°K$). There must be one value for each axial mesh cell.

5th CARD: LDATA, SHAPE(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

SHAPE = Power density shape in fuel assembly. There must be one value of SHAPE for each mesh cell in fuel assembly.

6th CARD: LDATA, SPPD(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

SPPD = Spacer pressure drop. There must be one value of SPPD for each mesh cell in fuel assembly. The code will treat the spacer pressure drop as:

$\Delta p = SPPD * \frac{\rho U^2}{2}$

7th CARD: LDATA, PPP(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

PPP = Radial power profile inside fuel pin. There must be one value of PPP for each fuel pin mesh cell, including gap and clad (i.e., there is NCF + 1 + NCLD values).

The power density at each fuel pin mesh cell will be the product of the power density specified in the boundary conditions, multiplied by the value of SHAPE for the corresponding fuel assembly mesh cell, multiplied by the value of PPP for the corresponding fuel pin mesh cell.

8th CARD: AD, APU, DIL (3E15.9)

AD = Fraction of theoretical density of fuel.

APU = Fraction of plutonium in full.

DIL = Fraction of helium in gap composition.

9th CARD: LPLNM(I), I = 1, NI (39I2)

LPLNM is an integer which indicates the axial composition of fuel pin. LPLNM = 0 indicates gas composition (for upper plenum). LPLNM = 1 indicates mixed oxide U,PuO_2. There must be one value of LPLNM for each axial node.

10th CARD: RADR, THC, THG (3E15.9)

RADR = Fuel pin outside radius (m).

THC = Clad thickness (m).

THG = Gap thickness (m).

4. Initial Conditions

1st CARD: LSS, TINIT (L1, E15.9)

LSS is a logical parameter to indicate steady-state or transient problem.

LSS = .FALSE. indicates transient problem.

LSS = .TRUE. indicates steady-state problem.

In case LSS is .TRUE., the remaining initial condition input data resume to the next card:

2nd CARD: PIN, POUT, TIN, TAV (4E15.9)

PIN = Pressure at fuel assembly inlet (N/m^2)

POUT = Pressure at fuel assembly outlet (N/m^2)

TIN = Inlet liquid temperature (°K)

TAV = An estimate of the average temperature in fuel assembly (°K)

In case LSS = .FALSE., the next cards follow:

2nd CARD: KO, TV, TL, P. ALFA (I5, 4E15.9)

3rd CARD: KO, UVZ, ULZ, UVR, ULR (I5, 4E15.9)

KO is the cell number. It appears in both cards to put a check in the input data. Each pair of cards correspond to the same mesh cell. The group is to be repeated for as many as the number of mesh cells.

TV = Vapor temperature (°K)

TL = Liquid temperature (°K)

P = Pressure

ALFA = Void fraction

UVZ = Axial vapor velocity (m/sec)

UVR = Radial vapor velocity (m/sec)

ULR = Radial liquid velocity (m/sec)

4th CARD: LDATA, TR(K) (L1, 5E15.9)

The same arrangement as the group of cards for DZ.

TR = Fuel pin temperature (°K).

This array must contain one value for each fuel pin mesh cell. The values of TR are ordered as:

TR(1) = Fuel centerline temperature at cell number 1.

TR(NCF + 1 + NCLD) = Surface clad temperature at cell number 1.

TR(NCF + 1 + NCLD + 1) = Fuel centerline temperature at cell number 2.

etc.

5th CARD: LDATA, TCAN(K) (L1, 5E15.9)

The same arrangement as the previous group of cards.

TCAN = Hex can initial temperature (°K).

There must be one value of TCAN for each axial node.

APPENDIX B

NATOF - 2D Programming Information

When NATOF-2D was programmed, it was recognized that the field of sodium boiling is presently the subject of a large effort of research, and therefore it can be expected that in the future this research will produce better correlations for the constitutive laws governing the sodium two-phase flow. In order to make changes in the program as easy as possible, NATOF-2D was programmed with its subroutines in a modular structure, particularly the parts of the program dealing with the constitutive laws.

In this way, the programmer working on modification of one particular subroutine does not have to worry about the rest of the program, provided the expressions introduced in that subroutine meet the requirements of consistency of the derivatives with respect to new time variables, which were discussed in chapter 2.

Following is a description of NATOF-2D subroutines, their functions and structure. The reader is referred to figure B1, which shows the structure of NATOF-2D.

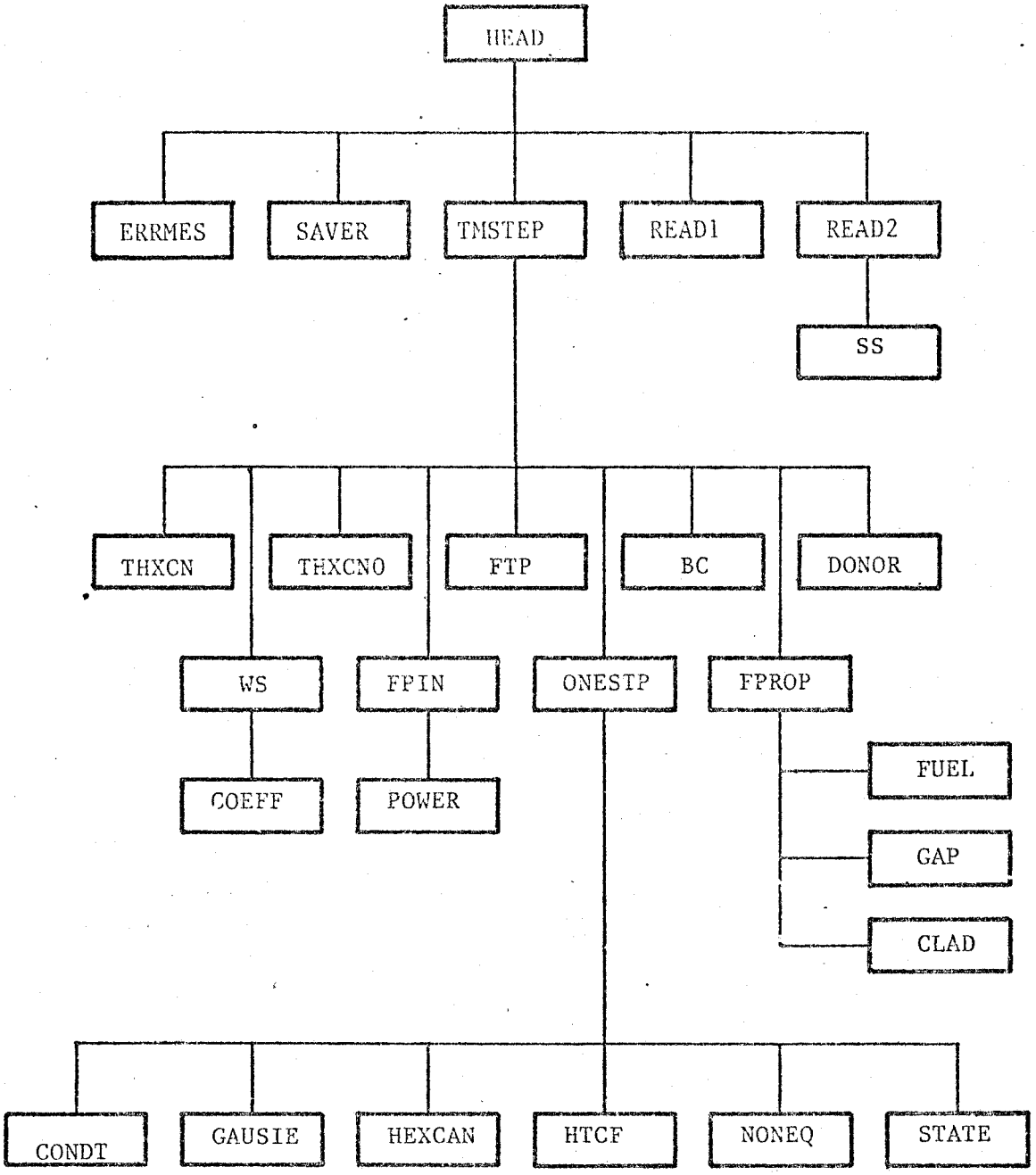

Figure B1. NATOF-2D Subroutine Structure

Main:

The main program's only function is to allocate memory storage space for the dimensioned arrays and transfer the control of the program to subroutine HEAD.

All arrays whose dimensions are a function of the number of mesh cells are placed within a single array ORBI. Individual arrays are located by pointers which determine the first element of each array. These pointers are grouped into the integer array M, and the correlation of the pointer to the variable is as following:

M(1)	=	P	=	New time, pressure, cell centered
M(2)	=	PO	=	Old time, pressure, cell centered
M(3)	=	TV	=	Vapor temperature, new time, cell centered
M(4)	=	TVO	=	Vapor temperature, old time, cell centered
M(5)	=	TL	=	Liquid temperature, new time, cell centered
M(6)	=	TLO	=	Liquid temperature, old time, cell centered
M(7)	=	ALFAN	=	Void fraction, new time, cell centered
M(8)	=	ALFAO	=	Void fraction, old time, cell centered
M(9)	=	ALFAZ	=	Void fraction, axial face centered
M(10)	=	ALFAR	=	Void fraction radial face centered
M(11)	=	RHOV	=	Vapor density, cell centered
M(12)	=	RHOL	=	Liquid density, cell centered

M(13) = RHOVZ = Vapor density, axial face centered

M(14) = RHOLZ = Liquid density, axial face centered

M(15) = RHOVR = Vapor density, radial face centered

M(16) = RHOLR = Liquid density, radial face centered

M(17) = EV = Vapor internal energy, cell centered

M(18) = EL = Liquid internal energy, cell centered

M(19) = EVZ = Vapor internal energy, axial face centered

M(20) = ELZ = Liquid internal energy, axial face centered

M(21) = EVR = Vapor internal energy, radial face centered

M(22) = ELR = Liquid internal energy, radial face centered

M(23) = UVZN = Axial vapor velocity, new time, axial face centered

M(24) = ULZN = Axial liquid velocity, new time, axial face centered

M(25) = UVRN = Radial vapor velocity, new time, radial face centered

M(26) = ULRN = Radial liquid velocity, new time, radial face centered

M(27) = UVZO = Axial vapor velocity, old time, axial face centered

M(28) = ULZO = Axial liquid velocity, old time, axial face centered

M(29) = UVRO = Radial vapor velocity, old time, radial face centered

M(30) = ULRO = Radial liquid velocity, old time, radial face centered

M(31) = UVRZ = Radial vapor velocity, axial face centered

M(32) = ULRZ = Radial liquid velocity, axial face centered

M(33) = UVZR = Axial vapor velocity, radial face centered

M(34) = ULZR = Axial liquid velocity, radial race centered

M(35) to M(62) = Implicit terms for the conservation equations

M(63) = DH = Axial flow hydraulic diameter

M(64) = DHR = Radial flow hydraulic diameter

M(65) = DV = Fuel pin specific surface area

M(66) = QSI = Maximum-to-average radial velocity coefficient

M(67) = TS = Saturation temperature, new time

M(68) = TW = Fuel pin wall temperature, new time

M(69) = DTW = Increment in heat transfer for unit increment in TW

M(70) = HCONV = Vapor heat transfer coefficient

M(71) = HCONL = Macroscopic liquid heat transfer coefficient

M(72) = HNB = Microscopic liquid heat transfer coefficient

M(73) to M(79) = Coefficients for the pressure problem

M(80) = TR = Fuel pin temperature

M(81) = DTR = Auxiliary array for fuel pin heat conduction

M(82) = TWO = Fuel pin wall temperature, old time

M(83) to M(89) = Auxiliary arrays

M(90) = SPPD = Localized pressure drop coefficienct

M(91) = TCAN = Hex can temperature

The storage space required by the array ORBI is given in double precision storage word by the formula:

$$[135 + 2(NCF + NCLD)]NI.NJ$$

HEAD: — Defines the pointers of array ORBI
— Controls the duration of the run
— Controls the printouts

READ 1: — Reads arrays' dimensions

READ 2: — Reads all other information
— Writes in FILE07 the input data for a restart
— Calculate parameters which will remain constant throughout the problem

SS: — Performs an initial guess for the steady-state problem

TMSTEP: — Advances one time step
— Controls convergence of the Newton iteration
— Controls time step size. The time step is always kept below the connective limit. If an instability occurs during the run, such as non-convergence of the iterative procedures or a variable outside range of validity, TMSTEP reduces the time step size by a factor of ten and the run is resumed. If the difficulty is removed, the time step will be increased slowly towards the convective limit again. If after three time step reductions the instability still persists, an error message will be printed and the execution terminated.

DONOR: — Transfers all centered quantities to face centered positions
— Calculates explicit terms in momentum equation

WS: — Calculates explicit terms for mass and energy equations

ONESTP: — Performs one step of Newton iteration
— Calculates new values of implicit variables
— Checks variables against range of validity

COEFF: — Calculates momentum exchange coefficients

BC: — Calculates boundary conditions as a function of time

HTCF: — Calculates heat transfer coefficients

STATE: — Calculates sodium thermodynamic properties and its derivatives. The code stability imposes two requirements on the expressions for the sodium functions of state: the expressions for the densities must account for the pressure dependence which corresponds to a real, positive, finite sonic speed.
The expressions for the property derivatives with respect to new time variables must be the analytic or numerical derivative of the expressions of the properties (but not approximated expressions).

NONEQ: — Calculates the mass and energy exchange rates and its derivatives. The same requirement applied to the derivatives of the properties in STATE also applies here.

CONDT: — Calculates the heat transfer between fluid and fuel pin and its derivatives. The requirement concerning the derivatives described above also applies here.

HEXCAN: — Calculates the heat transfer between fluid and hexcan walls, and its derivatives. The requirement concerning the derivatives described above also applies here.

FPROP: — Finds the fuel pin transport properties

FUEL: — Transport properties of fuel

GAP: — Transport properties of gap

CLAD: — Transport properties of clad

FPIN: — Solves first part of heat conduction in fuel pin

FTP: — Solves second part of heat conduction in fuel pin

THXCN: — Solves the first part of hexcan heat conduction

THXCNO: — Solves the second part of hexcan heat conduction

POWER: — Calculates the power density as a function of time

GAUSIE: — Solves the pressure problem

ERRMES: — Prints error messages

SAVER: — Saves fluid flow variables at the end of run for eventual restart

Functions

CONDL	—	Liquid thermal conductivity as function of temperature
CONDV	—	Vapor thermal conductivity as function of temperature
CPL	—	Liquid specific heat as function of temperature
HFG	—	Enthalpy of vaporization as function of pressure
PRL	—	Liquid Prandtl number as function of temperature
PRV	—	Vapor Prandtl number as function of temperature
SAT	—	Saturation temperature as function of pressure
DTSDP	—	Pressure derivative of saturation temperature as function of pressure
SURTEN	—	Surface tension as function of temperature
VISCV	—	Vapor viscosity as function of temperature
VISCL	—	Liquid viscosity as function of temperature

APPENDIX C

NATOF - 2D I/O EXAMPLES

Fortran unit numbers for the data files are as follows:

5 is the standard input unit

6 is for the printed output

7 is the dump file to restart

After a successful run, the program creates in file 7 an input data set corresponding to an initial value problem starting at the time the last run was finished. This is particularly useful in generating a transient problem input data set, which requires a substantial amount of information for the initial conditions. In this way, a steady-state problem, which requires a relatively small amount of information, produces in file 7 the input data for the transient problem. The user must only change the cards which describe the boundary conditions, to represent the desired transient conditions, and the desired sequence of printouts.

Following is an example of the input data set for a steady state problem, a transient problem, and an example of the printed output. These examples were taken from the 217-pin simulation described in section 4.

STEADY-STATE INPUT DATA SET

EXAMPLE

```
   12    5    4    2
    20.150000000D+01
    00.000000000D+00
         8       5000.100000000D+000.100000000D-010.100000000D-03
T        4.40000
0.000000000D+000.675750000D+060.000000000D+000.000000000D+00
0.000000000D+000.277600000D+060.000000000D+000.000000000D+00
0.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00
0.000000000D+000.244508686D+100.000000000D+000.000000000D+00
T        0.00000
    90.726440000D-020.584200000D-020.711200000D-03
   5   7   8   9   0   0   0   0   0   0   0   0   0   0   0   0   0   0
T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
T0.182900000D+000.182900000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.850000000D+040.850000000D+040.850000000D+040.850000000D+040.850000000D+04
T0.850000000D+040.850000000D+040.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.677403000D+000.838702000D+000.100000000D+01
T0.838702000D+000.677403000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.677403000D+00
T0.838702000D+000.100000000D+010.838702000D+000.677403000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.677403000D+000.838702000D+000.100000000D+010.838702000D+00
T0.677403000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.677403000D+000.838702000D+00
T0.100000000D+010.838702000D+000.677403000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.677403000D+000.838702000D+000.100000000D+010.838702000D+000.677403000D+00
T
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.610000000D+020.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.610000000D+020.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.610000000D+020.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.610000000D+020.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.869000000D+02
```

```
T
T
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.100000000D+010.100000000D+010.100000000D+010.100000000D+010.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.954000000D+000.250000000D+000.900000000D+00
 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.0 0
0.292100000D-020.381000000D-030.600000000D-04
T0.000000000D+00
0.675750000D+060.277600000D+060.661140000D+030.900000000D+03
```

TRANSIENT INPUT DATA SET

EXAMPLE

```
   12    5    4    2
   350.100000000D+00
    00.000000000D+00
         8      10000.100000000D+000.100000000D-010.100000000D-03
T        0.50000
-.822080000D+060.675750000D+060.000000000D+000.000000000D+00
-.180800000D+060.277600000D+060.000000000D+000.000000000D+00
0.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00
0.000000000D+000.661140000D+030.000000000D+000.000000000D+00
0.000000000D+000.244508688D+100.000000000D+000.000000000D+00
T        3.5
0.000000000D+000.264710000D+06
0.000000000D+000.187200000D+06
0.000000000D+000.000000000D+00
0.000000000D+000.661140000D+03
0.000000000D+000.661140000D+03
0.000000000D+000.244508688D+10
T        4.0
0.806340000D+060.272560000D+06
0.176800000D+060.189200000D+06
0.000000000D+000.000000000D+00
0.000000000D+000.661140000D+03
0.000000000D+000.661140000D+03
-.464566510D+100.244508688D+10
T        9.0
0.000000000D+000.675750000D+06
0.000000000D+000.277600000D+06
0.000000000D+000.000000000D+00
0.000000000D+000.661140000D+03
0.000000000D+000.661140000D+03
0.000000000D+000.122254330D+09
T        0.00000
    90.726440000D-020.584200000D-020.711200000D-03
   5   7   8   9   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
T0.182900000D+000.182900000D+000.182900000D+000.182900000D+000.182900000D+00
T0.182900000D+000.182900000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.850000000D+040.850000000D+040.850000000D+040.850000000D+040.850000000D+04
T0.850000000D+040.950000000D+040.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.677403000D+000.838702000D+000.100000000D+01
T0.838702000D+000.677403000D+000.000000000D+000.000000000D+000.000000000D+00
```

```
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.677403000D+00
T0.838702000D+000.100000000D+010.838702000D+000.677403000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.677403000D+000.838702000D+000.100000000D+010.838702000D+00
T0.677403000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.677403000D+000.838702000D+00
T0.100000000D+010.838702000D+000.677403000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.677403000D+000.838702000D+000.100000000D+010.838702000D+000.677403000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.610000000D+020.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.610000000D+020.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.610000000D+020.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.610000000D+020.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.869000000D+02
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
T0.100000000D+010.100000000D+010.100000000D+010.100000000D+010.000000000D+00
T0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
F0.000000000D+000.000000000D+000.000000000D+000.000000000D+000.000000000D+00
0.954000000D+000.250000000D+000.900000000D+00
 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0.292100000D-020.381000000D-030.600000000D-04
F0.000000000D+00
    10.661140000D+030.661140000D+030.675750000D+060.000000000D+00
    10.645364280D+010.645364280D+010.000000000D+000.000000000D+00
    20.661139976D+030.661139976D+030.457953361D+060.000000000D+00
    20.645364280D+010.645364280D+010.944392050D-030.944392050D-03
    30.709018194D+030.709018194D+030.440294610D+060.000000000D+00
    30.644028454D+010.644028454D+010.624381194D-030.624381194D-03
    40.768816955D+030.768816955D+030.422317592D+060.000000000D+00
    40.651813704D+010.651813704D+010.360111639D-030.360111639D-03
    50.840612698D+030.840612698D+030.404311805D+060.000000000D+00
    50.662503059D+010.662503059D+010.268289802D-030.268289802D-03
    60.901049391D+030.901049391D+030.386078761D+060.000000000D+00
    60.676132297D+010.676132297D+010.124816471D-030.124816471D-03
    70.949882204D+030.949882204D+030.367828564D+060.000000000D+00
    70.688248670D+010.688248670D+010.267044809D-040.267044809D-04
```

```
 80.949882146D+030.949882146D+030.349355177D+060.0000000000D+00
 80.698495222D+010.698495222D+01-.248258543D-03-.248258543D-03
 90.949882135D+030.949882135D+030.331413146D+060.0000000000D+00
 90.698859789D+010.698859789D+01-.991732707D-04-.991732707D-04
100.949882148D+030.949882148D+030.313474029D+060.0000000000D+00
100.699007538D+010.699007538D+01-.357622335D-04-.357622335D-04
110.949882121D+030.949882121D+030.295536496D+060.0000000000D+00
110.699063111D+010.699063111D+01-.821326883D-05-.821326883D-05
120.949882121D+030.949882121D+030.277600000D+060.0000000000D+00
120.699078638D+010.699078638D+010.000000000D+000.0000000000D+00
130.661140000D+030.661140000D+030.675750000D+060.0000000000D+00
130.645403526D+010.645403526D+010.000000000D+000.0000000000D+00
140.661139976D+030.661139976D+030.457944948D+060.0000000000D+00
140.645403526D+010.645403526D+010.134087592D-020.134087592D-02
150.709005430D+030.709005430D+030.440286984D+060.0000000000D+00
150.644199773D+010.644199773D+010.928251848D-030.928251849D-03
160.768786808D+030.768786808D+030.422313640D+060.0000000000D+00
160.651999493D+010.651999493D+010.548821310D-030.548821310D-03
170.840563281D+030.840563281D+030.404309099D+060.0000000000D+00
170.662673574D+010.662673574D+010.427459315D-030.427459315D-03
180.900987733D+030.900987733D+030.386077651D+060.0000000000D+00
180.676258771D+010.676258771D+010.215717403D-030.215717403D-03
190.949814287D+030.949814287D+030.367828379D+060.0000000000D+00
190.688324996D+010.688324996D+010.945914321D-040.945914321D-04
200.949814277D+030.949814277D+030.349357575D+060.0000000000D+00
200.698476007D+010.698476007D+01-.356041467D-03-.356041467D-03
210.949814285D+030.949814285D+030.331413956D+060.0000000000D+00
210.698812060D+010.698812060D+01-.149952101D-03-.149952101D-03
220.949814304D+030.949814304D+030.313474316D+060.0000000000D+00
220.698961889D+010.698961889D+01-.597175017D-04-.597175017D-04
230.949814279D+030.949814279D+030.295536554D+060.0000000000D+00
230.699028057D+010.699028057D+01-.182037349D-04-.182037349D-04
240.949814279D+030.949814279D+030.277600000D+060.0000000000D+00
240.699053848D+010.699053848D+010.000000000D+000.0000000000D+00
250.661140000D+030.661140000D+030.675750000D+060.0000000000D+00
250.645417845D+010.645417845D+010.000000000D+000.0000000000D+00
260.661139976D+030.661139976D+030.457935425D+060.0000000000D+00
260.645417845D+010.645417845D+010.155979087D-020.155979087D-02
270.709007651D+030.709007651D+030.440281298D+060.0000000000D+00
270.644169688D+010.644169688D+010.108494207D-020.108494207D-02
280.768792981D+030.768792981D+030.422310628D+060.0000000000D+00
280.651956761D+010.651956761D+010.638818786D-030.638818786D-03
290.840572981D+030.840572981D+030.404306936D+060.0000000000D+00
290.662642196D+010.662642196D+010.504013753D-030.504013753D-03
300.901000145D+030.901000145D+030.386076682D+060.0000000000D+00
```

```
300.676230243D+010.676230243D+010.256252601D-030.256252601D-03
310.949828329D+030.949828329D+030.367827986D+060.000000000D+00
310.688304299D+010.688304299D+010.127519496D-030.127519496D-03
320.949828094D+030.949828094D+030.349359311D+060.000000000D+00
320.698443939D+010.698443939D+01-.419776596D-03-.419776596D-03
330.949827962D+030.949827962D+030.331414574D+060.000000000D+00
330.698806424D+010.698806424D+01-.177593108D-03-.177593108D-03
340.949827916D+030.949827916D+030.313474560D+060.000000000D+00
340.698963622D+010.698963622D+01-.718548316D-04-.718548316D-04
350.949827871D+030.949827871D+030.295536625D+060.000000000D+00
350.699031921D+010.699031921D+01-.227451686D-04-.227451686D-04
360.949827871D+030.949827871D+030.277600000D+060.000000000D+00
360.699057870D+010.699057870D+010.000000000D+000.000000000D+00
370.661140000D+030.661140000D+030.675750000D+060.000000000D+00
370.645428949D+010.645428949D+010.000000000D+000.000000000D+00
380.661139976D+030.661139976D+030.457928041D+060.000000000D+00
380.645428949D+010.645428949D+010.179030824D-020.179030824D-02
390.709009385D+030.709009385D+030.440276865D+060.000000000D+00
390.644146447D+010.644146447D+010.124468169D-020.124468169D-02
400.768797831D+030.768797831D+030.422308292D+060.000000000D+00
400.651923359D+010.651923359D+010.726055283D-030.726055283D-03
410.840580594D+030.840580594D+030.404305235D+060.000000000D+00
410.662617911D+010.662617911D+010.579300377D-030.579300377D-03
420.901009888D+030.901009888D+030.386075914D+060.000000000D+00
420.676208049D+010.676208049D+010.294186726D-030.294186726D-03
430.949839319D+030.949839319D+030.367827626D+060.000000000D+00
430.688288602D+010.688288602D+010.177753970D-030.177753970D-03
440.949714824D+030.949714824D+030.349360678D+060.000000000D+00
440.698368498D+010.698368498D+01-.461539051D-03-.461539051D-03
450.949658968D+030.949658968D+030.331415063D+060.000000000D+00
450.698663932D+010.698663932D+01-.207414256D-03-.207414256D-03
460.949634152D+030.949634152D+030.313474756D+060.000000000D+00
460.698828621D+010.698828621D+01-.922642260D-04-.922642260D-04
470.949624131D+030.949624131D+030.295536684D+060.000000000D+00
470.698923257D+010.698923257D+01-.371568794D-04-.371568794D-04
480.949624131D+030.949624131D+030.277600000D+060.000000000D+00
480.698980314D+010.698980314D+010.000000000D+000.000000000D+00
490.661140000D+030.661140000D+030.675750000D+060.000000000D+00
490.541970490D+010.541970490D+010.000000000D+000.000000000D+00
500.661139976D+030.661139976D+030.457921379D+060.000000000D+00
500.541970490D+010.541970490D+010.000000000D+000.000000000D+00
510.699646325D+030.699646325D+030.440272850D+060.000000000D+00
510.548841579D+010.548841579D+010.000000000D+000.000000000D+00
520.747216772D+030.747216772D+030.422306195D+060.000000000D+00
520.559525484D+010.559525484D+010.000000000D+000.000000000D+00
```

530.803998842D+030.803998842D+030.404303690D+060.000000000D+00
530.569825826D+010.569885826D+010.000000000D+000.000000000D+00
540.851601173D+030.851601173D+030.386075215D+060.000000000D+00
540.581498449D+010.581498449D+010.000000000D+000.000000000D+00
550.890031705D+030.890031705D+030.367827221D+060.000000000D+00
550.590800788D+010.590800788D+010.000000000D+000.000000000D+00
560.890031690D+030.890031690D+030.349361862D+060.000000000D+00
560.598279362D+010.598279362D+010.000000000D+000.000000000D+00
570.890031694D+030.890031694D+030.331415520D+060.000000000D+00
570.596519113D+010.596519113D+010.000000000D+000.000000000D+00
580.890031699D+030.890031699D+030.313474959D+060.000000000D+00
580.595729676D+010.595729676D+010.000000000D+000.000000000D+00
590.890031688D+030.890031688D+030.295536764D+060.000000000D+00
590.595380175D+010.595380175D+010.000000000D+000.000000000D+00
600.890031688D+030.890031688D+030.277600000D+060.000000000D+00
600.595241215D+010.595241215D+010.000000000D+000.000000000D+00
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03
T0.661139976D+030.661139976D+030.661139976D+030.177692822D+040.169945358D+04
T0.148493467D+040.119305137D+040.887163285D+030.762281343D+030.739650519D+03
T0.718545367D+030.229407525D+040.220226250D+040.190408012D+040.143965213D+04
T0.989632560D+030.835144320D+030.807124783D+030.780994205D+030.270225402D+04
T0.262496466D+040.235072460D+040.175835324D+040.110423196D+040.920248747D+03
T0.886840533D+030.855684557D+030.249915538D+040.242058256D+040.215147977D+04
T0.165030075D+040.112246557D+040.968224687D+030.940205151D+030.914074572D+03
T0.223810093D+040.216198092D+040.192056129D+040.153494856D+040.112893154D+04
T0.100439023D+040.981759401D+030.960654249D+030.949882146D+030.949882146D+03
T0.949882146D+030.949882146D+030.949882146D+030.949882146D+030.949882146D+03
T0.949882146D+030.949882135D+030.949882135D+030.949882135D+030.949882135D+03
T0.949882135D+030.949882135D+030.949882135D+030.949882135D+030.949882148D+03
T0.949882148D+030.949882148D+030.949882148D+030.949882148D+030.949882148D+03
T0.949882148D+030.949882148D+030.949882121D+030.949882121D+030.949882121D+03
T0.949882121D+030.949882121D+030.949882121D+030.949882121D+030.949882121D+03
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03
T0.661139976D+030.661139976D+030.661139976D+030.177689940D+040.169942569D+04
T0.148491047D+040.119303284D+040.887149675D+030.762267718D+030.739636893D+03
T0.718531742D+030.229401948D+040.220220365D+040.190401737D+040.143960338D+04
T0.989601252D+030.835112961D+030.807093425D+030.780962946D+030.270219726D+04
T0.262490331D+040.235064620D+040.175826399D+040.110418122D+040.920197885D+03
T0.886789671D+030.855633695D+030.249907233D+040.242049368D+040.215137401D+04
T0.165020046D+040.112240299D+040.968161977D+030.940142441D+030.914011862D+03
T0.223799226D+040.216186803D+040.192044307D+040.153484906D+040.112883180D+04
T0.100432175D+040.981690925D+030.960585773D+030.949814277D+030.949814277D+03
T0.949814277D+030.949814277D+030.949814277D+030.949814277D+030.949814277D+03
T0.949814277D+030.949814285D+030.949814285D+030.949814285D+030.949814285D+03
T0.949814285D+030.949814285D+030.949814285D+030.949814285D+030.949814304D+03

```
T0.949814304D+030.949814304D+030.949814304D+030.949814304D+030.949814304D+03
T0.949814304D+030.949814304D+030.949814279D+030.949814279D+030.949814279D+03
T0.949814279D+030.949814279D+030.949814279D+030.949814279D+030.949814279D+03
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03
T0.661139976D+030.661139976D+030.661139976D+030.177690446D+040.169943059D+04
T0.148491472D+040.119303609D+040.887152066D+030.762270112D+030.739639288D+03
T0.718534136D+030.229403090D+040.220221571D+040.190403022D+040.143961337D+04
T0.989607666D+030.835119386D+030.807099849D+030.780969270D+030.270220841D+04
T0.262491536D+040.235066159D+040.175828151D+040.110419118D+040.920207874D+03
T0.886799659D+030.855643684D+030.249908909D+040.242051161D+040.215139534D+04
T0.165022069D+040.112241561D+040.968174625D+030.940155089D+030.914024510D+03
T0.223801488D+040.216189153D+040.192046768D+040.153486977D+040.112887741D+04
T0.100433600D+040.981705178D+030.960600027D+030.949828094D+030.949828094D+03
T0.949828094D+030.949828094D+030.949828094D+030.949828094D+030.949828094D+03
T0.949828094D+030.949827962D+030.949827962D+030.949827962D+030.949827962D+03
T0.949827962D+030.949827962D+030.949827962D+030.949827962D+030.949827916D+03
T0.949827916D+030.949827916D+030.949827916D+030.949827916D+030.949827916D+03
T0.949827916D+030.949827916D+030.949827871D+030.949827871D+030.949827871D+03
T0.949827871D+030.949827871D+030.949827871D+030.949827871D+030.949827871D+03
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03
T0.661139976D+030.661139976D+030.661139976D+030.177690842D+040.169943441D+04
T0.148491804D+040.119303864D+040.887153934D+030.762271982D+030.739641157D+03
T0.718536005D+030.229403988D+040.220222518D+040.190404032D+040.143962121D+04
T0.989612704D+030.835124431D+030.807104895D+030.780974316D+030.270221716D+04
T0.262492482D+040.235067368D+040.175829527D+040.110419900D+040.920215712D+03
T0.886807497D+030.855651522D+030.249910223D+040.242052567D+040.215141208D+04
T0.165023657D+040.112242552D+040.968184552D+030.940165015D+030.914034436D+03
T0.223803277D+040.216191012D+040.192048714D+040.153488615D+040.112888866D+04
T0.100434727D+040.981716450D+030.960611298D+030.949714824D+030.949714824D+03
T0.949714824D+030.949714824D+030.949714824D+030.949714824D+030.949714824D+03
T0.949714824D+030.949658968D+030.949658968D+030.949658968D+030.949658968D+03
T0.949658968D+030.949658968D+030.949658968D+030.949658968D+030.949634152D+03
T0.949634152D+030.949634152D+030.949634152D+030.949634152D+030.949634152D+03
T0.949634152D+030.949634152D+030.949624131D+030.949624131D+030.949624131D+03
T0.949624131D+030.949624131D+030.949624131D+030.949624131D+030.949624131D+03
T0.661139976D+030.661139976D+030.661139976D+030.661139976D+030.661139976D+03
T0.661139976D+030.661139976D+030.661139976D+030.175496958D+040.167823793D+04
T0.146656179D+040.117897741D+040.876804866D+030.751911989D+030.729281164D+03
T0.708176013D+030.225221059D+040.215821163D+040.185774891D+040.140409633D+04
T0.966699134D+030.812174698D+030.784155162D+030.758024583D+030.265753179D+04
T0.257657755D+040.228898081D+040.169079942D+040.106588978D+040.881819019D+03
T0.848410804D+030.817254929D+030.242750750D+040.234398530D+040.206164460D+04
T0.156873596D+040.107147044D+040.917126727D+030.889107191D+030.862976612D+03
T0.213527169D+040.205587076D+040.181242410D+040.144635784D+040.106777019D+04
T0.943126193D+030.920495368D+030.899390217D+030.890031690D+030.890031690D+03
```

```
T0.890031690D+030.890031690D+030.890031690D+030.890031690D+030.890031690D+03
T0.890031690D+030.890031694D+030.890031694D+030.890031694D+030.890031694D+03
T0.890031694D+030.890031694D+030.890031694D+030.890031694D+030.890031699D+03
T0.890031699D+030.890031699D+030.890031699D+030.890031699D+030.890031699D+03
T0.890031699D+030.890031699D+030.890031688D+030.890031688D+030.890031688D+03
T0.890031688D+030.890031686D+030.890031688D+030.890031689D+030.890031688D+03
T0.890031688D+030.890031688D+030.890031688D+030.890031688D+030.890031688D+03
F
T0.661139976D+030.661139976D+030.699646325D+030.747216772D+030.803998842D+03
T0.851601173D+030.890031705D+030.890031690D+030.890031694D+030.890031699D+03
T0.890031688D+030.890031688D+030.890031690D+030.890031694D+030.890031699D+03
F
```

PRINTED OUTPUT EXAMPLE

flow map at time = 1.9007 sec.

number of time steps = 192
number of iterations = 770
time step size = 0.1454D-02 sec.

inlet mass flow rate = 0.268297D+01 kg/sec inlet enthalpy flow = 0.234219D+07 watt
outlet mass flow rate = 0.914200D+01 kg/sec outlet enthalpy flow = 0.131079D+08 watt
total heat transfered = 0.272580D+07 watt

channel number 1

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.010783	851.598	851.598	1005.330	849.091	3.43381	3.37281	0.00000	0.00000
11	1.9192	0.010783	851.598	851.598	957.036	849.091	3.80309	3.92908	0.01375	0.01375
10	1.9053	0.120591	877.980	877.979	956.123	876.620	10.80335	8.29518	-0.04234	-0.04234
9	2.0836	0.778999	897.775	897.773	967.469	895.300	24.55758	8.75392	-0.80572	-0.79020
8	2.2962	0.983274	980.388	980.388	980.036	925.166	34.95135	5.32153	0.97598	0.93946
7	2.3966	0.939780	987.087	987.087	985.655	1070.378	3.46065	0.50722	0.40586	0.40327
6	2.4354	0.010680	989.274	989.274	987.774	997.784	0.85252	0.83565	0.06171	0.06170
5	2.4638	0.000000	886.795	886.795	989.308	898.459	0.89817	0.89817	0.00957	0.00957
4	2.4990	0.000000	717.602	717.602	991.196	726.899	0.86429	0.86429	0.00200	0.00200
3	2.5366	0.000000	559.583	559.583	993.184	567.597	0.81454	0.81454	0.00031	0.00031
2	2.5758	0.000000	388.000	388.000	995.231	388.000	0.81165	0.81165	-0.00020	-0.00020
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81165	0.81165	0.00000	0.00000

channel number 2

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.013694	850.618	850.618	1005.330	848.367	3.37639	3.30393	0.00000	0.00000
11	1.9180	0.013694	850.618	850.618	956.956	848.367	3.67583	3.82623	0.00991	0.00990
10	1.9052	0.161033	875.160	875.158	956.112	874.532	12.36248	8.03308	-0.17664	-0.17638
9	2.1451	0.990050	963.998	963.999	971.209	888.852	41.74311	6.83830	1.28744	1.21714
8	2.2468	0.942923	974.842	974.842	977.205	924.136	28.65856	6.20831	0.17670	0.16972
7	2.3659	0.941946	985.341	985.341	983.957	1056.445	7.71611	0.77429	0.91280	0.90757
6	2.4288	0.000272	985.940	985.940	987.414	994.873	0.98653	0.98567	0.05649	0.05649
5	2.4630	0.000000	885.666	885.666	989.268	897.504	0.93446	0.93446	0.00790	0.00790
4	2.4989	0.000000	717.325	717.325	991.187	726.660	0.87301	0.87301	0.00144	0.00144
3	2.5366	0.000000	559.447	559.447	993.183	567.473	0.81773	0.81773	0.00015	0.00015
2	2.5758	0.000000	388.000	388.000	995.233	388.000	0.81125	0.81125	-0.00050	-0.00050

1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81125	0.81125	0.00000	0.00000

channel number 3

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.008960	850.006	850.006	1005.330	847.885	3.31343	3.26321	0.00000	0.00000
11	1.9174	0.008960	850.006	850.006	956.916	847.885	3.71564	3.73243	0.01519	0.01519
10	1.9146	0.103637	871.035	871.034	956.734	871.663	9.99655	7.42375	-0.16207	-0.16200
9	2.1012	0.825045	889.776	889.775	968.551	886.600	22.08457	7.60331	0.54027	0.53450
8	2.2408	0.983656	977.425	977.426	976.855	924.390	34.67865	5.82414	0.72068	0.70740
7	2.3347	0.937245	983.685	983.685	982.215	1047.662	4.36154	1.16028	1.11469	1.10838
6	2.4263	0.000089	984.672	984.672	987.277	993.807	1.05234	1.05201	0.04685	0.04685
5	2.4627	0.000000	885.371	885.371	989.251	897.267	0.94564	0.94564	0.00587	0.00587
4	2.4988	0.000000	717.242	717.242	991.183	726.593	0.87624	0.87624	0.00104	0.00104
3	2.5366	0.000000	559.346	559.346	993.182	567.384	0.82035	0.82035	0.00008	0.00008
2	2.5758	0.000000	388.000	388.000	995.234	388.000	0.81083	0.81083	-0.00073	-0.00073
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81083	0.81083	0.00000	0.00000

channel number 4

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.002959	843.195	843.195	1005.330	842.154	3.07388	3.05353	0.00000	0.00000
11	1.9169	0.002959	843.195	843.195	956.883	842.154	3.16124	3.15590	0.01478	0.01478
10	1.9199	0.021862	850.868	850.868	957.078	854.891	7.19584	6.58415	-0.05237	-0.05237
9	2.0887	0.164075	850.842	850.840	967.783	848.647	11.37246	8.19370	-0.03159	-0.03150
8	2.2338	0.987239	976.981	976.981	976.449	918.343	32.79295	5.62849	1.45981	1.40642
7	2.3097	0.931968	981.770	981.770	980.301	1048.771	4.75434	1.59974	1.00605	1.00252
6	2.4250	0.000007	983.910	983.910	987.207	993.180	1.09089	1.09087	0.02652	0.02652
5	2.4626	0.000000	885.240	885.240	989.243	897.164	0.95099	0.95099	0.00315	0.00315
4	2.4988	0.000000	717.213	717.213	991.182	726.570	0.87760	0.87760	0.00058	0.00058
3	2.5366	0.000000	559.321	559.321	993.182	567.360	0.82085	0.82085	-0.00001	-0.00001
2	2.5759	0.000000	388.000	388.000	995.236	388.000	0.81020	0.81020	-0.00098	-0.00098
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	0.81020	0.81020	0.00000	0.00000

channel number 5

iz	p (bar)	void	tv	tl	tsat	twall	tcan	uvz (m/sec)	ulz (m/sec)
			(degree celsius)						
12	1.8720	0.000327	713.043	713.043	1005.330	709.701	713.043	2.29313	2.29167
11	1.9166	0.000327	713.043	713.043	956.861	709.701	713.043	1.75293	1.75221

10	1.9211	0.002928	741.112	741.112	957.157	735.525	741.112	4.04090	3.87783
9	2.0910	0.029919	803.879	803.879	967.923	795.314	803.879	8.19941	7.51813
8	2.2099	0.328978	856.139	856.136	975.053	846.926	856.136	6.59102	5.76649
7	2.2942	0.215748	862.126	862.123	979.923	867.617	841.491	2.56718	1.85150
6	2.4244	0.000000	783.632	783.632	987.179	793.138	767.650	0.86239	0.86239
5	2.4625	0.000000	710.987	710.987	989.240	724.489	700.308	0.71430	0.71430
4	2.4988	0.000000	603.494	603.494	991.181	614.385	595.840	0.66470	0.66470
3	2.5366	0.000000	499.709	499.709	993.182	508.341	494.512	0.63937	0.63937
2	2.5759	0.000000	388.000	388.000	995.238	388.000	388.000	0.67691	0.67691
1	2.6471	0.000000	388.000	388.000	1138.947	388.000	388.000	0.67691	0.67691

flow map at time = 2.0013 sec.

number of time steps = 265
number of iterations = 1138
time step size = 0.3378D-02 sec.

inlet mass flow rate = -.271222D+01 kg/sec inlet enthalpy flow = -.248236D+07 watt
outlet mass flow rate = 0.754283D+01 kg/sec outlet enthalpy flow = 0.109099D+08 watt
total heat transfered = -.562540D+06 watt

channel number 1

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.303258	861.964	861.963	1005.330	860.980	0.51501	3.07887	0.00000	0.00000
11	1.5388	0.303258	861.964	861.963	929.841	860.980	15.50110	5.88634	-0.14508	-0.14462
10	1.8744	0.994372	945.344	945.344	954.065	884.862	27.85684	3.51727	1.07253	1.05714
9	1.8967	0.970861	956.143	956.144	955.553	885.346	22.69684	3.38400	0.32808	0.31151
8	1.9247	0.973085	958.142	958.142	957.394	918.024	16.03301	2.50611	0.27615	0.26388
7	1.9403	0.971530	959.403	959.403	958.410	1102.302	6.95722	1.05196	0.33247	0.31702
6	1.9438	0.977325	959.525	959.525	958.641	1055.607	1.96098	-2.71013	0.37385	0.37203
5	1.9863	0.276728	961.461	961.461	961.380	965.323	-0.53806	-1.36656	0.12127	0.12115
4	2.1552	0.003745	806.046	806.046	971.812	803.191	-0.89807	-0.94945	0.00319	0.00318
3	2.3318	0.000193	637.440	637.440	982.050	634.469	-0.88206	-0.88490	0.00252	0.00252
2	2.5118	0.000005	424.457	424.457	991.872	417.601	-0.84238	-0.84243	-0.00068	-0.00068
1	2.6471	0.000000	388.000	388.000	1138.947	417.601	-0.84238	-0.84243	0.00000	0.00000

channel number 2

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.267247	859.178	859.177	1005.330	859.009	2.07274	3.82092	0.00000	0.00000
11	1.5528	0.267247	859.178	859.177	930.932	859.009	15.03800	9.07103	-0.25991	-0.25926
10	1.8534	0.993174	944.524	944.524	952.657	883.012	27.19286	3.42531	1.60127	1.58285
9	1.8781	0.961646	954.645	954.645	954.316	878.283	21.72918	3.35960	0.43395	0.41368
8	1.9107	0.969096	957.194	957.194	956.473	923.430	13.99242	2.38723	0.42054	0.39789
7	1.9230	0.973095	958.282	958.282	957.287	1093.160	6.96456	1.07655	0.56712	0.54060
6	1.9263	0.980335	958.405	958.405	957.497	1049.437	6.78303	-2.64147	0.77521	0.77175
5	1.9771	0.190750	960.883	960.883	960.797	964.921	-0.35509	-1.16214	-0.06259	-0.06248
4	2.1552	0.003013	803.565	803.565	971.809	801.283	-0.84814	-0.89399	0.00015	0.00015
3	2.3316	0.000176	636.337	636.337	982.037	633.663	-0.86065	-0.86327	0.00200	0.00200
2	2.5118	0.000005	424.035	424.035	991.876	417.296	-0.84150	-0.84154	-0.00216	-0.00216

1	2.6471	0.000000	388.000	388.000	1138.947	417.296	-0.84150	-0.84154	0.00000	0.00000

channel number 3

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.194987	847.012	847.011	1005.330	849.999	1.93011	3.94404	0.00000	0.00000
11	1.5650	0.194987	847.012	847.011	931.873	849.999	13.82167	8.91571	-0.29528	-0.29487
10	1.8382	0.992186	937.570	937.571	951.627	873.235	31.75979	3.99577	2.07276	1.95726
9	1.8649	0.962474	953.507	953.507	953.433	875.425	19.54338	3.28950	0.14674	0.14020
8	1.8976	0.955072	955.987	955.987	955.609	929.787	10.60926	2.26209	0.42229	0.40285
7	1.9086	0.972566	957.333	957.333	956.340	1083.821	2.22228	0.93556	0.63869	0.60924
6	1.9089	0.981059	957.346	957.346	956.360	1046.766	1.84264	-2.38606	0.95529	0.95168
5	1.9822	0.235475	961.237	961.237	961.115	965.307	-0.35914	-1.06574	0.02240	0.02241
4	2.1553	0.002785	802.564	802.564	971.816	800.549	-0.81983	-0.86247	-0.00133	-0.00133
3	2.3315	0.000164	635.795	635.795	982.032	633.262	-0.84754	-0.84997	0.00136	0.00136
2	2.5120	0.000004	423.787	423.787	991.883	417.115	-0.84167	-0.84170	-0.00321	-0.00321
1	2.6471	0.000000	388.000	388.000	1138.947	417.115	-0.84167	-0.84170	0.00000	0.00000

channel number 4

iz	p (bar)	void	tv	tl	tsat	twall	uvz (m/sec)	ulz (m/sec)	uvr (m/sec)	ulr (m/sec)
			(degree celsius)							
12	1.8720	0.057287	826.273	826.273	1005.330	830.661	2.56785	3.32214	0.00000	0.00000
11	1.5759	0.057287	826.273	826.273	932.711	830.661	9.94701	8.60545	-0.14279	-0.14277
10	1.7882	0.464318	845.903	845.899	948.201	841.528	9.54023	5.16638	0.03636	0.03632
9	1.8619	0.952944	935.220	935.219	953.233	860.605	11.26089	3.51069	0.24239	0.23809
8	1.8882	0.875778	927.424	927.424	954.991	926.111	7.09710	2.66816	0.16310	0.16036
7	1.8981	0.973212	956.609	956.609	955.648	1074.430	-2.40832	0.77884	0.64798	0.61870
6	1.8962	0.983523	956.620	956.620	955.518	1045.606	-0.37324	-2.25866	1.18517	1.17979
5	1.9827	0.091832	948.708	948.708	961.145	957.131	-0.58589	-0.98588	-0.01629	-0.01628
4	2.1554	0.002370	801.594	801.594	971.822	799.919	-0.80536	-0.84113	-0.00068	-0.00068
3	2.3314	0.000131	635.521	635.521	982.029	633.080	-0.84035	-0.84221	0.00045	0.00045
2	2.5121	0.000003	423.420	423.420	991.890	416.859	-0.84098	-0.84100	-0.00438	-0.00438
1	2.6471	0.000000	388.000	388.000	1138.947	416.859	-0.84098	-0.84100	0.00000	0.00000

channel number 5

iz	p (bar)	void	tv	tl	tsat	twall	tcan	uvz (m/sec)	ulz (m/sec)
			(degree celsius)						
12	1.8720	0.003825	763.699	763.699	1005.330	753.066	763.699	1.16509	1.25295
11	1.5793	0.003825	763.699	763.699	932.972	753.066	763.699	6.99345	6.85619

10	1.7879	0.020460	823.341	823.341	948.180	816.611	823.341	6.65369	6.37594
9	1.8577	0.105617	859.689	859.688	952.946	853.058	859.688	5.87165	5.69386
8	1.8855	0.093630	877.018	877.017	954.806	872.255	877.017	4.81453	4.76605
7	1.8880	0.308404	892.240	892.239	954.977	899.516	879.397	1.42155	1.66511
6	1.8766	0.318665	847.592	847.590	954.213	856.880	815.256	-1.69625	-2.11925
5	1.9834	0.007083	750.548	750.548	961.195	760.944	727.796	-0.84139	-0.88960
4	2.1554	0.000247	634.591	634.590	971.823	643.339	618.204	-0.82661	-0.82885
3	2.3314	0.000007	526.750	526.750	982.029	533.441	513.420	-0.81987	-0.81992
2	2.5122	0.000000	400.197	400.197	991.897	399.065	394.352	-0.61606	-0.61606
1	2.6471	0.000000	388.000	388.000	1138.947	399.065	394.352	-0.61606	-0.61606

APPENDIX D

NATOF - 2D PROGRAM LISTING

COMPILATION LISTING OF NATOF (>user_dir_dir>BOIL>Granziera>NATOF.fortran)

Compiled by: Multics New Fortran Compiler, Release 6
Compiled on: 04/29/80 1304.9 edt Tue
    Options: table card relocatable map

Main Program

```
 1 C          MAIN PROGRAM
 2 C
 3 C          THE MAIN PROGRAM HAS THE ONLY FUNCTION OF ALOCATING
 4 C          POSITIONS IN THE MEMORY FOR THE VARIABLES.
 5 C
 6 C          THE COMAND 'DIMENSION ORBI(XXXX)' ALOCATES MEMORY FOR
 7 C          ALL THE VARIABLES.FOR EACH PROBLEM,THE USER SHOULD
 8 C          SUPLY ITS DIMENSION,WHICH VALUE IS CALCULATED AS :
 9 C
10 C          XXXX = (131 + 2*(NCF + NCLD))*NI*NJ
11 C
12 C          WITH :
13 C
14 C          NI = NUMBER OF AXIAL MESH POINTS
15 C          NJ = NUMBER OF RADIAL MESH POINTS
16 C
17 C
18       IMPLICIT REAL*8 (A-H,O-Z)
19 C
20       DIMENSION ORBI(12000)
21       NORBI = 12000
22       DO 10 II = 1,NORBI
23    10 ORBI(II) = 0.D0
24 C
25       CALL HEAD(ORBI,NORBI)
26       STOP
27       END
```

```
      BLOCK DATA
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      REAL*8 ZERO/0.D0/,ONE/1.D0/,BIG/1.D+07/,SMALL/1.D-08/
      END
```

Subroutine head

```
      SUBROUTINE HEAD(ORBI,NORBI)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /BCX/ ULO
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /ERROR/ IERR
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /PNTR1/ K(100),M(100)
      DIMENSION ORBI(NORBI)
C
C          THE MATRIX M CONTAINS POINTERS TO THE MATRIX ORBI
C          WHICH CORRESPOND TO THE FIRST ELEMENT OF THE VARIABLE
C          DIMENSIONED ARAYS IN THE FOLLOWING EQUIVALENCE :
C
C     M( 1) = P              M( 2) = PO
C     M( 3) = TV             M( 4) = TVO
C     M( 5) = TL             M( 6) = TLO
C     M( 7) = ALFAN          M( 8) = ALFAO
C     M( 9) = ALFAZ          M(10) = ALFAR
C     M(11) = RHOV           M(12) = RHOL
C     M(13) = RHOVZ          M(14) = RHOLZ
C     M(15) = RHOVR          M(16) = RHOLR
C     M(17) = HV             M(18) = HL
C     M(19) = HVZ            M(20) = HLZ
C     M(21) = HVR            M(22) = HLR
C     M(23) = UVZN           M(24) = ULZN
C     M(25) = UVRN           M(26) = ULRN
C     M(27) = UVZO           M(28) = ULZO
C     M(29) = UVRO           M(30) = ULRO
C     M(31) = UVRZ           M(32) = ULRZ
C     M(33) = UVZR           M(34) = ULZR
C     M(35) = FUVZN          M(36) = FULZN
C     M(37) = FUVRN          M(38) = FULRN
C     M(39) = W(K( 1))       M(40) = W(K( 3))
C     M(41) = W(K( 3))       M(42) = W(K( 4))
C     M(43) = W(K( 5))       M(44) = W(K( 6))
C     M(45) = W(K( 7))       M(46) = W(K( 8))
C     M(47) = W(K( 9))       M(48) = W(K(10))
C     M(49) = W(K(11))       M(50) = W(K(12))
```

```
C      M(51) = W(K(13))        M(52) = W(K(14))
C      M(53) = W(K(15))        M(54) = W(K(16))
C      M(55) = W(K(17))        M(56) = W(K(18))
C      M(57) = W(K(19))        M(58) = W(K(20))
C      M(59) = W(K(21))        M(60) = W(K(22))
C      M(61) = W(K(23))        M(62) = W(K(24))
C      M(63) = DH              M(64) = DHR
C      M(65) = DV              M(66) = QSI
C      M(67) = TS              M(68) = TW
C      M(69) = DTW             M(70) = HCONV
C      M(71) = HCONL           M(72) = HNB
C      M(73) = DPN             M(74) = A1
C      M(75) = A2              M(76) = A3
C      M(77) = A4              M(78) = YP
C      M(79) = B               M(80) = TR
C      M(81) = DTR             M(82) = TWOLD
C      M(83) = BETA            M(84) = GAMMA
C      M(85( = QPP      C      M(86) = AVZD
C      M(87) = ALZD     C      M(88) = AVRD
C      M(89) = ALRD            M(90) = SPPD
C      M(91) = TCAN            M(92) =
C
       CALL READ1
C
       M(1) = 1
       DO 1001 L = 2,79
 1001  M(L) = M(L-1) + NN
       M(80) = M(79) + NB
       M(81) = M(80) + NTR
       M(82) = M(81) + NTR
       M(83) = M(82) + NN
       M(84) = M(83) + NN
       M(85) = M(84) + NN
       M(86) = M(85) + NN
       M(87) = M(86) + NN
       M(88) = M(87) + NN
       M(89) = M(88) + NN
       M(90) = M(89) + NN
       M(91) = M(90) + NN
       NCAN = 4*NI
       M(92) = M(91) + NCAN
C
C
C
       TIME = ZERO
```

```
      CALL READ2(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
     *           ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26)),
     *           ORBI(M(63)),ORBI(M(65)),ORBI(M(66)),ORBI(M(80)),
     *           ORBI(M(81)),TINIT,ORBI(M(68)),ORBI(M(90)),ORBI(M(91)),
     *           NP,NTR,NPIN,NPM1,NN,NCAN)
      DO 104 KO = 1,NN
      KL = KO - 1
      DO 103 L = 1,67
  103 K(L) = M(L) + KL
C
      ORBI(K(67)) = SAT(ORBI(K(1)))
      ORBI(K(2)) = ORBI(K(1))
      ORBI(K(4)) = ORBI(K(3))
      ORBI(K(6)) = ORBI(K(5))
      ORBI(K(8)) = ORBI(K(7))
      ORBI(K(30)) = ORBI(K(26))
      ORBI(K(27)) = ORBI(K(23))
      ORBI(K(28)) = ORBI(K(24))
      ORBI(K(29)) = ORBI(K(25))
  104 CONTINUE
C
C
C
C
      NPRI = 0
      TPRI = TINIT + TSET(1)
      TSHPRI = TINIT + TSHSET(1)
      LSH = 1
      L = 1
      NTS = 0
      NIT = 0
      TIME = TINIT
      LPRI = 0
      LSHPRI = 0
    1 CONTINUE
      NDT = 0
      IT3 = 0
      CALL TMSTEP(ORBI,NORBI,
     *            NN,NP,NB,NW,NTR,NPIN,NPM1,NCAN)
C
      NIT = NIT + IT3
      NTS = NTS + 1
C
      IF(IERR.NE.0) GO TO 7
      IF(TIME.LT.TPRI) GO TO 1
```

```
C
      GO TO 8
    2 CONTINUE
C
C
      LPRI = LPRI + 1
      IF(LPRI-NSET(L))3,4,4
    3 TPRI = TPRI + TSET(L)
      GO TO 1
    4 L = L + 1
      NPRI = NPRI + LPRI
      IF(NSET(L))6,6,5
    5 LPRI = 0
      TPRI = TPRI + TSET(L)
      GO TO 1
    6 CALL SAVER(ORBI(M(1)),ORBI(M(3)),ORBI(M(5)),ORBI(M(7)),
     *          ORBI(M(23)),ORBI(M(24)),ORBI(M(25)),ORBI(M(26)),
     *          ORBI(M(80)),ORBI(M(91)),TIME,NTR,NN,NCAN,NI)
      RETURN
    7 CALL ERRMES(TIME)
    8 CONTINUE
C
      QT = ZERO
      FMI = ZERO
      FME = ZERO
      FHI = ZERO
      FHE = ZERO
      KP = 0
      DO 9 J = 1,NJ
      KI = (J-1)*NI + 1
      KE = J*NI
      IF(ORBI(M(24)+KI-1).LT.ZERO) KI = KI + 1
C
      TV = ORBI(M(3)+KI-1)
      TL = ORBI(M(5)+KI-1)
      PP = ORBI(M(1)+KI-1)
      UV = ORBI(M(23)+KI-1)
      UL = ORBI(M(24)+KI-1)
      AA = ORBI(M(7)+KI-1)
      RV = ORBI(M(11)+KI-1)
      RL = ORBI(M(12)+KI-1)
      EV = ORBI(M(17)+KI-1)
      EL = ORBI(M(18)+KI-1)
C
C
```

```
      FMVJ = AA*RV*UV*DR4(J)
      FMLJ = (ONE - AA)*RL*UL*DR4(J)
      HV = EV + PP/RV
      HL = EL + PP/RL
C
      FMI = FMI + FMVJ + FMLJ
      FHI = FHI + FMVJ*HV + FMLJ*HL
C
      TV = ORBI(M(3)+KE-1)
      TL = ORBI(M(5)+KE-1)
      PP = ORBI(M(1)+KE-1)
      AA = ORBI(M(7)+KE-2)
      UV = ORBI(M(23)+KE-1)
      UL = ORBI(M(24)+KE-1)
      RV = ORBI(M(11)+KE-1)
      RL = ORBI(M(12)+KE-1)
      EV = ORBI(M(17)+KE-1)
      EL = ORBI(M(18)+KE-1)
C
C
      FMVJ = AA*RV*UV*DR4(J)
      FMLJ = (ONE - AA)*RL*UL*DR4(J)
      HV = EV + PP/RV
      HL = EL + PP/RL
C
      FME = FME + FMVJ + FMLJ
      FHE = FHE + FMVJ*HV + FMLJ*HL
C
      DO 9 I = 1,NIM2
      KP = KP + 1
      KO = (J-1)*NI + I
      QT = QT + ORBI(M(85)+KP-1)*ORBI(M(65)+KO)*DZ(I+1)*DR4(J)
    9 CONTINUE
      WRITE(6,200) TIME
      WRITE(6,201) NTS,NIT,DT
      WRITE(6,202) FMI,FHI,FME,FHE,QT
      DO 10 J = 1,NUM1
C
      WRITE(6,203) J
      WRITE(6,204)
C
      DO 10 I = 1,NI
      KI = NI - I +1
      KO = (J-1)*NI + KI
      KP = KO - 2*J + 1
```

```
      IF(KI.EQ.1) KP = KP + 1
      IF(KI.EQ.NI)KP = KP - 1
C
      PP = ORBI(M(1)+KO-1)/1.D+05
      TVP = ORBI(M(3)+KO-1) - 273.14
      TLP = ORBI(M(5)+KO-1) - 273.14
      TSP = ORBI(M(67)+KO-1) - 273.14
      TWP = ORBI(M(68)+KP-1) - 273.14
      AP = ORBI(M(7)+KO-1)
      UVZ = ORBI(M(23)+KO-1)
      ULZ = ORBI(M(24)+KO-1)
      UVR = ORBI(M(25)+KO-1)
      ULR = ORBI(M(26)+KO-1)
C
      WRITE(6,205) KI,PP,AP,TVP,TLP,TSP,TWP,
     *             UVZ,ULZ,UVR,ULR
   10 CONTINUE
C
      J = NJ
C
      WRITE(6,203) J
      WRITE(6,206)
C
      DO 11 I = 1,NI
      KI = NI - I +1
      KO = (J-1)*NI + KI
      KP = KO - 2*J + 1
      IF(KI.EQ.1) KP = KP + 1
      IF(KI.EQ.NI)KP = KP - 1
C
      PP = ORBI(M(1)+KO-1)/1.D+05
      TVP = ORBI(M(3)+KO-1) - 273.14
      TLP = ORBI(M(5)+KO-1) - 273.14
      TSP = ORBI(M(67)+KO-1) - 273.14
      TWP = ORBI(M(68)+KP-1) - 273.14
      TCP = ORBI(M(91)+KI-1) - 273.14
      AP = ORBI(M(7)+KO-1)
      UVZ = ORBI(M(23)+KO-1)
      ULZ = ORBI(M(24)+KO-1)
C
      WRITE(6,207) KI,PP,AP,TVP,TLP,TSP,TWP,TCP,
     *             UVZ,ULZ
   11 CONTINUE
C
      IF(IERR.NE.0) GO TO 6
```

```
      GO TO 2
  206 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
     *       5X,'TWALL',7X,'TCAN',6X,'UVZ',9X,'ULZ'/
     *       6X,'(BAR)',19X,'--------------(DEGREE CELSIUS)',
     *       '--------------',2(5X,'(M/SEC)')/)
  207 FORMAT(1X,I2,2X,F9.4,2X,F8.6,5(2X,F8.3),2(2X,F10.5))
  200 FORMAT(1H1,10X,'FLOW MAP AT TIME = ',F10.4,' SEC.'/)
  201 FORMAT(1X,'NUMBER OF TIME STEPS = ',I10/
     *       1X,'NUMBER OF ITERATIONS = ',I10/
     *       1X,'TIME STEP SIZE = 'D10.4,' SEC.'/)
  202 FORMAT(1X,'INLET MASS FLOW RATE  = ',D12.6,' KG/SEC',
     *       6X,'INLET ENTHALPY FLOW  = ',D12.6,' WATT'/
     *       1X,'OUTLET MASS FLOW RATE = ',D12.6,' KG/SEC',
     *       6X,'OUTLET ENTHALPY FLOW = ',D12.6,' WATT'/
     *       1X,'TOTAL HEAT TRANSFERED = ',D12.6,' WATT'/)
  203 FORMAT(1H0,40X,'CHANNEL NUMBER ',I5/)
  204 FORMAT(1X,'IZ',5X,'P',10X,'VOID',7X,'TV',8X,'TL',7X,'TSAT',
     *       5X,'TWALL',7X,'UVZ',9X,'ULZ',9X,'UVR',9X,'ULR'/
     *       6X,'(BAR)',19X,'---------(DEGREE CELSIUS)--------',
     *       4(5X,'(M/SEC)')/)
  205 FORMAT(1X,I2,2X,F9.4,2X,F8.6,4(2X,F8.3),4(2X,F10.5))
C
      END
```

Subroutine read1

```
      SUBROUTINE READ1
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /REA/ NN,NP,NB,NW,NTR,NPIN,NPM1,NSET(40),NSHSET(40)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /GRVTY/ GZ,GR
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /GAUSS/ NZ,NR,NZM1
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /ICONST/ NCF,NCC,NG
C
      READ(5,118) NI,NJ,NCF,NCLD
      WRITE(7,118)NI,NJ,NCF,NCLD
      NN = NI*NJ
      NIM1 = NI - 1
      NIM2 = NI - 2
      NJM1 = NJ - 1
      NP = (NI - 2)*NJ
      NB = 21*NN
      NW = 24*NN
      NNJ = NN - NI
      NNJJ = NNJ - NI
      NNI = NN - NJ
      NR = NJ
      NZ = NI - 2
      NZM1 = NZ - 1
      NG = NCF + 1
      NCC = NG + 1
      NPIN = NCC + NCLD
      NPM1 = NPIN - 1
      NTR = NPIN*NP
C
C
      L = 1
    1 CONTINUE
      READ(5,121) NSET(L),TSET(L)
      WRITE(7,121)NSET(L),TSET(L)
      L = L + 1
      IF(L.GT.50) GO TO 2
      IF(NSET(L-1)) 2,2,1
    2 CONTINUE
C
```

```
      GZ = 9.80665
      GR = 0.D0
C
C
      READ(5,119) ITM1,IGAUSS,DTMAX,EPS1,EPS2
      WRITE(7,119)ITM1,IGAUSS,DTMAX,EPS1,EPS2
      DT = DTMAX
C
      ITM2 = ITM1
  118 FORMAT(4I5)
  119 FORMAT(2I10,3D15.9)
  120 FORMAT(2D15.9)
  121 FORMAT(I5,D15.9)
      RETURN
      END
```

```
      SUBROUTINE READ2(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,DH,DV,
     *                 QSI,TR,DTR,TINIT,TW,SPPD,TCAN,
     *                 NP,NTR,NPIN,NPM1,NN,NCAN)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LP,LDATA,LSS
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
     *               PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
     *               ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
     *               TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
     *               OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
     *               LMAX,LP(51)
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /PINO/ RODR(20),VP(20),VM(20),RADR,PPP(20)
      COMMON /GCONST/ DIL,RADFU,RADCL
      COMMON /CCONST/ CA0,CA1,CA2,CA3,CB0,CB1,CB2,CB3
      COMMON /FCONST/ FA0,FA1,FA2,FA3,FB0,FB1,FB2,AD,APU,LPLNM(40)
      COMMON /ICONST/ NCF,NCC,NG
      COMMON /PD/ D4,POD2
      COMMON /POVERD/ R
      COMMON /HXCN/ ACOV
      COMMON /STST/ TAFP,LSS
      DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
     *          UVR(NN),ULR(NN),DH(NN),DV(NN),QSI(NN),TR(NTR),
     *          DTR(NTR),TW(NP),SPPD(NN),TCAN(NCAN)
      DIMENSION RAD(20),XIN(5),N(20)
C
      FA0 = 1.81D+06
      FA1 = 3.72D+03
      FA2 = -2.51D0
      FA3 = 6.59D-04
      FB0 = 10.8D0
      FB1 = -8.84D-03
      FB2 = 2.25D-06
C
      CA0 = 4.28D+06
      CA1 = 3.75D+02
      CA2 = -7.45D-03
      CA3 = ZERO
      CB0 = 16.27
      CB1 = ZERO
```

```
      CB2 = ZERO
      CB3 = ZERO
C
C
C
      TB(1) = ZERO
      L = 2
    2 CONTINUE
      READ(5,1001) LP(L),TB(L)
      WRITE(7,1001)LP(L),TB(L)
      IF(TB(L).LE.TB(L-1)) GO TO 3
      READ(5,1002) PNB1(L),PNB2(L),PNB3(L),OMP(L)
      READ(5,1002) PNT1(L),PNT2(L),PNT3(L),OMT(L)
      READ(5,1002) ALB1(L),ALB2(L),ALB3(L),OMA(L)
      READ(5,1002) TVB1(L),TVB2(L),TVB3(L),OMV(L)
      READ(5,1002) TLB1(L),TLB2(L),TLB3(L),OML(L)
      READ(5,1002) HNW1(L),HNW2(L),HNW3(L),OMH(L)
C
      WRITE(7,1002)PNB1(L),PNB2(L),PNB3(L),OMP(L)
      WRITE(7,1002)PNT1(L),PNT2(L),PNT3(L),OMT(L)
      WRITE(7,1002)ALB1(L),ALB2(L),ALB3(L),OMA(L)
      WRITE(7,1002)TVB1(L),TVB2(L),TVB3(L),OMV(L)
      WRITE(7,1002)TLB1(L),TLB2(L),TLB3(L),OML(L)
      WRITE(7,1002)HNW1(L),HNW2(L),HNW3(L),OMH(L)
C
      L = L + 1
      IF(L.GT.51) GO TO 3
      GO TO 2
    3 CONTINUE
      LMAX = L
      DO 4 KO = 1,NN
      QSI(KO) = (4.*D/(PITCH - D))**2
    4 CONTINUE
C
      READ(5,1003) NROW,PITCH,D,E
      WRITE(7,1003)NROW,PITCH,D,E
C
      POVD = PITCH/D
      POD2 = POVD*POVD
      D4 = 4./D
      R = -16.15 + 24.96*POVD - 8.55*POVD*POVD
C
      READ(5,1004) (N(J),J=1,19)
      WRITE(7,1004)(N(J),J=1,19)
      KRES = 0
```

```
469      5 CONTINUE
470        READ(5,1005) LDATA,(XIN(K),K=1,5)
471        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
472        IF(.NOT.LDATA) GO TO 205
473        DO 105 I = 1,5
474        KO = KRES + I
475        IF(KO.GT.NI) GO TO 5
476        DZ(KO) = XIN(I)
477    105 CONTINUE
478        KRES = KRES + 5
479        GO TO 5
480    205 CONTINUE
481  C
482        KRES = 3*NI
483    305 CONTINUE
484        READ(5,1005) LDATA,(XIN(K),K=1,5)
485        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
486        IF(.NOT.LDATA) GO TO 505
487        DO 405 I = 1,5
488        KO = KRES + I
489        IF(KO.GT.NCAN) GO TO 305
490        TCAN(KO) = XIN(I)
491    405 CONTINUE
492        KRES = KRES + 5
493        GO TO 305
494    505 CONTINUE
495        KRES = 0
496      6 CONTINUE
497        READ(5,1005) LDATA,(XIN(K),K=1,5)
498        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
499        IF(.NOT.LDATA) GO TO 206
500        DO 106 I = 1,5
501        KO = KRES + I
502        IF(KO.GT.NN) GO TO 6
503        SHAPE(KO) = XIN(I)
504    106 CONTINUE
505        KRES = KRES + 5
506        GO TO 6
507    206 CONTINUE
508        KRES = 0
509    306 CONTINUE
510        READ(5,1005) LDATA,(XIN(K),K=1,5)
511        WRITE(7,1005)LDATA,(XIN(K),K=1,5)
512        IF(.NOT.LDATA) GO TO 506
513        DO 406 I = 1,5
```

```
      KO = KRES + I
      IF(KO.GT.NN) GO TO 306
      SPPD(KO) = XIN(I)
  406 CONTINUE
      KRES = KRES + 5
      GO TO 306
  506 CONTINUE
C
      DZ1(1) = DZ(1)
      DO 7 I = 2,NI
      DZ1(I) = (DZ(I) + DZ(I-1))/2.D0
    7 CONTINUE
C
      A1 = DSQRT(3.D0)/2.D0
      A2 = 3.1415927/4.D0
      W = PITCH - D
C
      X = (PITCH*PITCH*A1 - (D*D + W*W)*A2)/A2/D
      XI = 4.D0/X
C
      DO 8 J = 1,NJM1
      DO 8 I = 1,NI
      KO = (J-1)*NI + I
      DH(KO) = X
      DV(KO) = XI
    8 CONTINUE
C
      DO 9 J = 2,NJM1
C
      N41 = N(J) - 1
      N42 = N(J-1) - 1
      DN4 = N41*N41 - N42*N42
      DR4(J) = DN4*X*A2*D*3.D0
C
      NX = N(J) - N(J-1)
      NX1 = 2*N41
      NX2 = (2*N42 + NX)*NX
      DNX1 = NX1
      DR1(J) = DNX1/NX2/PITCH/A1
      DR2(J) = 2.D0*N42/NX2/PITCH/A1
      DRO(J) = PITCH*A1*NX
    9 CONTINUE
C
      DN4 = (N(1) - 1)*(N(1) - 1)
      DR4(1) = DN4*X*A2*D*3.D0
```

```
C
      DR1(1) = 2.D0/PITCH/A1/(N(1)-1)
      DR2(1) = 0.D0
      DRO(1) = PITCH*A1*(N(1)-1)
C
      B1 = (N(NJM1) + NROW - 2)
      B2 = (NROW - N(NJM1))
      B3 = (NROW - 1)
C
      XX = B1*B2/2.D0 + B3/2.D0 + 1.D0/6.D0
      PT = B3*PITCH + (D/2.D0 + E)/A1 + A2*D*XX*4.D0
      AC = (B1*PITCH + (D/2.D0 + E)/A1)*(B2*PITCH*A1 + D/2.D0 + E)*
     *      0.5D0 - A2*(D*D + E*E)*XX
      Y = 4.D0*AC/PT
      PP = A2*D*XX*4.D0
      YY = PP/AC
      ARM = (ONE - A2/A1*(D*D + W*W)/(PITCH*PITCH))*
     *      (N(NJM1) - 1)*PITCH
C
      DR1(NJ) = ZERO
      DR2(NJ) = ARM/AC
      DRO(NJ) = B2*PITCH + D/2.D0 + E
      DR4(NJ) = AC*6.D0
      ACOV = (B3*PITCH + (D/2.D0 + E)/A1)/AC
C
      DO 10 I = 1,NI
      KO = NJM1*NI + I
      DH(KO) = Y
      DV(KO) = YY
   10 CONTINUE
C
      DR3(NJ) = DRO(NJ)
      DO 11 J = 1,NJM1
      DR3(J) = (DRO(J) + DRO(J+1))/2.D0
   11 CONTINUE
      KRES = 0
   12 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      WRITE(7,1005)LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 212
      DO 112 I = 1,5
      KO = KRES + I
      IF(KO.GT.NPIN) GO TO 12
      PPP(KO) = XIN(I)
  112 CONTINUE
```

```
      KRES = KRES + 5
      GO TO 12
  212 CONTINUE
C
      READ(5,1006) AD,APU,DIL
      READ(5,1007) (LPLNM(K),K = 1,39)
      READ(5,1008) RADR,THC,THG
C
      WRITE(7,1006)AD,APU,DIL
      WRITE(7,1007)(LPLNM(K),K = 1,39)
      WRITE(7,1008)RADR,THC,THG
C
      RADFU = RADR - THG - THC
      RADCL = RADFU + THG
      NCLD = NPIN - NCC
      DRF = RADFU/NCF
      DRC = THC/NCLD
      TAFP = RADFU*RADFU/D
C
      RAD(1) = ZERO
      DO 14 K = 1,NCF
      RAD(K+1) = RAD(K) + DRF
   14 CONTINUE
      RAD(NG+1) = RAD(NG) + THG
      DO 15 K = NCC,NPM1
      RAD(K+1) = RAD(K) + DRC
   15 CONTINUE
      DO 16 K = 1,NPM1
      IF(K.EQ.NG) RODR(K) = (RAD(K+1) + RAD(K))/2.D0
      IF(K.NE.NG) RODR(K) = (RAD(K+1)+RAD(K))/(RAD(K+1)-RAD(K))/2.D0
   16 CONTINUE
C
      VM(1) = ZERO
      VP(1) = DRF*DRF/8.D0
      RM = (RADR + RAD(NPM1))/2.D0
      VM(NPIN) = (RADR*RADR + W*W/4.D0- RM*RM)/2.D0
      VP(NPIN) = ZERO
      DO 17 K = 2,NPM1
      RP = (RAD(K+1) + RAD(K))/2.D0
      RM = (RAD(K) + RAD(K-1))/2.D0
      VP(K) = (RP*RP -RAD(K)*RAD(K))/2.D0
      VM(K) = (RAD(K)*RAD(K) - RM*RM)/2.D0
   17 CONTINUE
C
      READ(5,1009) LSS,TINIT
```

```
      TB(1) = ZERO
      IF (LSS) GO TO 19
      DO 1 KO = 1,NN
      READ(5,1000) KCHECK,TV(KO),TL(KO),P(KO),ALFA(KO)
      READ(5,1000) KCHECK,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
      IF(KCHECK.EQ.KO)GO TO 1
      IERR = 4
      RETURN
    1 CONTINUE
      KRES = 0
   13 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 213
      DO 113 I = 1,5
      KO = KRES + I
      IF(KO.GT.NTR) GO TO 13
      TR(KO) = XIN(I)
  113 CONTINUE
      KRES = KRES + 5
      GO TO 13
  213 CONTINUE
C
      KRES = 0
  313 CONTINUE
      READ(5,1005) LDATA,(XIN(K),K=1,5)
      IF(.NOT.LDATA) GO TO 513
      DO 413 I = 1,5
      KO = KRES + I
      K3 = KO + 2*NI
      IF(KO.GT.NI) GO TO 313
      TCAN(KO) = XIN(I)
      TCAN(K3) = XIN(I)
  413 CONTINUE
      KRES = KRES + 5
      GO TO 313
  513 CONTINUE
C
      DO 18 I = 1,NIM2
      DO 18 J = 1,NJ
      KP = (J-1)*NIM2 + I
      KT = KP*NPIN
      TW(KP) = TR(KT)
   18 CONTINUE
      RETURN
   19 CONTINUE
```

```
      READ(5,1010) PIN,POUT,TIN,TAV
      QPP = HNW2(2)*RADFU*RADFU/RADR/2.D0
      CALL SS (PIN,POUT,TIN,TAV,QPP,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
     *         TW,TR,DTR,DH,DV,NN,NP,NTR,NPIN,NPM1)
      RETURN
C
 1000 FORMAT(I5,4D15.9)
 1001 FORMAT(L1,F15.5)
 1002 FORMAT(4D15.9)
 1003 FORMAT(I5,3D15.9)
 1004 FORMAT(19I4)
 1005 FORMAT(L1,5D15.9)
 1006 FORMAT(3D15.9)
 1007 FORMAT(39I2)
 1008 FORMAT(3D15.9)
 1009 FORMAT(L1,D15.9)
 1010 FORMAT(4D15.9)
      END
```

Subroutine ss

```
      SUBROUTINE SS(PIN,POUT,TIN,TAV,Q,P,TV,TL,UVZ,ULZ,UVR,ULR,ALFA,
     *              TW,TR,DTR,DH,DV,NN,NP,NTR,NPIN,NPM1)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /GRVTY/ GZ,GR
      DIMENSION PROP(3,4)
      DIMENSION P(NN),TV(NN),TL(NN),UVZ(NN),ULZ(NN),UVR(NN),ULR(NN),
     *          ALFA(NN),TW(NP),TR(NTR),DTR(NTR),DH(NN),DV(NN)
C
C             SUBROUTINE SS PUTS AN INITIAL GUESS IN THE VARIABLES
C             TV,TL,P,UVZ,ULZ,UVR,ULR,ALFA AND TR, IN ORDER TO
C             ACCELERATE THE CONVERGENCE TO THE STEADY STATE PROBLEM.
C
      H = ZERO
      DO 1 I = 2,NI
      H = H + DZ1(I)
    1 CONTINUE
      DP = (PIN - POUT)/H
C
      CALL STATE (TAV,TAV,PIN,PROP,0)
      RHO = PROP(1,2)
      DPG = DP - RHO*GZ
C
      A = (RHO*DH(2)/VISCL(TAV))**.2*DH(2)/RHO/.1D0
      X = ONE/1.8D0
      V = (A*DPG)**X
C
      DO 2 J = 1,NJM1
      DO 2 I = 1,NI
      KO = (J-1)*NI + I
      ULZ(KO) = V
      UVZ(KO) = V
      ULR(KO) = ZERO
      UVR(KO) = ZERO
      ALFA(KO) = ZERO
    2 CONTINUE
C
      A = (RHO*DH(NNJ+2)/VISCL(TAV))**.2*DH(NNJ+2)/RHO/.1D0
      V = (A*DPG)**X
C
```

```
      DO 3 I = 1,NI
      KO = NNJ + I
      ULZ(KO) = V
      UVZ(KO) = V
      ULR(KO) = ZERO
      UVR(KO) = ZERO
      ALFA(KO) = ZERO
    3 CONTINUE
C
      TL(1) = TIN
      TV(1) = TIN
      P(1) = PIN
      DO 4 J = 1,NJ
      KO = J*NI - NIM1
      TL(KO) = TIN
      TV(KO) = TIN
      P(KO) = PIN
C
      DO 4 I = 2,NI
      KO = (J-1)*NI + I
      P(KO) = P(KO-1) - DP*DZ1(I)
      UXX = ULZ(KO)
      IF(UXX.EQ.ZERO) UXX = ONE
      TL(KO) = TL(KO-1) + Q*SHAPE(KO)*DV(KO)*DZ1(I)/RHO/UXX/
     /         CPL(TL(KO-1))
      TV(KO) = TL(KO)
    4 CONTINUE
C
      DT = .1D0
C
      DO 7 J = 1,NJ
      DO 7 I = 2,NIM1
      KO = (J-1)*NI + I
      KP = KO + 1 - J*2
      KT = (KP-1)*NPIN + 1
      KR = KP*NPIN
C
      TW(KP) = TL(KO)
      TS = SAT(P(KO))
      CALL HTCF (P(KO),TV(KO),TL(KO),ALFA(KO),PROP(1,1),
     *           PROP(1,2),PROP(1,3),PROP(1,4),DH(KO),TS,TW(KP),
     *           HCONV,HCONL,HNB,UVZ(KO),ULZ(KO))
C
      DO 5 K = 1,NPIN
      KTR = (KP-1)*NPIN + K
```

```
      TR(KTR) = TW(KP)
    5 CONTINUE
    6 CONTINUE
      TTR = TR(KT)
      CALL FPROP(TR(KT),NPIN,NPM1,I)
      CALL FPIN (TV(KO),TL(KO),TS,TW(KP),DTW,HCONV,HCONL,HNB,
     *           TR(KT),DTR(KT),DT,NPIN,NPM1,KO)
C
      TR(KR) = TW(KP)
      DO 16 KK = 1,NPM1
      KS = KR - KK
      TR(KS) = TR(KS) - DTR(KS)*TR(KS+1)
   16 CONTINUE
      TTR = DABS(TTR - TR(KT))/DT
      IF(TTR.GT.ONE) GO TO 6
    7 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TMSTEP(O,NO,
     *                  NN,NP,NB,NW,NTR,NPIN,NPM1,NCAN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /RHEA/ TSET(40),TSHSET(40),DTMAX,DTM1
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,IGAUSS
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /PNTR1/K(100),M(100)
C
C
      DIMENSION O(NO)
C
C
C
      DTLS = DT
      TMS = ZERO
      IERR = 0
      DO 100 J = 1,NJ
      DO 100 I = 2,NI
      KO = (J-1)*NI + I - 1
      K23 = KO + M(23)
      K24 = KO + M(24)
      K25 = KO + M(25)
      JO = J
C
      TSVZ = DABS(O(K23)/DZ1(I))
      TSLZ = DABS(O(K24)/DZ1(I))
      TSVR = DABS(O(K25)/DR3(J))
C
      TMS = DMAX1(TSVZ,TSLZ,TSVR,TMS)
  100 CONTINUE
C
      IF(TMS) 101,101,102
  101 DT = DTMAX
      GO TO 103
  102 DT = 0.95D0/TMS
      DT = DMIN1(DTMAX,DT,2.0*DTLS)
  103 CONTINUE
      IT2 = 0
      NDT = 0
```

```
      TIME = TIME+DT
C
      DO 104 KO = 1,NN
      KL = KO - 1
      K68 = M(68) + KL
      K82 = M(82) + KL
      O(K82) = O(K68)
  104 CONTINUE
C
      CALL DONOR(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 4)),O(M( 5)),
     *           O(M( 6)),O(M( 7)),O(M( 8)),O(M( 9)),O(M(10)),
     *           O(M(11)),O(M(12)),O(M(13)),O(M(14)),O(M(15)),
     *           O(M(16)),O(M(17)),O(M(18)),O(M(19)),O(M(20)),
     *           O(M(21)),O(M(22)),O(M(23)),O(M(24)),O(M(25)),
     *           O(M(26)),O(M(27)),O(M(28)),O(M(29)),O(M(30)),
     *           O(M(31)),O(M(32)),O(M(33)),O(M(34)),O(M(41)),
     *           O(M(42)),O(M(47)),O(M(48)),O(M(52)),O(M(53)),
     *           O(M(58)),O(M(59)),O(M(73)),O(M(86)),O(M(87)),
     *           O(M(88)),O(M(89)),NN,NP)
C
    1 CONTINUE
      CALL BC(O(M( 1)),O(M( 3)),O(M( 5)),O(M( 7)),TIME,
     *        O(M(24)),NN,NI,NIM1)
      CALL WS(O(M( 2)),O(M( 4)),O(M( 6)),O(M( 8)),O(M( 9)),
     *        O(M(10)),O(M(11)),O(M(12)),O(M(13)),O(M(14)),
     *        O(M(15)),O(M(16)),O(M(17)),O(M(18)),O(M(27)),
     *        O(M(28)),O(M(29)),O(M(30)),O(M(39)),O(M(40)),
     *        O(M(43)),O(M(44)),O(M(45)),O(M(46)),O(M(47)),
     *        O(M(48)),O(M(49)),O(M(50)),O(M(51)),
     *        O(M(54)),O(M(55)),O(M(56)),O(M(57)),O(M(58)),
     *        O(M(59)),O(M(60)),O(M(61)),O(M(62)),O(M(63)),
     *        O(M(65)),O(M(66)),O(M(90)),NN)
C
      DO 1001 I = 2,NIM1
      DO 1001 J = 1,NJ
      KO = (J-1)*NI + I - 1
      KP = KO + 1 -J*2
      KT = KP*NPIN
      K01 = M( 1) + KO
      K03 = M( 3) + KO
      K05 = M( 5) + KO
      K07 = M( 7) + KO
      K11 = M(11) + KO
      K12 = M(12) + KO
      K17 = M(17) + KO
```

```
      K18 = M(18) + KO
      K23 = M(23) + KO
      K24 = M(24) + KO
      K63 = M(63) + KO
      K67 = M(67) + KO
      K68 = M(68) + KP
      K69 = M(69) + KP
      K70 = M(70) + KP
      K71 = M(71) + KP
      K72 = M(72) + KP
      K80 = M(80) + KT
      K81 = M(81) + KT
      KF = KO + 1
C
      UV = (O(K23) + O(K23 + 1))/2.D0
      UL = (O(K24) + O(K24 + 1))/2.D0
C
      CALL HTCF (O(K01),O(K03),O(K05),O(K07),O(K11),O(K12),
     *           O(K17),O(K18),O(K63),O(K67),O(K68),O(K70),
     *           O(K71),O(K72),UV,UL)
      CALL FPROP(O(K80),NPIN,NPM1,1)
      CALL FPIN (O(K03),O(K05),O(K67),O(K68),O(K69),O(K70),
     *           O(K71),O(K72),O(K80),O(K81),DT,NPIN,NPM1,KF)
 1001 CONTINUE
C
      CALL THXCN(O(M(3)),O(M(5)),O(M(70)),O(M(71)),O(M(91)),
     *           DT,NN,NI,NJ,NCAN,NIM1,NIM2)
C
      IF(IERR.NE.0) RETURN
      IT2 = 0
    2 CONTINUE
      IT2 = IT2+1
      CALL ONESTP(O(M( 1)),O(M( 2)),O(M( 3)),O(M( 5)),O(M( 7)),
     *            O(M( 8)),O(M( 9)),O(M(10)),O(M(11)),O(M(12)),
     *            O(M(17)),O(M(18)),O(M(19)),O(M(20)),O(M(21)),
     *            O(M(22)),O(M(23)),O(M(24)),O(M(25)),O(M(26)),
     *            O(M(35)),O(M(36)),O(M(37)),O(M(38)),O(M(39)),
     *            O(M(65)),O(M(67)),O(M(68)),O(M(69)),O(M(70)),
     *            O(M(71)),O(M(72)),O(M(73)),O(M(74)),O(M(75)),
     *            O(M(76)),O(M(77)),O(M(78)),O(M(79)),O(M(83)),
     *            O(M(84)),O(M(86)),O(M(87)),O(M(88)),O(M(89)),
     *            O(M(91)),DT,NN,NB,NP,NW,NCAN)
      IF(IERR.NE.0) GO TO 5
      IF(RES.GT.EPS1) GO TO 4
      IT3 = IT3 + IT2
```

```
      CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
     *         O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
     *         NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      CALL THXCNO(O(M(91)),NCAN,NI)
      RETURN
    4 IF(IT2.LT.ITM2) GO TO 2
    5 CONTINUE
      NDT = NDT+1
      IT3 =IT3+IT2
      IT2 = 0
      TIME = TIME-DT
      DTO = DT
      DT = DT*0.1
      IF(DT.LT.1.D-07) IERR = 21
      TIME = TIME+DT
      DO 6 KO = 1,NN
      KL = KO - 1
      K01 = M( 1) + KL
      K02 = M( 2) + KL
      K03 = M( 3) + KL
      K04 = M( 4) + KL
      K05 = M( 5) + KL
      K06 = M( 6) + KL
      K07 = M( 7) + KL
      K08 = M( 8) + KL
      K23 = M(23) + KL
      K24 = M(24) + KL
      K25 = M(25) + KL
      K26 = M(26) + KL
      K27 = M(27) + KL
      K28 = M(28) + KL
      K29 = M(29) + KL
      K30 = M(30) + KL
      K73 = M(73) + KL
      K68 = M(68) + KL
      K82 = M(82) + KL
C
      O(K03) = O(K04)
      O(K05) = O(K06)
      O(K07) = O(K08)
      O(K23) = O(K27)
      O(K24) = O(K28)
      O(K25) = O(K29)
      O(K26) = O(K30)
      O(K01) = O(K02)
```

```
      O(K73) = ZERO
      O(K68) = O(K82)
    6 CONTINUE
      CALL FTP(O(M( 3)),O(M( 5)),O(M(67)),O(M(68)),O(M(70)),
     *         O(M(71)),O(M(72)),O(M(80)),O(M(81)),O(M(85)),
     *         NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      IF(IERR.GT.20) RETURN
      IF(NDT.GT.3) RETURN
      IERR = 0
      GO TO 1
      END
```

```
1007       SUBROUTINE DONOR(P,PO,TV,TVO,TL,TLO,ALFAN,ALFAO,ALFAZ,ALFAR,
1008      *                 RHOV,RHOL,RHOVZ,RHOLZ,RHOVR,RHOLR,
1009      *                 HV,HL,HVZ,HLZ,HVR,HLR,UVZN,
1010      *                 ULZN,UVRN,ULRN,UVZO,ULZO,UVRO,ULRO,
1011      *                 UVRZ,ULRZ,UVZR,ULZR,WZ1,WZ2,
1012      *                 WZ7,WZ8,WR1,WR2,WR7,WR8,DPN,AVZD,ALZD,
1013      *                 AVRD,ALRD,NN,NP)
1014        IMPLICIT REAL*8 (A-H,O-Z)
1015        COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
1016        COMMON / GRVTY/ GZ,GR
1017        COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
1018      *              DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
1019        COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
1020        DIMENSION P(NN),PO(NN),TV(NN),TVO(NN),TL(NN),TLO(NN),ALFAN(NN),
1021      *           ALFAO(NN),ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),
1022      *           RHOVZ(NN),RHOLZ(NN),RHOVR(NN),RHOLR(NN),
1023      *           HV(NN),HL(NN),HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
1024      *           UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),UVZO(NN),
1025      *           ULZO(NN),UVRO(NN),ULRO(NN),UVRZ(NN),ULRZ(NN),
1026      *           UVZR(NN),ULZR(NN),WZ1(NN),WZ2(NN),WZ7(NN),WZ8(NN),
1027      *           WR1(NN),WR2(NN),WR7(NN),WR8(NN),DPN(NN),
1028      *           AVZD(NN),ALZD(NN),AVRD(NN),ALRD(NN)
1029        DIMENSION PROP(3,4),S(5,2)
1030  C
1031        IFLAG = 0
1032        DTR = DT/DTLS
1033        DO 101 KO = 1,NN
1034        CALL STATE (TV(KO),TL(KO),P(KO),PROP,IFLAG)
1035  C
1036        IF(ALFAN(KO).GT.1.D-08) GO TO 100
1037        ALFAN(KO) = ZERO
1038        TV(KO) = TL(KO)
1039    100 CONTINUE
1040        TVO(KO) = TV(KO)
1041        TLO(KO) = TL(KO)
1042        PO(KO) = P(KO)
1043        RHOV(KO)   = PROP(1,1)
1044        RHOL(KO)   = PROP(1,2)
1045        HV(KO)     = PROP(1,3)
1046        HL(KO)     = PROP(1,4)
1047        ALFAO(KO)  = ALFAN(KO)
1048  C
1049        IF(DABS(UVRN(KO)).LT.1.D-10) UVRN(KO) = ZERO
```

```
      IF(DABS(ULRN(KO)).LT.1.D-10) ULRN(KO) = ZERO
      UVZO(KO) = UVZN(KO)
      UVRO(KO) = UVRN(KO)
      ULZO(KO) = ULZN(KO)
      ULRO(KO) = ULRN(KO)
  101 CONTINUE
      DO 1101 J = 1,NJ
      DO 1101 I = 2,NIM1
      KO = (J-1)*NI + I
      KP = (J-1)*NIM1 - J + I
 1101 DPN(KP) = P(KO) - PO(KO)
      DO 2101 I = 2,NI
      II = I - 1
      DO 2101 J = 1,NJM1
      JJ = J + 1
      KO = (J - 1)*NI + I
      KI = KO - 1
      KJ = KO + NI
C
      DZM = DZ(I) + DZ(II)
      DRM = DRO(J) + DRO(JJ)
C
      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM
      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
C
      ALFAR(KO) = (ALFAO(KO)*DRO(J) + ALFAO(KJ)*DRO(JJ))/DRM
      RHOVR(KO) = (RHOV(KO)*DRO(J) + RHOV(KJ)*DRO(JJ))/DRM
      RHOLR(KO) = (RHOL(KO)*DRO(J) + RHOL(KO)*DRO(JJ))/DRM
 2101 CONTINUE
C
      DO 3101 J = 1,NJ
      KO = (J - 1)*NI + 1
      ALFAZ(KO) = ALFAO(KO)
      RHOVZ(KO) = RHOV(KO)
      RHOLZ(KO) = RHOL(KO)
 3101 CONTINUE
C
      DO 4101 I = 2,NI
      KO = NNJ + I
      II = I - 1
      KI = KO -1
      DZM = DZ(I) + DZ(II)
C
      ALFAZ(KO) = (ALFAO(KO)*DZ(I) + ALFAO(KI)*DZ(II))/DZM
```

```
      RHOVZ(KO) = (RHOV(KO)*DZ(I) + RHOV(KI)*DZ(II))/DZM
      RHOLZ(KO) = (RHOL(KO)*DZ(I) + RHOL(KI)*DZ(II))/DZM
 4101 CONTINUE
      DO 102 J = 2,NJM1
      DO 102 I = 2,NIM1
      KO = (J-1)*NI+I
C
      UVRZ(KO) = (UVRO(KO)+UVRO(KO-1)+UVRO(KO-NI)+UVRO(KO-1-NI))/4.
      ULRZ(KO) = (ULRO(KO)+ULRO(KO-1)+ULRO(KO-NI)+ULRO(KO-1-NI))/4.
      UVZR(KO) = (UVZO(KO)+UVZO(KO+1)+UVZO(KO+NI)+UVZO(KO+1+NI))/4.
      ULZR(KO) = (ULZO(KO)+ULZO(KO+1)+ULZO(KO+NI)+ULZO(KO+1+NI))/4.
C
      KD = 0
      IF(UVZO(KO).GE.ZERO) KD = -1
      KN = KO+KD
      IO = I +KD
C
      HVZ(KO) = HV(KN)
      AVZO(KO) = ALFAO(KN)
      WZ1(KO) = ALFAO(KN)*RHOV(KN)
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = I+KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZO(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      KD = NI
      IF(UVRO(KO).GE.ZERO) KD = 0
      JO = J + KD/NI
      KN = KO + KD
C
      HVR(KO) = HV(KN)
      AVRO(KO) = ALFAO(KN)
      WR1(KO) = ALFAO(KN)*RHOV(KN)
      WR7(KO) = (UVRO(KN)-UVRO(KN-NI))/DRO(JO)*UVRO(KO)
      IF(J.EQ.NJM1) WR7(KO) = -UVRO(KN-NI)*UVRO(KO)/DRO(JO)
C
C
      KD = NI
```

```
      IF(ULRO(KO).GE.ZERO) KD = 0
      JO = J + KD/NI
      KN = KO+KD
C
      HLR(KO) = HL(KN)
      ALRO(KO) = ONE - ALFAO(KN)
      WR2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WR8(KO) = (ULRO(KN)-ULRO(KN-NI))/DRO(JO)*ULRO(KO)
      IF(J.EQ.NJM1) WR8(KO) = -ULRO(KN-NI)*ULRO(KO)/DRO(JO)
C
C
      KD = NI
      IF(UVRZ(KO).GE.ZERO) KD = 0
      KN = KO + KD
      JO = J - 1 + KD/NI
C
      WZ7(KO) = ((UVZO(KN)-UVZO(KN-NI))*UVRZ(KO)/DR3(JO) +
     +          WZ7(KO) + GZ)*ALFAZ(KO)*RHOVZ(KO)
C
      KD = NI
      IF(ULRZ(KO).GE.ZERO) KD = 0
      KN = KO + KD
      JO = J - 1 + KD/NI
C
      WZ8(KO) = ((ULZO(KN)-ULZO(KN-NI))*ULRZ(KO)/DR3(JO) +
     +          WZ8(KO) + GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
C
      KD = 0
      IF(UVZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = I + KD + 1
C
      WR7(KO) = ((UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO) +
     +          WR7(KO) + GR)*ALFAR(KO)*RHOVR(KO)
C
      KD = 0
      IF(ULZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = I + KD + 1
C
      WR8(KO) = ((ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO) +
     +          WR6(KO) + GR)*(ONE-ALFAR(KO))*RHOLR(KO)
  102 CONTINUE
C
C     TOP CELLS
```

```
1185 C
1186       DO 103 KO = NI,NN,NI
1187       KD = 0
1188       IF(UVZO(KO).GE.ZERO) KD = -1
1189       KN = KO+KD
1190       IO = NI+KD
1191 C
1192       HVZ(KO) = HV(KN)
1193       AVZD(KO) = ALFAO(KN)
1194       WZ1(KO) = ALFAO(KN)*RHOV(KN)
1195       WZ7(KO) = ((UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)+GZ)*
1196      *          ALFAZ(KO)*RHOVZ(KO)
1197 C
1198 C
1199       KD = 0
1200       IF(ULZO(KO).GE.ZERO) KD = -1
1201       KN = KO+KD
1202       IO = NI+KD
1203 C
1204       HLZ(KO) = HL(KN)
1205       ALZD(KO) = ONE - ALFAO(KN)
1206       WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
1207       WZ8(KO) = ((ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)+GZ)*
1208      *          (ONE-ALFAZ(KO))*RHOLZ(KO)
1209 C
1210 C
1211   103 CONTINUE
1212 C
1213 C          THE CENTERLINE CELLS
1214 C
1215       DO 110 KO = 2,NIM1
1216 C
1217       UVRZ(KO) = (UVRO(KO)+UVRO(KO-1))/4.
1218       ULRZ(KO) = (ULRO(KO)+ULRO(KO-1))/4.
1219       UVZR(KO) = (UVZO(KO)+UVZO(KO+1)+UVZO(KO+NI)+UVZO(KO+1+NI))/4.
1220       ULZR(KO) = (ULZO(KO)+ULZO(KO+1)+ULZO(KO+NI)+ULZO(KO+1+NI))/4.
1221 C
1222       KD = 0
1223       IF(UVZO(KO).GE.ZERO) KD = -1
1224       KN = KO+KD
1225       IO = KO +KD
1226 C
1227       HVZ(KO) = HV(KN)
1228       AVZD(KO) = ALFAO(KN)
1229       WZ1(KO) = ALFAO(KN)*RHOV(KN)
```

```
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)*UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = KO + KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZD(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      KD = 0
      IF(UVZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = KO + KD + 1
      WR7(KO) = (UVRO(KN+1)-UVRO(KN))*UVZR(KO)/DZ1(IO)
C
      KD = 0
      IF(ULZR(KO).GE.ZERO) KD = -1
      KN = KO + KD
      IO = KO + KD + 1
      WR8(KO) = (ULRO(KN+1)-ULRO(KN))*ULZR(KO)/DZ1(IO)
C
      IF(UVRO(KO))104,105,105
  104 KN = KO+NI
      JO = 2
C
      HVR(KO) = HV(KN)
      AVRD(KO) = ALFAO(KN)
      WR1(KO) = ALFAO(KN)*RHOV(KN)
      WR7(KO) = ((UVRO(KN) - UVRO(KO))/DRO(JO)*UVRO(KO)+
     +            WR7(KO)+GR)*ALFAR(KO)*RHOVR(KO)
C
C
      GO TO 106
  105 HVR(KO) = HV(KO)
      AVRD(KO) = ALFAO(KO)
      WR1(KO) = ALFAO(KO)*RHOV(KO)
      WR7(KO) = (UVRO(KO)/DRO(1)*UVRO(KO)+WR7(KO)+GR)*
     *          ALFAR(KO)*RHOVR(KO)
C
C
  106 CONTINUE
      IF(ULRO(KO)) 107,108,108
```

```
  107 KN = KO+NI
      JO = 2
C
      HLR(KO) = HL(KN)
      ALRD(KO) = ONE - ALFAO(KN)
      WR2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WR8(KO) = ((ULRO(KN) - ULRO(KO))/DRO(JO)*ULRO(KO)+
     +            WR8(KO)+GR)*(ONE-ALFAR(KO))*RHOLR(KO)
C
C
      GO TO 109
  108 HLR(KO) = HL(KO)
      ALRD(KO) = ONE - ALFAO(KO)
      RHOLR(KO) = RHOL(KO)
      WR2(KO) = (ONE-ALFAO(KO))*RHOLR(KO)
      WR8(KO) = (ULRO(KO)/DRO(1)*ULRO(KO)+WR8(KO)+GR)*WR2(KO)
C
C
  109 CONTINUE
C
      IF(UVRZ(KO)) 1108,2108,2108
 1108 WZ7(KO) = (WZ7(KO) + UVZO(KO+NI)*UVRZ(KO)/DR3(1) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
      GO TO 3108
 2108 WZ7(KO) = (WZ7(KO) + UVZO(KO)*UVRZ(KO)/DR3(1) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
C
 3108 CONTINUE
      IF(ULRZ(KO)) 4108,5108,5108
 4108 WZ8(KO) = (WZ8(KO) + ULZO(KO+NI)*ULRZ(KO)/DR3(1) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
      GO TO 6108
 5108 WZ8(KO) = (WZ8(KO) + ULZO(KO)*ULRZ(KO)/DR3(1) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
 6108 CONTINUE
C
C
  110 CONTINUE
C
C             THE WALL CELLS
C
      DO 111 I = 2,NIM1
      KO = NNJ+I
C
      UVRZ(KO) = (UVRO(KO-NI)+UVRO(KO-1-NI))/4.
```

```
      ULRZ(KO) = (ULRO(KO-NI)+ULRO(KO-1-NI))/4.
C
      KD = 0
      IF(UVZO(KO).GE.ZERO) KD = -1
      KN = KO+KD
      IO = I +KD
C
      HVZ(KO) = HV(KN)
      AVZD(KO) = ALFAO(KN)
      WZ1(KO) = ALFAO(KN)*RHOV(KN)
      WZ7(KO) = (UVZO(KN+1)-UVZO(KN))/DZ(IO)+UVZO(KO)
C
      KD = 0
      IF(ULZO(KO).GE.ZERO) KD = -1
      IO = I+KD
      KN = KO+KD
      HLZ(KO) = HL(KN)
      ALZD(KO) = ONE - ALFAO(KN)
      WZ2(KO) = (ONE-ALFAO(KN))*RHOL(KN)
      WZ8(KO) = (ULZO(KN+1)-ULZO(KN))/DZ(IO)*ULZO(KO)
C
C
      IF(UVRZ(KO)) 1110,2110,2110
 1110 WZ7(KO) = (WZ7(KO) - UVZO(KO)*UVRZ(KO)/DR3(3) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
      GO TO 3110
 2110 WZ7(KO) = (WZ7(KO) + (UVZO(KO)-UVZO(KO-NI))*UVRZ(KO)/DR3(2) +
     +            GZ)*ALFAZ(KO)*RHOVZ(KO)
C
 3110 CONTINUE
      IF(ULRZ(KO)) 4110,5110,5110
 4110 WZ8(KO) = (WZ8(KO) - ULZO(KO)*ULRZ(KO)/DR3(3) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
      GO TO 6110
 5110 WZ8(KO) = (WZ8(KO) +(ULZO(KO)-ULZO(KO-NI))*ULRZ(KO)/DR3(3) +
     +            GZ)*(ONE-ALFAZ(KO))*RHOLZ(KO)
 6110 CONTINUE
C
  111 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WS(PO,TVO,TLO,ALFAO,ALFAZ,ALFAR,RHOV,
     *              RHOL,RHOVZ,RHOLZ,RHOVR,RHOLR,HV,HL,
     *              UVZO,ULZO,UVRO,ULRO,
     *              WEV,WEL,WZ3,WZ4,WZ5,WZ6,WZ7,WZ8,WZ9,
     *              WZ10,WZ11,WR3,WR4,WR5,WR6,WR7,WR8,WR9,
     *              WR10,WR11,DH,DV,QSI,SPPD,NN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION PO(NN),TVO(NN),TLO(NN),ALFAO(NN),ALFAZ(NN),
     *          ALFAR(NN),RHOV(NN),RHOL(NN),RHOVZ(NN),RHOLZ(NN),
     *          RHOVR(NN),RHOLR(NN),HV(NN),HL(NN),UVZO(NN),
     *          ULZO(NN),UVRO(NN),ULRO(NN),WEV(NN),WEL(NN),
     *          WZ3(NN),WZ4(NN),WZ5(NN),WZ6(NN),WZ7(NN),WZ8(NN),
     *          WZ9(NN),WZ10(NN),WZ11(NN),WR3(NN),WR4(NN),
     *          WR5(NN),WR6(NN),WR7(NN),WR8(NN),WR9(NN),WR10(NN),
     *          WR11(NN),DH(NN),DV(NN),QSI(NN),SPPD(NN)
C
C
C
C           SUBROUTINE WS COMPLETE THE EVALUATION OF THE
C           EXPLICIT TERMS INVOLVED IN THE SOLUTION OF
C           THE PROBLEM STATED WITH SUBROUTINE DONOR.
C           HERE ARE SET THE TERMS CONTAINING THE TIME
C           INCREMENT DT.IT IS WRITTEN SEPARATELY FROM
C           SUBROUTINE DONOR IN ORDER TO ALLOW A CHANGE
C           IN THE VALUE OF DT WHEN THE PROBLRM DOES NOT
C           CONVERGE WITH THE PREVIOUS DT.
C           (SEE NEXT COMENT IN THIS SUBROUTINE.)
C
C
C
      DO 5 JO = 1,NJ
      DO 5 IO = 2,NI
      KO = (JO-1)*NI+IO
C
      WWZ1 = ALFAZ(KO)*RHOVZ(KO)
      WWZ2 = (ONE - ALFAZ(KO))*RHOLZ(KO)
      WWR1 = ALFAR(KO)*RHOVR(KO)
      WWR2 = (ONE - ALFAR(KO))*RHOLR(KO)
C
```

```
      CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
     *           ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
     *           RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
     *           SPPD(KO),WWZ1,WWZ2,WWR1,WWR2,
     *           FVZ,FLZ,FVR,FLR,C1Z,C1R)
C
      WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
      WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
C
      IF(NDT.NE.0) GO TO 1
C
C
C          SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
C          OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
C          IS NOT COMPLITED,WE PUT A CHECK HERE TO KNOW
C          IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
C          WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
C          BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
C          OLD DT AND ADD THEM BACK WITH THE NEW VALUE
C          OF DT.
C
      WZ4(KO) = C1Z
      WZ6(KO) = C1Z
      WR4(KO) = C1R
      WR6(KO) = C1R
C
      WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
      WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
      WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
      WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
C
C
      WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
      WZ8(KO) = WZ8(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
      WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
      WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
      GO TO 2
C
    1 DTC = ONE/DTO - ONE/DT
C
      WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
      WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
      WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
      WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
      WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC
```

```
1404        CALL COEFF(TVO(KO),TLO(KO),UVZO(KO),UVRO(KO),ULZO(KO),ULRO(KO),
1405       *           ALFAZ(KO),ALFAR(KO),RHOVZ(KO),RHOVR(KO),
1406       *           RHOLZ(KO),RHOLR(KO),DH(KO),DV(KO),QSI(KO),
1407       *           SPPD(KO),WWZ1,WWZ2,WWR1,WWR2,
1408       *           FVZ,FLZ,FVR,FLR,C1Z,C1R)
1409 C
1410        WEV(KO) = -(RHOV(KO)*HV(KO)+PO(KO))*ALFAO(KO)/DT
1411        WEL(KO) = -(RHOL(KO)*HL(KO)+PO(KO))*(ONE-ALFAO(KO))/DT
1412 C
1413        IF(NDT.NE.0) GO TO 1
1414 C
1415 C
1416 C           SINCE THE PROGRAM ALLOWS A CHANGE IN THE VALUE
1417 C           OF THE TIME INCREMENT DT,EVEN IF THE TIME STEP
1418 C           IS NOT COMPLITED,WE PUT A CHECK HERE TO KNOW
1419 C           IF SUCH A CHANGE DID OCCUR (IN THIS CASE NDT
1420 C           WOULD BE DIFFERENT THAN ZERO) IN CASE THE TEST
1421 C           BE TRUE,WE SUBTRACT THE TERMS WHICH HAVE THE
1422 C           OLD DT AND ADD THEM BACK WITH THE NEW VALUE
1423 C           OF DT.
1424 C
1425        WZ4(KO) = C1Z
1426        WZ6(KO) = C1Z
1427        WR4(KO) = C1R
1428        WR6(KO) = C1R
1429 C
1430        WZ3(KO) = WZ4(KO) + ALFAZ(KO)*RHOVZ(KO)/DT + FVZ
1431        WZ5(KO) = WZ6(KO) + (ONE-ALFAZ(KO))*RHOLZ(KO)/DT + FLZ
1432        WR3(KO) = WR4(KO) + ALFAR(KO)*RHOVR(KO)/DT + FVR
1433        WR5(KO) = WR6(KO) + (ONE-ALFAR(KO))*RHOLR(KO)/DT + FLR
1434 C
1435 C
1436        WZ7(KO) = WZ7(KO) - UVZO(KO)/DT*ALFAZ(KO)*RHOVZ(KO)
1437        WZ8(KO) = WZ6(KO) - ULZO(KO)/DT*(ONE-ALFAZ(KO))*RHOLZ(KO)
1438        WR7(KO) = WR7(KO) - UVRO(KO)/DT*ALFAR(KO)*RHOVR(KO)
1439        WR8(KO) = WR8(KO) - ULRO(KO)/DT*(ONE-ALFAR(KO))*RHOLR(KO)
1440        GO TO 2
1441 C
1442      1 DTC = ONE/DTO - ONE/DT
1443 C
1444        WZ7(KO) = UVZO(KO)*ALFAZ(KO)*RHOVZ(KO)*DTC + WZ7(KO)
1445        WZ8(KO) = ULZO(KO)*(ONE-ALFAZ(KO))*RHOLZ(KO)*DTC + WZ8(KO)
1446        WR7(KO) = UVRO(KO)*ALFAR(KO)*RHOVR(KO)*DTC + WR7(KO)
1447        WR8(KO) = ULRO(KO)*(ONE-ALFAR(KO))*RHOLR(KO)*DTC + WR8(KO)
1448        WZ3(KO) = WZ3(KO) - ALFAZ(KO)*RHOVZ(KO)*DTC
```

```
      WZ5(KO) = WZ5(KO) - (ONE-ALFAZ(KO))*RHOLZ(KO)*DTC
      WR3(KO) = WR3(KO) - ALFAR(KO)*RHOVR(KO)*DTC
      WR5(KO) = WR5(KO) - (ONE-ALFAR(KO))*RHOLR(KO)*DTC
C
    2 IF(WZ3(KO).GT.SMALL) GO TO 3
C
C
C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
C           VAPOR IN THE CELL AT THE PRESENT TIME STEP.
C           IN CASE THERE IS NO VAPOR NOR EVAPORATION
C           (WZ3 = ZERO),THE VAPOR MOMENTUM EQUATION
C           BECOMES TRIVIAL AND THE LIQUID EQUATION
C           STANDS ALONE.
C
      WZ11(KO) = ZERO
      WZ9(KO)  = ZERO
      WZ10(KO) = -(ONE-ALFAZ(KO))/DZ1(IO)/WZ5(KO)
      GO TO 5
C
    3 IF(WZ5(KO).GT.SMALL) GO TO 4
C
C
C           THIS TEST IS DONE TO CHECK THE PRESENCE OF
C           LIQUID IN THE CELL AT THE PRESENT TIME STEP.
C           IN CASE THERE IS NO LIQUID NOR CONDENSATION
C           (WZ5 = ZERO),THE LIQUID MOMENTUM EQUATION
C           BECOMES TRIVIAL AND THE VAPOR EQUATION
C           STANDS ALONE.
C
C
      WZ11(KO) = ZERO
      WZ10(KO) = ZERO
      WZ9(KO)  = -ALFAZ(KO)/DZ1(IO)/WZ3(KO)
      GO TO 5
C
    4 WZ11(KO) = WZ3(KO)*WZ5(KO)-WZ4(KO)*WZ6(KO)
      WZ10(KO) = -(ALFAZ(KO)*WZ6(KO)+(ONE-ALFAZ(KO))*WZ3(KO))/
     /             DZ1(IO)/WZ11(KO)
      WZ9(KO)  = -(ALFAZ(KO)*WZ5(KO)+(ONE-ALFAZ(KO))*WZ4(KO))/
     /             DZ1(IO)/WZ11(KO)
C
    5 CONTINUE
C
C
C           THE SAME TEST WHICH WAS DONE FOR THE
```

```
1494 C           Z-DIRECTION (SEE COMENTS ABOVE) IS
1495 C           DONE HERE FOR THE R-DIRECTION.NOTE
1496 C           THAT SINCE THE MOMENTUM EQUATIONS ARE
1497 C           EVALUATED AT DIFFERENT LOCATIONS FOR
1498 C           EACH DIRECTION,IT IS POSSIBLE THAT
1499 C           ONE PHASE IS ABSENT IN ONE DIRECTION
1500 C           EQUATIONS AND PRESENT IN THE OTHER
1501 C           DIRECTION EQUATIONS.
1502 C
1503 C
1504       DO 8 JO = 1,NJM1
1505       DO 8 IO = 2,NIM1
1506       KO = (JO-1)*NI + IO
1507 C
1508       IF(WR3(KO).GT.SMALL) GO TO 6
1509       WR11(KO) = ZERO
1510       WR9(KO)  = ZERO
1511       WR10(KO) = -(ONE-ALFAR(KO))/DR3(JO)/WR5(KO)
1512       GO TO 8
1513 C
1514     6 IF(WR5(KO).GT.SMALL) GO TO 7
1515       WR11(KO) = ZERO
1516       WR10(KO) = ZERO
1517       WR9(KO)  = -ALFAR(KO)/DR3(JO)/WR3(KO)
1518       GO TO 8
1519 C
1520     7 WR11(KO) = WR3(KO)*WR5(KO) - WR4(KO)*WR6(KO)
1521       WR10(KO) = -(ALFAR(KO)*WR6(KO)+(ONE-ALFAR(KO))*WR3(KO))/
1522      /             DR3(JO)/WR11(KO)
1523       WR9(KO)  = -(ALFAR(KO)*WR5(KO)+(ONE-ALFAR(KO))*WR4(KO))/
1524      /             DR3(JO)/WR11(KO)
1525     8 CONTINUE
1526       RETURN
1527       END
```

```
      SUBROUTINE ONESTP(PN,PO,TVN,TLN,ALFAN,ALFAO,ALFAZ,ALFAR,
     *                  RHOV,RHOL,HV,HL,HVZ,HLZ,HVR,HLR,
     *                  UVZN,ULZN,UVRN,ULRN,
     *                  FUVZN,FULZN,FUVRN,FULRN,W,DV,TS,
     *                  TW,DTW,HCONV,HCONL,HNB,DPN,A1,A2,A3,
     *                  A4,YP,B,BETA,GAMMA,AVZD,ALZD,AVRD,
     *                  ALRD,TCAN,DT,NN,NB,NP,NW,NCAN)
C
      IMPLICIT REAL*8 (A-H,O-Z)
C
C
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /ERROR/ IERR
      COMMON /DIM/ DZ(40),DZ1(40),DRO(40),DR1(40),DR2(40),DR3(40),
     *             DR4(40),NI,NJ,NIM1,NIM2,NJM1,NNI,NNJ,NNJJ
C
      DIMENSION EPSLON(9),RES(9)
      DIMENSION PN(NN),PO(NN),TVN(NN),TLN(NN),ALFAN(NN),ALFAO(NN),
     *          ALFAZ(NN),ALFAR(NN),RHOV(NN),RHOL(NN),HV(NN),HL(NN),
     *          HVZ(NN),HLZ(NN),HVR(NN),HLR(NN),
     *          UVZN(NN),ULZN(NN),UVRN(NN),ULRN(NN),
     *          FUVZN(NN),FULZN(NN),FUVRN(NN),FULRN(NN),
     *          W(NW),DV(NN),TS(NN),TW(NN),DTW(NN),
     *          HCONV(NN),HCONL(NN),HNB(NN),DPN(NN),
     *          A1(NN),A2(NN),A3(NN),A4(NN),YP(NN),B(NB),
     *          BETA(NN),GAMMA(NN),AVZD(NN),ALZD(NN),AVRD(NN),
     *          ALRD(NN),TCAN(NCAN)
C
      DIMENSION A(65),F(9),PROP(3,4),S(5,2),Q(4,2),K(30),M(30)
C
      IFLAG = 1
C
C                THE MOMENTUM EQUATIONS (Z-DIRECTION) AT THE BOTTON
C
      MM = NNJ + 2
      DO 4 KO = 2,MM,NI
      DO 1 L = 1,27
    1 K(L) = (L-1)*NN + KO
C
      IF(W(K(5)).GT.SMALL) GO TO 2
C
C
C           ONLY LIQUID PRESENT IN THE CELL
```

```
C
      V01 = (ONE-ALFAZ(KO))/DZ1(2)
      V05 = W(K(7))
C
      FUVZN(KO) = ZERO
      FULZN(KO) = -(W(K(7))*ULZN(KO) + (PN(KO)-PN(KO-1))*V01 +
     +            W(K(10)))/V05
      W(K(11)) = ZERO
      W(K(12)) = -V01/V05
      GO TO 4
C
    2 IF(W(K(7)).GT.SMALL) GO TO 3
C
C
C             ONLY VAPOR PRESENT IN THE CELL
C
      V02 = ALFAZ(KO)/DZ1(2)
      V03 = W(K(5))
C
      FUVZN(KO) = -(W(K(5))*UVZN(KO) + (PN(KO)-PN(KO-1))*V02 +
     +            W(K(9)))/V03
      FULZN(KO) = ZERO
      W(K(11)) = -V02/V03
      W(K(12)) = ZERO
      GO TO 4
C
C               BOTH PHASES PRESENT IN THE CELL
C
    3 CONTINUE
      V01 = (ONE-ALFAZ(KO))/DZ1(2)
      V02 = ALFAZ(KO)/DZ1(2)
      V03 = W(K(5))
      V04 = W(K(6))
      V05 = W(K(7))
      V06 = W(K(8))
      V07 = V04*V06 - V03*V05
C
      F(5) = W(K(5))*UVZN(KO) - W(K(6))*ULZN(KO) + (PN(KO)-PN(KO-1))*
     *       V02 + W(K(9))
      F(6) = W(K(7))*ULZN(KO) - W(K(8))*UVZN(KO) + (PN(KO)-PN(KO-1))*
     *       V01 + W(K(10))
C
      W(K(11))=(V05*V02+V04*V01)/V07
      W(K(12))=(V06*V02+V03*V01)/V07
      FUVZN(KO)=(F(5)*V05+F(6)*V04)/V07
```

```
1616        FULZN(KO)=(F(5)*V06+F(6)*V03)/V07
1617      4 CONTINUE
1618  C
1619  C             THE CENTRAL CELLS
1620  C
1621        A(4) = ZERO
1622        A(12)= ZERO
1623        A(20)= ZERO
1624        A(28)= ZERO
1625        DO 122 KO = 2,NIM1
1626        DO 5 L=1,27
1627      5 K(L) = (L-1)*NN+KO
1628        KM=KO+1
1629        KP = KO - 1
1630        CALL STATE (TVN(KO),TLN(KO),PN(KO),PROP,IFLAG)
1631        CALL NONEQ(ALFAO(KO),ALFAN(KO),TVN(KO),TLN(KO),PN(KO),
1632       *           RHOV(KO),RHOL(KO),TS(KO),S,IFLAG)
1633        CALL CONDT (TVN(KO),TLN(KO),PN(KO),ALFAO(KO),TS(KO),TW(KP),
1634       *            DTW(KP),HCONV(KP),HCONL(KP),HNB(KP),DV(KO),Q,KO)
1635        CALL IPHTC (HIF,ALFAN(KO))
1636  C
1637        V01=ALFAO(KO)/DT
1638        V02=(ONE-ALFAO(KO))/DT
1639        V03=ALFAN(KO)/DT
1640        V04=(ONE-ALFAN(KO))/DT
1641        V05=S(1,1)
1642        V06=S(2,1)
1643        V07=S(3,1)
1644        V08=S(4,1)
1645        V09=S(5,1)
1646        V10=W(K(3)+1)/DZ(KO)
1647        V11=W(K(3))/DZ(KO)
1648        V12=W(K(14))*DR1(1)
1649  C
1650        V14=W(K(4)+1)/DZ(KO)
1651        V15=W(K(4))/DZ(KO)
1652        V16=W(K(15))*DR1(1)
1653  C
1654        V18=HVZ(KM)*V10 + PO(KO)*AVZD(KM)/DZ(KO)
1655        V19=HVZ(KO)*V11 + PO(KO)/DZ(KO)*AVZD(KO)
1656        V20=HVR(KO)*V12 + PO(KO)*DR1(1)*AVRD(KO)
1657  C
1658        V22=HLZ(KM)*V14 + PO(KO)*ALZD(KM)/DZ(KO)
1659        V23=HLZ(KO)*V15 + PO(KO)*ALZD(KO)/DZ(KO)
1660        V24=HLR(KO)*V16 + PO(KO)*ALRD(KO)*DR1(1)
```

```
C
      V26=(TVN(KO)-TLN(KO))*HIF
      V27=V03*PROP(1,1)
      V28=V04*PROP(1,2)
      V29 = V01*RHOV(KO)
      V30 = V02*RHOL(KO)
      V31 = HV(KO)*V29
      V32 = HL(KO)*V30
C
C                THE RESIDUALS OF CONSERVATION EQUATIONS
C
      F(1) = V27 -V29 +UVZN(KM)*V10 -UVZN(KO)*V11 + UVRN(KO)*
     *       V12 -V05
      F(2) = PROP(1,3)*V27 - V31 +UVZN(KM)*V18 -
     *       UVZN(KO)*V19 + UVRN(KO)*V20 - S(1,2) - Q(1,1) +
     +       PO(KO)*(V03 - V01)
      F(3) = V28 - V30 + ULZN(KM)*V14 - ULZN(KO)*V15 +
     +       ULRN(KO)*V16 +V05
      F(4) = PROP(1,4)*V28 - V32 + ULZN(KM)*V22 -
     *       ULZN(KO)*V23 + ULRN(KO)*V24 + S(1,2) - Q(1,2) +
     +       PO(KO)*(V04 - V02)
      F(5) = W(K(5)+1)*UVZN(KM) -W(K(6)+1)*ULZN(KM) + (PN(KM)-PN(KO))*
     *       ALFAZ(KM)/DZ1(KM) + W(K(9)+1)
      F(6) = W(K(7)+1)*ULZN(KM) -W(K(8)+1)*UVZN(KM) + (PN(KM)-PN(KO))*
     *       (ONE-ALFAZ(KM))/DZ1(KM) + W(K(10)+1)
      F(7) = W(K(16))*UVRN(KO) - W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
     *       ALFAR(KO)/DR3(1) + W(K(20))
      F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
     *       (ONE-ALFAR(KO))/DR3(1) + W(K(21))
C
C
      A(1) =  PROP(1,1)/DT - V09
      A(9) =  (PROP(1,3)*PROP(1,1) + PO(KO))/DT - S(5,2)
      A(17) = -PROP(1,2)/DT + V09
      A(25) = -(PROP(1,4)*PROP(1,2) + PO(KO))/DT + S(5,2)
C
      A(2) =  PROP(2,1)*V03-V06
      A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
     -       Q(2,1) - S(2,2)
      A(18)=V06
      A(26) = S(2,2) - Q(2,2)
C
      A(3)  = -V07
      A(11) = - S(3,2)
      A(19)=PROP(2,2)*V04+V07
```

```
      A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*VQ4 -
     -       Q(3,2) + S(3,2)
C
C
      A(4) = ZERO
      A(12) = ZERO
      A(20) = ZERO
      A(28) = ZERO
C
      A(5)  = W(K(11))*V11
      A(13) = W(K(11))*V19
      A(21) = W(K(12))*V15
      A(29) = W(K(12))*V23
C
      A(7)  = W(K(11)+1)*V10
      A(15) = W(K(11)+1)*V18
      A(23) = W(K(12)+1)*V14
      A(31) = W(K(12)+1)*V22
      A(8) = W(K(22))*V12
      A(16) = W(K(22))*V20
      A(24) = W(K(23))*V16
      A(32) = W(K(23))*V24
C
      A(6)=PROP(3,1)*V03-V08-A(5)-A(7)-A(8)
      A(14)=(PROP(1,1)*PROP(3,3)+PROP(1,3)*PROP(3,1))*V03-
     -       S(4,2) - A(13) - A(15) - A(16)
      A(22)=PROP(3,2)*V04+V08-A(21)-A(23)-A(24)
      A(30)=(PROP(1,2)*PROP(3,4)+PROP(1,4)*PROP(3,2))*V04 - Q(4,2)
     *       + S(4,2) - A(29) - A(31) - A(32)
C
      IF(W(K(5)+1).GT.SMALL) GO TO 6
      FUVZN(KM) = ZERO
      FULZN(KM) = -F(6)/W(K(7)+1)
      GO TO 8
    6 IF(W(K(7)+1).GT.SMALL) GO TO 7
      FUVZN(KM) = -F(5)/W(K(5)+1)
      FULZN(KM) = ZERO
      GO TO 8
    7 CONTINUE
      FUVZN(KM) = -(W(K(7)+1)*F(5)+W(K(6)+1)*F(6))/W(K(13)+1)
      FULZN(KM) = -(W(K(8)+1)*F(5)+W(K(5)+1)*F(6))/W(K(13)+1)
    8 CONTINUE
      IF(W(K(16)).GT.SMALL) GO TO 9
      FUVRN(KO) = ZERO
      FULRN(KO) = -F(8)/W(K(18))
```

```
      GO TO 11
    9 CONTINUE
      IF(W(K(18)).GT.SMALL) GO TO 10
      FUVRN(KO) = -F(7)/W(K(16))
      FULRN(KO) = ZERO
      GO TO 11
   10 CONTINUE
      FUVRN(KO)=-(W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
      FULRN(KO)=-(W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
   11 CONTINUE
C
      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
C
C
C
      DO 111 L = 1,27
  111 K(L) = L*NN + KO
      IX2 = 1
      DO 12 IX1 = 8,24,8
      AUX = A(IX1+1)/A(1)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(1)*AUX
      DO 12 IX3 = 2,8
      IX4 = IX1 + IX3
   12 A(IX4) = A(IX4) - A(IX3)*AUX
      DO 13 L = 1,7
   13 B(K(L)) = -A(L+1)/A(1)
      B(KO) = F(1)/A(1)
C
      IF(DABS(A(10)).GT.SMALL) GO TO 16
C
C           ONLY LIQUID IN THE CELL
C
      B(K(8)) = ZERO
      B(K(9)) = ONE
      DO 14 L = 10,14
   14 B(K(L)) = ZERO
      B(K(15)) = F(4)/A(27)
      DO 15 L = 16,20
   15 B(K(L)) = -A(L+12)/A(27)
C
      AUX = A(19)/A(27)
```

```
      AUP = A(22) - A(30)*AUX
      A1(KP) = (A(20) - A(28)*AUX)/AUP
      A2(KP) = (A(21) - A(29)*AUX)/AUP
      A3(KP) = (A(23) - A(31)*AUX)/AUP
      A4(KP) = (A(24) - A(32)*AUX)/AUP
      YP(KP) = (F(3) - F(4)*AUX)/AUP
      GO TO 22
C
   16 CONTINUE
      IF(DABS(A(27)).GT.SMALL) GO TO 18
C
C           ONLY VAPOR IN THE CELL
C
      B(K(8)) = F(2)/A(10)
      B(K(15)) = B(K(8))
      B(K(9)) = ZERO
      DO 17 L = 10,14
      B(K(L)) = -A(L+2)/A(10)
      LL = L + 6
   17 B(K(LL)) = B(K(L))
C
      AUX = A(18)/A(10)
      AUP = A(22) - A(14)*AUX
      A1(KP) = (A(20) - A(12)*AUX)/AUP
      A2(KP) = (A(21) - A(13)*AUX)/AUP
      A3(KP) = (A(23) - A(15)*AUX)/AUP
      A4(KP) = (A(24) - A(16)*AUX)/AUP
      YP(KP) = (F(3) - F(2)*AUX)/AUP
      GO TO 22
C
C             BOTH PHASES PRESENT
C
   18 CONTINUE
      B(K(8)) = F(2)/A(10)
      DO 19 L = 9,14
   19 B(K(L)) = -A(L+2)/A(10)
C
      IX2 = 2
      DO 20 IX1 = 18,26,8
      AUX = A(IX1)/A(10)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(2)*AUX
      DO 20 IX3 = 1,6
      IX4 = IX1 + IX3
      IX5 = IX3 + 10
```

```
1841     20 A(IX4) = A(IX4) - A(IX5)*AUX
1842 C
1843 C
1844        B(K(15)) = F(3)/A(19)
1845        DO 21 L = 16,20
1846     21 B(K(L)) = -A(L+4)/A(19)
1847 C
1848        AUX = A(27)/A(19)
1849        AUP = A(30) - A(22)*AUX
1850        A1(KP) = (A(28) - A(20)*AUX)/AUP
1851        A2(KP) = (A(29) - A(21)*AUX)/AUP
1852        A3(KP) = (A(31) - A(23)*AUX)/AUP
1853        A4(KP) = (A(32) - A(24)*AUX)/AUP
1854        YP(KP) = (F(4) - F(3)*AUX)/AUP
1855 C
1856     22 CONTINUE
1857 C
1858        DDT = DABS(A1(KP)) + DABS(A2(KP)) + DABS(A3(KP)) +DABS(A4(KP))
1859        IF(DDT.GT.ONE) GO TO 58
1860 C
1861    122 CONTINUE
1862 C
1863 C              OUT OF THE BOUNDARIES
1864 C
1865        DO 46 J=NI,NNJ,NI
1866        JO=J/NI+1
1867        DO 46 I=2,NIM1
1868        KO=I+J
1869        KM = KO + 1
1870        KP = KO -1 - J/NI*2
1871        DO 23 L=1,27
1872     23 K(L) = (L-1)*NN+KO
1873        IO=I+1
1874 C
1875        CALL STATE(TVN(KO),TLN(KO),PN(KO),PROP,IFLAG)
1876        CALL NONEQ(ALFAO(KO),ALFAN(KO),TVN(KO),TLN(KO),PN(KO),
1877       *           RHOV(KO),RHOL(KO),TS(KO),S,IFLAG)
1878        CALL CONDT(TVN(KO),TLN(KO),PN(KO),ALFAO(KO),TS(KO),TW(KP),
1879       *           DTW(KP),HCONV(KP),HCONL(KP),HNB(KP),DV(KO),Q,KO)
1880        CALL IPHTC(HIF,ALFAN(KO))
1881 C
1882        V01 = ALFAO(KO)/DT
1883        V02 = (ONE-ALFAO(KO))/DT
1884        V03 = ALFAN(KO)/DT
1885        V04 = (ONE-ALFAN(KO))/DT
```

```
      V05 = S(1,1)
      V06 = S(2,1)
      V07 = S(3,1)
      V08 = S(4,1)
      V09 = S(5,1)
      V10 = W(K(3)+1)/DZ(I)
      V11 = W(K(3))/DZ(I)
      V12 = W(K(14))*DR1(JO)
      V13 = W(K(14)-NI)*DR2(JO)
      V14 = W(K(4)+1)/DZ(I)
      V15 = W(K(4))/DZ(I)
      V16 = W(K(15))*DR1(JO)
      V17 = W(K(15)-NI)*DR2(JO)
      V18 = HVZ(KM)*V10 + PO(KO)/DZ(I)*AVZD(KM)
      V19 = HVZ(KO)*V11 + PO(KO)/DZ(I)*AVZD(KO)
      V20 = HVR(KO)*V12 + PO(KO)*DR1(JO)*AVRD(KO)
      V21 = HVR(KO-NI)*V13 + PO(KO)*DR2(JO)*AVRD(KO-NI)
      V22 = HLZ(KM)*V14 + PO(KO)*ALZD(KM)/DZ(I)
      V23 = HLZ(KO)*V15 + PO(KO)*ALZD(KO)/DZ(I)
      V24 = HLR(KO)*V16 + PO(KO)*ALRD(KO)*DR1(JO)
      V25 = HLR(KO-NI)*V17 + PO(KO)*ALRD(KO-NI)*DR2(JO)
      V26 = (TVN(KO)-TLN(KO))*HIF
      V27 = V03*PROP(1,1)
      V28 = V04*PROP(1,2)
      V29 = V01*RHOV(KO)
      V31 = V02*RHOL(KO)
      V31 = HV(KO)*V29
      V32 = HL(KO)*V30
C
C
C
      F(1) = V27 - V29 + UVZN(KM)*V10 - UVZN(KO)*V11 +
     +       UVRN(KO)*V12 - UVRN(KO-NI)*V13 - V05
      F(2) = PROP(1,3)*V27 - V31 +UVZN(KM)*V18 -
     -       UVZN(KO)*V19 + UVRN(KO)*V20 - UVRN(KO-NI)*V21 -
     -       S(1,2) - Q(1,1) + PO(KO)*(V03 - V01)
      F(3) = V28 - V30 + ULZN(KM)*V14 - ULZN(KO)*V15 +
     +       ULRN(KO)*V16 - ULRN(KO-NI)*V17 + V05
      F(4) = PROP(1,4)*V28 - V32 + ULZN(KM)*V22 -
     +       ULZN(KO)*V23 +ULRN(KO)*V24 - ULRN(KO-NI)*V25 +
     +        S(1,2) - Q(1,2) + PO(KO)*(V04 - V02)
      F(5) = W(K(5)+1)*UVZN(KM) - W(K(6)+1)*ULZN(KM) +
     +       (PN(KM)-PN(KO))*ALFAZ(KM)/DZ1(I+1) + W(K(9)+1)
      F(6) = W(K(7)+1)*ULZN(KM) - W(K(8)+1)*UVZN(KM) +
     +       (PN(KM)-PN(KO))*(ONE-ALFAZ(KM))/DZ1(I+1) + W(K(10)+1)
```

```
C
      IF(J.LT.NNJ) GO TO 24
      CALL HEXCAN(TCAN(I),TCAN(NI+I),TVN(KO),TLN(KO),HCONV(KP),
     *            HCONL(KP),QVC,QLC,DQCDTV,DQCDTL)
      F(2) = F(2) + QVC
      F(4) = F(4) + QLC
      F(7) = ZERO
      F(8) = ZERO
      GO TO 25
   24 CONTINUE
      F(7) = W(K(16))*UVRN(KO) -W(K(17))*ULRN(KO) + (PN(KO+NI)-PN(KO))*
     *       ALFAR(KO)/DR3(JO) + W(K(20))
      F(8) = W(K(18))*ULRN(KO) - W(K(19))*UVRN(KO) + (PN(KO+NI)-PN(KO))*
     *       (ONE-ALFAR(KO))/DR3(JO) + W(K(21))
   25 CONTINUE
C
C
C
      A(1)=PROP(1,1)/DT - V09
      A(2)=PROP(2,1)*V03-V06
      A(3)=-V07
      A(4)=W(K(22)-NI)*V13
C
      A(9)=(PROP(1,3)*PROP(1,1) + PO(KO))/DT - S(5,2)
      A(10)=(PROP(1,1)*PROP(2,3)+PROP(1,3)*PROP(2,1))*V03 -
     -       Q(2,1) - S(2,2)
      A(11)= - S(3,2)
      A(12)=W(K(22)-NI)*V21
C
      A(17)=-PROP(1,2)/DT + V09
      A(18)=V06
      A(19)=PROP(2,2)*V04+V07
      A(20)=W(K(23)-NI)*V17
C
      A(25) = -(PROP(1,4)*PROP(1,2) + PO(KO))/DT + S(5,2)
      A(26)= S(2,2)
      A(27)=(PROP(1,2)*PROP(2,4)+PROP(1,4)*PROP(2,2))*V04 -
     -      Q(3,2) + S(3,2)
      A(28)=W(K(23)-NI)*V25
C
      A(5)  = W(K(11))*V11
      A(13) = W(K(11))*V19
      A(21) = W(K(12))*V15
      A(29) = W(K(12))*V23
C
```

```
      A(7)   = W(K(11)+1)*V10
      A(15)  = W(K(11)+1)*V18
      A(23)  = W(K(12)+1)*V14
      A(31)  = W(K(12)+1)*V22
C
C
      IF(J.GE.NNJ) GO TO 125
      A(8)   = W(K(22))*V12
      A(16)  = W(K(22))*V20
      A(24)  = W(K(23))*V16
      A(32)  = W(K(23))*V24
C
      GO TO 225
C
  125 A(8) = ZERO
      A(16) = ZERO
      A(24) = ZERO
      A(32) = ZERO
      A(10) = A(10) + DQCDTV
      A(27) = A(27) + DQCDTL
  225 CONTINUE
C
C
C
      A(6) = PROP(3,1)*V03 - V08 - A(4) - A(5) - A(7) - A(8)
      A(14)=(PROP(1,1)*PROP(3,3)+PROP(1,3)*PROP(3,1))*V03-
     -      S(4,2) - A(12) - A(13) - A(15) - A(16)
      A(22)=PROP(3,2)*V04+V08-A(20)-A(21)-A(23)-A(24)
      A(30)=(PROP(1,2)*PROP(3,4)+PROP(1,4)*PROP(3,2))*V04-Q(4,2)-
     *       A(28)-A(29)-A(31)-A(32) + S(4,2)
C
C
C
      IF(W(K(5)+1).GT.SMALL) GO TO 26
      FUVZN(KM) = ZERO
      FULZN(KM) = -F(6)/W(K(7)+1)
      GO TO 28
   26 IF(W(K(7)+1).GT.SMALL) GO TO 27
      FUVZN(KM) = -F(5)/W(K(5)+1)
      FULZN(KM) = ZERO
      GO TO 28
   27 CONTINUE
      FUVZN(KM) = -(W(K(7)+1)*F(5)+W(K(6)+1)*F(6))/W(K(13)+1)
      FULZN(KM) = -(W(K(8)+1)*F(5)+W(K(5)+1)*F(6))/W(K(13)+1)
   28 CONTINUE
```

```
      IF(JO.EQ.NJ) GO TO 31
      IF(W(K(16)).GT.SMALL) GO TO 29
      FUVRN(KO) = ZERO
      FULRN(KO) = -F(8)/W(K(18))
      GO TO 31
   29 CONTINUE
      IF(W(K(18)).GT.SMALL) GO TO 30
      FUVRN(KO) = -F(7)/W(K(16))
      FULRN(KO) = ZERO
      GO TO 31
   30 CONTINUE
      FUVRN(KO)=-(W(K(18))*F(7)+W(K(17))*F(8))/W(K(24))
      FULRN(KO)=-(W(K(19))*F(7)+W(K(16))*F(8))/W(K(24))
   31 CONTINUE
C
      F(1)=-F(1)-FUVZN(KM)*V10+FUVZN(KO)*V11-FUVRN(KO)*V12
     +     + FUVRN(KO-NI)*V13
      F(2)=-F(2)-FUVZN(KM)*V18+FUVZN(KO)*V19-FUVRN(KO)*V20
     +     + FUVRN(KO-NI)*V21
      F(3)=-F(3)-FULZN(KM)*V14+FULZN(KO)*V15-FULRN(KO)*V16
     +     + FULRN(KO-NI)*V17
      F(4)=-F(4)-FULZN(KM)*V22+FULZN(KO)*V23-FULRN(KO)*V24
     +     + FULRN(KO-NI)*V25
C
C
C
      DO 32 L = 1,27
   32 K(L) = L*NN + KO
      IX2 = 1
      DO 33 IX1 = 8,24,8
      AUX = A(IX1+1)/A(1)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(1)*AUX
      DO 33 IX3 = 2,8
      IX4 = IX1 + IX3
   33 A(IX4) = A(IX4) - A(IX3)*AUX
      DO 34 L = 1,7
   34 B(K(L)) = -A(L+1)/A(1)
      B(KO) = F(1)/A(1)
C
      IF(DABS(A(10)).GT.SMALL) GO TO 37
C
C           ONLY LIQUID IN THE CELL
C
      B(K(8)) = ZERO
```

```
      B(K(9)) = ONE
      DO 35 L = 10,14
   35 B(K(L)) = ZERO
      B(K(15)) = F(4)/A(27)
      DO 36 L = 16,20
   36 B(K(L)) = -A(L+12)/A(27)
C
      AUX = A(19)/A(27)
      AUP = A(22) - A(30)*AUX
      A1(KP) = (A(20) - A(28)*AUX)/AUP
      A2(KP) = (A(21) - A(29)*AUX)/AUP
      A3(KP) = (A(23) - A(31)*AUX)/AUP
      A4(KP) = (A(24) - A(32)*AUX)/AUP
      YP(KP) = (F(3) - F(4)*AUX)/AUP
      GO TO 43
C
   37 CONTINUE
      IF(DABS(A(27)).GT.SMALL) GO TO 39
C
C           ONLY VAPOR IN THE CELL
C
      B(K(8)) = F(2)/A(10)
      B(K(15)) = B(K(8))
      B(K(9)) = ZERO
      DO 38 L = 10,14
      B(K(L)) = -A(L+2)/A(10)
      LL = L + 6
   38 B(K(LL)) = B(K(L))
C
      AUX = A(18)/A(10)
      AUP = A(22) - A(14)*AUX
      A1(KP) = (A(20) - A(12)*AUX)/AUP
      A2(KP) = (A(21) - A(13)*AUX)/AUP
      A3(KP) = (A(23) - A(15)*AUX)/AUP
      A4(KP) = (A(24) - A(16)*AUX)/AUP
      YP(KP) = (F(3) - F(2)*AUX)/AUP
      GO TO 43
C
C            BOTH PHASES PRESENT
C
   39 CONTINUE
      B(K(8)) = F(2)/A(10)
      DO 40 L = 9,14
   40 B(K(L)) = -A(L+2)/A(10)
C
```

```
      IX2 = 2
      DO 41 IX1 = 18,26,8
      AUX = A(IX1)/A(10)
      IX2 = IX2 + 1
      F(IX2) = F(IX2) - F(2)*AUX
      DO 41 IX3 = 1,6
      IX4 = IX1 + IX3
      IX5 = IX3 + 10
   41 A(IX4) = A(IX4) - A(IX5)*AUX
C
C
      B(K(15)) = F(3)/A(19)
      DO 42 L = 16,20
   42 B(K(L)) = -A(L+4)/A(19)
C
      AUX = A(27)/A(19)
      AUP = A(30) - A(22)*AUX
      A1(KP) = (A(28) - A(20)*AUX)/AUP
      A2(KP) = (A(29) - A(21)*AUX)/AUP
      A3(KP) = (A(31) - A(23)*AUX)/AUP
      A4(KP) = (A(32) - A(24)*AUX)/AUP
      YP(KP) = (F(4) - F(3)*AUX)/AUP
C
   43 CONTINUE
C
      DDT = DABS(A1(KP)) + DABS(A2(KP)) + DABS(A3(KP)) +DABS(A4(KP))
      IF(DDT.GT.ONE) GO TO 58
C
   46 CONTINUE
C
      CALL GAUSIE(A1,A2,A3,A4,YP,DPN,BETA,GAMMA,NN)
C
C              CELL (2,1)
C
      KO = 2
      KP = KO - 1
      KQ = KP + NIM2
      DO 47 L = 1,27
      M(L) = (L-1)*NN + KO
   47 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KO) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
```

```
     +       B(K(12))*DPN(KP) + B(K(13))*DPN(KO) +
     +       B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +       B(K(5))*DPN(KP) + B(K(6))*DPN(KO) +
     +       B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
      UVZN(1)  = UVZN(KO)
      ULZN(1)  = ULZN(KO)
C
C                  CELLS (I,1) , I=3,NI-2
C
      DO 49 I = 3,NIM2
      KO = I
      KP = KO - 1
      KM = KO
      KQ = KP + NIM2
      KR = KP - 1
      DO 48 L = 1,27
      M(L) = (L-1)*NN + KO
   48 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(17))*DPN(KR) +
     +       B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
     +       B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
     +       B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +       B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
```

```
     +       B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
   49 CONTINUE
C
C                  CELL (NIM1,1)
C
      KO = NIM1
      KP = KO - 1
      KQ = KP + NIM2
      KR = KP - 1
      DO 148 L = 1,27
      M(L) = (L-1)*NN + KO
  148 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
```

```
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
C
C                  CELLS (2,J) , J = 2,NJ-1
C
C
      DO 51 J = NI,NNJJ,NI
      KO = J+2
      KP = KO - 1 - 2*J/NI
      KM = KP + 1
      KQ = KP + NIM2
      KR = KP - 1
      KS = KP - NIM2
      DO 50 L = 1,27
      M(L) = (L-1)*NN + KO
   50 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
```

```
2291      +           HNB(KP)*DTS)*DTW(KP)
2292 C
2293 C
2294       UVZN(KO) = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
2295       ULZN(KO)  = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
2296       UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
2297       ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
2298       UVZN(KO-1) = UVZN(KO)
2299       ULZN(KO-1) = ULZN(KO)
2300    51 CONTINUE
2301 C
2302 C          CELLS (I,J) , I=3,NI-2 , J=2,NJ-1
2303 C
2304       DO 53 J = NI,NNJJ,NI
2305       DO 53 I = 3,NIM2
2306       KO = I+J
2307       KP = KO - 1 - 2*J/NI
2308       KM = KP + 1
2309       KQ = KP +NIM2
2310       KR = KP - 1
2311       KS = KP - NIM2
2312       DO 52 L = 1,27
2313       M(L) = (L-1)*NN + KO
2314    52 K(L) = L*NN+KO
2315 C
2316 C
2317       DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
2318      +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM) + B(K(20))*DPN(KQ)
2319       DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
2320      +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM) +
2321      +      B(K(14))*DPN(KQ)
2322       DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
2323      +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM) +
2324      +      B(K(7))*DPN(KQ)
2325       PN(KO)  = PN(KO) + DPN(KP)
2326       IF(PN(KO).LT.1.D+04) GO TO 59
2327       IF(PN(KO).GT.4.D+07) GO TO 60
2328       TLN(KO) = TLN(KO) + DTL
2329       TVN(KO) = TVN(KO) + DTV
2330       ALFAN(KO) = ALFAN(KO) + DAL
2331       TX = SAT(PN(KO))
2332       DTS = TX - TS(KO)
2333       TS(KO) = TX
2334       TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
2335      +          HNB(KP)*DTS)*DTW(KP)
```

```
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
   53 CONTINUE
C
C          CELLS (NIM1,J) ,  J=2,NJ-1
C
      DO 153 J = NI,NNJJ,NI
      I = NIM1
      KO = I+J
      KP = KO - 1 - 2*J/NI
      KQ = KP +NIM2
      KR = KP - 1
      KS = KP - NIM2
      DO 152 L = 1,27
      M(L) = (L-1)*NN + KO
  152 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP)  + B(K(20))*DPN(KQ)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) +
     +      B(K(14))*DPN(KQ)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) +
     +      B(K(7))*DPN(KQ)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
```

```
      UVRN(KO) = W(M(22))*(DPN(KQ)-DPN(KP)) + FUVRN(KO) + UVRN(KO)
      ULRN(KO) = W(M(23))*(DPN(KQ)-DPN(KP)) + FULRN(KO) + ULRN(KO)
  153 CONTINUE
C
C           CELLS (I,NJ) , I=3,NI-1
C
      DO 55 I = 3,NIM1
      KO = I + NNJ
      KP = KO + 1 - 2*NJ
      KM = KP + 1
      KQ = KP + NIM2
      KR = KP -1
      KS = KP - NIM2
      DO 54 L = 1,27
      M(L) = (L-1)*NN + KO
   54 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) + B(K(17))*DPN(KR) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(11))*DPN(KR) + B(K(12))*DPN(KP) + B(K(13))*DPN(KM)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(4))*DPN(KR) + B(K(5))*DPN(KP) + B(K(6))*DPN(KM)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +          HNB(KP)*DTS)*DTW(KP)
      TCAN(I) = TCAN(I) + TCAN(NI + I)*(HCONV(KP)*DTV +
     +           HCONL(KP)*DTL)
C
C
      UVZN(KO) = W(M(11))*(DPN(KP)-DPN(KR))  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*(DPN(KP)-DPN(KR))  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = ZERO
      ULRN(KO) = ZERO
   55 CONTINUE
C
```

```
C          CELLS (NI,J) , J=1,NJ
C
      DO 57 KO = NI,NN,NI
      KR = KO -2*KO/NI
      DO 56 L = 1,27
      M(L) = (L-1)*NN + KO
   56 K(L) = L*NN+KO
C
C
      UVZN(KO) = FUVZN(KO) - W(M(11))*DPN(KR) + UVZN(KO)
      ULZN(KO) = FULZN(KO) - W(M(12))*DPN(KR) + ULZN(KO)
      TLN(KO) = TLN(KO-1)
      TVN(KO) = TVN(KO-1)
      ALFAN(KO) = ALFAN(KO-1)
   57 CONTINUE
C
C               CELL (2,NJ)
      KO = NNJ + 2
      KP = KO + 1 - 2*NJ
      KR = KP - 1
      KS = KP - NIM2
      KM = KP + 1
      DO 561 L = 1,27
      M(L) = (L-1)*NN + KO
  561 K(L) = L*NN+KO
C
C
      DTL = B(K(15)) + B(K(16))*DPN(KS) +
     +      B(K(18))*DPN(KP) + B(K(19))*DPN(KM)
      DTV = B(K(8)) + B(K(9))*DTL + B(K(10))*DPN(KS) +
     +      B(K(13))*DPN(KM) + B(K(12))*DPN(KP)
      DAL = B(KO) + B(K(1))*DTV + B(K(2))*DTL + B(K(3))*DPN(KS) +
     +      B(K(6))*DPN(KM) + B(K(5))*DPN(KP)
      PN(KO)  = PN(KO) + DPN(KP)
      IF(PN(KO).LT.1.D+04) GO TO 59
      IF(PN(KO).GT.4.D+07) GO TO 60
      TLN(KO) = TLN(KO) + DTL
      TVN(KO) = TVN(KO) + DTV
      ALFAN(KO) = ALFAN(KO) + DAL
      TX = SAT(PN(KO))
      DTS = TX - TS(KO)
      TS(KO) = TX
      TW(KP) = TW(KP) + (HCONV(KP)*DTV + HCONL(KP)*DTL +
     +         HNB(KP)*DTS)*DTW(KP)
      TCAN(2) = TCAN(2) + TCAN(NI + 2)*(HCONV(KP)*DTV +
```

```
     +          HCONL(KP)*DTL)
C
C
      UVZN(KO)    = W(M(11))*DPN(KP)  + FUVZN(KO) + UVZN(KO)
      ULZN(KO) = W(M(12))*DPN(KP)  + FULZN(KO) + ULZN(KO)
      UVRN(KO) = ZERO
      ULRN(KO) = ZERO
      UVZN(KO-1) = UVZN(KO)
      ULZN(KO-1) = ULZN(KO)
C
      DO 357 KO = 1,NN
      IF(ALFAN(KO).GE.ZERO) GO TO 257
      IF(ALFAN(KO).LT.-1.D-05) IERR = 3
      ALFAN(KO) = ZERO
  257 CONTINUE
      IF(ALFAN(KO).LE.ONE) GO TO 2257
      IF(ALFAN(KO).GT.1.00001) IERR = 3
      ALFAN(KO) = ONE
 2257 CONTINUE
      IF(TVN(KO).LT.4.D+02) IERR = 14
      IF(TVN(KO).GT.3.D+03) IERR = 15
      IF(TLN(KO).LT.4.D+02) IERR = 16
      IF(TLN(KO).GT.3.D+03) IERR = 17
  357 CONTINUE
      RETURN
   58 IERR = 2
      RETURN
   59 IERR = 12
      RETURN
   60 IERR = 13
      RETURN
      END
```

```
2503        SUBROUTINE COEFF(TV,TL,UVZ,UVR,ULZ,ULR,ALFAZ,ALFAR,
2504       *                 RHOVZ,RHOVR,RHOLZ,RHOLR,DH,DV,QSI,
2505       *                 SPPD,WZ1,WZ2,WR1,WR2,FRVZ,FRLZ,FRVR,
2506       *                 FRLR,C1Z,C1R)
2507        IMPLICIT REAL*8 (A-H,O-Z)
2508        COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2509        DATA TWO,PTWO,ADRY,CADRY/2.D0,.2D0,.957D0,0.043D0/
2510 C
2511 C            SUBROUTINE COEFF CALCULATES THE MOMENTUM EXCHANGE
2512 C            COEFFICIENTS.
2513 C            C1. ARE THE INTERPHASE MOMENTUM EXCHANGE COEFFICIENTS
2514 C            FOR THE TWO DIRECTIONS.
2515 C            FR.. ARE THE WALL FRICTION COEFFICIENTS FOR BOTH PHASES
2516 C            AND DIRECTIONS.
2517 C
2518        VV = VISCV (TV)
2519        VL = VISCL (TL)
2520 C
2521        AUVZ = DABS (UVZ)
2522        AUVR = DABS (UVR)
2523        AULZ = DABS (ULZ)
2524        AULR = DABS (ULR)
2525 C
2526        REVZ = WZ1*AUVZ*DH/VV +SMALL
2527        RELZ = RHOLZ*AULZ*DH/VL + SMALL
2528        REVR = WR1*AUVR*QSI*DH/VV + SMALL
2529        RELR = WR2*AULR*QSI*DH/VL + SMALL
2530 C
2531        FVZ = 0.180D0/REVZ**PTWO + SPPD*DH
2532        FLZ = 0.180D0/RELZ**PTWO + SPPD*DH
2533        FVR = PTWO/REVR**PTWO
2534        FLR = PTWO/RELR**PTWO
2535 C
2536        FRVZ = (ALFAZ - ADRY)/CADRY*RHOVZ*AUVZ*FVZ/TWO/DH
2537        FRVR = (ALFAR - ADRY)/CADRY*180.*VV/(DH*DH)*QSI
2538        FRLZ =       RHOLZ*AULZ*FLZ/TWO/DH
2539        FRLR =180.*VL/(DH*DH)*QSI
2540        XZ = (ONE - ALFAZ)/CADRY
2541        XR = (ONE - ALFAR)/CADRY
2542 C
2543        IF(ALFAZ.GT.ADRY) GO TO 1
2544        FRVZ = ZERO
2545        XZ = ONE
```

```
2546      1 CONTINUE
2547        IF(ALFAR.GT.ADRY) GO TO 2
2548        FRVR = ZERO
2549        XR = ONE
2550      2 CONTINUE
2551 C
2552        FRLZ = FRLZ*XZ
2553        FRLR = FRLR*XR
2554 C
2555        X = (ONE + (ONE-ALFAZ)*75.00)**.95*4.31
2556 C
2557        C1Z = ((ONE - ALFAZ)*DABS(UVZ - ULZ)*RHOVZ/TWO +
2558       +       VL/DH)*X/DH
2559        C1R = ((ONE - ALFAR)*DABS(UVR - ULR)*RHOVR/TWO +
2560       +       VL/DH)*X*QSI*QSI/DH
2561 C
2562        RETURN
2563        END
```

Subroutine bc

```
2564        SUBROUTINE BC(P,TV,TL,ALFA,TIME,UL,NN,NI,NIM1)
2565        IMPLICIT REAL*8 (A-H,O-Z)
2566        LOGICAL LP
2567        COMMON /BCX/ ULO
2568        COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
2569       *               PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
2570       *               ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
2571       *               TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
2572       *               OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
2573       *               LMAX,LP(51)
2574        DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN)
2575 C
2576 C
2577        L = 2
2578      1 CONTINUE
2579        IF(TIME.LE.TB(L)) GO TO 2
2580        L = L + 1
2581        IF(L.GT.LMAX) RETURN
2582        GO TO 1
2583      2 CONTINUE
2584        DTIME = TIME - TB(L-1)
2585 C
2586        PNB = PNB1(L)*DTIME + PNB2(L)
2587        PNT = PNT1(L)*DTIME + PNT2(L)
2588        ALB = ALB1(L)*DTIME + ALB2(L)
2589        TVB = TVB1(L)*DTIME + TVB2(L)
2590        TLB = TLB1(L)*DTIME + TLB2(L)
2591 C
2592        IF(LP(L)) GO TO 3
2593 C
2594        PNB = DEXP(OMP(L)*DTIME)*PNB + PNB3(L)
2595        PNT = DEXP(OMT(L)*DTIME)*PNT + PNT3(L)
2596        ALB = DEXP(OMA(L)*DTIME)*ALB + ALB3(L)
2597        TVB = DEXP(OMV(L)*DTIME)*TVB + TVB3(L)
2598        TLB = DEXP(OML(L)*DTIME)*TLB + TLB3(L)
2599 C
2600      3 CONTINUE
2601        DO 4 J = NI,NN,NI
2602        KO = J - NIM1
2603 C
2604        P(KO) = PNB
2605        P(J)  = PNT
2606        ALFA(KO) = ALB
```

```
2607          TV(KO) = TVB
2608          TL(KO) = TLB
2609        4 CONTINUE
2610          RETURN
2611          END
```

Function viscl

```
2612        FUNCTION VISCL(T)
2613        IMPLICIT REAL*8 (A-H,O-Z)
2614  C
2615  C             FUNCTION VISCL RETURNS THE SODIUM LIQUID VISCOSITY
2616  C             IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
2617  C             IN DEGREE CELSIUS
2618  C
2619        TK = T
2620        VISCL = DEXP(508.07/TK - 5.7316 - .4925*DLOG(TK))
2621        RETURN
2622        END
```

```
      FUNCTION VISCV(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           FUNCTION VISCV RETURNS THE SODIUM VAPOR VISCOSITY
C           IN (KG/M/SEC),AS A FUNCTION OF THE TEMPERATURE
C           IN DEGREE CELSIUS
C
      TK = T
      VISCV = 6.085D-09*TK + 1.261D-05
      RETURN
      END
```

```
      FUNCTION SURTEN (T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           FUNCTION SURTEN RETURNS THE SURFACE TENSION OF LIQUID
C           SODIUM IN NEWTON/METER
C           CORRELATION FROM GOLDEN AND TOKAR.
C
      TC = T - 273.14
      SURTEN = 2.067D-01 - 1.00-04*TC
      IF(SURTEN.LT.0.D0) SURTEN = 0.D0
      RETURN
      END
```

Function sat

```
2646        FUNCTION SAT(P)
2647        IMPLICIT REAL*8 (A-H,O-Z)
2648 C
2649        SAT = 12020./(21.9358 - DLOG(P))
2650        RETURN
2651        END
```

```
      FUNCTION DTSDP(P)
      IMPLICIT REAL*8 (A-H,O-Z)
C
C           CALCULATES THE DERIVATIVE OF THE SATURATION
C           TEMPERTURE WITH RESPECT TO THE PRESSURE
C
      X = 21.9358 - DLOG(P)
      DTSDP = 12020./(X*X*P)
      RETURN
      END
```

```
      FUNCTION CONDL(T)
      IMPLICIT REAL*8 (A-H,O-Z)
      DATA A1,A2,A3,X1,X2,X3 /54.306,-1.878D-02,2.0914D-06,1.8D0,
     *                        459.67D0,1.7307D0/
C
      TF = X1*T - X2
      T2 = TF*TF
      C = A1 + A2*TF + A3*T2
      CONDL = C*X3
      RETURN
      END
```

Function condv

```
2673        FUNCTION CONDV(T)
2674        IMPLICIT REAL*8 (A-H,O-Z)
2675        DATA A1,A2,A3,X1,X2,X3 /16.39D-04,3.977D-05,-9.697D-09,
2676       *       1.8D0,459.67D0,1.7307D0/
2677 C
2678        TF = X1*T - X2
2679        T2 = TF*TF
2680        C = A1 + A2*TF + A3*T2
2681        CONDV = X3*C
2682        RETURN
2683        END
```

Function cpl

```
2684        FUNCTION CPL(T)
2685        IMPLICIT REAL*8 (A-H,O-Z)
2686        DATA A1,A2,A3,X1,X2 /.389352D0,1.10599D-04,3.41178D-08,
2687       *     1.8D0,4.1869D+03/
2688 C
2689        TR = T*X1
2690        T2 = TR*TR
2691        CP = A1 - A2*TR + A3*T2
2692        CPL = X2*CP
2693        RETURN
2694        END
```

Function prv

```
      FUNCTION PRV(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      TX = T - 844.1
      PRV = .7596D0 + .810D-06*TX*TX
      RETURN
      END
```

Function prl

```
      FUNCTION PRL(T)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      PRL = CPL(T)*VISCL(T)/CONDL(T)
      RETURN
      END
```

Function hfg

```
      FUNCTION HFG(P)
      IMPLICIT REAL*8 (A-H,O-Z)
C
      T = SAT(P)
      HFG = 5.089D+06 - 1.043D+03*T
      RETURN
      END
```

Subroutine htcf

```
      SUBROUTINE HTCF (P,TV,TL,ALFA,RHOV,RHOL,HV,HL,DH,TS,TW,
     *                 HCONV,HCONL,HNB,UV,UL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /POVERD/ R
C
      HCONV = ZERO
      HCONL = ZERO
      HNB   = ZERO
C
      VV = VISCV(TV)
      VL = VISCL(TL)
      PV = PRV(TV)
      PL = PRL(TL)
      CV = CONDV(TV)
      CL = CONDL(TL)
      AUV = DABS(UV)
      AUL = DABS(UL)
      SIG = SURTEN(TL)
C
C                 COMPUTE QUALITY
C
      GV = ALFA*RHOV*AUV
      GL = (ONE-ALFA)*RHOL*AUL
      G  = GV + GL
      IF((UV-UL)*UL.LE.ZERO) GO TO 1
      X = GV/G
      GO TO 2
    1 CONTINUE
      X = ALFA*RHOV/(ALFA*RHOV + (ONE-ALFA)*RHOL)
    2 CONTINUE
C
C                 SINGLE PHASE : DITTUS-BOELTER CORRELATION (VAPOR)
C
      IF(ALFA.LE.0.96) GO TO 3
      REV = RHOV*AUV*DH/VV
      HCONV = 0.023*REV**0.8*PV**0.4*CV/DH
      RETURN
    3 CONTINUE
C
C                 SINGLE PHASE : SCHAD CORRELATION (LIQUID)
C
      REL = RHOL*AUL*DH/VL
```

```
      PEL = REL*PL
      IF(PEL.LE.150.) GO TO 4
      HCONL = PEL**0.3*R*CL/DH
      GO TO 5
    4 CONTINUE
      HCONL = 4.5*R*CL/DH
    5 CONTINUE
C
C                TWO PHASES : CHEN CORRELATION
C
      XTTI = (X/(ONE-X))**0.9*(RHOL/RHOV)**0.5*(VV/VL)**0.1
      F = (XTTI + .213)**0.736*2.35D0
      IF(F.LT.ONE) RETURN
      HCONL = F**0.375*HCONL
C
      IF(TW.LE.TL) GO TO 7
C
      FX = ONE
      GX = G
      IF(TL.LT.TS) GO TO 7
      IF(XTTI.GT.0.1) FX = F
      GX = GL
    6 CONTINUE
      REL = GX*DH/VL
      RETP = REL*FX**1.25*1.D-04
      S = 0.1D0
      IF(RETP.LT.70.D0.AND.RETP.GE.32.5D0) S = ONE/
     /      (ONE + RETP**0.78*0.42D0)
C
      IF(RETP.LT.32.5D0) S = ONE/(ONE + .12D0*RETP**1.14)
C
      HS = 1.22D-03*S*DSQRT(CL*CPL(TL)/SIG)/PL**.29*
     *     RHOL**.25*(CPL(TL)*RHOL/RHOV/HFG(P))**.24
C
      PWALL = DEXP(21.9358D0 - 12020.D0/TW)
      Z = DABS(PWALL - P)
C
      HNB = HS*(TW - TS)**.24*Z**.75
    7 CONTINUE
      IF(ALFA.LE.0.88) RETURN
C
      FAL = 12.D0 - 12.5D0*ALFA
      FAL = FAL*FAL*FAL
      REV = RHOV*AUV*DH/VV
      HCV = 0.023*(REV*REV*PV)**0.4*CV/DH
```

```
2803        HCONL = HCONL*FAL + HCV
2804        HN6 = ZERO
2805        RETURN
2806        END
```

Subroutine iphtc

```
2807        SUBROUTINE IPHTC (HIF,ALFA)
2808        IMPLICIT REAL*8 (A-H,O-Z)
2809        COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
2810 C
2811        HIF = 5.D+08
2812        RETURN
2813        END
```

```
      SUBROUTINE STATE (TV,TL,P,PROP,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION PROP(3,4)
      DATA RV0,RV1,RV2,RV22 /1.605D-02,2.51D-06,-3.23D-13,-6.46D-13/
      DATA RL0,RL1,RL2,RL3,RLP,RL22,RL33 /1.0116D+03,-0.2205,
     1     -1.9224D-05,5.6377D-09,2.26D-07,-3.8448D-05,
     2     1.6913ID-08/
      DATA EV0,EV1,EV2,EV3,EV22,EV33 /5.0215D+06,5.8714D+02,
     1     -.41672,1.54272D-04,-.83344,4.62816D-04/
      DATA EL0,EL1,EL2,EL3,EL22,EL33 /-6.75075D+04,1.63014D+03,
     1     -.41672,1.54272D-04,-.83344,4.62816D-04/
C
C           ALL PROPERTIES IN SI UNITS
C           PROPERTIES BASED IN
C           GOLDEN,G.H. AND TOKAR,J.V.,
C           THERMOPHYSICAL PROPERTIES OF SODIUM, ANL-7323
C           WITH THE ADDITION OF PRESSURE DEPENDENCE IN THE
C           LIQUID DENSITY.
C                THIS ADDITION WAS MADE BECAUSE THE NUMERICAL
C           STABILITY OF THE MODEL REQUIRES A NON ZERO,
C           POSITIVE VALUE OF THE PRESSURE DERIVATIVE OF
C           THE DENSITY .
C
C                ALSO A REQUIREMENT FOR THE NUMERICAL CONVERGENCE
C           IS THE DERIVATIVES OF PROPERTIES WITH RESPECT TO
C           TEMPERATURE AND PRESSURE BEING THE MATHEMATICAL
C           DERIVATIVES OF THE EXPRESSIONS FOR THE PROPERTIES
C
C
      TS = SAT(P)
      X1 = (RV2*P + RV1)*P + RV0
      PROP(1,1) = X1*TS/TV
      PROP(1,2) = ((RL3*TL + RL2)*TL + RL1)*TL + RL0 + RLP*P
      PROP(1,3) = ((EV3*TV + EV2)*TV + EV1)*TV + EV0 - P/PROP(1,1)
      PROP(1,4) = ((EL3*TL + EL2)*TL + EL1)*TL + EL0
C
      PROP(2,1) = -PROP(1,1)/TV
      PROP(2,2) = (RL33*TL + RL22)*TL + RL1
      PROP(2,3) = (EV33*TV + EV22)*TV + EV1
      PROP(2,4) = (EL33*TL + EL22)*TL + EL1
C
```

```
2857        PROP(3,1) = (X1*DTSDP(P) + (RV22*P + RV1)*TS)/TV
2858        PROP(3,2) = RLP
2859        PROP(3,3) = (P/PROP(1,1)*PROP(3,1) - ONE)/PROP(1,1)
2860        PROP(3,4) = ZERO
2861        RETURN
2862        END
```

```
      SUBROUTINE NONEQ(ALFAO,ALFA,TV,TL,P,RHOV,RHOL,TS,S,IFLAG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PD/ D4,POD2
      DIMENSION S(5,2)
      DATA AN,RGAS /1.33333330+07,.14469D+03/,HALF /0.5D0/
      DATA PI,SR3,CADRY,ADRY /3.141592654,3.464101616,0.043,0.957/
      DATA H0,H1 /5.089D+06,-.1043D+04/
      DATA RNU /6.D+03/
      DATA HL0,HL1,HL2,HL3 /-6.75075D+04,1.630140+03,
     *     -.41672D0,1.54272D-04/
C
C           SUBROUTINE NONEQ CALCULATES THE MASS AND ENERGY EXCHANGE RATES
C           AND ITS DERIVATIVES.
C           AN = 4/3*N, N = 1.0D+07 BUBLES/CUBIC METER
C           RGAS = SQUARE ROOT OF GAS CONSTANT FOR SODIUM OVER 2*PI
C           POD2 = PITCH TO DIAMETER RATIO SQUARED
C
C           S(1, ) =  EXCHANGE RATE          S( ,1) = MASS
C           S(2, ) = D/DTV                   S( ,2) = ENERGY
C           S(3, ) = D/DTL
C           S(4, ) = D/DP
C           S(5, ) = D/DALFA
C
      AX = ALFAO
      IF(ALFAO.LT.1.D-04) AX = 1.D-04
      IF(ALFAO.GT.0.9999) AX = 0.9999
C
      TS = SAT(P)
      HLG = H1*TS + H0
      X = ONE/(SR3*POD2 - PI)
C
      AM = 1.2D-07*PI*X*D4*D4
      IF(ALFAO.GT.0.6) GO TO 10
C
      XX = 3.*PI*AX*X
      GO TO 20
   10 CONTINUE
      Y = ONE
      IF(AX.GT.ADRY) Y = (ONE - AX)/CADRY
      XK = 1.8/(SR3*POD2*X - 0.6)
      XX = (SR3*POD2*X - AX)*X*Y*PI*XK
```

```
2906     20 CONTINUE
2907        A = DSQRT(XX)*D4
2908 C
2909     30 CONTINUE
2910        CE = A*RGAS*RHOV*RHOV
2911        CC = CE*(ONE - AX)
2912        CE = CE*AX
2913 C
2914        EL = ZERO
2915        CL = ZERO
2916        IF(TL.GT.TS) EL =.1.0D0
2917        IF(TS.GT.TV) CL = 5.D-03
2918 C
2919        CE = CE*EL
2920        CC = CC*CL
2921 C
2922        DDP = DTSDP(P)
2923        SRTS = DSQRT(TS)
2924        DTL = (TL - TS)/SRTS
2925        DTV = (TS - TV)/SRTS
2926 C
2927 C              MASS EXCHANGE RATE
2928 C
2929        SE = DTL*CE*(ONE - ALFA)
2930        SC = DTV*CC*ALFA
2931        S(1,1) = SE - SC
2932 C
2933 C              DERIVATIVES
2934 C
2935        S(2,1) = CC*ALFA/SRTS
2936        S(3,1) = CE*(ONE-ALFA)/SRTS
2937        DSEVAP = CE*(ALFA-ONE)*(TS+TL)/TS/SRTS*HALF*DDP
2938        DSCOND = CC*      ALFA*(TS+TV)/TS/SRTS*HALF*DDP
2939        S(4,1) = DSEVAP - DSCOND
2940        S(5,1) = -CE*DTL - CC*DTV
2941 C
2942 C                 ENERGY EXCHANGE RATE
2943 C
2944        U = A*CONDL(TV)*RNU*D4
2945        HL = ((HL3*TS + HL2)*TS + HL1)*TS + HL0
2946        HV = HL + HLG
2947        DHLDP = ((3.*HL3*TS + 2.*HL2)*TS + HL1)*DDP
2948        DHVDP = DHLDP + H1*DDP
2949 C
2950        S(1,2) = SE*HV - SC*HL + U*(TL - TV)
```

```
C
C                    DERIVATIVES
C
      S(2,2) = S(2,1)*HL - U
      S(3,2) = S(3,1)*HV + U
      S(4,2) = DSEVAP*HV + SE*DHVDP - DSCOND*HL - SC*DHLDP
      S(5,2) = -CE*DTL*HV - CC*DTV*HL
      RETURN
      END
```

```
      SUBROUTINE CONDT(TV,TL,P,ALFA,TS,TW,DTW,
     *                 HCONV,HCONL,HNB,DV,Q,KO)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LSS
      COMMON /STST/ TAFP,LSS
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION Q(4,2)
C
      Q(1,1) = (TW - TV)*HCONV*DV
      Q(1,2) = ((TW - TL)*HCONL + (TW - TS)*HNB)*DV
      Q(2,1) = (DTW*HCONV - 1)*HCONV*DV
      Q(2,2) = ZERO
      Q(3,1) = ZERO
      Q(3,2) = ((HCONL + HNB)*DTW - 1)*HCONL*DV
      Q(4,1) = ZERO
      Q(4,2) = ((HCONL + HNB)*DTW - 1)*HNB*DV*DTSDP(P)
      RETURN
      END
```

Subroutine hexcan

```
      SUBROUTINE HEXCAN(TCAN,DTC,TV,TL,HCONV,HCONL,QV,QL,
     *                  DQDTV,DQDTL)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /HXCN/ ACOV
C
C          SUBROUTINE HEXCAN CALCULATES THE HEAT TRANSFERED TO
C          THE HEXCAN AND ITS DERIVATIVES.
C
      QV = ACOV*HCONV*(TV - TCAN)
      QL = ACOV*HCONL*(TL - TCAN)
      DQDTV = ACOV*HCONV*(ONE - DTC*HCONV)
      DQDTL = ACOV*HCONL*(ONE - DTC*HCONL)
      RETURN
      END
```

Subroutine fprop

```
      SUBROUTINE FPROP(TRN,NPIN,NPM1,I)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /ICONST/ NCF,NCC,NG
      DIMENSION TRN(NPIN)
C
C           FUEL PROPERTIES
C
      DO 1 K = 1,NCF
      T = (TRN(K+1) + TRN(K))/2.D0
      CALL FUEL (T,K,I)
    1 CONTINUE
C
C           CLAD PROPERTIES
C
      DO 2 K = NCC,NPM1
      T = (TRN(K+1) + TRN(K))/2.D0
      CALL CLAD (T,K)
    2 CONTINUE
C
C           GAP CONDUCTIVITY
C
      T = (TRN(NG+1) + TRN(NG))/2.D0
      CALL GAP (T,TRN(NG),TRN(NG+1),NG)
      RETURN
      END
```

Subroutine fuel

```
3021        SUBROUTINE FUEL (T,K,I)
3022        IMPLICIT REAL*8 (A-H,O-Z)
3023        COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
3024        COMMON /PIN1/ CPIN(20),ROCP(20)
3025        COMMON /FCONST/ A0,A1,A2,A3,
3026       *                B0,B1,B2,AD,APU,LPLNM(40)
3027 C
3028        T2 = T*T
3029        T3 = T*T2
3030        X = 2.74D0 - 5.8D-04*T
3031 C
3032        CPIN(K) = (B0 + B1*T + B2*T2)*(ONE - (ONE - AD)*X)
3033        ROCP(K) = (A0 + A1*T + A2*T2 + A3*T3)*AD*(ONE + 0.045*APU)
3034        IF(LPLNM(I).EQ.0) ROCP(K) = 1.D+04
3035        RETURN
3036        END
```

Subroutine clad

```
      SUBROUTINE CLAD (T,K)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /CCONST/ A0,A1,A2,A3,B0,B1,B2,B3
C
      T2 = T*T
      T3 = T*T2
C
      CPIN(K) = B0 + B1*T + B2*T2 + B3*T3
      ROCP(K) = A0 + A1*T + A2*T2 + A3*T3
      RETURN
      END
```

Subroutine gap

```
      SUBROUTINE GAP (T,TF,TC,NG)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /GCONST/ DIL,RADFU,RADCL
C
      DATA ESB,HMIN /1.7D-08,3.705D+03/
      DATA C1,C2 /2.D0,1.5D+01/
      DATA G1,G2,G3 /1.32D-04,0.61D-04,1.8D+03/
C
C             CONDUCTION HEAT TRANSFER
C
      DGAP = RADCL - RADFU
      CG = C2**DIL*C1
      HG = ONE/((DGAP + G1)/CG + G2) + G3
C
C             RADIATION HEAT TRANSFER
C
      HR = (TF*TF + TC*TC)*(TF + TC)*ESB
      HGAP = HG + HR
      IF(HGAP.LT.HMIN) HGAP = HMIN
C
      ROCP(NG) = ZERO
      CPIN(NG) = HGAP
      RETURN
      END
```

```
      SUBROUTINE FPIN(TV,TL,TS,TW,DTW,HCONV,HCONL,HNB,
     *                TR,DTR,DT,NPIN,NPM1,KO)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LSS
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PIN0/ RODR(20),VP(20),VM(20),RADR,PPP(20)
      COMMON /PIN1/ CPIN(20),ROCP(20)
      COMMON /STST/ TAFP,LSS
      DIMENSION A1(20),A2(20),A3(20),B1(20)
      DIMENSION TR(NPIN),DTR(NPIN)
C
      CALL POWER(HEAT,KO)
C
      DTI = ONE/DT
      IF(LSS) DTI = ZERO
C
      A1(1) = ZERO
      A2(1) = RODR(1)*CPIN(1) + VP(1)*ROCP(1)*DTI
      B1(1) = VP(1)*HEAT*PPP(1) + VP(1)*ROCP(1)*TR(1)*DTI
      DO 1 K = 2,NPM1
      KM1 = K - 1
      A1(K) = -RODR(KM1)*CPIN(KM1)
      A2(K) = -A1(K) + RODR(K)*CPIN(K) + (VP(K)*ROCP(K) +
     +        VM(K)*ROCP(KM1))*DTI
      B1(K) = VP(K)*HEAT*PPP(K) + VM(K)*HEAT*PPP(KM1) +
     +        (VP(K)*ROCP(K) + VM(K)*ROCP(KM1))*TR(K)*DTI
    1 CONTINUE
C
      A1(NPIN) = -RODR(NPM1)*CPIN(NPM1)
      A2(NPIN) = -A1(NPIN) + VM(NPIN)*ROCP(NPM1)*DTI +
     +           RADR*(HCONV + HCONL + HNB)
      B1(NPIN) = VM(NPIN)*ROCP(NPM1)*TR(NPIN)*DTI +
     +           RADR*(HCONV*TV + HCONL*TL + HNB*TS) +
     +           VM(NPIN)*HEAT*PPP(NPM1)
C
      A1(NPIN+1) = ZERO
C
      A2(1) = ONE/A2(1)
      A3(1) = A1(2)*A2(1)
      B1(1) = B1(1)*A2(1)
C
      DO 2 K = 2,NPIN
      KM1 = K - 1
```

```
      A2(K) = ONE/(A2(K) - A1(K)*A3(KM1))
      A3(K) = A1(K+1)*A2(K)
      B1(K) = (B1(K) - A1(K)*B1(KM1))*A2(K)
    2 CONTINUE
C
      TW = B1(NPIN)
      DTW = A2(NPIN)*RADR
      DO 3 K = 1,NPM1
      TR(K) = B1(K)
      DTR(K) = A3(K)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FTP(TV,TL,TS,TW,HCONV,HCONL,HNB,TR,DTR,QPP,
     *               NI,NJ,NN,NP,NTR,NPM1,NIM2,NPIN)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      DIMENSION TR(NTR),DTR(NTR),TW(NP),TS(NN),TV(NN),TL(NN),
     *          HCONV(NP),HCONL(NP),HNB(NP),QPP(NN)
C
      TWMAX = ZERO
      TRMAX = ZERO
C
      DO 3 I = 1,NIM2
      DO 3 J = 1,NJ
      KO = (J-1)*NI + I + 1
      KP = (J-1)*NIM2 + I
      KR = KP*NPIN
C
      TR(KR) = TW(KP)
C
      DO 1 KK = 1,NPM1
      KTR = KR - KK
C
      TR(KTR) = TR(KTR) - DTR(KTR)*TR(KTR+1)
      IF(TRMAX.GT.TR(KTR)) GO TO 1
      TRMAX = TR(KTR)
      KTRMAX = KTR
    1 CONTINUE
C
      IF(TWMAX.GT.TW(KP)) GO TO 2
      TWMAX = TW(KP)
      KTWMAX = KO
    2 CONTINUE
C
      QPP(KP) = HCONV(KP)*(TW(KP) - TV(KO)) + HCONL(KP)*
     *          (TW(KP)-TL(KO)) + HNB(KP)*(TW(KP)-TS(KO))
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE THXCN(TV,TL,HCONV,HCONL,TCAN,DT,NN,NI,NJ,NCAN,
     *                 NIM1,NIM2)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LSS
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /STST/ TAFP,LSS
      DIMENSION TV(NN),TL(NN),HCONV(NN),HCONL(NN),TCAN(NCAN)
C
C
C         SUBROUTINE THXCN PERFORMS THE FIRT CALCULATION OF THE
C         HEXCAN TEMPERATURE.
C
      DTI = ONE/DT
      IF(LSS) DTI = ZERO
C
      DO 10 I = 2,NIM1
      KO = (NJ-1)*NI + I
      KP = (NJ-1)*NIM2 + I - 1
      K2 = NI + I
      K3 = K2 + NI
      K4 = K3 + NI
C
      TCAN(K2) = ONE/(TCAN(K4)*DTI + HCONV(KP) + HCONL(KP))
      TCAN(I) = (TCAN(K4)*TCAN(K3)*DTI + HCONV(KP)*TV(KO) +
     +           HCONL(KP)*TL(KO))*TCAN(K2)
   10 CONTINUE
      RETURN
      END
```

Subroutine thxcn0

```
      SUBROUTINE THXCN0(TCAN,NCAN,NI)
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION TCAN(NCAN)
C
C         SUBROUTINE THXCN0 TRANSFERS THE NEW VALUE OF THE HEXCAN
C         TEMPERATURE TO THE OLD HEXCAN TEMPERATURE ARRAY.
C
      TCAN(1) = TCAN(2)
      TCAN(NI) = TCAN(NI-1)
      DO 10 I = 1,NI
      K3 = 2*NI + I
      TCAN(K3) = TCAN(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POWER (HEAT,KO)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LP
      COMMON /ERROR/ IERR
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /PSHAPE/ SHAPE(100)
      COMMON /TEMPO/ TIME,DT,DTO,DTLS,NDT
      COMMON /BCOND/ TB(51),PNB1(51),PNB2(51),PNB3(51),OMP(51),
     *               PNT1(51),PNT2(51),PNT3(51),OMT(51),ALB1(51),
     *               ALB2(51),ALB3(51),OMA(51),TVB1(51),TVB2(51),
     *               TVB3(51),OMV(51),TLB1(51),TLB2(51),TLB3(51),
     *               OML(51),HNW1(51),HNW2(51),HNW3(51),OMH(51),
     *               LMAX,LP(51)
C
C
      L = 2
    1 CONTINUE
      IF(TIME.LE.TB(L)) GO TO 2
      L = L + 1
      IF(L.GT.LMAX) RETURN
      GO TO 1
    2 CONTINUE
      DTIME = TIME - TB(L-1)
      HEAT = HNW1(L)*DTIME + HNW2(L)
      IF(LP(L)) GO TO 3
      HEAT = DCOS(OMH(L)*DTIME)*HEAT + HNW3(L)
    3 CONTINUE
      HEAT = SHAPE(KO)*HEAT
      RETURN
      END
```

```
      SUBROUTINE GAUSIE (A1,A2,A3,A4,F,X,BETA,GAMMA,NC)
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /NUMBER/ ZERO,ONE,BIG,SMALL
      COMMON /GAUSS/ NZ,NR,NZM1
      COMMON /ERROR/ IERR
      COMMON /CNTRL/ EPS1,EPS2,RES,IT1,IT2,IT3,ITM1,ITM2,ITRMAX
      DIMENSION A1(NC),A2(NC),A3(NC),A4(NC),F(NC),X(NC),
     *          BETA(NC),GAMMA(NC)
C
      ITR = 0
    1 CONTINUE
C
C          NEW SOLUTION AT THE BOTTON
C
      I = 1
      GAMMA(1) = F(I) - A3(I)*X(I+1)
      BETA(1)  = ONE
C
      DO 2 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J) = (F(K)-A3(K)*X(K+1) - A1(K)*GAMMA(J-1))/BETA(J)
    2 CONTINUE
C
      K = (NR-1)*NZ + I
      CONV = DABS(X(K) - GAMMA(NR))
      X(K) = GAMMA(NR)
      DO 3 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
      DX = DABS(X(KX) - XA)
      IF(DX.GT.CONV) CONV = DX
      X(KX) = XA
    3 CONTINUE
C
C          NEW SOLUTION OUT OF THE BOUNDARIES
C
      DO 6 I = 2,NZM1
C
      GAMMA(1) = F(I) - A2(I)*X(I-1) - A3(I)*X(I+1)
```

```
      DO 4 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J)= (F(K) - A2(K)*X(K-1) - A3(K)*X(K+1) -
     -            A1(K)*GAMMA(J-1))/BETA(J)
    4 CONTINUE
C
      K = (NR-1)*NZ + I
      DX = DABS(X(K) - GAMMA(NR))
      IF(DX.GT.CONV) CONV = DX
      X(K) = GAMMA(NR)
C
      DO 5 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
      XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
      DX = DABS(X(KX) - XA)
      IF(DX.GT.CONV) CONV = DX
      X(KX) = XA
    5 CONTINUE
    6 CONTINUE
C
C          NEW SOLUTION AT THE TOP
C
      I = NZ
      GAMMA(1) = F(I) - A2(I)*X(I-1)
      DO 7 J = 2,NR
      K = (J-1)*NZ + I
      K1 = K - NZ
C
      BETA(J) = ONE - A1(K)*A4(K1)/BETA(J-1)
      GAMMA(J) = (F(K) - A2(K)*X(K-1) - A1(K)*GAMMA(J-1))/
     /                  BETA(J)
    7 CONTINUE
C
      K = (NR-1)*NZ + I
      DX = DABS(X(K) - GAMMA(NR))
      IF(DX.GT.CONV) CONV = DX
      X(K) = GAMMA(NR)
C
      DO 8 J = 2,NR
      K = NR - J + 1
      KX = (K-1)*NZ + I
```

```
3330        XA = GAMMA(K) - A4(KX)*X(KX+NZ)/BETA(K)
3331        DX = DABS(X(KX) - XA)
3332        IF(DX.GT.CONV) CONV = DX
3333        X(KX) = XA
3334      8 CONTINUE
3335 C
3336 C          CONVERGENCE TEST
3337 C
3338        IF(CONV - EPS2) 11,11,9
3339      9 IF(ITR - ITRMAX) 1,10,10
3340     10 IERR = 1
3341     11 CONTINUE
3342        RES = ZERO
3343        DO 12 L = 1,NC
3344        XX = DABS(X(L))
3345        IF(XX.GT.RES) RES = XX
3346     12 CONTINUE
3347        RETURN
3348        END
```

```
      SUBROUTINE ERRMES(TIME)
C
C                SUBROUTINE ERRMES PRINTS THE ERROR MESSAGES
C                WHENEVER THE EXECUTION OF THE PROGRAM HAS
C                BEEN TERMINATED DUE TO NUMERICAL ERRORS SUCH
C                AS INSTABILITY,VARIABLES OUT OF RANGE ETC.
C
C
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON /ERROR/ IERR
C
      WRITE(6,1100) TIME
C
      IF(IERR - 2) 1,2,100
  100 IF(IERR - 4) 3,4,101
  101 IF(IERR - 22) 21,22,102
  102 IF(IERR - 24) 23,24,103
  103 IF(IERR - 26) 25,26,104
  104 IF(IERR - 28) 27,50,50
C
    1 WRITE(6,1001)
      GO TO 200
    2 WRITE(6,1002)
      GO TO 200
    3 WRITE(6,1003)
      GO TO 200
    4 WRITE(6,1004)
      GO TO 200
   21 WRITE(6,1021)
      GO TO 200
   22 WRITE(6,1022)
      GO TO 200
   23 WRITE(6,1023)
      GO TO 200
   24 WRITE(6,1024)
      GO TO 200
   25 WRITE(6,1025)
      GO TO 200
   26 WRITE(6,1026)
      GO TO 200
   27 WRITE(6,1027)
      GO TO 200
   50 WRITE(6,1050)
```

```
C
  200 CONTINUE
      WRITE(6,1101)
1100 FORMAT(1H1,35(' *')//10X,'EXECUTION TERMINATED ON ERROR',
     *' CONDITION AT TIME ',F10.4//)
1001 FORMAT(1X,'THE PRESSURE MATRIX INVERSION DOES NOT CONVERGE'/
     *       1X,'IN THE MAXIMUM NUMBER OF ITERATIONS ALLOWED'//
     *       1X,'ERROR CONDITION NUMBER = 1'/)
1002 FORMAT(1X,'THE PRESSURE MATRIX IS NOT DIAGONAL DOMINANT'//
     *       1X,'ERROR CONDITION NUMBER = 2'/)
1003 FORMAT(1X,'THE VOID FRACTION TOOK A VALUE EITHER LOWER THAN'/
     *       1X,'ZERO OR GREATER THAN ONE'//
     *       1X,'ERROR CONDITION NUMBER = 3'/)
1004 FORMAT(1X,'THE INITIAL CONDITIONS INPUT DATA IS NOT IN THE'/
     *       1X,'PROPER ORDER'//
     *       1X,'ERROR CONDITION NUMBER = 4'/)
1021 FORMAT(1X,'THE TIME STEP SIZE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 21'/)
1022 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 22'/)
1023 FORMAT(1X,'THE PRESSURE TOOK A VALUE TOO HIGH'/
     *        1X,'ERROR CONDITION NUMBER = 23'/)
1024 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO SMALL'/
     *       1X,'ERROR CONDITION NUMBER = 24'/)
1025 FORMAT(1X,'THE VAPOR TEMPERATURE TOOK A VALUE TOO HIGH'/
     *       1X,'ERROR CONDITION NUMBER = 25'/)
1026 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO SMALL'/
     *        1X,'ERROR CONDITION NUMBER = 26'/)
1027 FORMAT(1X,'THE LIQUID TEMPERATURE TOOK A VALUE TOO HIGH'/
     *       1X,'ERROR CONDITION NUMBER = 27'/)
1050 FORMAT(1X,'A QUIT SIGNAL WAS ISSUED BY THE TERMINAL OPERATOR'/
     *       1X,'ERROR CONDITION NUMBER = 50'/)
1101 FORMAT(1X,35(' *'))
      RETURN
      END
```

```
      SUBROUTINE SAVER(P,TV,TL,ALFA,UVZ,ULZ,UVR,ULR,TR,TCAN,
     *                 TIME,NTR,NN,NCAN,NI)
      IMPLICIT REAL*8 (A-H,O-Z)
      LOGICAL LDATA
      DIMENSION P(NN),TV(NN),TL(NN),ALFA(NN),UVZ(NN),ULZ(NN),
     *          UVR(NN),TCAN(NCAN),TR(NTR),ULR(NN)
      DIMENSION XOUT(5)
      LDATA = .FALSE.
      WRITE(7,103) LDATA,TIME
      DO 1 KO = 1,NN
      WRITE(7,100) KO,TV(KO),TL(KO),P(KO),ALFA(KO)
      WRITE(7,100) KO,UVZ(KO),ULZ(KO),UVR(KO),ULR(KO)
    1 CONTINUE
      LDATA = .TRUE.
      KRES = 0
    2 CONTINUE
      DO 3 K = 1,5
      KM = KRES + K
      IF(KM.GT.NTR) GO TO 4
      XOUT(K) = TR(KM)
    3 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      KRES = KRES + 5
      GO TO 2
    4 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      LDATA = .FALSE.
      WRITE(7,102) LDATA
      LDATA = .TRUE.
      KRES = 2*NI
      K3 = 3*NI
    5 CONTINUE
      DO 6 K = 1,5
      KM = KRES + K
      IF(KM.GT.K3) GO TO 7
      XOUT(K) = TCAN(KM)
    6 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      KRES = KRES + 5
      GO TO 5
    7 CONTINUE
      WRITE(7,101) LDATA,(XOUT(KL),KL=1,5)
      LDATA = .FALSE.
```

```
3470        WRITE(7,102) LDATA
3471    100 FORMAT(I5,4D15.9)
3472    101 FORMAT(L1,5D15.9)
3473    102 FORMAT(L1)
3474    103 FORMAT(L1,D15.9)
3475        RETURN
3476        END
```